

**Submitted To: Zammad Farooq**

**Submitted By: Eman Amer Dar | 28692**

**Assignment - 2**

Table of Contents

[2.1 1](#_Toc477977701)

[Introduction 1](#_Toc477977702)

[Purpose 1](#_Toc477977703)

[Intended Audience and Reading Suggestions 1](#_Toc477977704)

[Product Scope 1](#_Toc477977705)

[Overall Description 1](#_Toc477977706)

[Product Perspective 1](#_Toc477977707)

[Product Functions 2](#_Toc477977708)

[User Characteristics 2](#_Toc477977709)

[Use Case Table 3](#_Toc477977710)

[ERD (Entity Relationship Diagram) 15](#_Toc477977711)

[Operating Environment 15](#_Toc477977712)

[External Interface Requirements 16](#_Toc477977713)

[Hardware Interfaces 16](#_Toc477977714)

[Software Interfaces 16](#_Toc477977715)

[Other Non-functional Requirements 16](#_Toc477977716)

[Performance Requirements 16](#_Toc477977717)

[Security Requirements 16](#_Toc477977718)

[2.2 17](#_Toc477977719)

[Components 17](#_Toc477977720)

[Entities and Attributes 17](#_Toc477977721)

[File Structure 17](#_Toc477977722)

[3.1 19](#_Toc477977723)

[Main-Page 19](#_Toc477977724)

[Login Page 19](#_Toc477977725)

[Specialist (Doctors) 20](#_Toc477977726)

[Patients 20](#_Toc477977727)

[3.2 22](#_Toc477977728)

[Relationship between Objects 22](#_Toc477977729)

[Java Docs 22](#_Toc477977730)

[3.3 23](#_Toc477977731)

[Control Structure 23](#_Toc477977732)

[3.4 24](#_Toc477977733)

[Pop-Up Message 24](#_Toc477977734)

[3.5 25](#_Toc477977735)

[IDE (Integrated Development Environment) 25](#_Toc477977736)

[Indentation 25](#_Toc477977737)

[4.1 26](#_Toc477977738)

[Critical Review 26](#_Toc477977739)

[Testing 26](#_Toc477977740)

[Input Testing 26](#_Toc477977741)

[JUNIT Testing 27](#_Toc477977742)

[4.2 28](#_Toc477977743)

[Independent Feedback 30](#_Toc477977744)

[Recommendations 30](#_Toc477977745)

[4.4 31](#_Toc477977746)

[User Documentation 31](#_Toc477977747)

[4.5 32](#_Toc477977748)

[Technical Documentation 32](#_Toc477977749)

[Maintenance 33](#_Toc477977750)

[Self-Criticism | Self-Evaluation 34](#_Toc477977751)

[Self-Criticism 34](#_Toc477977752)

[Self-Evaluation 35](#_Toc477977753)

# 2.1

## Introduction

The hospital management system for Darsons’ Hospital is designed and developed for storing the data and information of the hospital in the database that can be accessed online.

## Purpose

The Darsons’ Hospital needs a computerise form of the records to store them in the database online. By computerising the data it can easily be stored and will be done more efficiently with the help of database.

## Intended Audience and Reading Suggestions

This documentation is done for the Darsons’ Hospital. By reading this it will be easy to understand that how to use the hospital management system that is designed and developed for the hospital. If any problem/trouble occurs then by reading all this documentation the user will easily understand that how it can be used / processed and how to deal with the problem.

## Product Scope

The scope of this hospital management system is that it could be very essential in using it in the future. The data of the people that are coming for the checkups and getting an appointment i.e. patients and the employees and many other records such as Blood Bank, Bills etc. of the Darsons’ Hospital can be stored in the database and the backup of it is also provided so that in case of the loss of the data, the data can be restored.

# Overall Description

## Product Perspective

It is a proper hospital management system which is designed for **‘**Hospital**’** named as Darsons’ Hospital; it contains all the information of the hospital.

This hospital management system belongs to Darsons’ Hospital because it is designed and developed for storing the data of the patients, employees and etc. that are being managed by the hospital. It is can contain sub - systems but itself it is a proper hospital management system. Admin panel will include the BODs (Board of Directors) i.e. CEO, COO etc of the hospital.

