CHAPTER 1: INTRODUCTION

1.1 Introduction

Recently, the use of mobile phone is increasing day by day globally and subsequently the use of Short Message Service (SMS) has increased tremendously because texting is the cheapest and most popular mode of mobile phones communications. Billions of Short Message Service messages are sent daily and the value increases significantly. Short Message Service nowadays is the main profits generator for wireless carriers. Thus, the text entry into mobile devices has become an activity which is performed routinely by billions of users. This activity consists of composing and replying messages through SMS application. A lot of innovative applications and systems are now built based on the Short Message Service. With this thing in mind, there are various methods for text entry in the mobile phones. There are two main varieties such as predictive like Tegic’s T9 and non-predictive entry. Predictive text entry provides a fast way to enter text on a mobile phone device.

SMS is the service that sends messages from one mobile to another asynchronously, disjointedly and at a very low cost. Since the first SMS was sent in 1992, SMS transmission rates have increased rapidly and the mobile phone keypad has stay put limited in facility in terms of text entry convenience (Dunlop, 2007).

Mobile messaging is one of the popular ways to communicate and mobile phone instant messaging which enables sending messages from Malaysia to any other countries. It is a big challenge to write and display Malay language characters on mobile devices. There are many languages preference to send and receive messages in Malaysia because it is a multi-racial country. The facility of sending messages is available but, users who
are unable to understand English will not be able to use it for this purpose. Therefore, an increase in language preferences for these users may increase the usability of SMS.

Traditionally, text is entered on the mobile phone using the Multi-Tap method. This method uses a keypad which contains several characters mapped onto a single key and the mapping occur in typing delays as several keystrokes may be needed to outcome one letter. Developments have been made to ease text entry with the use of several methods. However, several algorithms have been implemented with the aim to trim down the usage of keypad when typing and simultaneously improve text entry speed. A language prediction is used to enhance speed and accuracy of text entry (Soukoreff, 2002) and develop techniques like the T9 (Tegic Communications, 2007). This research will explore on a existing predictive text algorithm for improved text entry and also will be implemented to enhance text entry convenience for Malay language.

In mobile phone, predictive text entry is a more successful method provided by T9 which allows users to enter text by pressing only one key per letter which is a useful feature to enter text messaging quickly. The method which is applied by T9 consists of an intuitive single tap technique using a technology of “Predictive text” by storing a dictionary predicting the key combinations. As a word is entered, the phone automatically compares all the possible letter combinations against a built-in dictionary of words and determines which word user intended to type. In case, it guesses correctly, the user can scroll through other possible words without re-typing the words.

How it allows user to enter the text is by single-tap keystrokes technique, for example pressing the numeric key “7” predicts “S”, the mostly used single letter word; continuing to press “3” leads to the word “Re”; pressing “3” again leads to “See” and finally pressing “3” leads to “Seed”. In summary 4 keystrokes – “7”, ”3”, ”3” and “3”
leads to the entry of “Seed”. If multi-tap would have needed 9 keystrokes (four times “7”, twice “3”, twice “3” and once “3”). The number of keystrokes typically has more than 100% reduction without the need for QWERTY type keyboard, compared to the traditional methods, when using predictive text entry. This method makes it much easier to enter long text messages faster. There are a handful of competing technologies in the mobile industry, including the AOL’s T9, Motorola’s ITAP and VICORP’s eZiText. However T9 is the most popular. eZiText competes with T9 and Itap is used by Motorola in some of its phones. A phone with T9 can exchange with an iTAP phone without any problem because there are no issues of compatibility.

There are not many choices to develop such applications, so the most commonly used platforms is Java 2 Micro Edition (J2ME) (James White, 2002) for Java enabled mobile phones and NetBeans IDE for mobile phone based on windows. The numbers of users who are using this platform are increasing day by day. This research focuses on the development of Malay Language Predictive Text Entry Tool (MLPTET) for short message service on mobile phone that incorporates prediction algorithm to assist the process of prediction of the text entry in Malay language.

In the sections that follow, this chapter presents problem statement, statement of objectives, research significance, research scope, project limitations, target users, mobile technologists and linguist, research methodology and thesis organizations.

1.2  Problem Statement
The predictive text entry method uses linguistic knowledge to predict the intended words of the user in mobile phones. The existing mobile phones have licensed the T9 input method that uses a dictionary as the basis for disambiguation. The Tegic
Communication Corporation has predictive text entry on mobile phones and has the method for compressing the dictionary in the memory. Based on the report given by Soukoreff et al. (1995), users use more comfortable words compared to what it appears in any corpus which the predictive text entry system is based on.

There is no doubt that predictive text entry has been very popular input technology. This popularity has been in spite of limitation of easy input mechanisms in mobile devices makes it very uncomfortable sending messages larger than 5-6 words. The issue of disambiguation failure happens when input text is more than one dictionary word for single sequence of keystrokes resulting misspelling and also limiting the effectiveness of predicting text implementations. The problem is multiple selection actions required. The are 2 types of consecutive selection. First make between group selection for example \{2,a,b,c,A,B,C\} and then make within group selection or disambiguation of automatic selection for example \{a\}.

1.2.1 Predictive Text Entry Current Problems

The current problems that relate to predictive text entry, will be addressed and discussed on the following issues.

- Predictive text entry feature for Short Message Service (SMS) is useful but it has some limitations. Nevertheless, it is instant and users can send message from one place to all over the world. However, the Short Service Message design which was meant to deliver short data allow maximum 160 characters to avoid overloading the system but the messages are split into several messages and the recipients receive the message in a single message. There is no language preference to send and receive messages. Although, the facility of sending
messages is also available but users who are unable to understand English which is commonly used, unable to use it to send message. Choices of language preference for these users may increase the usability of SMS. Therefore, this research work attempts to combine the functionalities of prediction text entry in Malay language and displaying the words on the windows based mobile phone and subsequently can download the search engine algorithm into a Nokia mobile device.

- The traditional method “multi-tap” text entry which is commonly used in mobile phone, in which user must press the key repeatedly in order to get the correct letter. Therefore, this method produces higher error rate and misspelled words in each text message. Essentially, it slows the text messaging process. The problem is, majority of users are still turning off the Predictive text mode and using the fashioned Multi-Tap for precise data entries not knowing the features. This grouping causes ambiguity and it is the problem of mobile phone keypad. The only way to overcome this problem is by correcting the ambiguous keypad.

- One of the problems is not all words that users need is included in the dictionary and language use varies with the circumstances and purpose for which it is used. According to the research conducted by Mackenzie (2002), it seems that the corpora does not based on data captured during input processing. Thus, the predictive text entry system does not recognize slang words. Therefore, predictive text entry cannot easily handle text written problems and support language use in mobile text messaging effectively.
Another issue is to reduce the ambiguities in predictive text which means that the dictionary based on disambiguation method will resolve the multi-tap problem by utilizing a small dictionary to predict words based on key combinations. The predictive text algorithm has ambiguity problem especially in languages such as Chinese, Japanese and most of the Unicode languages. The 9 button keypads, up to four letters are assigned to one button, requiring at worst 4 button presses to type just one letter. The same numerical strings could represent multiple words. Usually users do not get what they intended to type. In the case of typing in a language, such as Malay, with a 9 button keypads, predictive text entry ends up not being as efficient as simply typing without prediction. For example, entering a string of “4,6,6,3” on the keypad, the predictive text method will generate word candidates of “gone”, “home” and “good”. For such cases, it requires users to take traditionally key presses to select the correct one. Sometimes, it displays words which are not intended or meaningful during typing. The following sequence of key presses is required: 4677. The same sequence of key presses also matches the words “hopp”. The intended word is presented as the third choice. A solution is needed to eliminate “unwanted words” during predictive text entry. Though there are existing tools for predictive text entry, but prediction word completion of the keystrokes reduce is still required.

1.3 Statement of Objectives

As a result from the concerns brought from the statement of problem, this research therefore seeks to consider the following specific objectives. The project aims to develop a Malay Language Predictive Text Entry Tool (MLPTET) for Short Message Service on mobile phone interface. The following are the three objectives.
• To investigate the most efficient predictive text entry method (or algorithm).
• To design and develop Malay language predictive text entry tool based on the suitable existing algorithm.
• To evaluate the functionality, usability and accuracy factors of the developed Malay language predictive text entry tool.

1.4 Research Significance

This research is designed to shape an effective and usability predictive model of text entry in Malay language for SMS that permits a mobile phone user to send text messages, SMS, texts and any short messages. The predictive messaging have been introduced due to the difficulty in using and mastering the interface as the technology is being pressed into areas for which was not necessarily intended. Predictive text entry intended to facilitate interaction and simplify text entry and there are several issues potentially addressed. One the issues which have been considered the contribution of predictive messaging includes the speed of text entry.

Many countries have shown that it is not necessary to use English to achieve and enjoy the benefits of communication services. It is possible to develop comprehensive and innovative communication services using the national language.

The domain specificity of the system is to make predictions about the content of text input by the users. The mode of operation is as an interactive system with a flexible human-machine interface (system and / or user driven) where the system cooperates with the user in order to produce an effective and accurate user selectable language interface, which will reduce the ambiguity.
A predictive text entry will benefit the users whose preference language is Malay. This will increase the use of the text messaging application for those who do not understand English language. Based on the analysis and design made, a more suitable and flexible text prediction feature designed to overcome the existing text input issues feature and supported with Malay language. In addition, this research shows the importance to reduce the text entry work of the public users, as billion of messages are being sent out worldwide.

Another contribution is that the proposed prediction tool should be able to support in Nokia mobile phone and incorporated with the existing method. This tool mainly targets at students, business people, teenagers and those whom preferred language is Malay. The most common system of Short Message Service text input referred to as Multi-Tap. From Multi-Tap, a key or button is pressed multiple times to access the list of letters on that key. For instance, pressing the “2” key once displays an, “a”, twice displays a “b” and three times displays a “c”. To enter two consecutive letters that are on the same key, the user must either pause or hit a “next” button. A user can key in by pressing an alphanumeric keypad without looking at the screen display. Thus, multi-tap is easy to understand, and can be used without any visual feedback from the screen display. However, multi-tap is not very efficient and considered a difficulty by many users of electronic equipment. In addition, since all twenty-six letters of the English language are entered using only eight alphanumeric keys, it is difficult to use Short Messaging Service.

Predictive text improves on multi-tap. Predictive text entry dictionaries software applications generally reduce the number of key strokes that a user is required to enter or press to express a word. With predictive text dictionaries, users of electronic
equipment such as mobile phone do not need to press each keypad number several times in order to get the desired letter. The user just presses the number that corresponds to the letter and, as long as the word exists in the predictive text dictionary repository, the electronic equipment will search and generally recognize the word. The most extensively used systems of predictive text dictionaries are T9, iTAP and eZiText. Each of these systems requires the manufacturer to install a local dictionary of words and phrases for every input language that the electronic equipment supports such as mobile phone.

1.5 Research Scope

This research is on the implementation of a predictive text entry tool for SMS on a mobile device in Malay language. Predictive is a way to overcome the problem of ambiguity with adding the “linguistic knowledge” to the system. Prediction techniques attempt to reduce the input tediousness by predicting what user is intending. The predictive text entry algorithm will be implemented in J2ME platform. The foundation of the research and development of the text input prediction tool is to minimize the text input work by user as much as possible when SMS using mobile phone. In addition, the proposed text input prediction tool should incorporate all the features that have been determined to solve the shortcomings issues and problems relating to the existing SMS text input prediction and anticipated problem.

1.6 Project Limitations

This research is not to consider to measure or optimize the performance engine while entering the predictive text entry short message service in Malay language. Besides, to reduce the time message input using the algorithm over the standard predictive text entry model also is not part of this research’s scope. In short, it is to develop the
prediction tool in Malay language using J2ME program which is widely used software for mobile application.

### 1.7 Target User

The target user for the Malay Language Predictive Text Entry Tool (MLPTET) is for users who are able to speak and write in Malay language. The next target audience will be businesspersons, teenagers and virtually anyone of their field, age, gender as long as they know how to read and write in Malay language.

### 1.8 Mobile Technologists and Linguist

In mobile SMS technology, the mobile phone does not have to be active within the coverage range and the messages held for a number of days until the phone is active. SMS messages are sent and received (transmitted) within the same cell or to anyone have with roaming capability. This is because SMS is a store and forward service. There are many ways are available to send multiple short messages. In another words, short text messages are not send directly from sender to recipient but via a SMS Center. This means, each mobile telephone network that supports SMS has one or more messaging centers to handle and manage the short messages (GSM Association, 1999).

Predictive text entry algorithm significantly reduces the number of key strokes that need to be made to input a message. T9 by Tegic anticipates which word the user is trying to input in a message. The predictive text algorithm supports multiple languages including Malay language. The volume of usage in short messaging has increased with the contribution and introduction of standardized protocols in the SIM application toolkit and the Wireless Application Protocol (WAP).
However, the communicative options are limited to the nature of input device especially the mobile keypad. The set of characters physically found on the keypad will largely depend on the linguistic capacity to produce information by keystroke of the keypad (Crystal, 2001). The Malay language is a controlled language that has stable and well-established linguistic rules for Malay to put in practice. The government body named Dewan Bahasa dan Pustaka (DBP) and also known as Language and Literacy Agency of Malaysia has established and developed Standard Malay (SM). A new method developed to add Malay words in the MLPTET to enable the system to predict patterns for input text.

Most of the mobile phones use T9 technology, which is old, but it is a cool technology because it save a lot of time and it make look good. Predictive text entry objective is to make easier to type text messages. The reason to use predictive text is that it actually predicts in advance. Linguists encourages making local languages reasonable to use text messaging is one way of preventing from disappearing from face of Earth. This is one way to encourage to use a proper Malay language instead shortened the text messages where normally the words are not meaningful.

Based on the input technologies, a better predictive text will increase the number of input technologies that not in used. Predictive text will increase the number emails a user sends or posts to Facebook from their mobile.

The embedded Adaptxt software platforms like Symbian, windows mobiles, Linux are the key point technologies, which reinvents the predictive text. With these technologies, they will remove the inflexibility of language packs for the users who prefer to see the revenue for text entry. The Malay language predictive text entry
would make life easier. The predictive text entry is like Marmite; you either love it or hate it. There is no in between. Therefore, in the following chapters, the predictive text entry to be discussed in detailed.

1.9 Thesis Organizations

This research has been divided into eight chapters and each chapter discusses on specific areas. This research is organized in the following way where in Chapter 1 outlines the introduction of this research methodology. It starts by presenting the project definition, problem statement, statement of objectives, research significance, project scope and limitations, target user and thesis organization.

Chapter 2 outlines the literature review of a SMS mobile phone and the predictive text entry algorithm. This chapter also includes some information regarding existing text entry methodologies and the keypad design model. The review mostly discusses on the approach of implementing the dictionary repository for text prediction.

Chapter 3 contains the research methodologies, which describes the approaches and procedures used during this research in order to develop guidelines for the new text input prediction tool. Furthermore, the critical issues and shortcomings regarding the existing SMS text input prediction is outlined and discussed. This chapter also includes the techniques and methodologies to develop the mobile phone application, identify the target platform, and the core programming languages to develop the mobile phone application. Moreover, this chapter discusses the development strategy used to develop the tool.
Chapter 4 specifically contains the results of the proposed predictive text entry algorithm phase for the development of this tool. This chapter presents the issues and limitations of current SMS predictive text entry and proposes solution with features and concepts that the tool should incorporate.

Chapter 5 specifically contains the results of the requirements analysis phase for the development of this tool. In developing the requirements for the system, each new proposed features and concepts was examined, investigated separately and treated as a component. The requirements are elicited using use case diagram methodologies. This chapter is intended to form a foundation and basis for the design and implementation of the software product. It analyses the functional and non-functional requirements of predictive tool and Object Oriented analysis.

Chapter 6 discusses on the design phase of the new text input prediction tool based on the proposed predictive text entry algorithm and requirements elicited in Chapter 5 respectively. The design phase consists of the design activities that point up the structure, method and system specification that assure the functional requirements developed in the requirement phase.

Chapter 7 discusses the system environment engaged during the development, coding, system implementation and testing of the MLPTET. Then, it presents the selection of a tool-required implementation and execution of the tool are all discussed in detail. It presents system testing in general, the testing environment, methodology and process that are used during the testing phase. In addition, this chapter provides an assessment against the implementation.
Chapter 8 discusses system evaluation and presents the system evaluation results of MLPTET. The evaluation includes a tracing back to the requirements and objectives.

Chapter 9 presents a product to the work by explaining on problems encountered, solution applied and provides future system enhancement for the Malay Language Predictive Text Tool development.
CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Predictive Text Entry techniques endeavor to reduce text entry trouble by predicting what the user intends to enter. This is accomplished through analyzing great work of data collection of a corpus to establish the relative frequency of characters. Examples of such corpus include a diagram with words, sentences in the Malay language and a pair of characters that are used for statistical properties to predict letters or words as text is entered. Shannon (1951) published on the text prediction even though there are other ways to implement text prediction which will be discussed in the relevant session.

The Malay language has 26 characters, while the normal mobile keypad only has 8 to 13 available keys depends on the mobile phone manufacturer. As a result, there is an order to use the keypad for text entry. As shown in Figure 2.1, multiple letters must be assigned to the same key. For text entry, more than one key press is often needed to differentiate among the characters on each key.

Due to the appearance of mobile phone SMS messages, the importance of effective text entry on limited-size keyboards has increased. The entry of text in computer applications through Personal Computer (PC) is traditionally been carried out using a 102-key keyboard. These keyboards allow input characters in a completely unambiguous way using single keys or sometimes are key combinations. However, in the recent years, mobile phones have introduced a new demand for text entry methods. Mobile phones are usually minimal in size and weight. As a result, the keyboard button reduced to a minimal 12-button keyboard (MacKenzie et al., 2002).
The keypad consists of numerical keys 0-9 and two additional keys which is ‘*’ and ‘#’. The reasons to have these extra keys are to perform non-text entry functions for example navigating menus. The position of the space character may differ but it is normally placed on the zero keys. In Figure 2.1, the letters are arranged in alphabetical order from top to bottom and left to right using the keys 2-9. This design fits all 26 letters on only twelve keys which consist of three to four letters assigned to each key. As discussed the text input using the standard keypad is ambiguous (MacKenzie et al., 2002).

However, Multi-Tap method is the original text message technique for text entry on mobile phones and it requires the user to press key once or more in order to enter a character. So, segmentation problems occur when two characters which are mapped on the same key are entered repeatedly. According to the research conducted by Sirisena (2002), the most common solution to this problem is timeout. The timeout period elapses before the user enter another character from the same key. This is the typical method to wait a specific timeout period between the keystrokes sequence for the next character. Usually the period timeout duration is within milliseconds. Other alternative is by using the ‘Next’ button, before a timeout period expired, a button can be used to navigate through multi-page document to confirm the entry of a character and then
move to next character. The ‘Hold Down’, is like pressing a key repeatedly to sequence through the characters that are mapped on it.

The reduced keyboard button makes it hard for the user to enter text in an efficient way because user has to use multiple tapping or long key combinations to display and disambiguate the characters. The multiple tapping methods were the most universally implemented in mobile phone. To spare the user these elements of disturbance, a new class of text entry methods has appeared. It uses dictionaries repository in an attempt to resolve the word ambiguity and requires, in most cases, only one keystroke per character.

2.2 Short Message Service (SMS)

SMS stands for Short Message Service. This is one of technologies that enables the sending and receiving of message between mobile phone. The first short text message was believed to be appeared in Europe in 1992, over signalling channels of European GSM (Global System of Mobile Communication) standards from the beginning. The SMS and GSM standards were originally developed by ETSI (European Telecommunication Standards Institute). Since this successful trial, SMS usage has been the subject of tremendous growth (Jeff Brown, 2007).

In today’s market, there are many kinds of SMS application and many others which have been developed. There are some common applications which are available and will be discussed so that it gives some ideas of what can be performed with Short Message Service text messaging. This method is ubiquitous but limited. The most commonly used text messaging in Short Service Message application is a person-to-person text messaging that is simple and an important component for controlling the
total SMS traffic volumes. Besides, this is what the technology is originally designed for. Usually in this application, a mobile user types a text message using the keypad and inputs the recipients phone number then clicks on the button ‘Send’ or ‘OK’ to send the text message. Then when the recipient mobile phone receives the text message, it will be indicated by a ringing tone or a vibration.

The SMS is a basic service which allows the exchange of short text messages between subscribers (Brown, 2007). In wireless world, SMS has achieved a huge success with billions of text messages are sent daily. In mobile messaging technologies, many innovative applications are built on top of the Short Message Service technology. Apart from that, more are being developed by this technologies for example Short Message Service center and gateway. Short Message Service text messaging supports languages internationally and works fine with most of the languages like Chinese, Arabic, Tamil and many more supported by Unicode. In addition, Short Message Service messages can also carry binary data besides text. The main advantage of Short Message Service is because it is supported by GSM mobile phones.

