CHAPTER 5

System Design

5.1 Introduction

Based on the data analysis and data finding in chapter four, this chapter discusses the system design. The first part of this chapter is to determine system structure, which contains the system concept and the system modules. The second part of this chapter is the functional requirement for the system (system Use Case). The third part of this chapter is to determine the nonfunctional requirements that represent the system abilities and flexibilities, and fourth part of this chapter is the system interfaces and finally is the system tables and code.

5.2 System Structure

5.2.1 System Concept

An integrated health care management system is a computer application to manage the health care information allowing the user to access their information based on the user privileges. Moreover, this application can be used online and it typically offers a wide range of information content, applications and services, integrated into a single-theme interface that is easy to navigate and reflects interests of different users and allows them to access information from multiples sources based on their privileges (figure 5.1).

In addition, the application provides the following:

a. Offers an easy and flexible infrastructure that organizations do not need to develop.

b. Allow users to visually arrange components into a customized screen display.

c. Make it possible for non-technical users to choose information to be displayed.

d. The application presents a unified interface to back office business systems.
5.2.2 System modules

i. Patient module
This module is for managing all the patient information such as (personal information, consultation information, medication information and payment information).

ii. Doctor module
This module manages the doctors’ schedule and keeps track of the doctors’ status as well as availability.

iii. Appointment module
This module is for managing the patient’s appointments with the doctors.

iv. Product module
This module for managing the products (medicine) in the health care and keep tracking for all the information related to the product such as (expire date, suppliers name, availability in the health care pharmacy, etc).

v. Client module
This module is for managing all the information related to health care clients.

vi. Document module
This module manages all the information related to health care documents and all the documents related to the patients.

vii. Task module
All the user tasks inside the health care is managed by this module.
viii. **Account module**
This module for managing all the accounts information in the health care

ix. **User module**
This module for managing all the information related to the system users and their privileges.

### 5.2.3 System Environment

This application developed using Oracle Application Express (Apex), which provides everything necessary to develop and implement any application to share information. The following is the scales and the tools that have been used in apex to develop this web-based:

i. HTML for building the web-based forms.

ii. SQL for the inquiry purpose from the database.

iii. PL/SQL is the programming code.

iv. Java script, for system structure and validation.

v. CSS for the interface design to make the system interface more attractive.

vi. FUSION Chart for reports analysis chart.

The benefit for using Apex to develop the web based is:

i. Provides a complete and integrated application platform.

ii. Enable a fast, scalable Web Site with pre-assembled set of scripting, runtime, and monitoring tools.

iii. Simplify information sharing, internally and externally, with Instant web based and Enterprise web based.

iv. Provide easy access to data stored in databases. A perfect complement to Oracle Database 10g Standard Edition One.

v. Platform flexibility to pick the best, most cost-effective hardware. Choose Linux, Windows, or both.
5.3 Functional Requirement

The functional requirement (Use Case) is detailed out to provide the explanation for the system function and details about system requirement specification. System Use Case (Appendix C) is a describing the interaction between the user and the system functions. In this system there is a user call administrator created by the developer during system installation in the computer and has full privilege to create the system user in groups (Figure 5.2).

![System users as group](image)

Figure 5.2 System users as group

5.4 Non functional Requirement

It is defining the emergent of system properties such as integrity, flexibility, security, maintainability etc. that is not directly related to the specific functions provided by the system.

5.4.1 Integrity

It is to prevent any mistake or errors may happen from the users or operators while using the system. For instance, the integrity in this system is verifies and checks the patient appointment. After the nurses make sure about the doctor availability, the nurse entered the patient appointment in the system and then the patient can use the system correctly and confirm the appointment on date. This integrity process to prevent any errors may happen during entering the patient appointment.
5.4.2 **Flexibility**

Developing this system is based on the problem statement and the user requirement. Consequently after two, three years or less or more, the system user will have a new requirements such as extra database, external users or extra functions such as (patient, set the appointment by themselves, laboratory module and online consultations, etc). So the system should be flexible enough to upgrade, modify any part of the system also flexible to change the users interface and can extend DB capacity. So it’s flexible for any requirements regarding to the user requirements.

5.4.3 **Security**

Security it is to prevent any unauthorized user to access the system. The system users have username and password provided by authorize person (admin) to let the users accessing the system. So if the user doesn’t have username and password, cannot access the system. This process to avoid any problem might happen from illegal users. In addition, even if the user has username and password, they have limited authorization, which means each user have special privilege based on their job level (doctor, nurse and admin etc).

5.4.4 **Maintainability**

Maintainability it is one of the must important points in any system or part of the system that have the ability for modifying and/or correction to improve the system performance. System can support the maintainability by supporting the re-engineering or redeveloper to change the format of the existing system without changing the system function.
5.5 System Interface

System interface is the tool that allows the system users to interact with the system. These interfaces such as the following:

5.5.1 User Interface

This Interface describes the progress of the system to understand how the system works. Basically there are four different users in the system listed below:

i. Patient: can use the interface for confirming the appointment

ii. Nurse: user in the health care can use the system based on the privileges provided by the system admin.

iii. Doctor: user in the health care can use the system based on the privileges provided by the system admin.

iv. Administrator: the system admin has full privilege to manage the system.