## Product Functions

The functions of Property Management System are the following:

* Sign in
* Adding patients, employees etc.
* Updating patients, employees etc.
* Viewing the blood bank, employees’ etc. data.
* Checking and calculating the bills.
* Deleting patients, employees etc.
* Searching patients, employees etc.

## User Characteristics

**User Characteristics**

* Education level must be high for the administrator to fulfill the requirement.
* The Administrator are the experts i.e. they should have the know-how of the technology i.e. Controlling Database, Computer skills etc.

**Use Case**

**USECASE_JAVA (2).png**

## Use Case Table

**Admin**

|  |  |
| --- | --- |
| **Name** **of Use Case** | Sign in |
| **X-Ref** | UseCase-01 |
| **Action** | Sign In |
| **Pre-Conditions** | * Knows the Administrator ID and the Password |
| **Task Sequence** | * Go to the panel * Enter required ID * Enter required Password |
| **Post Conditions** | Admin is logged into the management system |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Adding |
| **X-Ref** | UseCase-02 |
| **Action** | Add |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Enter the entries * Click add to add patients, resources etc. |
| **Post Conditions** | Adding function is performed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Updating |
| **X-Ref** | UseCase-03 |
| **Action** | Update |
| **Pre-Conditions** | * Logging into hospital management system |
| **Task Sequence** | * Log into the system * Enter the entries that are needed to be update and also the id according to which the data is updated * Click update to update the data. |
| **Post Conditions** | Update function is performed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Searching |
| **X-Ref** | UseCase-04 |
| **Action** | Search |
| **Pre-Conditions** | * Logging into a hospital management system |
| **Task Sequence** | * Log into the system * Enter the name of the record * Click on the search to search the record |
| **Post Conditions** | Details are viewed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Deleting |
| **X-Ref** | UseCase-05 |
| **Action** | Delete |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Delete the record by entering a specific id * Click on delete to delete the record |
| **Post Conditions** | Record deleted |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Manage Laboratory |
| **X-Ref** | UseCase-06 |
| **Action** | manage |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Enter Lab Room |
| **Post Conditions** | Laboratory managed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Add Resource |
| **X-Ref** | UseCase-07 |
| **Action** | Adding Resource |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the resource by entering a resource name and quantity * Click on add to add the resource |
| **Post Conditions** | Resource added |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Add and Display Blood Bank Details |
| **X-Ref** | UseCase-08 |
| **Action** | Adding |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the blood bank detail by entering a type of blood and quantity and show the details * Click on add to add the details and to show by click on show button |
| **Post Conditions** | Details added and displayed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Bills |
| **X-Ref** | UseCase-09 |
| **Action** | Billing |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the quantity and price of the items * Click on calculate to calculate |
| **Post Conditions** | Bill Calculated |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Diagnosis Details |
| **X-Ref** | UseCase-10 |
| **Action** | Adding Diagnosis Details |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the diagnosis details by entering a type of diagnosis and patient’s ID and name * Click on add to add the details |
| **Post Conditions** | Details added |
| **Alternative path** | No alternative path |