2.2.1 Concatenated Short Message Service

SMS has limitation whereby one SMS message can only carry a very limited amount of data. The solution to this limitation is an extension is developed which is called concatenated Short Message Service. Actually, a concatenated SMS text message contains more than 160 Malay language characters. Basically, the sender’s mobile phone breaks down a long message into smaller parts and sends each of them as a single text messaging. When it reaches the destination, the receiver will automatically combine the broken parts back to one complete message.
2.2.2 Structure of a Message Segment

A message segment is associated with a number of parameters. These parameters indicate the message type, class, coding group, etc. In addition, parameters also contain the message content, which is provided by the subscriber, the content provider, or content, which is automatically generated by a machine.

A data structure where there will be an English character for each Malay Unicode character. There will be no difference between the English letter and Malay letter since there is no need for key mapping when user input the Malay letters. Figure 2.5 shows keypad for Malay text entry.

![Figure 2.2: Text Entry Techniques](image)

2.2.3 Text Compression and message submission

The text entry of a message is compressed during the transmission of the text message. However, in the market currently, none of the handsets available supports text compression. Therefore it is not advised to generate messages with compressed text. According to the research conducted by Gwenael (2005), a message with compressed text cannot be displayed properly by a Message Submission that does not support text decompression. In Short Message Service terms, submission refers to sending a message segment from the destination SME (Short Message Entity) to the destination SMSC (Short Message Service Center). Figure 2.6 shows interactions between an originator SME and an SMSC for the submission of a message.
2.3 Current Text Entry Methods

This section discusses the text entry methods which are currently in use and some methods which are still under development. All the mentioned methods use a keyboard with 12 buttons. As a measurement of the efficiency of the different text entry methods, we will use the number of Key strokes per Character also known as KSPC and it is straightforward to compute a value for the method by dividing the number of key presses required by the number of characters in the corpus (MacKenzie, 2002). A completely unambiguous keyboard enables a KSPC of one; text prediction methods may reduce this number even further. KSPC values for other methods are 1.1500 for LetterWise, 1.1141 for T9, and 2.0342 for Multi-Tap with the use of a timeout kill key. It should be noted, however that the KSPC is not the only parameter affecting the speed of the input. For example, if the layout is unfamiliar and confusing, the speed with the keystrokes are performed will be significantly lower thus affecting the final text entry speed.

2.3.1 Multi-Press Input Methods

The multi-press method requires more than one keystroke in order to enter a character. This method allows unambiguous typing of characters. It can be used independently or
as a fallback for systems which are using more complex text entry methods. The multiples press method is well suited to type words which are not available in the dictionary repository.

Methods based on repeated key pressed included Multi-Tap. The most common way to enter text on a mobile phone is by using the Multi-Press method (also called Multi-Tap) technique. In the keypad, each key represents more than one characters but accessing to the characters is performed using multiple key presses. For example, the letter ‘a’, ‘b’ and ‘c’ share the same key, the user presses it once to enter an ‘a’, twice to enter a ‘b’, and three times to enter a ‘c’. To enter the word dog, the user presses the sequence of keys “36664” (Haselgren, 2003). This method of entering is called Multi-Tap. However, this way of entering text is not efficient and user friendly. Moreover, there will be more error in typing and users might overlook on the intended characters due to over -pressing or pressing too slow which results to the time-out. When there are two consecutive characters of a word which share a same key, as for example the word “no” where both ‘n’ and ‘o’ are assigned to 6, a timeout algorithm is needed to determine when to stop shifting the letters and display a new character. This method results in a KSPC of 2.0342 if English text is entered (MacKenzie, 2002).

Based on a study by Fitt’s Law (1954) a mathematical model based for predicting the speed of text input method for a mobile phones and subsequently proposed by Silverberg et al. (2000), there are three types of predictions for multi press which include the next button, timeout and two key.

Basically when a user chooses to type an intended word there is possibilities for the user to use the same key on the keypad. Therefore, the timeout decides when a user is completing by pressing multiple times to retrieve the letters on the key. This is a
technique called **Multi-press with timeout** that uses a fixed timeout. When the timeouts starts? The timeout starts the moment the user presses a key and before the timeout expires approximately from 1 to 1.5 seconds, if the same key is pressed in cycle until the letters available in the key.

Next is the **Multi-press with next button** techniques which is replaced by the timeout with next button. In this case the user presses the next button to show that they have completed the cycle of the letters on that key instead of waiting for the following letters.

Finally is the Two-key technique which has a different approach compared to the previous two methods. How it functions? The first press indicates the intended key and the second key indicates the position of the key. Basically this technique uses only two presses to enter for every letter. However this technique is not suitable for special characters and this is because users intend to view all the letters mapped to each key on the keypad to determine the position of the key they intended. Based on their study, they compared their results with the modeled prediction by James and Reischel (2001) and proven that text entry rates for multi press is slower compared to predictions. Based on performance predictive model is considered to be the best among the real users comparative to multi-press.

### 2.3.2 Remapped Keyboard

On current mobile phone keyboards button, characters are assigned alphabetically to numerical key buttons. For instance, the most frequent character in English, ‘e’ is displayed using two taps. Remapped keyboards assign a single key to the most frequent characters. The remaining characters are grouped into sets that share a same key. This method decreases the KSPC because frequent characters can be entered with only one
keystroke. The program MessageEase (Saied, 2001) uses the idea of the remapped keyboard technique. MessageEase results in a $KSPC$ at 1.8210 (MacKenzie, 2002).

### 2.4 Predictive Text Entry Methods

Predictive text entry method allows overcoming the difficulties that are usually faced. The reason to present few predictive models is because a user presses only one single key per letter to perform the complete words. This means it will remove the multiple key presses and solve the problem of the conventional method in the mobile phone. However, there will be a large usage of word list leads to uncertainty and more unresolved ambiguities. Looking at it, there will be only some words which will be matched based on the combination of key presses and the key stroke sequence. Predictive text entry can suggest matched words to the user from the list. Hence, the frequency of word occurrence allow only the most common word will be suggested when multiple words match the keystrokes. Furthermore, this approach requires a big list of words to be stored for the language in the mobile phone.

With the predictive text entry method, the user presses one key per character and the program matches the key sequence to words in a dictionary (Haestrup, 2001). Even if several characters mapped to the same key, in many cases, only one word is possible given the sequence. This method makes it possible to reduce the $KSPC$ roughly.

This method uses the same keypad layout like multi-tap. It requires only one key press per letter. As the user enter the text by pressing only one key per character, the system will automatically compares the key sequence against words in a dictionary. The predictive text entry has a number of restrictions. Look at the layout mobile phone, there are several characters which are shared to the same key but in many cases only
one word is possible to be given in the sequence. If the key sequence corresponds to two or more words, the user can browse through the resulting word list and choose the word user intended to write.

For example, the user intended to enter the word *come*, by first pressing 2 at the mobile keypad. The program will then propose the word ‘a’ because it matches the entered sequence. When the user presses 6, 6, and 3, the program might propose the words *an*, *con* and finally *come*. The words *bone*, *bond*, and *anod* (and some more), also fit the given sequence. The user can access these words by pressing a next key. However, problem arises when multiple words match the same key. In such situation, the user can access these words by pressing a next-key or (*). The words that are not present in the dictionary need to be handled. In this situation, such words are assumed and the user enters the word using the multi-tap technique and then adding to the dictionary.

Although predictive text entry method has significant advantages over the traditional text entry method (multi-tap), it has certain drawbacks in terms of usability. Basically, when the output of predictive method and traditional method are analyzed, the users will not know which letter will appear until the key is pressed. Many new mobile phones use this method. The most widely used implementation is T9 by Tegic (Tegic Communication, 2007). Other implementations are eZiText by Zi Corporation (Zi Corporation, 2002) and iTAP by Motorola (Lexicus Division, 2002). Most implementations only match words with the same length as the key sequence, resulting in a *KSPC* (Key strokes per Character) of slightly greater than one when the user types words that are available in the dictionary.
2.4.1 WordWise

WordWise developed by Eatoni Ergonomics uses an auxiliary key (Eatoni, 2009), attempt to reduce drawback of the disambiguation by allowing entering one of the letter for each key (‘b’, ‘e’, ‘h’, ‘k’, ‘n’, ‘r’, ‘v’ and ‘z’) ambiguously through the use of an auxiliary key. A character on a key is selected explicitly by concurrently pressing the key corresponding to the character and the auxiliary key indicating the position of the character on the key. This reduces the number of matching words for a key sequence considerably because the user explicitly disambiguates some characters in the sequence. A drawback is that two keys must press concurrently. With a limited space keyboard, this can be difficult to some users.

2.4.2 LetterWise

LetterWise also by Eatoni Ergonomics, is a different approach, which eliminates the need for a large dictionary (MacKenzie et al., 2001). It works by guessing the most probable next letter based on what the users entered in the beginning and it only considers the letter diagram probabilities. In English, the letter ‘t’ is often followed by ‘h’ and hardly ever by ‘g’. The program selects the most probable letter knowing the previous one. The user can browse and change the characters by pressing the ‘Next’ key.

The LetterWise method which has a KSPC of 1.1500 (MacKenzie, 2002) uses information on the frequency of letter occurrence. Based on what has been written so far, the system retrieves the letter on highest occurrence that matches current key press. One of its main advantages is, it only needs small amount of memory. Another advantage is the fact that it is just as easy to enter words, which are not in the dictionary.
Therefore, this could be a suitable fallback method to produce faster text input instead of the multi-tap methods.

2.4.3 Predictive Word Completion

In order to improve the predictive text entry method, the predictive word completion is proposed. This method attempts to complete the current word based on the previous word. Thus, when a user starts typing a word or combination of letters the most likely word given to the current word could be proposed for auto-completion.

Recently, numbers of mobile input interfaces include a word completion facility, in which a partially completed word can become fully completed word as intended by user. iTAP technology offers the word completion, in which after entering a few letters, it offers several complete word from its main dictionary. For example, if a user enters letters with ‘mo’, iTAP suggests moving, move, moved and mood. Thus, it also has the facility to turn on or off the dictionary. Besides that ZiCorp corporation also uses word completion in its input method namely eZiText.

To examine current word prediction that uses the previous word that was entered by the user because the user has so far entered only partial key sequence. Thus, we can try to visualize the full word that user intended based on the partial key sequence entered and the previous word. The language modeling is by combining the letter n-gram model with a bigram word model to make a prediction. The n-gram uses the previous N-1 words to predict next one and on the other hand the bigram model, where N=2, uses previous word to predict the following word.
The probability of the current word given the previous word and input for the current word is calculated using the formula shown in Figure 2.4 by MacKenzie (2002).

\[
P(\text{currWord} | \text{prevWord} & \text{KeySeq}) = \frac{C(\text{prevWord}, \text{currWord})}{\sum \limits_{\text{we words (keyseq)}} C(\text{prevWord}, w)}
\]

**Figure 2.4: Equation (MacKenzie, 2002)**

### 2.4.4 T9

What does T9 stands for? T9 stands for “Text on 9 keys.” It is a similar to predictive text technology. Predictive text is almost flawless in the way it works; this is because it retrieves the word from a dictionary. This is to prevent user from tapping the same key on the mobile phone 4 times to get the letter ‘S’. The objective of T9 is to make it easier for the user to type text messages. T9 software is developed by Tegic Communications. After each word has been entered, by single key for each letter then it combines the group of letters and will look up for the most likely word to be used in the dictionary. The rest of the features are similar to predictive text entry.

### 2.4.5 Keystroke Level Modeling (KLM)

According to the research conducted by Crossan (2000) use Keystroke Level Model (KLM) were developed for three entry methods, this is to predict the execution time of a task and the keystroke action performed by the. The sum together the time required by each action. Basically, this is to gain the potential performance of the text entry methods. The reason to propose a further extension for word completion is because to speed-up the text entry. The estimation of average time is by modeling interaction down to individual keystroke using the operations K, H and M for Key press, Homing time for the hand to move to the keyboard and mental preparation time for executing physical actions. In this equation, it uses the estimated time given by KLM for each operation.
The efficiency of the text entry methods used based on the keystroke model and time model. The keystroke model is a good indicator for efficiency and the time the model gives accurate measurement efficiency. The KLM for mobile phone user describes different types of actions that user performs when doing the text entry in a mobile.

2.5 Predictive Text Problems

Predictive text is an input technology which is most commonly used on mobile phones, and for any convenience accessibility. The technology allows some general words to be entered by a key press for each letter, as different to the multiple key press approach used in the older age of mobile phones. The objective is to ease the process of typing text messages, e-mail, calendar, and entries into mobile phone address book. The number of keystrokes per character, on average, is comparable to using a QWERTY keyboard, provided that all words used which include jargon, proper nouns, abbreviations, Uniform Resource Locator (urls), foreign-language words and so on are in the dictionary, provided that no spelling mistakes or typing error and punctuation is ignored.

According to the research conducted by Starner (2004), the typing rate of today’s text entry methods using the mobile phone’s keypad two to five times slower than that expected with QWERTY keyboard. The fastest mobile text messaging “text” entered 29 words per minute (Stanner, 2004). Problem with predictive text where the typing application was not widely used was found by Grinter and Eldridge and they point out that “predictive typing can interfere with an experts knowledge of the interface (Grinter and Eldridge, 2001). User must take a leap faith when entering words into dictionary-based system, (Kobel et al. 2001). As an example, reader was asked to consider the word “golf”.
Table 2.1: Predictive Dictionary Failures (Kobel et al. 2001)

<table>
<thead>
<tr>
<th>Keystroke</th>
<th>Display</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>I</td>
<td>Wrong first letter</td>
</tr>
<tr>
<td>6</td>
<td>in</td>
<td>Still wrong</td>
</tr>
<tr>
<td>5</td>
<td>Ink</td>
<td>Still wrong</td>
</tr>
<tr>
<td>3</td>
<td>Hold</td>
<td>Wrong word appear</td>
</tr>
<tr>
<td>*</td>
<td>Hole</td>
<td>Adjust- wrong word</td>
</tr>
<tr>
<td>*</td>
<td>Gold</td>
<td>Adjust- wrong word</td>
</tr>
<tr>
<td>*</td>
<td>Golf</td>
<td>Adjust- correct word</td>
</tr>
<tr>
<td>0</td>
<td>Golf</td>
<td>Accept word</td>
</tr>
</tbody>
</table>

He mentions that perceptual and cognitive processes are clear as work by user consider the system to response to each keystroke.

2.5.1 Language feature used in predictive text

Predictive text is called as one key to use the languages is the availability of technology which reduces the number of key taps necessary to create a word when using a limited keypad. If the prediction is good then, predictive text entry technologies have the ability to reduce the effort required to enter text (Soukoreff et al., 2002). Then, in order to make prediction good, Mackenzie and Soukoreff points out a few requirements to be considered based on language model (Soukoreff et al., 2002). These standards are the corpus that is not necessarily representative for user language, the process of editing process for corpus is not reflected and there input modalities of input devices is not reflected in the existing corpus. The key combination of the traditional written texts for both formal and informal purpose in T9 causing worries among users. Moreover, the predictive text input system does not accept slang or dialect. In multi-tap technique there many possibility to type in new words and input them into the memory. A question was raised, what effect predictive text entry will have on the development of the “Finnish” language, or which influence it will have on the personal style of their users. Is it the device or the user that will adapt? (Kasesniemi and Rautiainen, 2002).
Similar question are raised on how the development of prediction text entry is applied for the Malay language.

2.6 Competitors Comparison

Nokia and Motorola are the top two manufacturers of mobile phones in the world. Both have increased respective market share in the mobile phone. This is based on a research by Gartner Inc. (2008) which reveals the numbers. In the early month of 2008, Motorola achieved the mobile industry worldwide market share of 84% compared to year 2007 dropped to 13% while Nokia market share hits 40%. Both are making aggressive improvement on their performances in the mobile phone market. In the mobile industry market share article published on 8 November, 2008 announced that the Samsung overtakes the Motorola market in US mobile phones. In few months, Samsung reached to the market share of 22.4% compared to Nokia 21.1% and 20.5 LG Electronics.

The Table 2.2 shows an overview of the market share and in-built predictive text entry for mobile phones sales in the fourth quarter of 2008. Nokia being in the first ranking shown in Table 2.3 is the mobile industry and subsequently Samsung finished with a good performance and LG jump two places to third and Sony Ericsson dropped to fourth and Motorola slipped to fifth place in worldwide ranking.

Predictive text is developed and marketed in a variety of competing products and the most widely used systems of predictive text dictionaries are T9 and iTap. Tegic Communication’s T9 is the market leader and other products include Motorola’s iTap. Each mobile requires a linguistic database for every input language to be supported. T9 is a multilingual rich text rendering engine and text editor that can help people to communicate in many languages. There are over 70 languages in market. Some mobile
manufacturers implement permanent user database, while others implement one when it needed. The competing Eatoni products do not use a dictionary. Based on the existing technology and features it used as a guidance to develop a tool in the Malay language. Many mobile manufacturers modify the features of phones to be different from the competitors, usually at the accuracy and usability of the device.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Competitors</th>
<th>Worldwide Market Share 2008 %</th>
<th>Predictive Text Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Nokia</td>
<td>39.1</td>
<td>T9</td>
</tr>
<tr>
<td>2</td>
<td>Samsung</td>
<td>14.4</td>
<td>T9</td>
</tr>
<tr>
<td>3</td>
<td>LG</td>
<td>8</td>
<td>T9</td>
</tr>
<tr>
<td>4</td>
<td>Sony Ericsson</td>
<td>7.5</td>
<td>T9</td>
</tr>
<tr>
<td>5</td>
<td>Motorola</td>
<td>10.2</td>
<td>iTap</td>
</tr>
<tr>
<td>6</td>
<td>Other</td>
<td>20.8</td>
<td>SureType, Eatoni</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

2.7 Predictive Text Entry Techniques

There are a variety of techniques have been proposed for facilitating query entry using the mobile text entry by Soukoreff (2002). The most popular techniques are to use language prediction. The idea to use language prediction is to input language and automatically suggest what the user might be entering based on what the user has input thus far. Additionally, language prediction tools have the potential feature to facilitate text input with a good predictions.

Another important aspect is in the designing of text entry. For mobile phone interface, design plays a main role to reduce the movement of the finger to faster typing speeds (Soukoreff, 2002). In order to minimize the movement there are two common mathematical models.
2.7.1 Fitts’s Law Model

This model is of human movement, predicting the time required to move from target to destination area. This function measures the distance and size of the target. Fitts’ law is used to model the act of pointing (Fitts, 1954).

\[ MT = a + b \log_2 \left( \frac{2A}{W} \right) \]

**Figure 2.5: Equation (Fitts Law, 1954)**

Means:
- \( MT \) is average time taken to complete the movement
- \( a, b \) regression coefficients
- \( A \) is distance of movement from start to target
- \( W \) is width target measured on the axis motion and allowed user error tolerance and final point of the motion fall within \( W/2 \) of target.

The equation shown in Figure 2.5 measures the speed-accuracy associated with pointing, with smaller target whereby requires more time to acquire (MacKenzie, 1992).

2.7.2 Hick Hyman’s Law

This model describes the human reaction time taken by a user to make decision within the given choices as a Human Computer Interaction under uncertainty. The decision time (T) increases with uncertainty about the judgment or decision to be made.

Given \( n \) equally probable choices, the average reaction time required to choose (Hick, 1951). The following are the computation:

- \( T = kH \), where \( H \) is the entropy of the decision and \( k \) is a constant.
- \( H = \sum_{i=1}^{n} p_i \log_2 (1/ p_i +1) \)
  - \( p_i \) is the probability of making i-th choice
- \( H = \log^2(n+1) \), if probabilities are equal \( n = \) total number of choices)

2.8 Methods

The current study sample consisted of 150 participants. From the 150 sets of questionnaire total of 101 sets responded by the entire range of age groups. First, a
group of teenagers, 10 were selected with relatively who uses SMS frequently to send
text messaging to take part in the MLPTET. The participants was observed by the
researchers between periods of movement and the time of resting, walking and standing
and individual interviews were conducted. A survey questionnaire was conducted on
the target audience to collection information on the predictive text entry, were assisted
by research assistants by using email. This method permits the collection of the most
extensive data on each person questioned (Crossan, 2000). Prior to this, a briefing on
the objective of the research study was given. The willingness of the participants in the
study was asked before preceding using a survey questionnaire. A token of appreciation
was given to the participants upon completion of the questionnaire.

The research method to develop the Malay Language Predictive Text Entry Tool
(MLPTET) consists of conducting interview and distributing questionnaires to elicit
users’ requirements. The system was developed using object-oriented approach. The
questionnaire forms part of the process of the evaluating the MLPTET. A questionnaire
survey was carried out to evaluate the functions and features in the MLPTET.