Some of the system functions such as (medication history, consultation, consultation history, allergies, task manager, documentation, payment) were added based on the user requirement that collected during the system testing (feedback).

1- Login interface:

This is the first page in the system and the user can use this page to login into the system (Figure 5.3).

![Login interface](image)

Figure 5.3 Login interface

Notes: For patient only (Username=Patient ID & Password = I/C or passport number).
2- Today Queue (Today’s consultation queue) (Home Page).

Users: Admin/nurse/doctor

- This page (Figure 5.4) appears by clicking on (today’s queue) in the system navigator. It’s the home page for the system user except the patient. Here the user can view the patients queue list, which have been sent to the doctor for consultation purpose and they are ready for treatment. Here for more efficient work the patient name held with the doctor name and the consultation status.
- Through this page the doctor will be able to go through patient consultation history and by clicking on view the details for particular consultation will appear.

This is the list of patients list that has been sent by the nurse to the doctors for medical consultation purpose and these patients are ready to meet the doctor.

Figure 5.4 Today Queue
3- Search Patient

Users: Admin/nurse/doctor

Search patient (Figure 5.4.1) it helps user to search about patient using different criteria.

- This page appear by clicking on (Search Patient) in the system navigator in the home page. Here the user can search for any patient in the system by the followings:
  - Patient Name
  - IC No / Passport
  - Telephone
  - Fax No

![Figure 5.4.1 Search Patients]
4- Create new Patient

Users: Admin/nurse/doctor

- This page (Figure 5.4.2) appear by clicking on (Create new patient) in the system navigator in the home page or from the search page by clicking on button new patient.
5- View appointment

Users: Admin/nurse/doctor

- This page (Figure 5.4.3) appears by clicking on (View appointment) in the system navigator in the home page.
- From this page (Appointment calendar) the user can view all the patients’ appointments for each doctor for a specific date.
- By clicking on view list the list of patients who have appointment with selected doctor will appear (Figure 5.4.3. (a)).

Figure 5.4.3 View appointments
Figure 5.4.3 (a) Appointment list

From this page the user can view the list of appointment that they have in particular day, and by clicking on edit button the next page (Figure a1) will appear to allow user to edit the appointment.

Figure a1 Edit Appointments
7- View Documents

Users: Admin/nurse/doctor

- This page (Figure 5.4.4) appears by clicking on (View Documents) in the system navigator in the home page.
- From this page, the user can manage all the documents (download, upload), based on the document categorize.
- This function helps the user to keep tracking and archiving all the documentation they have.
- By clicking on download button the page (Figure 5.4.4. (a)) will appears.
- By clicking on upload button the page (Figure 5.4.4. (b)) will appear.
- By clicking on the manage file button the page (Figure 5.4.4. (c)) will appear.

Figure 5.4.4. (a) View Documents
Figure a1 Downloading document

Figure 5.4.4.b Upload file
Figure b1 Locate upload file

Manage file button is for managing the files editing/deleting; (Figure 5.4.4. (c)).

By clicking on edit button the next page (Figure c1) will appear for editing or deleting the file.

Figure 5.4.4. (c) Managing files
Figure c1 Editing /Deleting file

Apply changes button to save the changes after editing

Delete button to delete file
8- My Tasks

Users: Admin/nurse/doctor

- This page (Figure 5.4.5) appears by clicking on (View My Tasks) in the system navigator on the home page. Here the user can view the list of the tasks that has to do it based on the priority.
- The importance of this function came from the user work environment. So they needs to schedule their tasks based on the task priority.
- By clicking on add new task or edit button the next form (Figure 5.4. 5 (a)) will appear to create a new task or to edit task.

Figure 5.4.5 View My Tasks
Figure 5.4.5 (a) Editing/Deleting task

Figure 5.4.5 (b) Creating a new task
9- Doctor Schedule

Users: Admin/nurse/doctor

This page (Figure 5.4.6) for managing the doctor status and through this page the user will be able to:

- View the history of doctor status by clicking on view button the page (Figure 5.4.6) will appear.

Figure 5.4.6 Doctor Schedule
Figure 5.4.6 (a) Doctor Schedule History

To edit the doctor status press on edit (Figure 5.4.6 (b))

The schedule details for Selected doctor

Figure 5.4.6 (b) Editing Doctor Status
Doctor status:
- Completed: the status is completed.
- Postponed: the status postponed to next time.
- Cancelled: the status cancelled.
- Attending patient: doctor attending in the clinic to treat the patients
- Meeting: doctor in meeting.
- On-leave: doctor not available

Figure 5.4.6 (c) Creating New Status

Figure 5.4.6 (d) Doctor Schedule and status history
10 – Reports

Users: Admin/nurse/doctor

• This page (Figure 5.4.7) appears by clicking on (View Report) in the system navigator in the home page.