**Receptionist**

|  |  |
| --- | --- |
| **Name** **of Use Case** | Sign in |
| **X-Ref** | UseCase-01 |
| **Action** | Sign In |
| **Pre-Conditions** | * Knows the Administrator ID and the Password |
| **Task Sequence** | * Go to the panel * Enter required ID * Enter required Password |
| **Post Conditions** | Admin is logged into the management system |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Adding |
| **X-Ref** | UseCase-02 |
| **Action** | Add |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Enter the entries * Click add to add patients etc. |
| **Post Conditions** | Adding function is performed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Updating |
| **X-Ref** | UseCase-03 |
| **Action** | Update |
| **Pre-Conditions** | * Logging into hospital management system |
| **Task Sequence** | * Log into the system * Enter the entries that are needed to be update and also the id according to which the data is updated * Click update to update the data. |
| **Post Conditions** | Update function is performed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Searching |
| **X-Ref** | UseCase-04 |
| **Action** | Search |
| **Pre-Conditions** | * Logging into a hospital management system |
| **Task Sequence** | * Log into the system * Enter the name of the record * Click on the search to search the record |
| **Post Conditions** | Details are viewed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Deleting |
| **X-Ref** | UseCase-05 |
| **Action** | Delete |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Delete the record by entering a specific id * Click on delete to delete the record |
| **Post Conditions** | Record deleted |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Add and Display Blood Bank Details |
| **X-Ref** | UseCase-08 |
| **Action** | Adding |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the blood bank detail by entering a type of blood and quantity and show the details * Click on add to add the details and to show by click on show button |
| **Post Conditions** | Details added and displayed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Bills |
| **X-Ref** | UseCase-09 |
| **Action** | Billing |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the quantity and price of the items * Click on calculate to calculate |
| **Post Conditions** | Bill Calculated |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Diagnosis Details |
| **X-Ref** | UseCase-10 |
| **Action** | Adding Diagnosis Details |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the diagnosis details by entering a type of diagnosis and patient’s ID and name * Click on add to add the details |
| **Post Conditions** | Details added |
| **Alternative path** | No alternative path |