All the analyzed material and findings are combined together to form and structure a
proposed predictive text entry algorithm against the weaknesses of the current SMS text
input. For each of these issues, the solutions, features and new concepts have been
investigated to address the weaknesses of the current SMS text input. Then, carried out
the findings of the literature review with existing algorithm that can be utilized for
Malay language predictive text entry. The result of the findings of the literature review
formulates a conceptual diagram to illustrate how predictive text entry algorithm for
search and Malay language structure are link together based on the development process
of the MLPTET.
The available technology of text input prediction are studied and analyzed expansively to suit with the development of the new text input prediction tool. Furthermore, ideas related to the materials have been extracted and analyzed to generate new ideas and thoughts. Formulas, standards and procedures are specifically identified. All the analyzed material and findings as stated are collected to form and structure a proposed predictive text entry algorithm against the existing system weaknesses. For each of the issues, the solutions, features and new concepts have been investigated to overcome the weaknesses of the existing technologies. These newly proposed solutions become the features that the text prediction tool should be incorporated with. Then, the new and modified models, formulas, and algorithm are to be developed based on a component basis.

These newly proposed solutions become the features that the prediction tool should be incorporate with. Each module will represent the information model to make the development tasks easier. The dependencies among the modules are also recognized and highlighted.

A survey was conducted by Tanaka (2001), on the usefulness of the predictive text entry program for users having severe spelling problems and helped in the key saving aspects to speed up in the creation of text messaging. With this, Malay predictive text entry produce written work with the substantial reduction in the number if spelling errors. This provided great motivation for the user commented on the improvement the language and vocabulary development.

This MLPTET will dramatically reduce the time it takes to complete text entry on the 12 button keypad and most important spell words accurately. This tool includes two
predictive text input system, MLPTET for intelligent multi-tap entry and MLPTET for prediction.

The MLPTET have all combined in one package, with features of an unique functionality such as word prediction, learning and personalization to provide user with simpler and faster input over the standard multi-tap entry. This tool developed to provide users with intuitive text input that are familiar, east to learn and fast to use. MLPTET also increases the literacy level by the use of linguistic prediction.

According to the research conducted by MacKenzie and Soukoreff (2002) who argue that the most two popular approaches for prediction text entry are the movement minimization and language prediction. Movement minimization refers to how to reduce the movements of the finger in interacting with mobile phones to enter text and the language predictions utilize the nature of language to predict the user’s intended letters or words. Apart from this there is also a method called hybrid approach. This is one of the ways to reduce the input by predicting on what the user is entering. In these moments, the system maps key presses against an embedded database of language. Therefore, the system analyzes a large collection of documents to establish the predictions system in Malay language. This information is used to predict characters or words as text is entered by the user. There is another advantage to this method that is if a required word is not in the database, it can be added using “multi-tap” in the sequence of key presses in the dictionary stored in the memory of the phone device.

Summarized methods is shown Table 2.3:

T9 by Tegic Communication, the dictionary based predictive text entry method available from AOL only require one key press for each character text entry. Research has been conducted for improving the performance of disambiguation algorithms. The
dictionary based text entry method was developed by Masui (1999). Later Rau and Skiena (1994) enhanced dictionary based disambiguation by matching unknown words then subsequently Hasselgreen, Montnemery and Svenson (2003) further their investigation on this disambiguation algorithm which uses the matching words technique. The LetterWise that uses a stored dictionary to retrieve the intended letter based on key press was developed by MacKenzie et al. (2001).

Table 2.3: Summarized Methods (MacKenzie et al., 2001)

<table>
<thead>
<tr>
<th>Predictive Method</th>
<th>Key strokes per Character</th>
<th>Use of Next Key</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>T9</td>
<td>1.0072</td>
<td>Required</td>
<td>Dictionary based</td>
</tr>
<tr>
<td>T12</td>
<td>Varies</td>
<td>Required</td>
<td>String matching</td>
</tr>
<tr>
<td>iTAP</td>
<td>No Formal Evaluation</td>
<td>No Formal</td>
<td>Prediction and word</td>
</tr>
<tr>
<td>eZiText</td>
<td>No Formal Evaluation</td>
<td>No Formal</td>
<td>Prediction and word</td>
</tr>
<tr>
<td>Letterwise</td>
<td>1.150</td>
<td>Not required</td>
<td>Uses a database of</td>
</tr>
<tr>
<td>WordWise</td>
<td>No Formal Evaluation</td>
<td>No Formal</td>
<td>Words can be entered</td>
</tr>
<tr>
<td>POBox</td>
<td>No Formal Evaluation</td>
<td>No Formal</td>
<td>Word completion</td>
</tr>
</tbody>
</table>

2.9 Result

In brief, the following result were taken into consideration into the analysis of the MLPTET, they are as follows:

- Through the theoretical analysis, it is shown to the real world situations that it is common for typing errors, lack in the speed of text entry using the multi press method and dictionaries are incomplete.

- The linguistics text entry methods however have speed the text entry but even if dictionary words are correctly entered with a just a few key presses, a greater problem is created in the practice. This is because it is necessary to allow entry of non dictionary words in which common used words with proper names, slang, abbreviations and professional terms.
To measure the query rate and lookup error rate.

To have the predictive text entry algorithm and this solution enables users to enter any letter sequences with a speed typing.

2.10 Algorithm for Prediction

Algorithm play a central role in predictive performance. The predictive text entry is the process of adoption of an intelligent algorithm. From the in built dictionary, the sequence of characters is compared with possible letter combinations. Thus, whenever the algorithm predicts a similarity usually when user enter the intended words.

When the algorithm written in Java, method that takes a string which contains a set of words (a sentence). Then it rotates those words X number of places to the right. When a word in the last position of the sentence is rotated it should show up at the front of the resulting string. Basically, understanding and choose the suitable data structure. Then identify the order of the algorithm.

For example given array of Strings of length 3. One of the string in the array is marked as Start string and another one as End string. In order to resolve the data structure problems, by converting the Start string to end string, given the condition that the intermediate string which you will make should differ from its previous string by only one character and the string should be present in the input array. Predictive text entry algorithm is similar to a spell checker. An algorithm searches the dictionary for a list of possible words that match the keypress combination. This method used in the mobile phone to complete words for creating Short Message Service. Moreover, each letter key on the mobile is mapped to a set of three to four letters. Given a sequence of key
strokes, a simple algorithm is to generate all combinations of letters form the corresponding types. Thus, the suggestion is only on that combination that exits in a dictionary of words. Figure 2.6 shows the predictive editor algorithm flowchart. This is a pseudocode converted to a flow chart.

![Figure 2.6: Predictive editor Flowchart](image)

For example, keypad number 2 consists of (‘A’, ‘B’, “C’) and 3 consist of (‘D’,’E’,’F’). Sequence ‘24’ can be any of **ag, ah, ai, bg, bh, bi, cg, ch, ci**. The dictionary will have “abang”, “cakap” and “guna”. The words are in alphabetical order and in the order of most frequently user, last used. Figure 2.7 shows the sample of a real predictive text entry on the mobile phone device screen.
There are two main functions in this approach. `getString` returns a list of alternative for a given word. It also requires a dictionary words and the type’s structure of keypad. It is a recursive function where all action happens. For example, assume the word has 3 letters, where the first two letters have been filled and now user is intending the 3rd letter. The way the algorithm pick is each letter in the current block and append it to generate string and use as accumulator variable. However, the drawback of this method is all the invalid or unmatched words that are generated only to be discarded. For a word length $n$, if there is an average $m$ option for each letter, then the number of words generated is $m$ to the power $n$. This is the time complexity of this program. Unless if the dictionary were already sorted into different types according to the key sequences. This is possible because mobile phone can come loaded with such a data structure that facilitates the quick lookup. In this case, for a given sequence, mainly have look up the data structure for all matching words and return one or more.
Hence, the suggested function that implements this improved the algorithm.

<table>
<thead>
<tr>
<th>Pseudocode:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display selection menu</td>
</tr>
<tr>
<td>Read a list (A) of first/last/word search selection</td>
</tr>
<tr>
<td>Prompt for result browse command</td>
</tr>
<tr>
<td>For A in loop</td>
</tr>
<tr>
<td>If f is not null then</td>
</tr>
<tr>
<td>Search first name</td>
</tr>
<tr>
<td>Display word length in string</td>
</tr>
<tr>
<td>Display result</td>
</tr>
<tr>
<td>Elsif l is not null then</td>
</tr>
<tr>
<td>Search last name</td>
</tr>
<tr>
<td>Display word length in string</td>
</tr>
<tr>
<td>Display result</td>
</tr>
<tr>
<td>Endif</td>
</tr>
<tr>
<td>End Loop</td>
</tr>
</tbody>
</table>

**Figure 2.8: The pseudocode for the algorithm**

The dictionary has 2 steps where each character in the keypad is mapped types number. For each word in the dictionary, it concatenates the list of each character in the word. There if O (number of character in the letter) to generate the key. Hence dictionary has n words of average O (m * n ).

### 2.11 Malay Language and inherent issues

Apart from the algorithm, this research also focus on the Malay Language to find and design the best database for the Malay Language Predictive Text Entry. The Malay linguistic background will be discussed. The research does require the implementation of the whole library in the dictionary module. This is also called as database. The Malay dictionary can contain more than 1000 words most common used Malay words and almost 81% accuracy after correcting of the mismatch rules. Malay language is a language that has been used by countries in ASEAN as their national language such as Malaysia, Indonesia and Brunei (Asmah, 1983).
The basic knowledge of Malay language linguistics such as Malay word structure will be proceed for discussion. Thus, it is also important to note that the words in the dictionary are taken from Dewan Bahasa and Pustaka. Standard Malay is written in 26 Latin alphabets consists of six vowels, nineteen primary consonant, native consonant sounds and eight secondary consonants (consonants borrowed from other languages) This implies that the spelling, words, phrasing, grammar, pronunciation, punctuation, sentences, abbreviations, acronyms, capital letters, numbering and style of the language are already standardized. In this research, it is important to emphasize that the output of predictive text entry system will only in standard Malay.

Malay morphology is defined as a study of word structures in Malay language. The morphology function assists user to search translation of words entered in different grammatical forms then it retrieves the corresponding word from the dictionary with of the best matched word. A morpheme is the smallest unit in a language that is meaningful (Jurafsky, 2000). A Malay word can be comprised of one of more morphemes.

The national language of Malaysia is Malay Language but however English is language being used commonly for text messaging. The motivation to develop with MLPTET is due to lack of Malay language mobile dictionary in market compared to the amount mobile dictionaries available in other world languages such as English, Chinese-Mandarin and Japanese. By developing the MLPTET is excepted to meet the needs of wide range of users such as language learner, native speakers and all professional in Malaysia. MLPTET can offer a potential benefits for users where it is quick and easy to use.
2.11. 1 The problems to be solved

- The first problem arises when multiple words match with the same key sequence therefore the linguistics disambiguation is not perfect
- There a higher tendency to use abbreviations, emoticons and dialects in their messages.
- This type of message mostly written by teenage users are very informal and normally exchanged among friends as form of keeping in touch unlike the older users prefers to use text in a more formal manner.
- Evolving SMS language becomes tedious especially using various abbreviations and symbols being able to text messaging

2.11. 2 Predicting Word Comparison between English and Malay

- Malay does not have a grammatical subject in the sense that English does. Such as the noun comes before the verb. The vowels in Malay language is different from English Language. There are five front vowels in English compared to three only in Malay.

2.12 Summary of the Chapter

From the observation, there are predictive text entry methods proposed to improve the algorithm of prediction of text entry on mobile system for SMS using input technology. However, user who uses Multi-Tap or Multi-Press mode had to type a maximum of three times per character. Therefore the user’s speed and efficiency decreased to a greater extent compared to a QWERTY user.
This research begins with a discussion on the difficulties in applying traditional text entry method for Malay language. The proposed a predictive text entry with word completion method based on a careful analysis on the structure of current T9 system. The reason to choose T9 system is because it allows users to enter text quickly and easily using the one key press per letter. This patented technology enables faster, simpler and a leading choice for text entry on today’s sophisticated mobile devices. This system is twice faster than the traditional Multi-Tap entry (non-predictive text input).

As opposed to the non-predictive text input method used on most mobile phones, users only have to press the number key once with T9 method that has a built in dictionary, and it will predict or guess which words from the sequence of the keys pressed. In this way, the user will avoid to make entry laborious and time consuming. This technology can support global languages including the complex scripts like Chinese and Japanese with low system requirement.

However, the importance of having an input text prediction still remains as an issue that requires improvement with the incorporation of the new algorithm to improve the speed of the text input. The algorithm is on the sequential index search. The sequential searching technique searches through data in sequential order and has to pass the n-1 data to form the word completion. In this way, user key presses only once for each letter to build the words sequentially. As and when user key presses to add letters in word, he algorithm searches sequentially moving from one possibility to another in added word list.

2.13 Conclusion

The organization of this study was to improve the predictive text entry algorithm for the Malay language, which is the most common method used to type text SMS messages. The proposed improved approach for Malay language message to make the
predictive text entry method efficient, faster and easier to use. After evaluating the proposed keypad, the results have showed good improvements in the predictive text entry method by reducing the number of keypress required to type Malay text message. This chapter provides a review of related literature and other research carried out in this field.

In the next Chapter 3, the research methodology of MLPTET will be discussed in detailed. The comparison of the existing tool and survey will be outlined.
CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

There are several approaches have been adopted in this research in order to develop the predictive tool. Firstly, a literature review is conducted which focuses on three main areas in literature review. Basically, the best practice in this project, literature review and questionnaires, observation and methods were used to gather and analyses data. The first area presents the SMS and its definition, the available models for it and existing tools that support it.

Next is on the Malay structure support in the predictive text entry which to be discussed comprehensively. Finally the predictive text entry definition, features, characteristics, support for the Short Message Service are all discussed along with some examples of the existing tools which support the algorithm for search discussions.

In order to develop the Malay Language Predictive Text Entry Tool (MLPTET), an Object-Oriented approach has been adopted in conjunction with Iterative development methodology. The object-oriented approach combines data and methods. In the system development methodology, object-oriented techniques are used while questionnaire was used in the system evaluation. Figure 3.1 illustrates the flow of research methodology of this research.

As shown in Figure 3.1, the research methodology consist of literature review, observation, questionnaires and methods. Then, the researcher proceeds to the system development which comprises of the proposed predictive text entry algorithm,
requirement analysis, design, implementation and testing. Moreover there are also some major steps such as the study of the mechanism for the prediction techniques and the comparison between methods and the existing techniques. The evaluation, end product and outcome are described in detailed.

Figure 3.1: Research Methodology Process Flow

Figure 3.2: Iterative development methodology model
The MLPTET is developed based on Iterative development methodology. The iterative development methodology is an approach to developing the system can be repeated in loops and can be cycled an unlimited number of times before completing the required task. Figure 3.2 shows that each iteration is considered a mini project by itself that is contain of activities such as analysis, design, implementation and testing (Bunea, 2003). In the following, the Iterative development model of MLPTET is described. The cycle starts from the system analysis until system testing. Any modification at any phase results in a new iteration.

3.2 Research Methodology Description

3.2.1 Literature Review

Prior to this research study, literature review is performed based on some subjects related to the SMS text entry method and text input prediction technique. Next, researcher reviewed on the current approach on implementation of the dictionary to predict the text entry. Typically, the review focuses comprehensively on the text input procedure, its relationship with keypad, the mobile keypad design that influences the word entry method, and how the dictionary algorithm functions to predict the text. The text input prediction approach that is currently available is for review. This information is gathered using online search engines via the internet for journal and articles that are related to SMS.

In the Literature Review, different types of text entry method such as Predictive Text Entry, Word Completion and LetterWise are analyzed. According to researches, some subjects related to the text input prediction aspects are initially reviewed. The text prediction support tools currently available are reviewed as well. Moreover, the findings
cover up the importance of kind of language and input method available. The current technologies used for developing the SMS are investigated. Thus, it is selected to be used in the development of the tool. Further the justification is provided in Chapter 1 and Chapter 6.

All the information above is collected using on-line search engine via the internet. Online databases namely:

- **Scientific Literature Digital Library and Search Engine** at http://citeseerx.ist.psu.edu/. The appropriate published Journals, Proceedings and Newsletter, are reviewed.


- Besides, the information is also obtained from Nokia forum (http://www.forum.nokia.com/), J2ME books and mobile development books.

A background study was also conducted and some of the important information gathered are as listed:

- Proposed method predictive text entry
- How Short Message Service (SMS) performed
- Language used
- The problem in predictive text entry
3.2.1.1 Survey

In this research, survey questionnaire were used to gather information for both analysis and system evaluation which will be discussed in details. The goal of survey is to adaptive to innovative text entry.

- Distribution of Survey Questionnaire

The questionnaires were distributed to the public in Malaysia including students, working adults, friends to gather information on the user’s preferences and requirements for the development of MLPTET. A total of one hundred fifty (150) sets of questionnaires were distributed through electronic mail and by hand. Out of the one hundred and fifty sets of questionnaires distributed, only hundred and one (101) sets of questionnaire were returned, giving a response rate of 67%. A set of the questionnaire can be found in Appendix D.

The text input method surveys. There are some surveys and overview of the text entry methods that is turned up using Google. This text entry is a web page with a survey of the text entry methods by John Williamson (2005).

3.2.1.2 Observation

According to the research conducted by Roger’s (1995), observation is used as one of successful techniques of requirement elicitation. For this research, direct observation and face-to-face communication with participants has given support to gather the requirements to develop the MLPTET.

3.2.1.3 Methods

There are many options for text messaging but this demonstration is particularly used for the predictive text entry that uses the Malay language dictionary that is a built-in
application in the mobile devices. There are several types of software which have been
developed for predictive text entry. However, overall aim of prediction systems is to
reduce time and effort spent on entering text especially, through a number of different
methods and techniques.

3.2.2 The Proposed Predictive Text Entry Algorithm

This phase specifically contains the results of the analysis phase for the development of
this tool. This phase presents analysis of the MLPTET. In this phase, an analysis
performed to identify the issues from the findings that are helping to achieve realistic
goals set by them.

In the proposed predictive text entry algorithm process phase, an object-oriented
analysis is used to increase the understanding of the problem domain that is to support
the needs to develop a Malay language Predictive Text Entry tool by identifying the
current issues and limitations of SMS predictive text entry.

In this section that follow, comparison of the existing tool was conducted to propose
solution with features and concepts that the tool should incorporate.

3.2.3 Requirement Analysis

This phase presents the requirements Analysis. Since each case can be examined
separately, we decided that use cases will be used to capture the system requirement of
these tools. For each component of the system, the use cases will be converted into a list
of requirements for that component to be implemented successfully.
In this phase, it analyses the functional and non-functional requirement of predictive tool. The first important thing to be performed is to understand what the system is supposed to do. One convenient way of determining the functional requirements for a system is to identify its use cases. In this chapter, functional specification and use case diagram are produced to model the functionalities of the system. There are several advantages of viewing a system in terms of its use cases.

The reasons are as follows:

Since there are few connections from one use case to another, the use case can be examined separately. It could be understood without having to know all the details of the larger system. Particularly, a user can understand the proposed functionality without having to learn about the functionality of other use cases. The use cases can be used as the basis for estimating how much time and effort will be needed to design and code the system. The system development can be tracked in terms of use cases. That is to follow the progress of each use case as it is designed, coded and tested.

Since use cases can be used to identify the requirements of a system and since each use case can be examined separately, it has been decided that use cases will be used to capture the system requirement of prediction tool. For each module of the system, the use cases will be converted into a list of requirements for that module to be implemented successfully.

### 3.2.4 System Design

In this research, the conceptual design gives attention to the system functions while the technical design expresses the form the system which includes the hardware and software aspects.
This phase describes on the architectural design and data design. The design phase consists of overall design activities that point up the structure, method and system specification that assure the functional requirements developed in the requirement phase. Unified Modeling Language presents the interaction of components. Then, the sequence diagrams are used to show the detailed of the MLPTET. The sequence diagrams are used to show dynamic behavioral aspect of MLPTET. Finally the user interface design is illustrated in details. The design phase of the MLPTET is illustrated in Chapter 6.

3.2.5 System Implementation and Testing

System implementation will elaborate the constructed MLPTET design and translate it into an interface that can be used on a mobile phone with predictive text entry in Malay language which includes coding.

The implementation and testing phase describes the implementation of the complete tool. The testing is carried out throughout the implementation process. Testing is performed to measure the functionality, accuracy and usability of the tool.

System implementation which involves requirements and design then will be converted into a workable program codes. The implementation and coding of the prediction tool was performed purely in Java programming using the J2ME (Java 2 Platform, Micro Edition Enterprise). The reason to develop this tool is Java because can write codes and have it run on small devices. It improves the ability of mobile phones to support applications vastly. There is some sample code which is provided in Appendix C.
The testing general has three main types which are Unit testing, integration testing, and system testing which are carried out to ensure that the system is error-free and it will fulfill the user requirements. Testing is conducted to compare MLPTET with the existing ones in terms of usability and accuracy. Testing is then carried out to observe user’s evaluation for the text entry in terms of speed and easiness. As a result, this will determine whether the predictive text entry algorithm in Malay language can be used for short message service text messaging. System Testing will be elaborated heavily in Chapter 7. After the source code is developed, the software must be tested to detect the program code error. Then, testing is conducted by performing a validation of the implementation, to show if it complies with the original requirements, specifications and design.