• From this page the user can view different reports and can export reports based on specific criteria such as appointments, patient’s race, gender etc.

• The user can search for specific report by using the search criteria.

• And by clicking on the list of report option, the next pages will appear.

Figure 5.4.7. (a) View Report (Appointment)
By clicking on breakdown of patient by gender, the report about patient gender will appear.

Figure 5.4.7. (b) Patients Gender Report

By clicking on breakdown of patient by Race, the report about patient Race will appear.

Figure 5.4.7. (c) Patients Race Report
By clicking on break down the payment the report about payment (monthly and daily).

Figure 5.4.7. (d) Payment Report

By clicking on break down the payment method the report about payment method will appear.

Figure 5.4.7. (e) Payment method Report
By clicking on summary of doctor consultation the report about doctor consultation per month will appear.

Figure 5.4.7. (f) Doctor Consultation Report

By clicking on breakdown of client type a report about client type will appear.

Figure 5.4.7. (g) Client Report
By clicking on breakdown of document a report about the document uploaded will appear.

Figure 5.4.7. (h) Document Report

By clicking the patient complaints a report about the patient complaints will appear.

Figure 5.4.7. (i) The Patient Complaint Report
By clicking the patient treatment a report about the patient’s treatment will appear.

Figure 5.4.7. (j) The Patient Treatment Report

By clicking diagnosis category a report about the diagnosis category will appear.

Figure 5.4.7. (k) The Diagnosis Category Report
By clicking the total product use a report about the total product use will appear.

Figure 5.4.7. (I) The Total Product Used Report
11- The patient consultation

The patient consultation managed in steps. The following is the details.

- (Figure 5.4.8) is contained the patient list and by clicking on view button the next page (Figure 5.4.8. (a)) will appear.

![Patients list](image1)

**Figure 5.4.8 Patients list**

![Patient details](image2)

**Figure 5.4.8. (a) Patient details**

By clicking on view/details in the patient options list, full patient details will appear (Figure 5.4.8. (b)).
By clicking on previous illness in the patient options list, the previous illness for that’s particular patient will appear (Figure 5.4.8. (c)).
Figure 5.4.8. (c) Patient previous illness details

By clicking on consolation history in the patient option list, a list of history consultation will appear (Figure 5.4.8. (d)).

Figure 5.4.8. (d) Patient consultation history
In this page the user can view or edit all the consultation information related to the patient for particular consultation.

Figure d1 The Patient Consolation Details.

This page is for patient diagnosis details.

Figure d2 Patient Consolation Diagnosis

By clicking on medication button in the consultation option list, the medication page (Figure d3) will appear.
By clicking on the payment button in the consultation option list, the payment page (Figure d4) will appear.

Figure d3 Patient Medication

By clicking on prescribe new medication add medication page will appear and the user can add the medication or patient or edit. Once the user finished prescribing the medicine, user need to press on submit button than the prescribed medicine will be appear as roe in the upper page.

This page is for user to key in the consultation and the medication payment for patient.

Figure d4 Patient Payment page
12- Patient appointment history

This page (Figure 5.4.9) for managing the patient appointment

- By clicking on appointment history button in the patient option all the appointments history for the particular patient will appear (Figure 5.4.9).

- By clicking on new appointment the page for creating new appointment will appear, (Figure 5.4.9.(a)), and the same form will appear if you press on edit button for editing selected appointment will appear.

This is the main page for the appointment for particular patient.

Figure 5.4.9 Patient Appointment History
13- Patient allergies

Patient allergies are the last button in the patient option list. By pressing on this button the allergies page (Figure 5.4.10) will appear.
14 – Administration

User: Admin

This is the home page (Figure 5.4.11) for the system administration, from this page the admin can manage all the information related to the system and the system user.

15 – User management

User: Admin

From the admin home page the admin has the admin task list, by clicking on user management the next page (Figure 5.4.11. (a)) will appear and the admin can manage the entire system user from this page.
In this field the admin can search for any user.

By clicking on add new user button, new form will appear (Figure 5.4.11. (a)) to the admin to create a profile for the new user.

By pressing on admin button the admin will be able to edit the user profile (Figure 5.4.11. (b))

Figure 5.4.11. (a) User Management

Figure 5.4.11. (b) Edit/ Create user
16 – Group management

User: Admin

By clicking on Group management the next page (Figure 5.4.12) will appear and the admin can manage the entire system group from this page.

![Group Management Interface](image)

**Figure 5.4.12 Group Management**

By clicking on edit button the admin can edit the group information (Figure 5.4.12. (a)).

By clicking on Create button the admin can create a new group (Figure 5.4.12. (a)).

**Figure 5.4.12. (a) Edit/ Create group**
17 – System Management

User: Admin

By clicking on system management the next page (Figure 5.4.13) will appear and the admin can manage all the information related to the system.