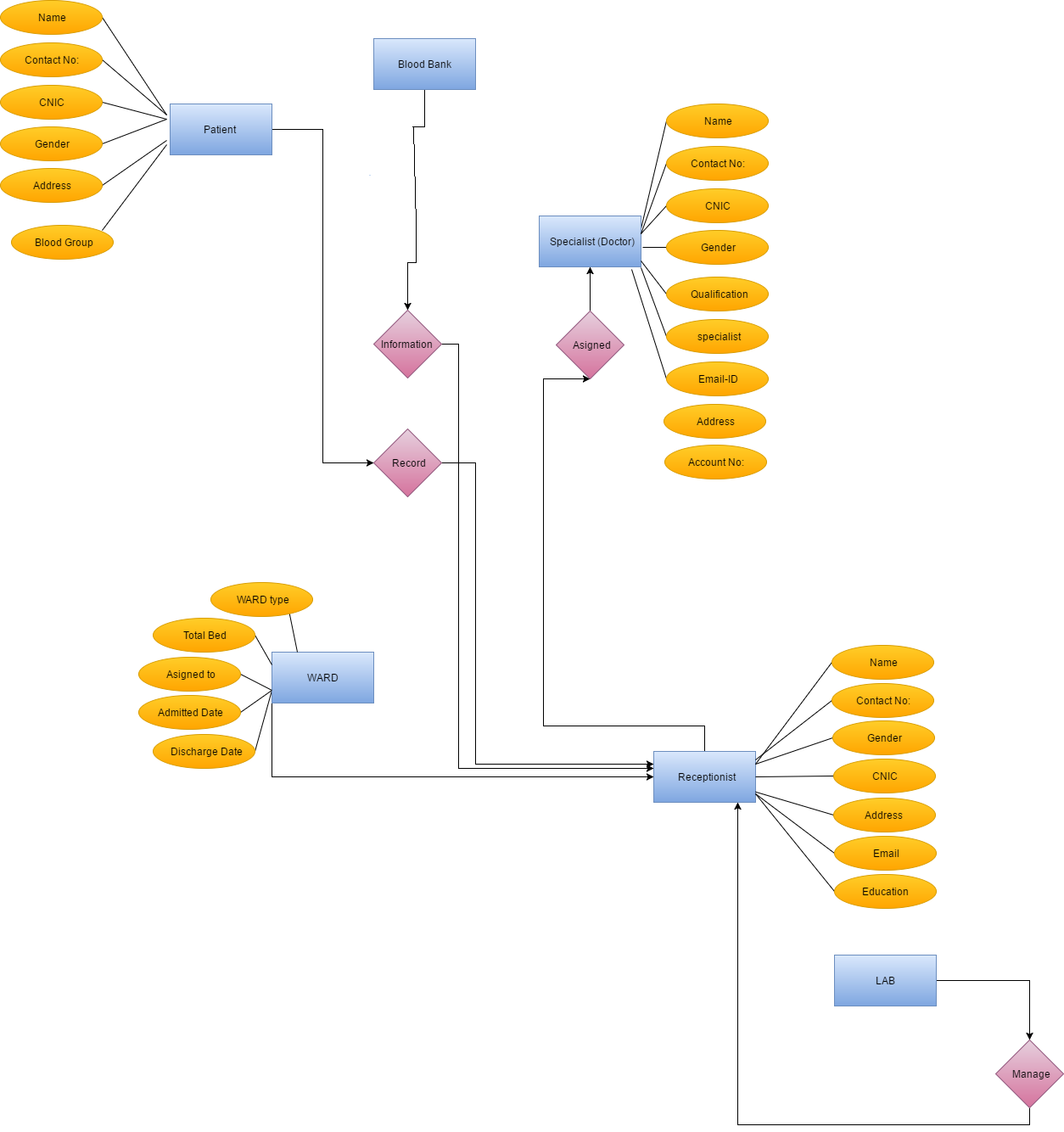
**Laboratory Manager**

|  |  |
| --- | --- |
| **Name** **of Use Case** | Sign in |
| **X-Ref** | UseCase-01 |
| **Action** | Sign In |
| **Pre-Conditions** | * Knows the Administrator ID and the Password |
| **Task Sequence** | * Go to the panel * Enter required ID * Enter required Password |
| **Post Conditions** | Admin is logged into the management system |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Manage Laboratory |
| **X-Ref** | UseCase-06 |
| **Action** | manage |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Enter Lab Room |
| **Post Conditions** | Laboratory managed |
| **Alternative path** | No alternative path |

|  |  |
| --- | --- |
| **Name** **of Use Case** | Add Resource |
| **X-Ref** | UseCase-07 |
| **Action** | Adding Resource |
| **Pre-Conditions** | * Logging into the hospital management system |
| **Task Sequence** | * Log into the system * Add the resource by entering a resource name and quantity * Click on add to add the resource |
| **Post Conditions** | Resource added |
| **Alternative path** | No alternative path |

## ERD (Entity Relationship Diagram)



## Operating Environment

The environment in which the system is going to be operated includes a hardware platform, an operating system. The environment is a hospital management system in which data is added, deleted, searched and updated etc. The hardware required for it must be at least 8GB (Giga Byte) RAM and about more than 4 TB (terra byte) storage. The operating system will be most probably Microsoft Windows 10.

# External Interface Requirements

## Hardware Interfaces

The devices that are able to support this hospital management system must be advanced according to the technology that is being used.

## Software Interfaces

The operating system on which this hospital management system is working is Microsoft Windows 10.

# Other Non-functional Requirements

## Performance Requirements

System’s performance i.e. working of the system is very efficient. There is no lagging in the system which makes it efficient to use. There are no such difficulties in using the system i.e. it is easy to use. The accuracy of the performance of the system is full 100% which makes it very efficient.

## Security Requirements

The personal information of the patients and staff members (employees) etc. are safe and secured in this hospital management system. The data is stored in it and the safety precautions are taken place by the installation the firewall. The authentication is essential to use the system. Without authenticating the admin cannot perform administrator functions and receptionist cannot perform its system functions and this authentication is done by the logging in.

# 2.2

### Components

The components that are needed in designing and developing the interface in java are **JFrame**, **JDialog**, **JButton**, **Label**, **JTextfield** and etc. The tool used for implementing is **Eclipse**. JFrame is used for designing a frame of an interface in which nothing will be available except the blank message box. JDialog is used for designing a dialog in which buttons like, OK and CANCEL will be available from the initial point i.e. even when it is called they are available there. The interface can be designed and developed with the help of both JFrame and JDialog. The components that are used in JDialog, JFrame are JButton, JLabel and absolute layer etc. The absolute layer is applied in the JDialog, JFrame and JPanel so that the components added in these can be adjusted by dragging otherwise it will be automatically adjusted and won’t move further. There are many other components used in designing the interface in java. The IDE (integrated development environment) is Eclipse.