3.2.6 System Evaluation

The evaluation phase conducted based on the respondents. System evaluation is a necessary step in the system development process to measure the successfulness of the tool. A system evaluation was conducted to assess the users’ satisfaction towards the tool and for decision support system. The evaluation is important to indicate the deficiencies and improve the tool in future. Questionnaires are the inquiry methods widely used by researchers to gain quantitative data (Folmer et al., 2004). However there age of group user has been requested to participate to use the MLPTET to verify and fulfill the system the developed MLPTET system. This method of this evaluation were sent to all the age group starting from below twenty (20) to above fifty (50). The age group profiles of the respondents are hundred and one (101). Total of hundred and fifty (150) questionnaires were distributed to measure the three key factors, functionality, accuracy and usability. Hence, out of the hundred and one respondents, 54 (53%) are male respondents while 47 (47%) are female respondents.
There are many methods available for the purpose of system evaluation in terms of the functionality, accuracy and usability satisfaction of the tool. The objective of this evaluation is to view the results of the MLPTET. The most important is to identify the ease of use of the tool. Then, the system is measured based on the usability and accuracy. This evaluation will be elaborated in the Chapter 8 heavily. Then, next the end product and outcome phase of the MLPTET ends the development processes.

### 3.2.7 End Product

A text prediction application is produced as a result of this research development. It should be able to predict all message input process according to the database repository provided and auto complete the SMS input message based on the dictionary controlled by the user. It should be able to predict all message input process according to the database repository provided.

### 3.2.8 Outcome

Malay language Predictive text entry tool for SMS is developed in Java 2 platform Micro Edition (J2ME) the most Ubiquitous Application platform for mobile devices. This is one of the best practices to improve the usability and accuracy of the MLPTET. Predictive text entry method is used to improve the efficiency functionality of the text entry rate over multi-tap.

The predictive text entry tool with the search algorithm tool is proposed to adopt the traditional method such as ‘Multi-Tap’, which made the text entry search using the data structure and retrieves it faster. An additional feature is the development of a self-based dictionary to support the Malay language. The application will provide the users with a prediction tool to send messages through SMS, add functionality such adding new
word, update and retrieve from the data storage. The MLPTET will help users to reduce the time of multiple pressing on the mobile keypad. There will no restriction in using the MLPTET and this will reduce the users’ frustration on pressing the keypad to write their message.

Mainly it helps the user to predict their intended word and even though the word is not found in the dictionary the tool will, the system allows user to save the word for future use. The tool is developed based on the combination of the existing techniques. The tool will be developed for user convenience and speed in typing the text messages for day to day use.

Generally the prediction helped to reduce error by providing suggestions based on the text input. For those who have difficulty in using the old fashioned method, this prediction tool assist user in learning how to navigate the interface in a simple way with easy instruction.

In the next Chapter 4, the proposed predictive text entry algorithm of the mobile phone SMS text input especially the issues that influence the adoption of the word prediction techniques. The chapter also includes the technique to develop the mobile application, identify the target platform, and the core programming languages to develop the application.
CHAPTER 4: THE PROPOSED PREDICTIVE TEXT ENTRY ALGORITHM

4.1 Introduction
This chapter is designed to look into the analysis of the MLPTET. The proposed predictive text entry is conducted to list the need to develop a Malay Language predictive text entry tool. An existing tool comparison was made at the initial stages in the software development life cycle of the MLPTET. The results in conjunction with the analysis of the literature review conducted in Chapter 2 formulates a conceptual diagram to illustrate process flow on the how the predictive text entry operates in Malay language for Short Service Message on mobile phones. Following that sub sections, the methods, result, algorithm and the types of combinations of words of the MLPTET are presented.

4.2 Proposed problems and Solutions
As the first step, one must understand the interface from the user’s side and the purpose of text entry mechanism. The prediction text entry tools have been developed for many languages. Each language has a different structure. For those who speak in Malay, they would like to apply the spoken language term in the text messaging. This is especially for those students who like to use slang, abbreviations and some of their dialect words. However, the predictive text entry system does not accept the characteristics with informal spoken language. Since they prefer to use such characteristics for text messaging communicates then it will not be useful even though the predictive text entry tool is developed. The reason for Short Message Service is mainly uses for the purpose of communicating in a fast manner and at the same time to reduce their financial cost. The collection of words is extracted mostly from the Malay dictionary and stored in the
mobile database. The most frequently used words have to be selected. Basically, for Short Message Service communication, the grammar does not really play an important role.

The questionnaires was distributed to identify the frequently used words. There subjects comes from various categories. The survey shows that students prefer to use slang and abbreviations. The adults like to type the words with proper grammar. To cater for both the group, the dictionary will be designed in such a way that the abbreviation and Malay dictionary words will be allowed and stored into the database. Adding new words will be allowed in the system. This is due to the fact that the main purpose of predictive text entry is to input text message rapidly and easily. This will resolve the problem which is not all words that users require is included in the database. The new dictionary will adapt to the word entered by users.

The second issue is on the disambiguation problem. Currently all existing system with the predictive method has problem in identifying the most frequently intended word. This has been discussed in detailed in the Literature Review which includes a chapter on how the sequence of the key presses work. In actual fact, the research study on how to enhance the disambiguation has been carried out. However, to avoid the disambiguation a “tradition based dictionary” will be incorporated for the user to use the mobile phone for sending text message purposes. This will lead to an increase of the storage capacity. Since there are new technology of mobile grows, the limited sized memories of mobile phone can be increased. The growth or storage will also increase since there will be a lot of intended words. An increase in the memory size can overcome this problem. The text messaging consists of a concise dictionary of Malay words with frequency counts. An N-level index search mechanism to search a word. A bigram grammar
structure has a adaptive mechanism with grammar rules and verb sub grammar where facility for user to enter the text message in a mobile. A part of this the GUI also consisting of an editor. (Afsaneh, 2002).

4.3 Process of the MLPTET Algorithm

The MLPTET input in the mobile phone that helps user type their messages quickly and easily. With MLPTET you can write words by pressing keys just once because MLPTET recognizes each letter by predicting the intended the word you want to type. So you can be relieved from pressing keys repeatedly to get letter. According to the spelling of the word that you want to type of each letter, press key just once based on the Malay letters which are displayed on the keypad of your phone. MLPTET will intelligently select and display the desired words. So how the MLPTET works? Imagine that you want to type the words in you message. The MLPTET has an intelligent words selection features will automatically select the enter letter and display the letter in the mobile phone. What’s now with the new Malay?

You must be wondering what is new with MLPTET? Well the new MLPTET has been designed to suit your needs by just pressing one key for all the Malay letters mapped on that key will be displayed on your screen. So you don’t have to search for the right key for the right letters. Every time you do this the new Malay MLPTET also displays all the others words that can be formed by the same key stroke one after the other. So you can choose the word that you are looking for by simply pressing the ‘Next’ button key until you find it. Not only that the MLPTET is smarter than you think. If you want to type the words that have 5 letters or more, all you have to do is to press the letter of the word.
The new MLPTET automatically displays all words of 5 or more letters which start with that letter and you can select the word the ‘Top’ Arrow button. So the longer the word the lesser of number of keystrokes you use to type it. Even if you don’t know how to spell the words you wanted, there’s nothing to worry since the new MLPTET displays all the words that can be found from the first letter. You just have to choose the word you wanted to type. The new MLPTET has another advantage which it does not only predict Malay words but also English words at the same time. So if you want to use both Malay and English words in your text message words in your text messages which first choose Bilingual SMS option. With this MLPTET predicts the English words just as well it predicts all your Malay words. Now that you know that how simple it is to use the MLPTET and must have realize that it is the smarter way to SMS (Short Message Service) because you can send more messages in lesser time and with lesser number of keystrokes presses. So what are you waiting for? Get on the fast lane and have fun messaging with MLPTET.

4.4 The Conceptual Diagram of Predictive Mechanism

Most of the mobile applications are programmed in Java 2 Micro Edition (J2ME). In order to standardized Java Runtime environment in Java Application Programming (API), the Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIPD) are used. The system has a mechanism to interpret the input which is pressed and translates it into a letter, when user key press as input. Then it also should take into consideration of the letters that are pressed multiple times and the timing between each key press (N.N MIDP, 2009). Subsequently when the first key letter is pressed, it will use the first letter as the keyword to search for possible matches with the word stored in the persistent storage. Then it stores the word into the persistent
storage which was entered the user for further predictive use. After storage, the predictive system retrieves the matched words and temporarily stores it to a data structure. Then users will input the next letter and the searching process will be continued in the data structure. The search will be faster through data structure compared to the search through a persistent storage. The reason to use this technique is because it will improve the responding time. Well the system is smart where the words will be sorted when user is retrieving the words. The Figure 4.1 illustrates on how the proposed predictive text entry works.

**Figure 4.1: Conceptual Diagram of Predictive Mechanism**

### 4.5 Summary of this Chapter

This chapter presented the proposed predictive text entry algorithm part of the MLPTET. The proposed solution for MLPTET have been completely identified. The functional and non functional requirements of the MLPTET explanation will be illustrated in the Chapter 5. An object oriented analysis approach was studied and it helped to produce use cases of the system.
CHAPTER 5: REQUIREMENT ANALYSIS

5.1 Introduction

Raw data collected is insufficient to establish the usability and accuracy of an existing system or the requirements for a new system. The next stage is to convert and manipulate the collected data so that it is usable for the analysis and requirements phase.

The requirement analysis initiates with a number of general assumptions and user characteristics. In developing the requirements for the system, each component of the MLPTET and its new proposed features and concept is examined and investigated separately and is treated as a component. Use case diagrams are used as they are capable of capturing the systems functionality from the user’s perspectives. Each use case for the component will be transformed into a list of requirements for that component in order to be implemented successfully.

This chapter is dedicated to the elicitation of the requirements for the MLPTET. It provides a description of what this system should do to solve the identified problems in the current Short Message Service (SMS) problem. It fulfills the objectives of capturing the systems concept and requirements. Generally, this chapter is intended to form a foundation and basis for the design of the software product.

5.2 General Assumptions

The following assumptions have been made in the elicitation of the requirements. It is assumed that the users of the SMS understand well the concept of the current SMS message input methodologies, and the keypad design.
5.3 User Characteristics

The system users here refer to the individual public who participate sending out the SMS by using the Malay language. Users must be aware of how to improve the predictability, quality and productivity of their work. The users should understand well the concept of the current SMS word prediction problem and how the propose system solve out the problem.

5.4 Scope of the Research

MLPTET should be designed for user particularly to predicate and auto complete the word being entered only if exists in the dictionary database repository which controllable by the user themselves. The foundation of the MLPTET is to minimize the effort of user to do the message entry work by auto complete as much as possible base on the database repository. It also should provide the flexibility for the user to switch on and off the automation prediction feature.

5.5 Object Oriented Analysis

This section presents a complete object oriented analysis which models the system based on the identified functional requirements. The following sub section present a use case analysis on the MLPTET.

5.5.1 Users Requirements

Use Case Diagrams for the MLPTET start from typing message in the editor of the mobile interface. There are three actors which are briefly explained here as use case analysis. Therefore, the analysis use case represents them to expose their dynamic within the MLPTET. Figure 5.1 shows the case diagram for Text Input in Malay language.
Figure 5.1: Use Case Diagram of MLPTET

a. **Text Input.** A user is required to type message in preferred word in Malay. There is an option for the prediction to switched on or switched off. User can type the message in traditional way or on the dictionary called “predict on” to get the intended word from the dictionary.

b. **Select Character.** A user is required to type the first letter of the intended word to retrieve the matching word from dictionary.

c. **Search Word.** A user selects a search type (i.e. by alphabetical or by random) from the Search option. Then, the user can type in the intended word that they want to search.
d. **Add new word.** The user is allowed to add new words into the MLPTET. In case the commonly used words are not available then, user can enter the new word for their future use.

![Use Case Diagram Display Malay Text](image)

**Figure 5.2: Use Case diagram display Malay text**

**Use Case:** Display Malay Text

**Primary Actors:** User.

**Stakeholders:** User – wants to see what she/he typing

**Precondition:**
- User must type the letters on the keypad to display

**Post condition:**
- Display text as written.
- Flow of event(s):
  - System has the specific information and key values
  - System checks the rule within the key value and positioning.
  - Analysis the rules of the Malay grammar.

**Use case ends.**
Use Case: Analyze Entered Text

Primary Actors: MLPTET

Stakeholders: System → Passive (Analyze the entered message and search using Index from the dictionary)

Precondition: User request

Post condition: Display text as written.

Flow of event(s):

- After user entered the text messaging the system start to search by index and string.
- The words are sorted in alphabetical order
- The match the intended word and display on the interface of the mobile

Use case ends

5.5.2 Browse Dictionary Component

The user should be able to browse through the dictionary in order for the user to do the checking on the dictionary. Figure 5.4 illustrates the use case diagram for the Browse Dictionary component while Table 4 in Appendix A gives the description in relation to the use case.
Figure 5.4: Use Case Diagram for the Browse Dictionary Component

For the implementation of the Browse Dictionary Component, thirteen requirements are elicited, as shown in Table 5.1.

Table 5.1: The Requirements for the Browse Dictionary Component

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MLPTET shall provide a screen that enables the user to browse all the data in the dictionary.</td>
<td>MLPTET-Mod04-01</td>
</tr>
<tr>
<td>2.</td>
<td>MLPTET shall provide an import function for the user to import all the embedded data into the database if the database is empty</td>
<td>MLPTET-Mod04-02</td>
</tr>
</tbody>
</table>

5.5.3 Add New Word Component

The user should be able to add new word into the dictionary at any time needed. Figure 5.5 illustrates the use case diagram for the component while Table 5 in Appendix A gives the description in relation to the use case.

Figure 5.5: Use Case Diagram for the Add New Word Component
The use case diagram and its description defined in this section lead to ten requirements for implementing the module, as described in Table 5.2:

Table 5.2: The Requirements for the Add New Word Component

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MLPTET shall provide a screen for the user to add new word into the database</td>
<td>MLPTET-Mod05-01</td>
</tr>
<tr>
<td>2.</td>
<td>Alert shall be generated if the data being added or fails to be added</td>
<td>MLPTET-Mod05-02</td>
</tr>
</tbody>
</table>

5.6 System Requirements

Use Cases are useful way to capture the functional requirements of a system. Thus, use case is also a technique for documenting the potential requirements of a new system. Each use case provides one or more scenarios to convey how the system should interact with the user and system to achieve a specific goal. The technique were used for MLPTET.

5.6.1 Use Case diagram of the system

The use case shown in Figure 5.6 is an overview use case diagram of the system. As shown in the Figure 5.6 the Actor is outside the system that interact with the system. A primary actor is one having a goal requiring the assistance of the system.

![Use Case diagram of the system](image)

**Figure 5.6: Use Case diagram of the system**

**Use Case:** Type message in Malay

**Primary Actors:** User
Stakeholders: User – wants to write message in Malay.

Precondition:
- User must have knowledge of how to write in Malay
- Input character with 160 characters or more depends on the mobile device’s word capacity.

Post condition:
- Characters are correctly formatted and the user intended word.

Flow of event(s):
- User will open the menu from his/ her mobile phone.
- Then choose write message option from the menu
- An option to choose prediction or non prediction (on/off)
- An editor will open to type the message
- To write words user will enter some letters from the mobile keypad
- System will search the intended word from the dictionary.

Extensions:
At any time the intended word is not searched than save the typed word in the dictionary

Special Requirements:
None

Use case ends.
5.6.2 Malay Predictive Text Entry Tool Components

- **Auto Search Option Component.** The auto search option component will ensure the auto complete feature being on for the whole text input process.

- **Text Input Component.** The text input component will allow the text being entered and will be auto completed base on the prediction option and the data in dictionary repository.

- **SMS Send Component.** SMS send component is ready to deploy into real mobile phone in order to complete up the SMS process.

- **Browse Dictionary Component.** Browse dictionary repository send component here will display up every word exist in the database which is entered by the user.

- **Add New Word Component.** Add new word component provides an interface to the user to do enter the new word into the dictionary.

5.6.3 Auto Search Option Component

A good auto completion tool is good if it allows the user to have the choice to switch on and switch off the feature as they desire. Figure 5.7 illustrates the use case diagram for the component Table 1 in Appendix A gives the description in relation to the use case.

![Figure 5.7: Use Case Diagram for the Auto Search Option Component](image-url)
Based-on the use case diagram and its description, the following requirements are derived as shown in Table 5.3.

**Table 5.3: The Requirements for the Auto Search Option Component**

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MLPTET shall provide a switch setting screen for the user to switch on and off the auto prediction feature.</td>
<td>MLPTET-Mod01-01</td>
</tr>
<tr>
<td>2.</td>
<td>The switch setting will be default set to switch on</td>
<td>MLPTET-Mod01-02</td>
</tr>
<tr>
<td>3.</td>
<td>The switch setting shall able to affect the text input process.</td>
<td>MLPTET-Mod01-03</td>
</tr>
</tbody>
</table>

**5.6.4 Text Input Component**

This is the main component of the system which performs the text input prediction. Figure 5.8 illustrates the use case diagram for the component Table 2 in Appendix A gives the description in relation to the use case.

![Figure 5.8: Use Case Diagram for the Text Input Prediction Component](image)

Few requirements must be fulfilled to implement this module, as illustrated in Table 5.4.

**Table 5.4: The Requirements for the Text Input Component**

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MLPTET shall provide a message input screen that allows user to enter the text message within the text length allowed</td>
<td>MLPTET-Mod02-01</td>
</tr>
<tr>
<td>2.</td>
<td>The input will be auto completed base on the information from the dictionary which is stored in the MIDlet database</td>
<td>MLPTET-Mod02-02</td>
</tr>
<tr>
<td>3.</td>
<td>The auto prediction feature will only be available if the auto search feature is set to on</td>
<td>MLPTET-Mod02-03</td>
</tr>
<tr>
<td>4.</td>
<td>The search button shall be provided in the text entering screen to allow the user to auto complete the word whether the prediction feature is on or off</td>
<td>MLPTET-Mod02-04</td>
</tr>
</tbody>
</table>
5.6.5 SMS Send Component

SMS Send component here will be for the demo purpose as it needs a suitable device which is able to run the MIDP and also has the port to send out the message. Figure 5.9 illustrates the use case diagram for the component Table 3 in Appendix A gives the description in relation to the use case.

![Figure 5.9: Use Case Diagram for the SMS Send Component](image)

Based on the use case diagram and its description, the following requirements are derived, as shown in Table 5.5.

**Table 5.5: The Requirements for the SMS Send Component**

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement Description</th>
<th>Requirement ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MLPTET shall provide a phone number entry screen that allows user to enter the phone number for sending.</td>
<td>MLPTET-Mod03-01</td>
</tr>
<tr>
<td>2.</td>
<td>MLPTET shall provide a send button for the user to send out the SMS.</td>
<td>MLPTET-Mod03-02</td>
</tr>
<tr>
<td>3.</td>
<td>MLPTET shall provide a cancel button for the user to cancel the message sending action</td>
<td>MLPTET-Mod03-03</td>
</tr>
<tr>
<td>4.</td>
<td>Alert shall be displayed if the message being sent</td>
<td>MLPTET-Mod03-04</td>
</tr>
</tbody>
</table>

5.7 Functional Requirements

Functional requirements capture the intended behavior of the mobile phone system. Based on the modules, each component will be investigated to capture the system
requirement. The functional requirements of a system describe what the system or portion of a system must do in order to meet the business requirements. The functionality of any application is primarily depicted according to three key components which are system, non participant and participant. The capitalizing the scenario of the conceptual diagram into a complete analysis, the study has to determine a proximate identification of all the functional and non functional requirements of the MLPTET. The first session starts by describing the first actor which is the system itself and then, the core functional requirements which covers the following topics.

- The users who deal with SMS to text messaging of the MLPTET relating to the dictionary in the mobile phones.
- System support which presents a determination of the system space for projects creation which reflects the mobile device and the support of data types.
- Project Management which relates to the Management of creation, deletion and customization.
- File control of active project which relates to the management of files upload of words depends on the size for the dictionary in the mobile devices. It seeks a compression before uploading all the commonly used words.

5.8 Non-Functional Requirements

Apart from the functional requirement, the Malay Language Predictive Text Entry Tool (MLPTET) also consists of the non-functional requirements such as design of a system. The non-functional requirements can be used to view the operation of the system instead of the behaviors. According to the research conducted by Lawrence et al. (2001), requirement describes the activities of system like reaction to input and state of each entity in the system before or after the activity occurs. Non-functional
requirements are requirements which identify criterion which is used to measure the operation aspect of the certain system, rather than behavioral aspect. In general, non-functional describes how a system is supposed to be. Sometimes they are referred as qualities of a system. Other expressions for non-functional requirements are quality attributes.