By clicking on profile button the admin will be able to view the system information (Figure 5.4.13. (a))

By clicking on code control button the admin will be able to edit the system code information (Figure 5.4.13. (b))

By clicking on edit button the admin will be able to view the system information (Figure 5.4.13. (b))

Figure 5.4.13 System Management

Figure 5.4.13. (a) System Profile
By clicking edit button the admin can edit the selected code (Figure b1)

Figure 5.4.13.b System Code Control

Figure b1 Edit/ Create System Code
18 – Account management

User: Admin

From this page (Figure 5.4.14) the admin can manage all the accounts system. In the administrator home page by clicking on account management on the page navigator, the list of accounts will appear, for managing the accounts. By clicking on edit button to edit the account and by clicking on add new account button for adding a new account.

Figure 5.4.14 Account Management

Figure 5.4.14. (a) Edit/ Create account management
19 – Client management

User: Admin

From this page the admin can manage the health care clients (Figure 5.4.15)

In the administrator home page by clicking on Client management on the page navigator, the list of client will appear for managing the clients. By clicking on edit button to edit the client account and by clicking on add new account button for adding a new account client.

![Figure 5.4.15 Client management](image-url)
Figure 5.4.15. (a) Edit/Create Client
20 – Vendor management

User: Admin

From this page the admin can manage the vendors (supplier) profile (Figure 5.4.16). In the admin home by clicking on the Vendor management on the page navigator, the list of vendors will appear, for managing the vendors (suppliers). By clicking on edit can editing the vendor file and by clicking on add new can create new vendor (supplier) profile. This will help the management to keep tracking about the supplier who dealing with them.

Figure 5.4.16 Vendor management
Figure 5.4.16. (a) Edit/ Create vendor
21 – Product management

User: Admin

From this page the admin can manage the products profile (Figure 5.4.17). In the admin home by clicking on the Product management, the list of the entire medicine product will appear. This is will help the management to manage the medicines. By clicking on edit button for editing the Product file and by clicking on add new for creating a new Product file.
Figure 5.4.17.(a) Create / Edit product
15- Confirmation Appointment

User: Patient

After login to the system using username and password

(Username = MRN & Password = NRIC or passport No) the Appointment Confirmation page appears to the patient.

- Thorough this page, the patient can confirm or cancel or postponed the appointment. In addition can view the doctor availability.
- By clicking on the view today appointment button the patient can view the current day appointment.

![Image of Patient Home Page]

Figure 5.4.18 Patient Home Page
By clicking on view the next page (Figure 5.4.18. (a)) will appear.

![Image](image1)

**Welcome To The Health Care Management System**

This buttons use to confirm or cancel or postponed the appointment.

Figure 5.4.18. (a) Patient View the Appointment

This buttons use to confirm or cancel or postponed the appointment.

Figure 5.4.18. (b) Patient Confirm the Appointment
5.5.2 Hardware Interface

The hardware interface is the hardware devices that use to communicate with each other such as the wires, plugs, computers and sockets, etc.

5.5.3 Communications Interfaces

The communication interfaces is the tools that help the user to interact with the system user interface and establishing the communications.

i. The web browser

The web browser is a software application installed by default with any windows (XP, Vista, etc) and Linux. This application enable user to interact with any information (text, mage, music, etc) located on the web page in the World Wide Web or in the local area network. Health care management system recommended working on internet explorer because this system developed using apex oracle, which support windows internet explorer more than the other browsers and it support to work over mozilla firefox or any kind of web browser. To configure Microsoft Internet Explorer to connect to the Oracle Database XE Database Home Page:
1. From the Start menu, select Control Panel, then Internet Options.

2. In the Internet Options dialog box, click the Security tab.

3. Under Security, select Local Intranet and then select Sites.

4. In the Local Intranet dialog box, select Advanced.

5. Under Add this Web site to the zone, enter the following site: 127.0.0.1

6. Click OK.

To configure Firefox to connect to the Oracle Database XE Database Home Page:

1. Under the Tools menu, select Options.

2. From the General tab, select Connection Settings.

3. If you are using a proxy server, add the following setting to the No Proxy For setting: 127.0.0.1

ii. The protocol

The web browser initially uses HTTP (hypertext transfer protocol) protocol. The http supports the web browser to interact with the web server.

iii. Connection ports

This is the most ports used for oracle application server and oracle database:

- 7777 it is Oracle HTTP Server port and listener port for Oracle Application Server.
- 8080 its Oracle XMLDB HTTP port Oracle Database.

5.6 System Tables

System is designed in 17 tables (see appendix D).

1) Appointment Table: it is contains all the appointments information.

2) Clinic Profile Table: it is contains information about the health care profile.

3) CMS Account Table.

4) CMS Tasks Table: it is contains all the system users’ tasks information.
5) Consultation Table: it is contains all the information related to the patient consultations.

6) Patient Diagnosis Table: it is contains all the information related to the patient diagnosis.