### Entities and Attributes

The entities used in the interface are Blood Bank, Patients, Other Employees (Staff Members), Specialist (Doctor), Bills, Receptionist, Diagnosis Details, Resources, Lab Manager and the attributes of these entities used in database are Blood\_ID, Blood\_Type, Blood\_Total, Patient\_ID, Patient\_Name, Patient\_Type, Patient\_CNIC, Patient\_DOB, Patient\_Number, Patient\_Ward, Patient\_Gender, Patient\_Address, OE\_ID, OE\_Name, OE\_Type, OE\_CNIC, OE\_DOB, OE\_Contact, OE\_Salary, OE\_Ward, OE\_Gender, OE\_Address, Specialist\_ID, Specialist\_Name, Specialist\_Type, Specialist\_Salary, Specialist\_DOB, Specialist\_Ward, Specialist\_Gender, Specialist\_CNIC, Specialist\_Address, B\_ID, B\_Items, B\_Total, B\_Type, Rec\_ID, Rec\_Name, Rec\_Salary, Rec\_Address, Rec\_Gender, Rec\_DOB, Rec\_Contact, Rec\_CNIC, Diagnosis\_ID, Type\_Diagnosis, Patient\_ID, Patient\_Name, ID\_Res, Total\_Res, Items\_Res, L\_ID, L\_Name, L\_ Address, L\_Contact, L\_CNIC, L\_DOB, L\_Salary, L\_Gender respectively.

### File Structure

File structure is that how to implement the data and store it more specifically how to handle the data that is stored.

For storing the data there are two ways:

* File Handling
* Database

File handling is a very complex way of storing data that a programmer gets in big trouble in storing data through file handling. File handling is a way of storing data in a text file that is not an efficient way of storing data. The database is an efficient way of storing data the data is inputted and stored in a database which removes the complexity, saves time and by assigning the attributes to the entities the work is done efficiently and it is very effective and easy to use as compared to file handling process of storing data.

# 3.1

## Main-Page



## Login Page



## Specialist (Doctors)



## Patients



## 

The interface of the Hospital Management System is like this the other pages are bills, receptionist, resources etc. All the pages that are designed are according to the requirements that are given by the Hospital named as **“**Darsons’ Hospital**”** to design and develop a hospital management system. All the functions are also according to the requirements that are given by the Hospital that is **“**Darsons’ Hospital**”**.

# 3.2

## Relationship between Objects

The relationships defined in the ERD are of receptionist, administrator and etc. Blood bank has a relationship with Receptionist the blood bank’s information is received from the receptionist. Ward details are also available that can be accessed through receptionist. The appointments assigned to the doctors are done by the receptionist.

## Java Docs

Java Docs makes an API file in the format of HTML which consists of coding with respect to Java and etc.