In this research, three quality attributes are considered which are as follows:

- **Usability.** The MLPTET has to provide simple graphical user interface to make sure that the participants or users of the system are able to understand how to work with it. The percentage of participants responded on the usability options such as simple, complexity of the tool and others related question. According to this functionality the user has responded that 82% the tool is user friendly.

- **Accuracy.** The MLPTET performance is efficient; therefore the speed in typing text messaging and retrieving from the dictionary is important. Analyzing the typing speed compared to several input techniques. Reducing the spelling error, typing characters, all the typed words that are not in dictionary will be saved in the dictionary. The participants has responded that the accuracy of MLPTET is 85% effective and 14% average and 5% is ineffective.

- **Functionality.** The MLPTET functionality should be easy to modify for future enhancement. This is to support more languages in the mobile devices. The functionality with sophisticated search and new word that be added into dictionary is very useful. Based on the statistic 80% of user has responded that this functionality is effective and 29 % user has responded that this functional is ineffective and difficult to use. This function assists user to search translation of
the words entered in different grammatical forms like plural nouns, different tense.

5.9 System Requirements

System requirement consists of hardware and software specification to run any application. The important role of the system requirement is to ensure that the best practices environment to run the application. In this research, the suggested tool and programming languages is J2ME (Java 2 Micro Edition Enterprise).

<table>
<thead>
<tr>
<th>Table 5.6: Software Requirements</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Software</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows XP Professional and above</td>
</tr>
<tr>
<td>Database</td>
<td>Record Store Management</td>
</tr>
<tr>
<td>Runtime</td>
<td>Java Runtime Environment, NetBean IDE 6.5</td>
</tr>
<tr>
<td>Web based</td>
<td>Plug-In</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5.7: Hardware Recommendation</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>Intel Pentium Compatible Processor (1.8 GHz)</td>
</tr>
<tr>
<td>RAM</td>
<td>4GB more</td>
</tr>
<tr>
<td>Hard Drive Space</td>
<td>5GB</td>
</tr>
<tr>
<td>CD ROM Drive</td>
<td>1</td>
</tr>
</tbody>
</table>

5.10 Summary of the Chapter

This chapter has documented the requirements of the MLPTET, a tool that will provide text input prediction feature for the message input in SMS. Additional requirements are derived such that the user can make use of the MLPTET properly. The requirements are elicited through the application of use case scenarios. Many of the use cases are based on the review and analysis in Chapter 4.
The requirements elicited in this chapter will be used as the basis for the design and implementation of the MLPTET in Chapters 6 and 7. Each requirement has been portioned such that it can be traced and accounted for, in the evaluation of the finished product.
CHAPTER 6: SYSTEM DESIGN

6.1 Introduction

System design is the essential key element for a successful software development process that converts the functional and non-functional specifications of the software into solution model. The solution model consists of three components of architectural representation which are software structure, behavior, components and flow. In this research, Unified Modelling Language (UML) is the main modelling tool utilised to represent the Predictive text entry for Short Messaging Service on mobile structure and its interaction in conjunction with object-oriented design.

This chapter describes the design strategy of Short Messaging Service Prediction based on the requirements described in the previous chapter, Chapter 5. The first segment of the chapter will discuss on the outline of Short Messaging Service Prediction. Then, the development strategy is discussed in terms of how the system will be developed using the screen map structure.

6.2 System Architecture

In software engineering, system architecture refers to overall structure of the system and its relationship. This is the initial design to identify main components and sub-system communicates with the system. The Short Messaging Service Prediction system is developed from MIDlet. Figure 6.1 illustrates the system architecture of the system MLPTET.
The technology consists of a few core modules that work together to provide the core engine. The Graphical User Interface (GUI) consists of the forms which will present modules for the user. The module performs the entire pattern matching functions. Figure 6.2 illustrates the application the technical architecture.

GUI provides the front-end point of the communication and interaction between the system or device and the user. Basically in the context, the graphical user generates the module and handles the processes requests by retrieving text message or data from the database. Usually the main dictionaries are the no dynamic data and the only dictionary which is different is the user dictionary which contains the entries by user. The functions that are available such as add, update, delete and retrieval of the data.

Figure 6.1 illustrates the architecture of the mobile phones with the external software that in used to send the message. Figure 6.2 shows the big picture on how the internal architecture sends and receives the characters that entered by user is retrieved from the dictionary and sends back to the GUI which text message editor. As the user enters, the MLPTET sends the alphabetic characters over the system and the possible Malay
characters are predicted and complied and eventually sent back to the MLPTET. This information is used to capture some meaning about the technical structure and it helps to understand sending and receiving characters.

The J2ME platform consists of a set of layers, on top of which lies MIDP. We develop J2ME applications on top of MIDP; thus, the applications are called MIDlets. Every J2ME application must extend the MIDlet class so the application management software can control it.

When the application management software invokes the startApp(), pauseApp(), or destroyApp() method, the MIDlet's state changes. For example, when pauseApp() is invoked, the MIDlet changes from an active to a paused state (Keogh, 2003).
Here is a blueprint of the MIDlet:

```java
public class Main extends MIDlet implements CommandListener {
    Display display;
    private List menu;
    private Command commandSelect;
    private Command commandExit;
    ...
    public Main() {
        // The constructor.
        ...
        // Data initialization.
        // Read saved data from RMS.
        // Create UI components and the first screen (menu).
    }
    public void startApp() {
        // Enter the active state.
        // Display the first screen.
        display.setCurrent(menu);
    }
    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
        ...
        // Clean up data streams etc.
    }
    public void commandAction(Command c, Displayable s) {
        notifyDestroyed();
    }
    // Other customized methods....
}
```
6.3 **Sequence Diagram**

The sequence diagram is called Interaction Diagrams which is belongs to a group of UML diagram. This diagram describe how objects interact over the course of time through exchange messages. Moreover the single sequence diagram represents the flow of events for single use case.

6.3.1 **Sequence Diagram of MLPTET**

Figure 6.4 and Figure 6.5 shows the sequence diagram that illustrates the sequence actions that occur in the system. It shows graphical constructs that represents the existence of the entities with its roles and also operation involved over a period of time (Pilone and Pitman, 2005).

![Sequence Diagram of MLPTET](image)

- **Return a text box property so that user can write message**
- **MLPTET will take each character of the textbox and will come up with the specific key value for the character**

**Figure 6.4: Sequence Diagram of the Use Case for Overview of MLPTET**
Figure 6.5: Sequence Diagram of the Use Case for Overview of MLPTET

a. Text Input

The type message in functions allows user to choose the option predictive option to be switched ON or OFF. If user chooses, the predictive option ON then it will switch to search letter as what the user has typed in. If user chooses the predictive OFF then users are allowed to type like Multi-Tap text messaging.

b. Select Character

The Select Character function allows user to choose the letters contains in the user’s intended word. Figure 6.6 shows the sequence diagram Select Character.

Figure 6.6: Sequence Diagram of Select Character
c. Search Word

This function enables a user to search for a word that they intend and displays the match word. If the word is not found, the message will be displayed that the word is not found. Figure 6.7 shows the sequence diagram of Search Word. A user can search a word according to what they intend, either in alphabetical order or randomly. To perform a search according to the alphabetical order or by random order, the user enters or chooses the first letter from the intended word from a list of words which will be displayed. Similarly, if the user chooses to type a letter randomly, then they can select the letter like ‘G’ and see the list of matching words in ‘G’ will be displayed. Users can type more than one letter to get the search word and display it on the text message screen.

![Figure 6.7: Sequence Diagram of Search Word](image)

---

d. New Word

The users are allowed to add new words into the Malay Language Predictive Text Entry Tool (MLPTET). If the commonly used words are not available, then the user can enter the new word for their future use. Figure 6.8 shows the sequence diagram for Add New Word.
6.3.2 Activity Diagram of MLPTET

The activity diagram is used to describe the behavior of an object in response to internal processing and the flow of its processes (Pilone and Pitman, 2005).

Figure 6.9 shows the activity diagram of Search Word. Firstly a user selects the search type. Then the users select the search letter or type the complete word. If the word found in the dictionary then the results is displayed on the text message screen. If not the prompt message will shows “Word not found”. This means that a particular word intended is not available in the dictionary.
Figure 6.9: Activity Diagram of Search Word

Figure 6.10 shows the activity diagram of Add New Word. User types the new words not found in the dictionary and save to the database or dictionary.

Figure 6.10: Activity Diagram of Add New Word
6.4 General Design Overview

GUI APIs are defined in MIDP rather than CLDC. Short Message Service Prediction will use both high-level user interface APIs (such as Alert, Form, and exclusive components like Command). Screen is the super class of all high-level interface APIs. Figure 6.11 shows a screen map of MLPTET.

![Figure 6.11: MLPTET screen map](image)

6.5 Record Management

The Mobile Information Device Profile (MIDP) is a persistent storage. In the Figure 6.12 shows a record store which is in a database. There are many set of data in pieces which is called the record. In a single MIDlet there are many record store has been shared with MIDlet. There can have multiple records are stored within the MIDlet. Each record store is identified by a name which is unique by White (2002).

![Figure 6.12: MIDlet Record Management (White, 2002)](image)
6.6 Structural Design

This section addresses the structure design of each module described in Chapter 6.

6.6.1 Auto Search Option Component

The auto search option component will ensure the auto complete feature being on for the whole text input process. The structure of this component consists of attributes and methods as illustrated in Figure 6.13.

![Auto Search Option Component Diagram](image)

**Figure 6.13: Auto Search Option Component**

Based on the following figure methods satisfy the requirements elicited for this component where:

- The `genOptionScr()` method will generate the option screen. This option screen consists of Form, ChoiceGroup and Command button. The Back Command button will fire the CommandListener action and the ChoiceGroup will fire the ItemStateListener.

- `itemStateChanged(Item item)` method will reset the global search value according to the ChoiceGroup selection when the ItemStateListener being fired.
6.6.2 Text Input Component

The text input component will allow the text being entered and will be auto completed base on the prediction option and the data in dictionary. The structure of this module consists of attributes and methods as illustrated in Figure 6.14.

![Text Input Component Diagram]

**Figure 6.14: Text Input Component**

Based on the following figure satisfy the requirements elicited for this component where:

The genSearchScr () method will generate the message input screen. This screen consists of Textbox and Command button. The Back Command, Search Command and Send Command button will fire the CommandListener action.

SimpleTimerTask() class will be called if the search option is switched on. The method used is scheduleAtFixedRate which set to 3000 milliseconds. Timer will always be reset if as the replacement process might be longer than the interval second Command Listener (Item item) method will fire the commandAction(Command c, Displayable) method. Split() method will split out the sentence according in order to get the last word which will be used for auto replacement.

The replaceMsg(String msg, String l, boolean local) method will be called is the auto complete feature being called. The method will call the SimpleComparator class, SimpleFilter class and RecordEnumeration class.
6.6.3 SMS Send Component

SMS send component here is a ready component that enables to deploy into any mobile phone in order to complete up the SMS process. This component will have the close relationship with the text input module. The structure of this component consists of attributes and methods as illustrated in Figure 6.15.

![Figure 6.15: Send SMS Component](image)

Based on the following figure satisfy the requirements elicited for this component where:

- The genPhoneScr() method will generate the phone number entry screen. This screen consists of Textfield, and Command button. The Back Command button will fire the CommandListener.

- The SMSender() class will be called if the port being set and the device is supportable.

6.6.4 Browse Dictionary Component

Browse dictionary send component here will display every word which exists in the database which is entered by the user. The structure of this component consists of attributes and methods as illustrated in Figure 6.16.
Based on the following figure satisfy the requirements elicited for this component where:

- The genNameListScr() method will generate the dictionary’s data screen. This screen consists of List, and Command button. The Back Command button will fire the CommandListener.
- The SimpleComparator() class, SimpleFilter() class and RecordEnumeration() class will be called in order to get the data from the database and ensure that the display order is alphabetical order.
- ImportEntry() method will import the data list in to the database

6.6.5 Add New Word Component

Add new word component provides an interface to the user to enter the new word to the dictionary. The structure of this module consists of attributes and methods as illustrated in Figure 6.17.
Based on the following figure satisfy the requirements elicited for this component where:

The genEntryScr() method will generate the dictionary/s data entry screen. This screen consists of Textfield, and Command button. The Back Command button and the Add Command button will fire the CommandListener.

The addEntry() method will call up the SimpleRecord.createRecord in order to add in the new record from the Textfield.

### 6.7 Mobile Application Flow

The following Figure 6.18 shows the Mobile Application for Malay language predictive text entry tool for Short Message Service (SMS) on mobile phone.
6.8 Algorithm

There are two types of algorithm which are available in the market and commonly used. The two types of algorithm include Multi-Tap and Predictive Text Entry methods. The old-fashioned method is the multi-tap which is used for precise text messages. The Predictive text entry is the quicker and easy way for text messaging. This method facilitates user to type in a fast manner. The optimal solution is to reduce the number of key strokes and reducing errors. However, improvement is still required.

Figure 6.18: Mobile Application Flow
6.8.1 Implementation of Dictionary

In the current dictionary a list of 700 most frequently used Malay words was compiled into an excel software and then were arranged the frequently used word and imported the data to the dictionary. The dictionary can support more than 700 words. When user presses a key from the mobile phone, the key’s number would search; the words that match the first typed key sequence would be retrieved from a dictionary. There is a facility for new word addition. Whenever the user intended words which are not found in the dictionary, it allows user to save the new word. This is purely based on the user’s choice. If the user uses the word frequently, then user can save this word for future use. Figure 6.19 shows the flowchart of the algorithm for Predictive Text Entry.

Figure 6.19 shows the words are searched from the dictionary. The array and string function is used to retrieve the words from the dictionary. The dictionary searches fetch the matching string and then display the result.
Figure 6.19: MLPTET Flow Chart
6.9 Summary of the Chapter

This chapter discusses on the Predictive Text Entry models. There are two architecture designs which include system architecture and technical architecture. The data design was presented using UML diagrams and flow chart of predictive algorithm.

Overall this chapter provided an overview of the design of MLPTET, which was based on the requirements elicited in Chapter 5. All the components of the MLPTET appeared at the design level, some of the components are more easily presented at an implementation level in Chapter 7.
CHAPTER 7: SYSTEM IMPLEMENTATION AND TESTING

7.1 Introduction

This chapter discusses on the development of environment employed during the development of MLPTET and the actual environment to run the MLPTET. According to Lawrence (2001), the system implementation converts design requirements and designs it into a workable code. The implementation of MLPTET was prepared in several levels. In the following sections, MLPTET’s development environment, implementation and coding are discussed.

7.2 Development Environment

This section discusses the programming languages along with the development tools used to develop MLPTET as well as the record management system and operating system employed during the development of MLPTET. The development environment is crucial for the development of any software tool. Selection of suitable software and hardware not only reduces development but also ensures the product’s success. The actual operating environment is also presented in Section 7.4.

7.2.1 Programming Language

The MLPTET is developed by using an SDK, however Java is the core language. J2ME is the inherited version of Java targeted at devices that have limited memory, display, and processing power such as mobile phones and PDA. J2ME is compatible with all Java-enabled devices and it is a device that runs the Java Virtual Machine. In addition, J2ME also provides powerful security features that is found in the Java language (Salleh, 2006).
7.2.2 Development Toolkit

To develop the mobile application, the Java 2 Platform, Micro Edition (J2ME) is the best toolkit. The mobile application the toolbox of J2ME is consists of Connected Limited Device Configuration and Mobile Information Device Profile (MIDP) is designed to run on mobile device. This toolkit consists of many features such as the emulation performance optimization, tuning and documentation to speed up the development time. (Salleh, 2006)

7.2.3 Development Environment

The MLPTET is developed using J2ME as the core programming language with the NetBean 6.5 as the development tool. NetBeans IDE is a free-of-charge integrated development environment (IDE) primarily focused on making it easier to develop Java applications. It provides support for all types of Java applications, from rich desktop clients to multi-tier enterprise applications to applications for Java-enabled handheld devices. NetBeans IDE consists of a modular architecture that allows for plug-ins. However, since the range of features in the basic installation is so huge that it can start using the IDE for your work without plug-ins. The IDE is written in Java; therefore this program can run on any operating systems provided there is a Java 2 Standard Edition JDK. The job of IDE's is to make the edit-compile-debug cycle smoother by integrating the tools for these activities. There are many features available in IDE to ease programming and identifies the syntax error in the coding. It detects the error and immediately marks them in Source Editor. Besides that it also has the help functionality to code faster code completion and word matching. Most importantly it provides a visual navigation window and keyboard navigation shortcuts. In this research, an object-oriented programming technique with bottom up coding approach has been
adopted to develop the MLPTET. This approach is chosen to enable the smaller units module to be reused since the units are often invokes other modules.

7.3 Record Management System (RMS)

The Mobile Information Device Profile (MIDP) of J2ME provides a mechanism through MIDlets with the Record Management System (RMS) that has a records-based persistent storage system. Record Management System (RMS) can also store and retrieve data persistently. RMS comprises multiple record stores and also called as data dictionary repository. The MIDLet provides several methods to manage inserting and deleting records in a record store. Each record store consists of a collection of records which remain persistent storage of MIDlet. (Mourad Debbabi, 2007). The interfacing of RMS and MIDlet is shown in Figure 7.1.

![Figure 7.1: The Interfacing of RMS and MIDlet](image)

7.4 System Implementation

In Figure 7.2 the interface of the MLPTET is shown. This section mostly shows the interface of each module. Then, the implementation of each module will be verified against the requirement elicited in Chapter 5 and Chapter 6.
7.4.1 Auto Search Option Component

Auto search option component provides the interface for the user to switch on or switch off the auto predict feature during the text input process. Table 7.1 describes the implementation of the component as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.
<table>
<thead>
<tr>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLPTET shall provide a functionality to allow the user to on and off the auto new project.</td>
<td>MLPTET-Mod01-01</td>
<td>Completed, implemented using genOptionScr() method</td>
</tr>
<tr>
<td>This option will affect the whole process of auto prediction when the text being entered.</td>
<td>MLPTET-Mod01-02</td>
<td>Completed, identified by private int search = 1</td>
</tr>
</tbody>
</table>

**7.4.2 Text Input Component**

Text input component provides the interface for the user to enter the message for sending out. Table 7.2 describes the implementation of the component as elicited in the requirement list in Chapter 4 based on the Requirement ID provided.

![Message Input Screen](image)

**Figure 7.4: Message Input Screen**

**Table 7.2: Message Input Component**

<table>
<thead>
<tr>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLPTET shall provide a work space for the user to enter the message to send out the SMS</td>
<td>MLPTET-Mod02-01</td>
<td>Completed, implemented using genSearchScr() method</td>
</tr>
<tr>
<td>The input will be auto completed if the search option is set to “On” and the prediction exist in the database</td>
<td>MLPTET-Mod02-02</td>
<td>Completed, implemented using SimpleTimerTask() and replaceMsg() method</td>
</tr>
<tr>
<td>The prediction word will be based on the word that exists in the device database only</td>
<td>MLPTET-Mod02-03</td>
<td>Completed, implemented using SimpleComparator() class</td>
</tr>
</tbody>
</table>
7.4.3 SMS Send Component

SMS send component here is actually the dummy in order to complete up the SMS process. The coding will not be tested in actual as device and port needed. Table 7.3 describes the implementation of the component as elicited in the requirement listed in Chapter 5 based on the Requirement ID provided.

![Figure 7.5: SMS Phone Number Screen](image)

Table 7.3: SMS Send Component

<table>
<thead>
<tr>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLPTET shall provide a work space for the user to enter the phone number in order to</td>
<td>MLPTET-Mod03-01</td>
<td>Completed, implemented using genPhoneScr () method</td>
</tr>
<tr>
<td>complete up the SMS process</td>
<td></td>
<td></td>
</tr>
<tr>
<td>One dummy send button will be provided for user to send out the message</td>
<td>MLPTET-Mod03-02</td>
<td>Completed, implemented using SMSender () class</td>
</tr>
<tr>
<td>Alert should be displayed when the SMS has been send</td>
<td>MLPTET-Mod03-03</td>
<td>Completed, implemented using alert () method</td>
</tr>
</tbody>
</table>

7.4.4 Browse Dictionary Component

Browse dictionary send component here will display up every word exist in the database which is entered by the user. Table 7.4 describes the implementation of the component as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.
Table 7.4: Browse Dictionary Component

<table>
<thead>
<tr>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLPTET shall provide a screen for the user to view all the record exist in the database which is used for auto prediction.</td>
<td>MLPTET-Mod04-01</td>
<td>Completed, implemented using genNameListScr() method</td>
</tr>
<tr>
<td>The list should be displayed in the alphabetical order</td>
<td>MLPTET-Mod04-02</td>
<td>Completed implemented using SimpleComparator() method</td>
</tr>
<tr>
<td>Alert should be display if the database consist empty record</td>
<td>MLPTET -Mod04-03</td>
<td>Completed, implemented using alert() method</td>
</tr>
<tr>
<td>Import function should be provided if the database consist empty record</td>
<td>MLPTET-Mod04-04</td>
<td>Completed, implemented using importEntry() method</td>
</tr>
</tbody>
</table>

7.4.5 Add New Word Component

Add new word component provides an interface to the user to do enter the new word to the dictionary. Table 7.5 describes the implementation of the component as elicited in the requirement list in Chapter 5 based on the Requirement ID provided.