7) Vendor Table: it is contains all the information related to products suppliers’ information.

8) CMS Client Table: it is contains all the information related to health care client’s information.

9) CODE Type Table: it is contains all the information related to the system codes.

10) Library CODE Table: it is contains all the information about the documents.

11) Patient Table: it is contains all the information patients.

12) Patient Medication Table: it is contains all the information about patient’s medications.

13) CODE Table.

14) Patient Allergies Table: it is contains all the information related to the patient allergies.

15) Product Table: it is contains all the information about the products.

16) Doctor Schedule Table: it is contains all the information related to the doctor’s schedule.

17) Doctor schedule history: for managing the doctor status history.
5.7 System Code

The system code in this system is PL/SQL and it contains four parts (Table 5.1):

1- System Sequence Code
2- System Triggers Code
3- Functions Code
4- Packages (Chart Code)

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
</table>
| **CMS_SEQ**         | CREATE SEQUENCE cms_seq  
INCREMENT BY 1  
START WITH 6141  
MINVALUE 1  
MAXVALUE 9999999999999999999999999999999 
NOCYCLE  
NOORDER  
CACHE 20  
/ | This is the CMS sequences  
Started by 1, incremented with 6141 and the max value is 9999999999999999999999999999999 |
| code_ctrl_seq       | CREATE SEQUENCE code_ctrl_seq  
INCREMENT BY 1  
START WITH 2420  
MINVALUE 1  
MAXVALUE 9999999999999999999999999999999 
NOCYCLE  
NOORDER  
CACHE 20  
/ | This is the Code Control Sequence. Started with 2420, incremented by 1 and the max value is 9999999999999999999999999999999 |
| **LIBRARY_DOCSEQ**  | CREATE SEQUENCE library_docs_seq  
INCREMENT BY 1  
START WITH 161  
MINVALUE 1  
MAXVALUE 9999999999999999999999999999999 
NOCYCLE  
NOORDER  
CACHE 20  
/ | This is the Library code sequences  
Started by 161, incremented by 1 and the max value is 9999999999999999999999999999999 |
| MAL_MEDICINE_LIST_SEQ | CREATE SEQUENCE mal_medicine_list_seq  
INCREMENT BY 1  
START WITH 5521  
MINVALUE 1  
MAXVALUE 9999999999999999999999999999999 
NOCYCLE  
NOORDER  
CACHE 20  
/ | This is the Medicine Code sequences  
Started by 5521, incremented by 1 and the max value is 9999999999999999999999999999999 |
| PAT_SEQ             | CREATE SEQUENCE pat_seq  
INCREMENT BY 1 | |
This is the Patients Sequences
Started by 1141, incremented by 1 and the max value is 999999999999999999999999999

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
</table>
| CREATE OR REPLACE TRIGGER appointment_bi | BEFORE INSERT OR UPDATE ON appointment REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW declare v_patient_name varchar2(300); begin select pat_full_name into v_patient_name from patient where pat_id = :new.apt_pat_prn; if inserting and :new.apt_id is null then select cms_seq.nextval into :new.apt_id from dual; end if; if inserting then :new.apt_status := 'New'; :new.search := :new.apt_pat_prn || ' | ' || v_patient_name; :new.patient_name := v_patient_name; end if; if updating then if trunc(:new.apt_date) < trunc(sysdate) then :new.apt_status := 'Closed'; end if; if :new.apt_duration is not null then :new.APT_END_DATETIME := to_date(:new.apt_DATE + (:new.apt_Duration/1440), 'DD-MON-YYYY HH24:MI'); end if; :new.search := :new.apt_pat_prn || ' | ' || v_patient_name; :new.patient_name := This is the appointment trigger, started before inserting or updating data.

So it's referring to the row new as new and the old as old.

Begin select full Patient Name from patient table, where The patient ID = new appointment Patient ID.

If the inserting value is null back to the system sequence and select the next value.

If the data is updating trunc the current appointment and set the new status date.

In the app duration is nut null set the appointment end date time and the appointment duration.
v_patient_name;
    end if;
end;
/

CREATE OR REPLACE TRIGGER bd_library_docs
    BEFORE DELETE
    ON library_docs
    REFERENCING NEW AS NEW OLD AS OLD
    FOR EACH ROW
BEGIN

    insert into library_docs_history
    (id, filename, date_deleted, deleted_by, old_owner, created_date, created_by, PROJECT_CODE, file_id)
    values
    (library_docs_seq.nextval, :old.filename, sysdate, v('APP_USER'), :old.upload_user, sysdate, v('APP_USER'), :old.file_upload_id, :old.id);

END;
/

CREATE OR REPLACE TRIGGER bi_consultation
    BEFORE INSERT OR UPDATE
    ON consultation
    REFERENCING NEW AS NEW OLD AS OLD
    FOR EACH ROW
BEGIN

    if inserting and :new.id is null then
        select "CMS_SEQ".nextval into :NEW."ID" from dual;
    end if;

    if inserting then
        :new.created_by := v('USER');
        :new.created_date := sysdate;
    end if;
end;
/

This is library trigger starting before deleting
And it's referring to the row new as new and the old as old.