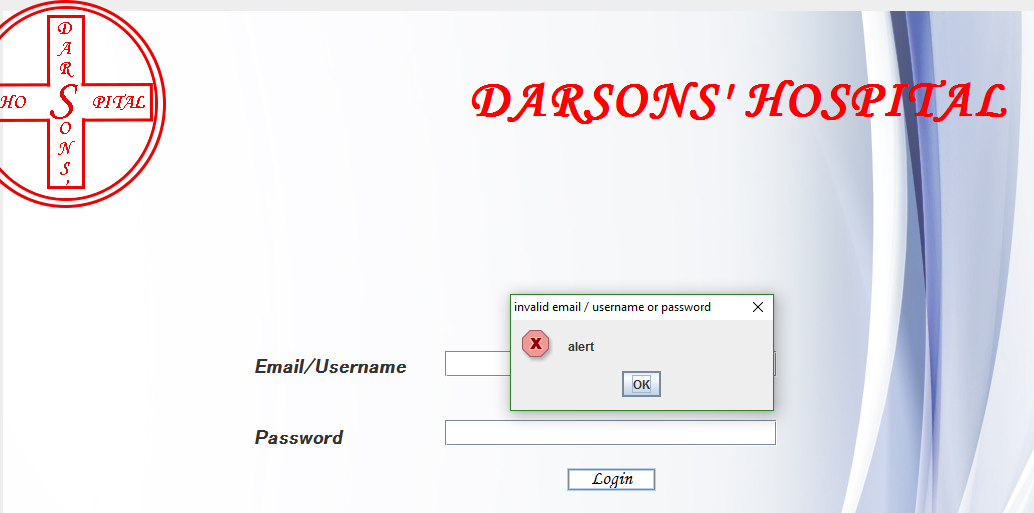
# 3.3

## Control Structure

The control structure is the coding part of the interface. The loops and conditions that are used in the designing and developing the interface is a part of control structure. The actions that are performed on the buttons are also a part of control structure e.g. moving from one page to another page i.e. Bills s = new Bill and then s.setVisible (true); here bill is a page on which the user has to move. setVisible is a function of Boolean data type so its parameter can be true or false and to move the parameter must be true. **“**dispose()**”** function is used behind the Cancel button to close the respective page and while moving from one page to another if dispose is also used then the page on which the user is will be closed and will move towards the other page. In this interface different functions are used two of them are these functions that are mentioned earlier.

# 3.4

## Pop-Up Message



In the login page if no data is entered in both the fields then the alert message will be displayed of invalid email / username or password. This is a message box or popup message that is shown to give user information and by providing authentication the hospital management system can be accessed otherwise it can’t be accessed which is one good approach to secure the system.

# 3.5

## IDE (Integrated Development Environment)

The integrated development environment used in designing and developing the management system is Eclipse Luna Service Release 2 (4.4.2) Version that is specially made for java developers. It is very effective to use as it provides the graphical representation of the components that are to be used and when it is applied in the JDialog or JFrame or JPanel the source code of it is automatically generated at the back-end of the component.

## Indentation

The indentation is that when the space is provided in the code while doing the coding.



*Indentation*

## 

# 4.1

## Critical Review

The hospital management system is working properly throughout as it is required. The requirements that are the Darsons’ Hospital has to manage patients, doctors, resources, bills, blood bank etc. The records are stored in the database after accessing the entities that are mentioned according to the entities that are gathered from the hospital. As per the requirements there should be the attributes of the patients that are patient’s name, patient’s address etc. so the hospital can have the track of it. Different kind of people also needs to be handled. The people are patients and the doctors etc. of the hospital. So, to fulfill this requirement pages are made having different data mentioned in it. As every person must have a name, contact number so the hospital can have a track of it. For the patients there must be information of the diagnosis handled by the receptionist. The employee can be of few types i.e. Specialist (Doctors) or Receptionist, Lab Manager and etc.

## Testing

### Input Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Text-field | Data Type | Input | Result |
| Patient (ID) | Number  -----  ----- | strNJjloi  +92ZZEE  3520208957321 | Pass Fail  Pass Fail  Pass Fail |
| Receptionist (Contact Number) | Number  -----  ----- | EKNHJzadf  +91BJNwwd  0321889743221 | Pass Fail  Pass Fail  Pass Fail |

|  |  |  |  |
| --- | --- | --- | --- |
| Text-field | Data Type | Input | Result |
| Patient (Address) | varchar  -----  ----- | Null  +92UUbxxs  657722344 | Pass Fail  Pass Fail  Pass Fail |
| Receptionist (CNIC) | varchar  -----  ----- | OPtvtubu  Null  89-767-8978 | Pass Fail  Pass Fail  Pass Fail |

### JUNIT Testing

The JUNIT testing is done to test the methods and techniques used in the code. The functions used are tested with the help of JUNIT testing.

## 

# 4.2

|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | Test Type | Expected Outcome | Actual Outcome |
| 1 | Adding | When the add button is clicked the data must be inserted into the database. | When the add button is clicked the data is inserted into the database. |
| 2 | Deleting | When the delete button is clicked the data that is available must be deleted from the database. | When the delete button is clicked the data that is available is deleted from the database. |
| 3 | Updating | When the update button is clicked the data that is available must be updated from the database. | When the update button is clicked the data that is available is updated from the database. |
| 4 | Searching | When the search button is clicked the data that is available in the database must be searched from the database. | When the search button is clicked the data that is available in the database is searched from the database. |
| 5 | Page Movements | When the specific page button is clicked it must move to that specific page | When the specific page button is clicked it moves to that specific page |
| 6 | Login Page | When the user logs in the user should have the access of the functions of the system otherwise the access must not be granted. | When the user logs in the system the user gets the access of the functions of the system otherwise the access is not granted. |