Figure 7.7: Add New Word for Dictionary Screen
Table 7.5: Add New Word Component

<table>
<thead>
<tr>
<th>Requirement Description</th>
<th>Requirement ID</th>
<th>Implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLPTET shall provide a screen for the user to enter the new word.</td>
<td>MLPTET-Mod05-01</td>
<td>Completed, implemented using genEntryScr () method</td>
</tr>
<tr>
<td>The word will be entered into the database</td>
<td>MLPTET-Mod04-02</td>
<td>Completed, implemented using addEntry () method</td>
</tr>
<tr>
<td>Alert should be display if the data being added successfully</td>
<td>MLPTET-Mod04-03</td>
<td>Completed, implemented using alert () method</td>
</tr>
<tr>
<td>Alert should be display if the data can not be added</td>
<td>MLPTET-Mod04-04</td>
<td>Completed, implemented using alert () method</td>
</tr>
</tbody>
</table>

7.5 SYSTEM TESTING

This chapter also discusses on system testing in the software development cycle of MLPTET. In the first section of the chapter, it presents system testing in general, the testing environment, methodology and process which were used during the testing phase. Testing deals with the validation of software at various levels. Testing is also a critical phase of software quality assurance in a software development life cycle. The objective is to execute a program with the intention to discover errors, evaluate the systems abilities and most importantly, to determine whether it complies according to its requirements and functional specifications.

The aim of this research was to test the new interface and algorithm on Malay Language Predictive Text Entry Tool (MLPTET). During the development of MLPTET, after completing the coding phase, testing began from individual program modules and progressed towards the entire system. There are four stages in the testing phase of MLPTET; unit testing, module testing, system testing and detecting coding error.

7.6 Test Environment

In this testing environment is important to ensure that hardware, software and operating system are accepted in the user requirements. For MLPTET, Java ME SDK Device
Manager is used. For the MLPTET the testing environment is under the Java NetBean IDE 6.5’s emulator is shown in Figure 7.8.

![Java Sun Emulator](image)

**Figure 7.8: Java Sun Emulator**

### 7.7 Testing Methodology

There are four stages being implemented in the testing phase of MLPTET. As shown in Figure 7.8, MLPTET testing included Unit Testing, Module Testing, System and Detecting Coding Error.

![MLPTET Testing Steps](diagram)

**Figure 7.9: MLPTET Testing Steps**

#### 7.7.1 Unit Testing

Unit testing is conducted to ensure that proper functionality and code coverage of MLPTET have been achieved during coding. This specification based testing verifies
each functions in the test cases where correctly input and the output is as expected based on the requirements. During the development of MLPTET, there are several steps which are carried out which include to:

- Examine and review the code correctness of every line of codes by reading line by line and correctness of the algorithm.
- Perform a test to each button and link to ensure that it functions as required.
- Develop test cases from the output and show the input is properly processed to the expected result.
- Ensure testing functions do exist the boundary conditions and limiting the process
- Test data and capture all error using the error handling paths

The exit criterion for this milestone is code-complete. A complete set of application with the logical and physical components were completed and made available for module testing.

7.7.2 Module Testing

A module is a collection of dependent components in which related process encapsulates into components. Testing can be defined in requirement phase which is performed at every phase. Each module unit is tested to ensure that the listed functions is working successfully based the test cases and output is as expected. The test cases is verified for correctness.
Table 7.6: Test Plan for Input Prediction

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Expected Result</th>
<th>Actual Result</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Enter the first three (2) characters of a word that exist in the database</td>
<td>Word will be auto replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
</tr>
<tr>
<td>2 Enter the first three (2) characters of a sentence that exists in the database</td>
<td>Sentence will be auto replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
</tr>
<tr>
<td>3 Enter the word that is not exist in the database</td>
<td>Word will not be replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
</tr>
<tr>
<td>4 Enter the first three (3) characters of a word that exists in the database</td>
<td>Word will be only auto replace when the [Search] button being clicked</td>
<td>Same as expected</td>
<td>The auto predict feature is off</td>
</tr>
<tr>
<td>5 Enter the space after the word</td>
<td>The word before the space will not be auto replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
</tr>
</tbody>
</table>

Table 7.7: Test Plan for Dictionary Component.

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Expected Result</th>
<th>Actual Result</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Click on the import button to import the data</td>
<td>Data will be imported</td>
<td>Same as expected</td>
<td>The database is totally empty</td>
</tr>
<tr>
<td>2 Enter the new word in the [Add New Word] screen</td>
<td>New word will be added</td>
<td>Same as expected</td>
<td>-</td>
</tr>
<tr>
<td>3 Click on the [Browse All Words] button</td>
<td>All words exist in the database will be displayed</td>
<td>Same as expected</td>
<td>The time to display all the result might be long as the database is large</td>
</tr>
</tbody>
</table>
7.7.3 System Testing

System testing in MLPTET involves testing against integrated hardware and software system in order to verify that the system meets the specified requirements as described in the requirement specifications. Also, it involves a series of different test designed to fully exercise the system to uncover its limitation and measure its capabilities. The MLPTET testing takes place at a higher level, whereby the testing focuses on behavior rather than functional structure.

7.7.4 Detecting Coding Error

The NetBean IDE 6.1 has the ability to detect an error in the line. These errors are displayed at the output windows which can be corrected instantly. However, some errors can only be detected during the runtime. When errors are detected and warning alters will be displayed with a hint of error code to ease the correction. When the user double-clicks the error message it jumps to the location in the source where the error has occurred. Thus, programming Java in this software eases the coding and errors which are found in MLPTET program codes can be immediately corrected.

7.8 Testing Results

During the functionality testing for Text Input Prediction, the following scenarios were tested in the J2ME platform using the Emulator.
Table 7.8: Test Results Text Input Prediction

<table>
<thead>
<tr>
<th>Test Scenario</th>
<th>Expected Result</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1    Click on the import button to import the data</td>
<td>Data will be imported</td>
<td>Data imported. All the words available in the dictionary are displayed as shown in Figure 7.10</td>
</tr>
<tr>
<td>2    Enter the new word in the [Add New Word] screen</td>
<td>Add new word ‘kolej’, and ‘lambat’</td>
<td>The two new word ‘Kolej’ and ‘lambat’ are added and the result is show in Figure 7.11. Figure 7.12 shows the new words are added. Word is entering one at a time.</td>
</tr>
<tr>
<td>3    Click on the [Browse All Words] button</td>
<td>All words exist in the database will be displayed</td>
<td>Figure 7.13 show the new word that added are browsed</td>
</tr>
</tbody>
</table>
| 4    Enter the first four character of word that is exist in the database    | Word will be auto replaced
Enter ‘lamb’                                                                        | In Figure 7.14 shows that first three character were input and then the system automatically predict the complete word ‘lambat’          |
| 5    Enter the space after the word. Enter the ‘hari’                        | The word before the space will not be auto replaced. In the dictionary, added two words which are commonly used. For example ‘Hari ini’ | The result is shown in Figure 7.15 on the prediction of completion word. The word ‘hari ini’ is displayed. System predicts the next word. |
| 6    Enter the first two characters of a word that exist in the database      | Word will be auto replaced
Enter ‘gu’                                                                              | The word displayed is ‘guna’ is show in Figure 7.16                                                                                 |
| 7    Enter short letters. Enter the character ‘dkt’ instead of ‘dekat’ add as new record | The added word ‘dkt’ appears on text message screen. | The short word ‘dkt’ is retrieve from dictionary which is shown in Figure 7.17                                                      |

Based on Table 7.8, the actual results for test case one is shown in Figure 7.10 where all the data is imported from the dictionary.
Based on Table 7.8, the actual results of the test case number two is shown in Figure 7.11 to add a new word in the dictionary.

Based on Table 7.8, the actual results of the test case number two is shown in Figure 7.12 where a message is prompt that new word is added into the dictionary.
Based on Table 7.8, the actual results of the test case number three is shown in Figure 7.13, all the words from the dictionary can be browsed.

Based on Table 7.8, the actual results of the test case number four is shown in Figure 7.14, to enter the intended word and the MLPTET predicts the word and provide the completion of word.
Based on Table 7.8, the actual results of the test case number five is shown in Figure 7.15. MLPTET have ability to automatically predict the word string and display the word completion.

Based on Table 7.8, the actual results of the test case number six is shown in Figure 7.16. When user only enter the first two letters than MLPTET automatically display the next word in the keystrokes sequence.
Based on Table 7.8, the actual results of the test case number seven is shown in Figure 7.17, when user enters short letter or abbreviation without having any meaning. If this word is added in the dictionary then when the character is typed the word automatically will display text messaging screen.
7.9 Testing Process

Each type of testing stage in the MLPTET uses the testing process as illustrated in Figure 7.18.

![Testing Process in MLPTET](image)

**Figure 7.18: Testing Process in MLPTET**

As shown in Figure 7.18, all the test cases are documented in Test Plans which are derived from MLPTET requirements and functional specifications. Each of the test cases is developed and the expected results are written first. The test outputs are compared with expected outputs. If there exists any differences between the test outputs and the expected outputs, the program code will be checked to discover bugs.

Based on the test result, the functionality, the MLPTET can predicts the words successfully retrieve from the dictionary and displays the input text on the mobile phone interface.
7.10 Summary of the Chapter

This chapter on system implementation has provided a detailed report on the implementation of the MLPTET. This tool is developed by using J2ME as the core language and developed under the NetBean IDE 6.5. NetBean IDE 6.5 included the easy configured and implemented emulator. This MIDP application is build up from the MIDlet and NetBean tool is used for compiled and build up the application.

In software development, testing plays an important role in which the requirement of the product is successfully implemented and functionality wise, bug free. This determines the quality if the system. There were four levels of testing which are carried out in this phase that include unit testing, system testing, module testing and detecting error program code. The testing focuses on the prediction tool for SMS in Malay language and testing was successfully conducted. The results of the testing and evaluation of prediction tool are represented thorough graphical charts.

This chapter has discussed system testing in the software development cycle of MLPTET. In the first section of the chapter, it presents system testing in general, and the testing environment, methodology and process which were used during the testing phase.
CHAPTER 8: EVALUATION

8.1 Introduction

Firstly, this chapter evaluates the MLPTET implementation. There are few subjects of evaluations which the MLPTET will be evaluated on. System evaluation is a necessary module in the development process. This chapter aims at evaluating the MLPTET by focusing specifically on the usability and accuracy objective and future enhancements. The usability factors are efficiency, learn ability, error prevention, accuracy and effectiveness have identified for the evaluation process. Techniques to measure various usability in the following characteristics.

- Goal to improve a system usability
- The participants are real users
- Test feedback

The first key subject evaluation is performed in Chapter 8, whereby during the implementation of MLPTET, all the requirements elicited in Chapter 6 were verified and fulfilled according to its module and requirement ID.

8.2 User Evaluation

The objective of this evaluation can be seen in its effort to tackle the following important areas of MLPTET:

- To identify the system ease of use.
- To identify and measure the functionality components, usability and accuracy.
- To design and develop Malay language predictive text entry tool based on the suitable existing algorithm.
To present recommendations for future enhancement of the implemented Malay Language Predictive Text Entry Tool.

Predictive text entry method was designed and implemented using the 12 key mobile phone prototype interfaces. Based on the comparison, the results show that the existing method model provides better keystrokes and quicker.

8.3 Comparison MLPTET with Existing Tool

8.3.1 Multi-Tap Model

This follows the normal English keypad layout of 3 and 4 characters per key. A transition time between characters can be set and it is flexible. In order to get the intended letter, it is important to press the key twice or thrice within the transition time. The same methods are used to construct words for text messaging. Let us consider the word “guna”. Key 4 is pressed once to get the letter ‘g’; Key 8 is pressed twice, key 6 twice and subsequently press the same key 1 once. Therefore, the total number of key pressed needed here is **6 times**. Figure 8.1 shows the number of times the key sequences performed in Multi-Tap method.

<table>
<thead>
<tr>
<th></th>
<th>ABC 2</th>
<th>DEF 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHI 4</td>
<td>JKL 5</td>
<td>MNO 6</td>
</tr>
<tr>
<td>PQRS 7</td>
<td>TUV 8</td>
<td>WXYZ 9</td>
</tr>
<tr>
<td>*+</td>
<td>0</td>
<td>#</td>
</tr>
</tbody>
</table>

Figure 8.1: Multi-Tap Method

8.3.2 Predictive Text Entry (Dictionary in Malay)

In this method, to get the intended word, users have to press key in which the letter is present only once. Key 4 is pressed once then key 8 is pressed once and key 6 is pressed
once. After key 6 is pressed then the word “guna” will be displayed. If the word “guna” is found in dictionary find the match word. Therefore, the total number of key presses needed is 3 times. Figure 8.2 shows the number of times the key sequences performed in Predictive Text Entry method.

![Keyboard layout]

**Figure 8.2: Predictive Text Entry Method**

Table 8.1 shows the number of key pressed is reduced from 6 times to 3 times which is 50% reduction in key presses as well as time.

<table>
<thead>
<tr>
<th>Method of Implementation</th>
<th>Number of Key pressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Tap</td>
<td>6</td>
</tr>
<tr>
<td>Predictive Text Entry</td>
<td>3</td>
</tr>
<tr>
<td>Predictive Text Entry (proposed algorithm)</td>
<td>2</td>
</tr>
</tbody>
</table>

Comparatively the proposed algorithm is faster because within key 4 and 8 is pressed, if the intended word is found or match in the dictionary then it display the word within key presses twice. The MLPTET (Malay Language Predictive Text Entry Tool) has been developed using J2ME. The execution is viewed through the Toolkit Emulator mobile phone called NetBean IDE 6.1. Since the words are embedded in the memory, it makes the search for word faster. Since there is a drastic reduction in number of key presses, it saves a lot of and reduces the number of key presses. Language used feature problem are resolved because the proposed predictive text entry and intensive based dictionary accept new word to be added likely the language regularly change.
The data is gathered by making the user participates in the project by providing them with a mobile phone to use the mobile interface for a period of time. The participants are in different range of groups. The software is being designed as on going trials for the Malay language Predictive text entry for short message service on mobile phone in which valuable data are collected based on the users’ preferences and the frequently used words by the students. In this research, the need of the predictive text entry and examined the problems involved in the developing predictive text solution. MLPTET uses internal database of Malay words can contain approximately 500 words and can be added more than 700 words. In future the number of words can be increased to make it suitable for messaging because the feature is readily available to be used in real mobile devices. Besides, this software is supported and it can be transferred to proposed mobile device.

### 8.3.3 Comparison between MLPTET and Existing Tool

Table 8.2 shows the summary of the comparison of features between the developed tool (MLPTET) and existing tool for text messaging which was reviewed in Literature Review. The functionality of the MLPTET is that when a new word is entered it automatically captures the new entered word and allow user to store in the persistent storage (RecordStore). Basically, it has a function to check whether this new word exists in the dictionary before adding. However, it is user driven tool. User can choose to save the new word or skip the option. Another point to look at is that users can select any words or browse any words from the dictionary. Users have the option to choose to switch ON or OFF the predictive feature button before they start to write their SMS messages.
Another feature available in the system is to add new words and delete words. The developed tool has the facility to delete the words from the dictionary if necessary. There is a hidden option in the tool where system allows user to delete the word one by one or delete all the words in the persistent storage. The reason to have this option is to allow users to have the flexibility to store their frequently used words and delete the unintended word. This will reduce the storage volume. If the storage volume in the dictionary is lesser than the faster the word will be responded by the prediction tool. All the methods summarized in the Table 8.2, comparison on the functionality was discussed. Users finding that it is time consuming to write a simple message. It will be more difficult for those who is not good at their typing skill. This will again cause the error rate is high. What is unique in this MLPTET? The MLPTET uniqueness is that it has the capability in predicting and suggesting intended words. There is a facility to switch user-preferred language. It also has the capacity to add more than one language.

Table 8.2: Comparison between MLPTET and existing methods

<table>
<thead>
<tr>
<th>Scope</th>
<th>Multi-Tap</th>
<th>Dictionary based disambiguation</th>
<th>LetterWise</th>
<th>MLPTET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td>None</td>
<td>Character by Character</td>
<td>Character by Character</td>
<td>Intended Word Completion Word</td>
</tr>
<tr>
<td>Language</td>
<td>Non dictionary</td>
<td>Dependent on Dictionary</td>
<td>Non dictionary</td>
<td>Independent on dictionary</td>
</tr>
<tr>
<td>Key presses</td>
<td>Multiple</td>
<td>Single</td>
<td>Single</td>
<td>Multiple</td>
</tr>
<tr>
<td>Learning</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Addition of New words</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Deletion of word</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Additionally, MLPTET catered for the possibility of storage overflow when the dictionary reaches its maximum capacity. As an outcome, MLPTET is designed with a deletion function to enable users to delete any unused words from the dictionary. This option is not displayed in the interface but the feature has been catered. In earlier sections, an evaluation was conducted and discussed on the results. The MLPTET was evaluated against usability functions such as functionality, usability and accuracy and
last but not least the user satisfaction. The next chapter discusses on the results in a
closer scope.

8.3.4 Different Language and Techniques

The Table 8.3 illustrates the differences between languages and techniques which have
been developed in mobile application system.

<table>
<thead>
<tr>
<th>Foundation</th>
<th>Symbian</th>
<th>Java ME</th>
<th>.NET Compact</th>
<th>Microbrowser Based</th>
<th>Pocket PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Curve</td>
<td>C++</td>
<td>Java</td>
<td>C#, VB.NET</td>
<td>XHTML (WAP 2.0), WML</td>
<td>C, C++</td>
</tr>
<tr>
<td>Debuggers available</td>
<td>Difficult (unusual C++)</td>
<td>Average</td>
<td>Average</td>
<td>Varies by Server-side</td>
<td>Average (excellent)</td>
</tr>
<tr>
<td>Emulator available</td>
<td>Good on latest version.</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Good</td>
<td>Excellent</td>
</tr>
<tr>
<td>Integrated Development</td>
<td>Free Emulator, Sun Java</td>
<td>Bundled with IDE</td>
<td>Many</td>
<td>Bundled with IDE</td>
<td></td>
</tr>
<tr>
<td>Cross-Platform Deployment</td>
<td>Many choices</td>
<td>Eclipse, LMA NetBeans Mobility Pack</td>
<td>Visual Studio</td>
<td>Many</td>
<td>Visual Studio</td>
</tr>
</tbody>
</table>

8.4 Methodology

This method of this evaluation were sent to all the age group starting from below twenty
(20) to above fifty (50). However, there are different age of group users has been
requested to participate to use the MLPTET to verify and fulfill the system the
developed MLPTET system. Before assigning the task, a brief explanation and
guidelines are provided. However, the utilization of the text messaging is very essential
therefore it is easy to use the MLPTET. The aim is to get their response and perception
towards the questionnaire which is provided in Appendix D. The age group profiles of
the respondents are shown in Table 8.4. As mentioned earlier, out of the hundred and
one respondents, 54 (53%) are male respondents while 47 (47%) are female
respondents. Their age groups are shown in Figure 8.3.
Table 8.4: Age Group Profiles Respondents

<table>
<thead>
<tr>
<th>Age Group</th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
<th>Total %</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;20</td>
<td>10</td>
<td>8</td>
<td>18</td>
<td>17.8%</td>
</tr>
<tr>
<td>21-30</td>
<td>23</td>
<td>25</td>
<td>48</td>
<td>47.5%</td>
</tr>
<tr>
<td>31-40</td>
<td>16</td>
<td>10</td>
<td>26</td>
<td>25.7%</td>
</tr>
<tr>
<td>41-50</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>5.9%</td>
</tr>
<tr>
<td>&gt;50</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3.0%</td>
</tr>
<tr>
<td>Total</td>
<td>54</td>
<td>47</td>
<td>101</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

Majority of the respondents were in the group of 21-30 years old (48 respondents, 46%), followed by 31-40 years old (26 respondents, 26%), below 20 years old (18 respondents, 18 %), 41-50 years old (6 respondents, 6%) and above 50 years old (3 respondents, 3 %). The next section illustrates the evaluation results in graphs.

Figure 8.3: Age Groups of Respondents

8.5 Evaluation Methods and Results

For predictive text messaging, there are two main evaluation method which consist of speed and accuracy (Soukoreff et al., 2002). The text entry speed is measured as the number of characters entered by per second or average the number of trails SMS text
messaging with CPS (Continual Preparedness System). This is used to convert cps to words per minute. Words consist of characters regardless of letters, spaces, number or punctuation marks. For example, words per minute are then obtained by multiplying cps by sixty (seconds per minute) and dividing by five characters per word (MacKenzie, 2002).