Begin Than for inserting the data into library doc (the data required)
The values.

This is a consultation trigger starting before inserting data
And it's referring to the row new as new and the old as old.

Begin If the inserted ID in null than back to the system sequence and take the new ID from the Dual file.

If inserting than appear the user name and the date of creating the data (system date).
This is a MEDICINE_LIST trigger starting before inserting data to the medicine list
And it's referring to the row new as new and the old as old.

Begin If the selected medicine ID is null back to the system sequence and select a new ID.

This is the Task trigger starting before inserting or updating data.
And it's referring to the row new as new and the old as old.

Begin If the selected item ID is null back to the system sequence and select a new ID from the table.
If the data entered is inserting appear the user name and the system date and the new task call it "new".
If the data entered is updating.
(if the task complete is not null, the task status is "Closed").
CREATE OR REPLACE TRIGGER clinic_profile_biu
BEFORE INSERT OR UPDATE
ON clinic_profile
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
begin
  if inserting and :new.id is null then
    select cms_seq.nextval into :new.id from dual;
  end if;
  if inserting then
    :new.clinic_address := :
      new.CLINIC_ADD_1 ||' ||
      ':new.CLINIC_ADD_2 ||'' ||
      ':new.CLINIC_CITY ||' ||
      ':new.CLINIC_STATE ||' ||
      ':new.CLINIC_POSTCODE ||' ||
      ':new.CLINIC_TEL_1 ||' ||
      ':new.CLINIC_FAX_1;
  end if;
  if updating then
    :new.clinic_address := :
      new.CLINIC_ADD_1 ||' ||
      ':new.CLINIC_ADD_2 ||'' ||
      ':new.CLINIC_CITY ||' ||
      ':new.CLINIC_STATE ||' ||
      ':new.CLINIC_POSTCODE ||' ||
      ':new.CLINIC_TEL_1 ||' ||
      ':new.CLINIC_FAX_1;
  end if;
end;
/

CREATE OR REPLACE TRIGGER cms_account_bi
BEFORE INSERT
ON cms_account
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
begin
  if inserting and :new.id is null then
    select cms_seq.nextval into :new.id from dual;
  end if;
end;
/

This is the Clinic Profile trigger starting before inserting or updating data.
And it's referring to the row new as new and the old as old.

Begin If inserting data and the ID is null than back to the system sequence and select the next value.
And the inserting the address(" ")
If the inserting is updating than the address(" ")

This is the account trigger starting before inserting data.
And it's referring to the row new as new and the old as old.

Begin If inserting the new ID is null than back to the system sequence and select the next value
CREATE OR REPLACE TRIGGER cms_clients_bu
BEFORE
INSERT OR UPDATE
ON cms_clients
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
begin
  if inserting and :new.id is null then
    select cms_seq.nextval into :new.id from dual;
  end if;
  if inserting then
    :new.created_by := nvl(v('USER'), 'Mohammed');
    :new.created_date := sysdate;
    :new.search :=
      :new.CLIENT_CODE || ' | ' ||
      :new.CLIENT_NAME || ' | ' ||
      :new.CLIENT_STATE || ' | ' ||
      :new.CLIENT_TEL_1 || ' | ' ||
      :new.CLIENT_CONTACT_NAME || ' | ' ||
      :new.CLIENT_CONTACT_DID || ' | ' ||
      :new.CLIENT_CONTACT_MOBILE;
  end if;
  if updating then
    :new.updated_by := v('USER');
    :new.updated_date := sysdate;
    :new.search :=
      :new.CLIENT_CODE || ' | ' ||
      :new.CLIENT_NAME || ' | ' ||
      :new.CLIENT_STATE || ' | ' ||
      :new.CLIENT_TEL_1 || ' | ' ||
      :new.CLIENT_CONTACT_NAME || ' | ' ||
      :new.CLIENT_CONTACT_DID || ' | ' ||
      :new.CLIENT_CONTACT_MOBILE;
  end if;
end;
/

CREATE OR REPLACE TRIGGER code_types_bd
BEFORE
DELETE
ON code_types
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
BEGIN
  delete from codes where cod_cot_id = :old.cot_id;
END;
/

This is the Client trigger starting before Update data.

And it's referring to the row new as new and the old as old.

Begin if the inserted ID is null, back to the sequence and select the next value.

If inserted, than the user name is null, appear 'mohammed' and appear the created date "system date"

If update appear the updated username and the date of updating "system date"

This is the Code Types trigger starting before deleting data.

And it's referring to the row new as new and the old as old.

Begin Delete the Code where the cod_cot_id = :old.cot_id;
This is the Patients Allergies trigger starting before Inserting data.
And it's referring to the row new as new and the old as old.