According to the research conducted by (MacKenzie et al., 2002) during a text entry experiment, users were given a sentence to enter using the text entry method. Based on the text strings it is determined how many errors have been made. So, to do the text entry error analysis, there are two techniques such as minimum string distance error rate (MSD error rate) and keystrokes per character (KSPC) (MacKenzie et al., 2002). For predictive text entry method, speed of typing is measured and it is required but the measurement of speed is not part of this research. All the information gathered throughout the questionnaire is analyzed thoroughly in the following sections:

### 8.5.1 Users Response

User evaluation forms were distributed to a group of teenagers and others identified as adults in order to get their feedback on the MLPTET. Based on the evaluation form, refer to Appendix C a summary of the outcomes are presented and discussed in detail. The evaluation form is designed to get the users’ respond in terms of the systems.

In this research, a study was conducted using 18 teen participants’ 75 adults who can read and write Malay Language. Most of the participants’ responses were obtained through email. A total of hundred and fifty sets of the questionnaire were distributed to the Malaysian. From the 150 sets of questionnaire total of 101 sets responded by the entire range of age groups. The rest 49 did not respond to the questionnaire. The adult
participants responded quite well and responded the questionnaire but a few questionnaire forms have to be discarded due to incompleteness. Figure 8.4 and Table 8.5 shows that 101 participants have responded and 49 of them did not respond to the user’s evaluation.

Table 8.5: Participants responded and not responded by Groups

<table>
<thead>
<tr>
<th>Participants</th>
<th>Teenagers</th>
<th>Adults – Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responded</td>
<td>18</td>
<td>75</td>
</tr>
<tr>
<td>Not Responded</td>
<td>20</td>
<td>29</td>
</tr>
</tbody>
</table>

![Figure 8.4: Participants Responded by Groups](image)

Most of the users communicated through Internet and email due to difficulty to get in person to do the testing, due to the complexity not all users able to do the “hands-on” testing. However, the system is tested by 18 teenagers and 75 adults and others.

8.5.2 System Ease of Use

According to the research conducted by Drs. Erik (1998) in the Software Usability Measurement Inventory (SUMI) user evaluation questionnaire is one of the approach assessing qualities of the ease of use of the system by the end user. However, all different approaches stated in SUMI is tested and proven.
Figure 8.5 shows the usability of the predictive text entry tool. 82% of respondents found the tool to be simple to use and 5% found it is complex to use the predictive tool. 14% of the respondents found the tool is performing average and 60 % like the tool. Based on respondent’s outcomes, overall the tool proved to have a higher and good usability.

![Response of Usability](image)

**Figure 8.5: User Response on Tool Usability**

Based on the usability evaluation, Table 8.6 shows the percentage of participants responded on the usability options such as simple, complex to use the tool, tool is responding on functionality of the tool.

<table>
<thead>
<tr>
<th>Feedback</th>
<th>Number of Participants in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>82</td>
</tr>
<tr>
<td>Average</td>
<td>14</td>
</tr>
<tr>
<td>Complex</td>
<td>5</td>
</tr>
</tbody>
</table>

**Table 8.6: User Response for Usability of the MLPTET**

Figure 8.6 show that the respondents of age group 20-40 of preference on the predictive tool. In Figure 8.7 also indicates difference for those people who use predictive text and those who do not use. Overall, predictive text entry users reported 75 users use
prediction tool, while 26 non-users reported prefer to use multi-tap. While the difference falls higher statistical of 74% for user preferred predictive users, which significance the prediction tool approach and found 26% of respondents dislike the Non Dictionary method. Instead of challenges of representing the language use via mobile phone keypad, messaging text is a now routine in the daily lives of many people.

Based on the evaluation the hypothesis that has been set through the test was whether the use of prediction tool for text messages compared with SMS sent by those users using multi-tap input.

![Preferred Predictive Dictionary or Traditional](image)

**Figure 8.6: Evaluation on Preferred Method by Age Group**

Figure 8.7 shows user responded for the language preferred. The choice of language chosen by participants were mix language which is English and Malay language which is 21% and then next choice of preferences is English 43% and 36% of Malay language. This show that even the tool is available but the choice of language also plays an important role in the predictive input for text messaging.
As shown in Figure 8.8, 84% of the respondents found that the accuracy is effective and excellent for predictive text input messaging because they like to use the dictionary, the application is easy to learn, and it shortens the key pressing which is fascinating. 5% of the users found that the accuracy is ineffective because it is very complicated, not familiar to use, finding it difficult to get the intended word and this take time to add into the dictionary. Besides that they have also commented on their preferred mixed language and ‘slang’ words during the text messaging. 10% of the respondents found that the accuracy is average. However, the majority of the respondents found the overall accuracy of the MLPTET is effective.
The aim of the following is an additional test of the functionality of the MLPTET. The participants were given a questionnaire to answer whether the MLPTET is effective, average or ineffective. The effectiveness of the developed system has been evaluated by sending the questions to the participants. Figure 8.9 shows the response from the participants. However, it has resulted that a reasonable score for of 89 was achieved on the functionality effectiveness of MLPTET components overall and also the process. Next, 9 participants responded as average and a lower score of 2 was for the ineffective of the MLPTET.

![Figure 8.9: User Response on Functionality of MLPTET](image)

Figure 8.9: User Response on Functionality of MLPTET

Figure 8.10 shows 72% of the user responded that also on the performance with a good score.

![Figure 8.10: User Response on Performance of MLPTET](image)

Figure 8.10: User Response on Performance of MLPTET
Figure 8.11 shows 80% of the user responded that the Search and Add New Word into dictionary is very useful because they like to add their own words which they frequently use. The respondents found that editing to dictionary is good because they can edit the slang and their commonly use words. However, there are 20% of user found that this functionality is not very useful for them. A small group of user finds it difficult to edit word because not all the words are available in the dictionary.

![Figure 8.11: Search and Add New Word Function](image)

The results of words evaluation of MLPTET test has been presented in Table 8.7 present the results of the total error and its percentage for each question from question from 1 to 80. There are altogether 8 questions answered more than 15% of errors. That means 30 questions or 81.94% of the percent within 15% of errors. The table 8.7 shows that the total average percentage error is 18.06% that means the average percentage of correct score is 81.94%.
Table 8.7: Words Evaluation test Results predicted by MLPTET

<table>
<thead>
<tr>
<th>No</th>
<th>Words</th>
<th>Correct Answer</th>
<th>Never Answer</th>
<th>Total Error</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>abang</td>
<td>30</td>
<td>0</td>
<td>15</td>
<td>33.33</td>
</tr>
<tr>
<td>2</td>
<td>ada</td>
<td>42</td>
<td>0</td>
<td>3</td>
<td>6.67</td>
</tr>
<tr>
<td>3</td>
<td>adik lelaki</td>
<td>19</td>
<td>10</td>
<td>16</td>
<td>35.56</td>
</tr>
<tr>
<td>4</td>
<td>Ahad</td>
<td>39</td>
<td>3</td>
<td>3</td>
<td>6.67</td>
</tr>
<tr>
<td>5</td>
<td>air</td>
<td>41</td>
<td>1</td>
<td>3</td>
<td>6.67</td>
</tr>
<tr>
<td>6</td>
<td>aiskrim</td>
<td>10</td>
<td>8</td>
<td>27</td>
<td>60.00</td>
</tr>
<tr>
<td>7</td>
<td>ajak</td>
<td>35</td>
<td>1</td>
<td>9</td>
<td>20.00</td>
</tr>
<tr>
<td>8</td>
<td>akan</td>
<td>29</td>
<td>12</td>
<td>4</td>
<td>8.89</td>
</tr>
<tr>
<td>9</td>
<td>akhirnya</td>
<td>18</td>
<td>10</td>
<td>17</td>
<td>37.78</td>
</tr>
<tr>
<td>10</td>
<td>aku</td>
<td>31</td>
<td>6</td>
<td>8</td>
<td>17.78</td>
</tr>
<tr>
<td>11</td>
<td>alamat</td>
<td>31</td>
<td>8</td>
<td>6</td>
<td>13.33</td>
</tr>
<tr>
<td>12</td>
<td>ambil</td>
<td>31</td>
<td>8</td>
<td>6</td>
<td>13.33</td>
</tr>
<tr>
<td>13</td>
<td>anak dara</td>
<td>15</td>
<td>9</td>
<td>21</td>
<td>46.67</td>
</tr>
<tr>
<td>14</td>
<td>anak lelaki</td>
<td>16</td>
<td>12</td>
<td>17</td>
<td>37.78</td>
</tr>
<tr>
<td>15</td>
<td>anda</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>16</td>
<td>antara</td>
<td>39</td>
<td>2</td>
<td>4</td>
<td>8.89</td>
</tr>
<tr>
<td>17</td>
<td>apa</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>18</td>
<td>apa khabar?</td>
<td>5</td>
<td>20</td>
<td>20</td>
<td>44.44</td>
</tr>
<tr>
<td>19</td>
<td>apel</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>20</td>
<td>atas</td>
<td>36</td>
<td>2</td>
<td>7</td>
<td>15.56</td>
</tr>
<tr>
<td>21</td>
<td>atau</td>
<td>35</td>
<td>3</td>
<td>7</td>
<td>15.56</td>
</tr>
<tr>
<td>22</td>
<td>awak</td>
<td>35</td>
<td>4</td>
<td>6</td>
<td>13.33</td>
</tr>
<tr>
<td>23</td>
<td>ayah</td>
<td>40</td>
<td>2</td>
<td>3</td>
<td>6.67</td>
</tr>
<tr>
<td>24</td>
<td>bagaimana</td>
<td>38</td>
<td>2</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>25</td>
<td>bagus</td>
<td>40</td>
<td>0</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>26</td>
<td>baiklah</td>
<td>38</td>
<td>0</td>
<td>7</td>
<td>15.56</td>
</tr>
<tr>
<td>27</td>
<td>balik</td>
<td>26</td>
<td>0</td>
<td>19</td>
<td>42.22</td>
</tr>
<tr>
<td>28</td>
<td>banyak</td>
<td>42</td>
<td>0</td>
<td>3</td>
<td>6.67</td>
</tr>
<tr>
<td>29</td>
<td>bebas</td>
<td>26</td>
<td>6</td>
<td>13</td>
<td>28.89</td>
</tr>
<tr>
<td>30</td>
<td>belajar</td>
<td>42</td>
<td>3</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>31</td>
<td>berbahaya</td>
<td>30</td>
<td>11</td>
<td>4</td>
<td>8.89</td>
</tr>
<tr>
<td>32</td>
<td>besar</td>
<td>28</td>
<td>9</td>
<td>8</td>
<td>17.78</td>
</tr>
<tr>
<td>33</td>
<td>boleh</td>
<td>41</td>
<td>4</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>34</td>
<td>cakap</td>
<td>35</td>
<td>0</td>
<td>10</td>
<td>22.22</td>
</tr>
<tr>
<td>35</td>
<td>cuti</td>
<td>44</td>
<td>0</td>
<td>1</td>
<td>2.22</td>
</tr>
</tbody>
</table>
Table 8.7: Words Evaluation test Results predicted by MLPTET, continued

<table>
<thead>
<tr>
<th>No</th>
<th>Words</th>
<th>Correct Answer</th>
<th>Never Answer</th>
<th>Total Error</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>dalam</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>37</td>
<td>dapat</td>
<td>43</td>
<td>0</td>
<td>2</td>
<td>4.44</td>
</tr>
<tr>
<td>38</td>
<td>dua</td>
<td>32</td>
<td>3</td>
<td>10</td>
<td>22.22</td>
</tr>
<tr>
<td>39</td>
<td>empat</td>
<td>32</td>
<td>7</td>
<td>6</td>
<td>13.33</td>
</tr>
<tr>
<td>40</td>
<td>guna</td>
<td>35</td>
<td>2</td>
<td>8</td>
<td>17.78</td>
</tr>
<tr>
<td>41</td>
<td>hari ini</td>
<td>10</td>
<td>18</td>
<td>17</td>
<td>37.78</td>
</tr>
<tr>
<td>42</td>
<td>hari itu</td>
<td>10</td>
<td>25</td>
<td>10</td>
<td>22.22</td>
</tr>
<tr>
<td>43</td>
<td>Ini</td>
<td>39</td>
<td>0</td>
<td>6</td>
<td>13.33</td>
</tr>
<tr>
<td>44</td>
<td>itu</td>
<td>36</td>
<td>0</td>
<td>9</td>
<td>20.00</td>
</tr>
<tr>
<td>46</td>
<td>lambat</td>
<td>40</td>
<td>0</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>47</td>
<td>lima</td>
<td>40</td>
<td>0</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>48</td>
<td>luar</td>
<td>21</td>
<td>15</td>
<td>9</td>
<td>20.00</td>
</tr>
<tr>
<td>49</td>
<td>makan</td>
<td>39</td>
<td>2</td>
<td>4</td>
<td>8.89</td>
</tr>
<tr>
<td>50</td>
<td>nak</td>
<td>20</td>
<td>6</td>
<td>19</td>
<td>42.22</td>
</tr>
<tr>
<td>51</td>
<td>pergi</td>
<td>20</td>
<td>0</td>
<td>25</td>
<td>55.56</td>
</tr>
<tr>
<td>52</td>
<td>rehat</td>
<td>43</td>
<td>0</td>
<td>2</td>
<td>4.44</td>
</tr>
<tr>
<td>53</td>
<td>rumah</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>54</td>
<td>satu</td>
<td>44</td>
<td>0</td>
<td>1</td>
<td>2.22</td>
</tr>
<tr>
<td>55</td>
<td>saya</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>56</td>
<td>sayap</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>57</td>
<td>sekolah</td>
<td>24</td>
<td>10</td>
<td>11</td>
<td>24.44</td>
</tr>
<tr>
<td>58</td>
<td>senyap</td>
<td>39</td>
<td>1</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>59</td>
<td>suka</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>60</td>
<td>sungguh</td>
<td>38</td>
<td>1</td>
<td>6</td>
<td>13.33</td>
</tr>
<tr>
<td>61</td>
<td>tadi</td>
<td>43</td>
<td>1</td>
<td>1</td>
<td>2.22</td>
</tr>
<tr>
<td>62</td>
<td>tanpa</td>
<td>44</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>63</td>
<td>telah</td>
<td>20</td>
<td>11</td>
<td>14</td>
<td>31.11</td>
</tr>
<tr>
<td>64</td>
<td>tidak</td>
<td>25</td>
<td>7</td>
<td>13</td>
<td>28.89</td>
</tr>
<tr>
<td>65</td>
<td>tiga</td>
<td>25</td>
<td>5</td>
<td>15</td>
<td>33.33</td>
</tr>
<tr>
<td>66</td>
<td>tikar</td>
<td>44</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>67</td>
<td>tinggal</td>
<td>37</td>
<td>3</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>68</td>
<td>tolong</td>
<td>25</td>
<td>5</td>
<td>15</td>
<td>33.33</td>
</tr>
<tr>
<td>69</td>
<td>tua</td>
<td>40</td>
<td>0</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>70</td>
<td>tulang</td>
<td>5</td>
<td>15</td>
<td>25</td>
<td>55.56</td>
</tr>
<tr>
<td>71</td>
<td>tulis</td>
<td>26</td>
<td>2</td>
<td>17</td>
<td>37.78</td>
</tr>
<tr>
<td>72</td>
<td>turun</td>
<td>25</td>
<td>4</td>
<td>16</td>
<td>35.56</td>
</tr>
<tr>
<td>73</td>
<td>tutup</td>
<td>27</td>
<td>8</td>
<td>10</td>
<td>22.22</td>
</tr>
<tr>
<td>74</td>
<td>umur</td>
<td>10</td>
<td>11</td>
<td>24</td>
<td>53.33</td>
</tr>
<tr>
<td>75</td>
<td>untuk</td>
<td>32</td>
<td>1</td>
<td>12</td>
<td>26.67</td>
</tr>
<tr>
<td>76</td>
<td>warna</td>
<td>25</td>
<td>6</td>
<td>14</td>
<td>31.11</td>
</tr>
<tr>
<td>77</td>
<td>wayang</td>
<td>20</td>
<td>6</td>
<td>19</td>
<td>42.22</td>
</tr>
<tr>
<td>78</td>
<td>ya</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>79</td>
<td>sayang</td>
<td>40</td>
<td>0</td>
<td>5</td>
<td>11.11</td>
</tr>
<tr>
<td>80</td>
<td>yang</td>
<td>40</td>
<td>2</td>
<td>3</td>
<td>6.67</td>
</tr>
</tbody>
</table>

Total Error 335 650 1444.44
Average Error 18.06
Total Correct 81.94

Finally, as shown in Figure 8.12, overall the respondents are very satisfied with the Predictive Text Entry Tool in Malay language which is called as MLPTET. Based on the evaluation the respondents voted 70% very satisfied of user acceptance and 23%
satisfied and remaining 7% found not satisfied with the system. However majority has voted the system is very satisfactory.

![User Acceptance on MLPTET](image)

**Figure 8.12: User Acceptance on MLPTET**

8.6 System Strength

Generally, based on the participants’ feedback, overall the MLPTET have demonstrated its capabilities which are hardly found in the Multi-Tap SMS and also other methods that have been compared in the previous sections. Based on the evaluation the MLPTET has achieved to predictive text entry in Malay language and overall users have shown the satisfaction on the MLPTET. Next is that MLPTET can accommodate more than one languages mainly English because there is no need for remapped keypad is required. If the words are available in the persistent storage then it will predict the intended words. MLPTET is a simple interface which is easy to operate and user-friendly.

8.7 System Weaknesses

Principally, apart from the strengths all the system has it weaknesses. There are words where users intended to use in their text messaging not available in the persistent storage. If a user intends to add the new word which is a frequently used word than the
capacity of the storage will be exceeded and may cause the performance of the predictive text to be lower. The MLPTET is not able to predict all the words entered by user. There are still rooms for improvement because of the disambiguation. The user enters an ambiguous sequence. As for ambiguous sequence, the word-based entry from among various predictive methods. This system has dictionary driven feature. Thus, this system can be used for evaluation of dictionary.

8.8 Summary of the Chapter

This chapter presents an evaluation of the implementation of MLPTET which is based on few subjects of evaluation. This chapter evaluation techniques has been designed in the language of text entry research. The MLPTET was evaluated against the requirements elicited from the previous chapters. Generally the outcomes of the evaluation were very positive. The participants have pointed out some of the features than can be added in the future enhancement. This research is performed where all the requirements elicited in Chapter 5 are verified and fulfilled according to its module and requirement ID.
CHAPTER 9: CONCLUSION AND OUTCOME

9.1 Research Summary

This research and development work highlights on the development of SMS Message Prediction tool namely MLPTET.

MLPTET is a tool to assist individual public to simplify the writing of text by auto completing the text entered by the user. It is a tool that allows users to enter the personal predicated word in their database which will be used for their personal message entry prediction.

This chapter summarizes the thesis by determining how MLPTET has satisfied the objectives as identified in Chapter 1. Then, it discusses on the research outcome, limitation and the future enhancement of the message prediction tool.

9.2 Achievement of Objectives

Chapter 1 identified objectives of this research for Malay Predictive Text Entry Tool (MLPTET).

The first objective is to develop and design a tool with the Predictive text entry algorithm for Short Message Service on mobile phone. The outcome is, the objective was fulfilled where by the MLPTET has successfully created a database which allows the dictionary data being stored accordingly. A search algorithm was designed in MLPTET to automatically capture the words type and display on the mobile text messaging screen. This has reduced the multiple key presses.
Besides having the capabilities of the word prediction, it is also enhanced with the additional function of retrieving word and sentence, which allows user to browse all the prediction words that exist in their dictionary and browse all prediction word. For example, if the dictionary consist of “atas ( di atas )”, when user entered “atas”, it will become “atas ( di atas )”. Other than that, the millisecond feature is also incorporated to predict the data will be reduced to speed up the data entry process. Currently the timer is set to 3000 milliseconds interval. It is time consuming to split the text, lower case the text for better comparison in the database.

The second objective is to develop Predictive Text Entry Tool in Malay language. To achieve this objective, a predictive tool in Malay language dictionary are developed. A self-dictionary is provided for user to enter the new word in future. Meanwhile, if the dictionary consist empty record, one import function also provided for the user to import the data which has been embedded in the system. Allow to import the excel data in the system. Currently the sample data is under the embed array in order for demonstration purpose.

The third objective is to test and evaluate the implementation approach of the Malay language prediction tool in Java platform based on the usability and accuracy functionality. As best practices for mobile application, this system is developed in Java platform (J2ME). The predictive text entry is developed and the effort required to implement Malay language prediction tool for SMS on mobile phone in J2ME platforms is also evaluated as part of the objective. To measure the predictive text entry, testing tools are used to evaluate the predictive text entry. The MLPTET recognized as a usable predictive text entry.
The research objective and objectives that are stated in chapter one is fulfilled by the MLPTET (Malay Language Predictive Text Entry Tool)

9.3 Future Enhancement

The MLPTET still has room for improvements because the functionalities are kept only to develop a predictive text entry for text message, an optimized engine for Malay Language. Besides that there are need for an improvement on the time message input over the standard predictive text entry model and improved measure on the usability and accuracy of the search engine.

9.4 Conclusions

This chapter wraps-up the research that was carried out in this research. The conclusion shows how the objectives of the research were fulfilled. The research study was mainly on predictive text entry in Malay language for texting messages using mobiles phones. The main factors are the satisfaction of users in texting message in Malay language which will eventually increase the usage of National language without using abbreviation words. It was concluded that the implementation, testing and evaluation of MLPTET has fulfilled the objectives of providing a tool with a reasonably good prediction support in Malay language for texting messages. Thus, the task of programming in Java platform has increased the knowledge on the mobile application.
Afsaneh Fazly (2002). *The Use of Syntax in Word Completion Utilities*. A thesis submitted in Conformity with the requirements for the degree of Master of Science, Graduate Department of Computer Science, University of Toronto.


http://www.associatedcontent.com/article/448581/study_text_messaging_while_driving.html
(Accessed on February 2009).


### Table 1: Use Case Description for the Auto Search Option Component

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Search Option Setting</td>
<td><strong>Actor</strong>:</td>
</tr>
<tr>
<td></td>
<td>User</td>
</tr>
<tr>
<td></td>
<td>Goal:</td>
</tr>
<tr>
<td></td>
<td>On/Off auto search setting</td>
</tr>
<tr>
<td><strong>Pre-condition</strong>:</td>
<td>Click on the main screen</td>
</tr>
<tr>
<td><strong>Success post-condition</strong>:</td>
<td>The user switch on/off the auto search function</td>
</tr>
<tr>
<td><strong>Main Success Scenario</strong>:</td>
<td>1. The user select the ‘Predict Option’ item</td>
</tr>
<tr>
<td></td>
<td>2. The user click on ‘Select’ button</td>
</tr>
<tr>
<td></td>
<td>3. The user select the on/off option</td>
</tr>
<tr>
<td></td>
<td>4. The user click ‘Enter’ button</td>
</tr>
<tr>
<td></td>
<td>5. The search option value will be reset</td>
</tr>
<tr>
<td><strong>Extensions</strong>:</td>
<td>None</td>
</tr>
</tbody>
</table>

### Table 2: Use Case Description for the Text Input Component

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Input Component</td>
<td><strong>Actor</strong>:</td>
</tr>
<tr>
<td></td>
<td>User</td>
</tr>
<tr>
<td></td>
<td>Goal:</td>
</tr>
<tr>
<td></td>
<td>Predict the message enter in the input screen</td>
</tr>
<tr>
<td><strong>Pre-condition</strong>:</td>
<td>Click on the main screen</td>
</tr>
<tr>
<td><strong>Success post-condition</strong>:</td>
<td>The word will be auto completed for just few character entered by the user.</td>
</tr>
<tr>
<td><strong>Main Success Scenario</strong>:</td>
<td>1. The user select the ‘Message’ item</td>
</tr>
<tr>
<td></td>
<td>2. The user click on ‘Select’ button</td>
</tr>
<tr>
<td></td>
<td>3. The user select key in the characters</td>
</tr>
<tr>
<td></td>
<td>4. The system auto replace the characters into word which consist in the dictionary</td>
</tr>
<tr>
<td><strong>Extensions</strong>:</td>
<td>If predict option is set to ‘no’, the word will not be auto predicted.</td>
</tr>
</tbody>
</table>
Table 3: Use Case Description for the SMS Send Component

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
</table>
| SMS Send Component  | **Actor**: User  
|                     | Goal: Predict the message enter in the input screen  
|                     | Pre-condition: The user has entered the message and click on the ‘Send’  
|                     | Success post-condition: The message will be sent.  
|                     | **Main Success Scenario:**  
|                     | 1. The user select the ‘Send’ item  
|                     | 2. The system displays the ‘Phone Number’ entry screen  
|                     | 3. The user enter the phone number and ‘OK’  
|                     | 4. The ‘SMS has been sent’ alert screen will be displayed  
|                     | 5. The user will be directed to the main screen  
|                     | **Extensions:** None |

Table 4: Use Case Description for the Browse Dictionary Component

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
</table>
| Browse Dictionary     | **Actor**: User  
| Component             | Goal: Browse all the word consist in the dictionary  
|                       | Pre-condition: Click on the main screen  
|                       | Success post-condition: All the words exist in the database will be displayed.  
|                       | **Main Success Scenario:**  
|                       | 1. The user select the ‘Browse all word’ item  
|                       | 2. The user click on ‘Select’ button  
|                       | 3. The system will display all the word exist in the database  
|                       | **Extensions:**  
<p>|                       | If the database has no record, the ‘Import Data’ button will be display for the user to insert the existing words exist in the system to the database |</p>
<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
</table>
| Add New Word Component   | **Actor:** User  
|                          | **Goal:** Predict the message enter in the input screen  
|                          | **Pre-condition:** Click on the main screen  
|                          | **Success post-condition:** The word entered by the user will be inserted into the database.  
|                          | **Main Success Scenario:**  
|                          | 1. The user select the ‘Add New Word’ item  
|                          | 2. The user click on ‘Select’ button  
|                          | 3. The user enter the word  
|                          | 4. The user click ‘Add’ button  
|                          | 5. The system will display the ‘Record Added’ message  
|                          | 6. The user will be directed to the main screen  
|                          | **Extensions:** None |
TEST PLAN FORM

<table>
<thead>
<tr>
<th>No</th>
<th>Test Scenario</th>
<th>Expected Result</th>
<th>Actual Result</th>
<th>Remarks</th>
<th>Completion Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter the first 2 character of word that is exist in the database</td>
<td>Word will be auto replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
<td>12/12/2009</td>
</tr>
<tr>
<td>2</td>
<td>Enter the first 2 character of sentence that is exist in the database</td>
<td>Sentence will be auto replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
<td>12/12/2009</td>
</tr>
<tr>
<td>3</td>
<td>Enter the word that is not exist in the database</td>
<td>Word will not be replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
<td>12/12/2009</td>
</tr>
<tr>
<td>4</td>
<td>Enter the first 2 character of word that is exist in the database</td>
<td>Word will be only auto replaced when the [Search] button being clicked</td>
<td>Same as expected</td>
<td>The auto predict feature is off</td>
<td>12/12/2009</td>
</tr>
<tr>
<td>5</td>
<td>Enter the space after the word</td>
<td>The word before the space will not be auto replaced</td>
<td>Same as expected</td>
<td>The auto predict feature is on</td>
<td>12/12/2009</td>
</tr>
</tbody>
</table>
private Screen genEntryScr() {
    if (entryScr == null) {
        entryScr = new Form("Add New Word");
        entryScr.addCommand(cmdCancel);
        entryScr.addCommand(cmdAdd);
        entryScr.setCommandListener(this);
        e_lastName = new TextField("Word:", "", LN_LEN, TextField.ANY);
        entryScr.append(e_lastName);
    }
    e_lastName.delete(0, e_lastName.size());
    display.setCurrent(entryScr);
    return entryScr;
}

private Screen genSearchScr() {
    if (s_lastName == null) {
        display = Display.getDisplay(this);
        s_lastName = new TextBox("0/255", "", LN_LEN, TextField.ANY);
        s_lastName.addCommand(cmdSearchLocal);
        s_lastName.addCommand(cmdPhone);
        s_lastName.addCommand(cmdBack);
        s_lastName.setCommandListener(this);
    }
    display.setCurrent(s_lastName);
    aTimer = new Timer();
    aTimerTask = new SimpleTimerTask();
    aTimer.scheduleAtFixedRate(aTimerTask, 0, 3000);
    return s_lastName;
}
private void replaceMsg(String msg, String l, boolean local) {
    SimpleComparator sc;
    SimpleFilter sf = null;
    RecordEnumeration re = null;

    sc = new SimpleComparator(SimpleComparator.SORT_BY_LAST_NAME);
    sf = new SimpleFilter(l);
    try {
        re = addrBook.enumerateRecords(sf, sc, false);
    } catch (Exception e) {
        s_lastName.delete(0, msg.length());
        s_lastName.setString("Could not create enumeration: " + e);
    }

    String word = msg;
    String original = word;
    word = word.substring(0, word.length() - l.length());
    if (re.hasNextElement()) {
        try {
            if (re.hasNextElement()) {
                byte[] b = re.nextRecord();
                word = (word + SimpleRecord.getLastName(b));
            }
        } catch (Exception e) {
            s_lastName.delete(0, msg.length());
            s_lastName.setString("Error while building word list: " + e);
        }
        s_lastName.delete(0, msg.length());
        s_lastName.setString(word + "+/255");
    } else {
        s_lastName.setTitle(msg.length() + "+/255");
    }

    aTimer.cancel();
    aTimerTask.cancel();
    aTimer = new Timer();
    aTimerTask = new SimpleTimerTask();
    aTimer.scheduleAtFixedRate(aTimerTask, 0, 3000);
}
QUESTIONNAIRE SURVEY

THIS SECTION CONSISTS OF QUESTIONS REGARDING:

(A) Post – test questionnaire

(B) User Evaluation form for Malay Predictive Text Entry

You are required to complete all sections in this questionnaire form. The aim of this questionnaire is to gather requirements prior to development and User Evaluation.

Thank you for your kind co-operation.

Reference No: [ ] [ ]
(A) POST-TEST QUESTIONNAIRE

Please tick (✓) ONE of the following answers that best suits you:

1. Please tick on the Gender
   - Female
   - Male

2. Please tick on the Age Group
   - 20 and below
   - 21-30
   - 31-40
   - 41-50
   - 50 and above

3. Category
   - Teenagers
   - Adults
   - Others

4. Please tick on the race
   - Malay
   - Chinese
   - Indian
   - Others, please specify______________________

5. If you use a mobile phone, please answer the following.
   a. What kind of mobile phone do you currently use:____________________
   b. Do you enter SMS or email messages on your mobile phone?
      - Yes
      - No
   c. If you use SMS or email messaging on your mobile phone, which of the following do you use to enter text on your mobile phone? Check all that apply.
      - Multi-tap
      - T9 predictive text entry option (Dictionary)
   d. What language do you prefer to Short Message Service (SMS) on your mobile phone?
5. If you have ever used predictive text entry, how good are you at this method of text entry?

☐ Very good
☐ Good
☐ Average
☐ Poor
(B) USER EVALUATION FORM FOR MLPTET

6. How would you evaluate your overall experience using this form of predictive text entry for mobile phone?
   □ Simple
   □ Average
   □ Complex

7. How quickly were you able to master entering text in the mobile phone using the provided method?
   □ Simple
   □ Average
   □ Complex

8. How would you rate on the search and adding new words into dictionary? Do you find this feature useful?
   □ Yes
   □ No

9. How do you find the Predictive Text Entry Tool in Malay Language and what is the level of Acceptance?
   □ Very Satisfied
   □ Satisfied
   □ Not Satisfied

10. What do you measure the speed of (MLPTET) predictive text entry?
    i. What is the speed of the text entry using predictive text entry and Multi-Tap? How many characters or words in length able to enter?
       a. □ Predictive entry
          ________________________________
       b. □ Multi-Tap
          ________________________________
       c. □ Used both techniques interchangeably.

    ii. Do you use dialect or slang in Malay Language? What is the common dialect used? For example ‘lah’
        □ Always
           If other than above, please specify ____________________________
        □ Sometimes
        □ Never
iii. Do you use special characters selections such as comma, dash?
  
  □ Always  
  □ Sometimes  
  □ Never

iv. Do you use abbreviations when SMS in Malay language?
  
  □ Always  
  Please provide the abbreviations used__________________________
  □ Sometimes  
  □ Never

v. Is there anything else you want from the predictive text entry (MLPTET)?


11. Efficiency
  
i. Help to select appropriate words more quickly?
  
  □ Effective  
  □ Average  
  □ Ineffective

ii. Response time of the MLPTET is acceptable?
  
  □ Good  
  □ Average  
  □ Poor

12. Error Prevention
  
i. Easy to make mistakes?
  
  □ Simple  
  □ Average  
  □ Complex

ii. Easy to correct the mistakes?
  
  □ Simple  
  □ Average  
  □ Complex

13. How effective is the use of MLPTET components in facilitating on the functionality?
  
  □ Effective  
  □ Average  
  □ Ineffective
14. How effective is the use of MLPTET components in facilitating on the accuracy?
   ☐ Effective
   ☐ Average
   ☐ Ineffective

15. What do you hate about the current predictive text entry? (you can tick more than one box)
   ☐ Stupid suggestions
   ☐ It never learns
   ☐ Difficult to add new words
   ☐ It’s not intelligent
   ☐ It’s not really predicting, its just completing words

16. How is the overall evaluation?
   i. Likely recommended to other users?
      ☐ Strongly
      ☐ Average
      ☐ Strongly Disagree
   
   ii. Overall, I am satisfied with the MLPTET system?
       ☐ Strongly
       ☐ Average
       ☐ Strongly Disagree

17. In your opinion, what are the possible enhancements which could be added to improve the MLPTET?

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________

18. Do you know any other MLPTET which are available elsewhere? Any comment about it?

   ____________________________________________________________
   ____________________________________________________________
   ____________________________________________________________
19. Please tick (√) any options of the following answers that best describes your perception towards the existing SMS methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Multi-Tap</th>
<th>Dictionary based Disambiguation</th>
<th>Letter Wise</th>
<th>MLPTET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction Methods</td>
<td>A. None</td>
<td>A. None</td>
<td>A. None</td>
<td>A. None</td>
</tr>
<tr>
<td></td>
<td>B. By Character</td>
<td>B. By Character</td>
<td>B. By Character</td>
<td>B. By Character</td>
</tr>
<tr>
<td></td>
<td>C. Word completion</td>
<td>C. Word completion</td>
<td>C. Word completion</td>
<td>C. Word completion</td>
</tr>
<tr>
<td>Language used</td>
<td>A. Non Dictionary</td>
<td>A. Non Dictionary</td>
<td>A. Non Dictionary</td>
<td>A. Non Dictionary</td>
</tr>
<tr>
<td>Is key presses</td>
<td>A. Single</td>
<td>A. Single</td>
<td>A. Single</td>
<td>A. Single</td>
</tr>
<tr>
<td></td>
<td>B. Multiple</td>
<td>B. Multiple</td>
<td>B. Multiple</td>
<td>B. Multiple</td>
</tr>
<tr>
<td>Is this easy to learn</td>
<td>A. Yes</td>
<td>A. Yes</td>
<td>A. Yes</td>
<td>A. Yes</td>
</tr>
<tr>
<td></td>
<td>B. No</td>
<td>B. No</td>
<td>B. No</td>
<td>B. No</td>
</tr>
<tr>
<td>Is the dictionary has the ability to add new words</td>
<td>A. Yes</td>
<td>A. Yes</td>
<td>A. Yes</td>
<td>A. Yes</td>
</tr>
<tr>
<td></td>
<td>B. No</td>
<td>B. No</td>
<td>B. No</td>
<td>B. No</td>
</tr>
<tr>
<td>Is the dictionary has the ability to delete words</td>
<td>A. Yes</td>
<td>A. Yes</td>
<td>A. Yes</td>
<td>A. Yes</td>
</tr>
<tr>
<td></td>
<td>B. No</td>
<td>B. No</td>
<td>B. No</td>
<td>B. No</td>
</tr>
</tbody>
</table>

20. Please tick (√/×) in box next to the predictive text inputted words in the Malay Language. From the 80 words, test the system how many MLPTET able to predict.

<table>
<thead>
<tr>
<th>Words</th>
<th>√ / ×</th>
<th>Words</th>
<th>√ / ×</th>
<th>Words</th>
<th>√ / ×</th>
<th>Words</th>
<th>√ / ×</th>
</tr>
</thead>
<tbody>
<tr>
<td>abang</td>
<td></td>
<td>atas</td>
<td></td>
<td>guna</td>
<td></td>
<td>sungguh</td>
<td></td>
</tr>
<tr>
<td>ada</td>
<td></td>
<td>atau</td>
<td></td>
<td>hari ini</td>
<td></td>
<td>tadi</td>
<td></td>
</tr>
<tr>
<td>adik lelaki</td>
<td></td>
<td>awak</td>
<td></td>
<td>hari itu</td>
<td></td>
<td>tanpa</td>
<td></td>
</tr>
<tr>
<td>Ahad</td>
<td></td>
<td>ayah</td>
<td></td>
<td>Ini</td>
<td></td>
<td>telah</td>
<td></td>
</tr>
<tr>
<td>air</td>
<td></td>
<td>bagaimana</td>
<td></td>
<td>itu</td>
<td></td>
<td>tidak</td>
<td></td>
</tr>
<tr>
<td>aiskrim</td>
<td></td>
<td>bagus</td>
<td></td>
<td>ke</td>
<td></td>
<td>tiga</td>
<td></td>
</tr>
<tr>
<td>ajak</td>
<td></td>
<td>baiklah</td>
<td></td>
<td>lambat</td>
<td></td>
<td>tikar</td>
<td></td>
</tr>
<tr>
<td>akan</td>
<td></td>
<td>balik</td>
<td></td>
<td>lima</td>
<td></td>
<td>tinggal</td>
<td></td>
</tr>
<tr>
<td>akan</td>
<td></td>
<td>banyak</td>
<td></td>
<td>luar</td>
<td></td>
<td>tolong</td>
<td></td>
</tr>
<tr>
<td>akhirnya</td>
<td></td>
<td>bebas</td>
<td></td>
<td>makan</td>
<td></td>
<td>tua</td>
<td></td>
</tr>
<tr>
<td>aku</td>
<td></td>
<td>belajar</td>
<td></td>
<td>nak</td>
<td></td>
<td>tulang</td>
<td></td>
</tr>
<tr>
<td>alamat</td>
<td></td>
<td>berbahaya</td>
<td></td>
<td>pergi</td>
<td></td>
<td>tulis</td>
<td></td>
</tr>
<tr>
<td>ambil</td>
<td></td>
<td>besar</td>
<td></td>
<td>rehat</td>
<td></td>
<td>turun</td>
<td></td>
</tr>
<tr>
<td>anak dara</td>
<td></td>
<td>boleh</td>
<td></td>
<td>rumah</td>
<td></td>
<td>tutup</td>
<td></td>
</tr>
<tr>
<td>anak lelaki</td>
<td></td>
<td>cakap</td>
<td></td>
<td>satu</td>
<td></td>
<td>umur</td>
<td></td>
</tr>
<tr>
<td>anda</td>
<td></td>
<td>cuti</td>
<td></td>
<td>saya</td>
<td></td>
<td>untuk</td>
<td></td>
</tr>
<tr>
<td>antara</td>
<td></td>
<td>dalam</td>
<td></td>
<td>sayap</td>
<td></td>
<td>warna</td>
<td></td>
</tr>
<tr>
<td>apa</td>
<td></td>
<td>dapat</td>
<td></td>
<td>sekolah</td>
<td></td>
<td>wayang</td>
<td></td>
</tr>
<tr>
<td>apa khabar?</td>
<td></td>
<td>dua</td>
<td></td>
<td>senyap</td>
<td></td>
<td>ya</td>
<td></td>
</tr>
<tr>
<td>apel</td>
<td></td>
<td>empat</td>
<td></td>
<td>suka</td>
<td></td>
<td>yang</td>
<td></td>
</tr>
</tbody>
</table>

END OF QUESTIONNAIRE
~ THANK YOU ~
1.1 Installation Guide Sequence

To install the Malay Language Predictive Text Entry Tool (MLPTET) please follow the guidelines.

- Install NetBeans IDE 6.1
- Java(TM) ME platform SDK 3.0, EA
- J2ME 3.0
- Sun Java (TM) Wireless Toolkit 2.5.2 for CLDC
- Plugin 2 category
- Mobility
- Visual Mobility Designer
- Mobile Application

1.2 User Guide

To start the mobile emulator in Java double click on the java, double click the respective icon on the desktop.

Once it is loaded, the mobile emulator will be prompted as shown in Figure 1.1.
Figure 1.1: Mobile Emulator

Click on Launch to display Main Menu

Click Select to choose the option from the Menu.

Use down arrow ↓ and ↑ up arrow key to select the options

Select Message to write text messages

Select Add New Word to add new word in the dictionary

Select Predict Option to ON and OFF

Select Browse all word to display all the words in the dictionary

Select Exit to exit from the Menu

Click on the right top button to exit from the emulator
Please enter your text message on the text screen.

Select Search to search any word in the dictionary.

After completing writing the message, select SEND to send the message.

Select Back to return to the Main Menu.

Exit from Emulator.
Enter the new word in the box and Select Add to add new

Select the browse all option to browse all the words from the dictionary

Select Back to return to Main Menu