Begin if the inserted ID is null back to the system sequence and select new value.

This is the Patients trigger starting before Inserting or Update data.
And it's referring to the row new as new and the old as old.

Declaring the new values (v_prn, d_cuur_date, v_sec_group_id, v_group_id, v_user_id, p_password, v_prn_prefix)

Begin
If the inserted patient ID is null Back to the patient sequence and select new value.

The patient ID =The Patient Registration Number 'PRN' = The Medical Registration Number followed by '00' "MRN00"

If inserting name, appear the username if not appear the 'admin' and the created date.
If the Patient ID is not null then the Password= IC number
In update, in the system user set the type is 'CMS' and change the password in the first use where the username = PRN

If updating the modified by (username) and the current updating date.

This is the Patients Diagnosis Trigger starting before Inserting or Update data.

And it's referring to the row new as new and the old as old.

Begin
If the inserted ID is null, back to the system sequence and select new value
| PATIENT_MEDICATIONS_BI | CREATE OR REPLACE TRIGGER patient_medications_bi  
|                        | BEFORE INSERT ON patient_medications  
|                        | REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW  
|                        | begin  
|                        | if inserting and :new.id is null then  
|                        | select cms_seq.nextval into :new.id from dual;  
|                        | end if;  
|                        | if inserting then  
|                        | :new.created_by := v('USER');  
|                        | :new.created_date := sysdate;  
|                        | end if;  
|                        | end;/  
| PRODUCTS_BI | CREATE OR REPLACE TRIGGER products_bi  
|                        | BEFORE INSERT ON products  
|                        | REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW  
|                        | begin  
|                        | if inserting and :new.id is null then  
|                        | select cms_seq.nextval into :new.id from dual;  
|                        | end if;  
|                        | end;/  
| TRG_ALL_TEXT_LIBRARY_DOCS | CREATE OR REPLACE TRIGGER trg_all_text_library_docs  
|                        | BEFORE INSERT OR UPDATE ON library_docs  
|                        | REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW  
|                        | BEGIN  
|                        | :NEW.ALL_TEXT := UPPER(:NEW.FILENAME || '' ||  
|                        | :NEW.STATUS || '' ||  
|                        | :NEW.UPLOAD_DATE|| '' ||  
|                        | :NEW.UPLOAD_USER || '' ||  
|                        | :NEW.DESCRIPTION || '' ||  
|                        | :NEW.MIME_TYPE || '' ||  
|                        | :NEW.FILE_TYPE || '' ||  
|                        | :NEW.KEYWORD);  
|                        | END;/  
|                        | This is the Patients Medication trigger starting before Inserting data.  
|                        | And it's referring to the row new as new and the old as old.  
|                        | **Begin**  
|                        | If the inserted ID is null, back to the system sequence and select new value  
|                        | Than appear the user name and the created date.  
|                        | This is the Products Trigger starting before Inserting data.  
|                        | And it's referring to the row new as new and the old as old.  
|                        | **Begin**  
|                        | If the inserted ID is null, back to the system sequence and select new value  
|                        | This is the Library Doc Trigger starting before Inserting or Update data.  
|                        | And it's referring to the row new as new and the old as old.  
|                        | **Begin**  
|                        | Insert the file information |
| VENDOR_BIU | CREATE OR REPLACE TRIGGER | biu_vendor | BEFORE INSERT OR UPDATE ON vendor REFERENCING NEW AS NEW OLD AS OLD FOR EACH ROW begin if inserting and :new.id is null then select cms_seq.nextval into :new.id from dual; end if; if inserting then :new.created_by := nvl(v('USER'),'Mohammed'); :new.created_date := sysdate; :new.search := :new.VEND_CODE || ' ' || :new.VEND_NAME || ' ' || ' ' || :new.VEND_STATE || ' ' || ' ' || :new.VEND_TEL_1 || ' ' || ' ' || :new.VEND_CONTACT_NAME || ' ' || ' ' || :new.VEND_CONTACT_MOBILE; end if; if updating then :new.updated_by := v('USER'); :new.updated_date := sysdate; :new.search := :new.VEND_CODE || ' ' || ' ' || :new.VEND_NAME || ' ' || ' ' || :new.VEND_STATE || ' ' || ' ' || :new.VEND_TEL_1 || ' ' || ' ' || :new.VEND_CONTACT_NAME || ' ' || ' ' || :new.VEND_CONTACT_MOBILE; end if; end; /
| BIU_DOCTOR_SCHEDUL_E | CREATE OR REPLACE TRIGGER | biu_doc_schedule BEFORE INSERT OR UPDATE ON doctor_schedule REFERENCING OLD AS OLD NEW AS NEW FOR EACH ROW declare v_patient_id varchar2(300); begin if inserting and :new.sched_id is null then select cms_seq.nextval into :new.sched_id from dual; end if; if inserting then :new.status := 'NEW'; :new.created_date := sysdate; :new.created_by := nvl(v('USER'),'USER');(' | This is The Vendor Trigger starting Before Inserting or Updating data. And it's referring to the row new as new and the old as old. Begin If inserting the data and the ID is null, back to the system sequence and select new value. If inserted then appear the username and if the username is null appear "Mohammed" If updating appear the user name and the created date. This is The Doctor Schedule Trigger starting Before Inserting and updating data. And it's referring to the row new as new and the old as old. Begin If the inserted ID is null than back to the system sequence and take the new ID from the Dual file. If inserting than appear the new status value as new, the |
if :new.duration is not null
then
:new.STATUS_END_DATE := (:new.STATUS_START_DATE + :new.duration);
end if;
if :new.status is not null
then
insert into
doctor_schedule_hist (hist_id, sched_id, patient_id, created_by, created_date, prior_status, status, doctor, sched_date)
values (cms_seq.nextval, :new.sched_id, :new.patient_id, nvl(v('USER'), 'USER'), (:sysdate, :new.prior_status, :new.status, :new.doctor_name, :new.status_start_date);
:new.prior_status := :new.status;
:new.status := null;
end if;
end if;

if updating then
if :new.status is not null
then
insert into
doctor_schedule_hist (hist_id, sched_id, patient_id, created_by, created_date, prior_status, status, doctor, sched_date)
values (cms_seq.nextval, :new.sched_id, :new.patient_id, nvl(v('USER'), 'USER'), (:sysdate, :new.prior_status, :new.status, :new.doctor_name, :new.status_start_date);
:new.prior_status := :new.status;
:new.status := null;
end if;
end if;

created date is the system date, and the created by is the user.
The new duration value is not null so the status end date = status start date+ new duration.

New status not null so insert into the
doctor_schedule_hist the following:
(sched_id, patient_id, created_by, created_date, prior_status, status, doctor, sched_date( )
The following values
:new.prior_status := :new.status;
:new.status := null;
)

In the updating case

If the value of status is not null insert into the
doctor_schedule_hist
(hist_id, sched_id, patient_id, created_by, created_date, prior_status, status, doctor, sched_date)
if :new.status = 'POSTPONED' then
  :new.status_start_date := :new.postponed_date;
  :new.prior_status := :new.status;
  :new.status := null;
  :new.postponed_date := null;
end if;

if :new.duration is not null then
  :new.STATUS_END_DATE := (:new.STATUS_START_DATE + :new.duration);
end if;

end if;

end;

The following values.
((cms_seq.nextval, :new.sched_id, :new.patient_id, nvl(v('USER'),'USER'), ('sysdate, :new.prior_status, :new.status, :new.doctor_name, :new.status_start_date);
)
And if the new status is POSTPONED
The new status start date = new postponed date
And new priority status = new status,

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGE_NOW</td>
<td>CREATE OR REPLACE FUNCTION age_now(dd in varchar2, mm in varchar2, yyyy in varchar2) return varchar2 as</td>
<td>This is the age function</td>
</tr>
<tr>
<td></td>
<td>dob date;</td>
<td>The age is Integer</td>
</tr>
<tr>
<td></td>
<td>age pls_integer;</td>
<td>Begin</td>
</tr>
<tr>
<td></td>
<td>age_now number;</td>
<td>The dob = (DD,MM,YYYY)</td>
</tr>
<tr>
<td></td>
<td>begin</td>
<td>The age is</td>
</tr>
<tr>
<td></td>
<td>dob:=to_date(valid_dob(dd,mm,yyyy), 'DDMMRRRR');</td>
<td>trunc(months_between(sysdate,dob)/12 )</td>
</tr>
<tr>
<td></td>
<td>age_now := (trunc(months_between(sysdate,dob)/12 ));</td>
<td></td>
</tr>
<tr>
<td></td>
<td>return age_now;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>end;</td>
<td></td>
</tr>
</tbody>
</table>

4- Packages (Chart Code) Refer to the appendix (E)
5.8 Conclusion

System designing started by looking at the analysis result, which determines the system boundaries based on the system need and the user requirement. In this research, the result of analysis was developing an integrated health care management system to manage the entire health care information. The first stage for system design was determining the system modules and UseCase than designing the system tables and system interface and finally writing the system codes.

The way of developing this integrated web based was using apex oracle, which provide the flexibility to use more than one tool such as (HTML, SQL, PL/SQL, JavaScript, CSS and FUSION) for developing the system. Therefore the main module designed and developed in this system is patient module, doctor module, appointment module, product module, client module, document module, task module, account module and user module.

5.9 Summary

System design is presenting system structure, details about system modules and system interface. Therefore, system structure covers the system concept, which covers the research objectives through designing the system modules. Those modules developed using (Oracle Apex) environment that provides necessary development tools. This first stage of developing this system started by specifying the function requirements (use case) and non-function requirements (integrity, flexibility, etc), then developing the system interfaces, system tables and system codes. The next chapter is presenting the research discussion and conclusion.