The analysis taken after doing the testing and getting the outcomes that were expected in actual outcomes is good. The inserting of the data in the database is done with the help of add button, deleting is done with the help of delete button, updating is performed on the data that is available in the database with the help of update button and searching is performed by clicking the search button available against the search bar given. Login page is working properly and is very essential for authentication and making the record safe and secured. And the moving of page from one to another is working perfectly according to the testing done. 4.3

## Independent Feedback

The feedback that is received from doing this is that the testing performed is very valuable and can be very efficient and effective. Because through testing it is checked that the working of the database is accurate and no problem / trouble are occurring while performing the functions in it. The input testing is also performed for checking the data. Security precautions are also done perfectly for securing the database that is a very good, efficient and effective approach.

## Recommendations

The database can be more efficient if the processing speed of it is increased. The functions performed must be separately done in the database. The storage of the database could be increased to make it more efficient than before i.e. 4TB to 8TB.

# 4.4

## User Documentation

* To access the hospital management system the user needs to login first and after the login user can access all the features of management system.
* The attributes of the entities that are required by the hospital are mentioned.
* The pages are designed and developed for it according to which the attributes are made in which the data is inserted, deleted, updated and searched.
* The Add button is given in the pages designed and developed according to the specific required entity. With the help of this the data that needs to be added is inserted as a record in the database.
* The Update button is given in the pages that are designed and developed for the respective entity. With the help of this button the data that is already there in the database tables can be updated if needed and the changed / updated record.
* The Delete button is available in the pages that are designed and developed with respect to the database tables that are given in the designed and developed database. With the help of this button the data that is no more important or essential to be there in the record can be deleted that deleted data will vanished from the tables made.
* The Search bar is given in designed and developed against the respective database tables in the database. With the help of this searching the data can be searched in the database.

# 4.5

## Technical Documentation

|  |  |  |
| --- | --- | --- |
| Table Name | Attribute | Primary Key |
| blood\_bank | Blood\_Type Blood\_Total  Blood\_ID | Blood\_ID |
| other\_employees | OE\_ID  OE\_Name  OE\_Type  OE\_CNIC  OE\_DOB  OE\_Contact  OE\_Salary  OE\_Ward  OE\_Gender  OE\_Address | OE\_ID |
| patients | Patient\_ID Patient\_Name Patient\_Type Patient\_CNIC Patient\_DOB Patient\_Number Patient\_Ward Patient\_Gender Patient\_Address | Patient\_ID |
| specialists | Specialist\_ID Specialist\_Name Specialist\_Type Specialist\_Salary Specialist\_DOB Specialist\_Ward Specialist\_Gender Specialist\_CNIC Specialist\_Address | Specialist\_ID |
| bill | B\_Items  B\_Total  B\_ID | B\_ID |
| receptionist | Rec\_ID Rec\_Name Rec\_Address Rec\_Gender Rec\_DOB Rec\_Contact Rec\_CNIC  Rec\_Salary | Rec\_ID |
| diagnosis | Type\_Diagnosis  Diagnosis\_ID  Patient\_ID Patient\_Name | Diagnosis\_ID |
| resources | Total\_Res  Items\_Res  ID\_Res | ID\_Res |

## Maintenance

The maintenance is done in many ways some of them are:

* **Back-up**
  + The data that is in the database is secured with the help of backing up the data. By backing up the data it remains safe that if the data is loss unfortunately at some occasion the backing up of the data can help in restoring the data.
* **Link Checking**
  + If there a problem occurs then the links created can be checked so that there stays no problem / trouble of crashing.

# Self-Criticism | Self-Evaluation

## Self-Criticism

The work done in the report could be done more efficiently. There should be more research papers because if there are more research papers according to the topic then the work done could be more authentic. Criticizing of the work is done very effectively. The software developer has given the details about the java docs and JUNIT but they are not done effectively because of this the readers won’t be able to understand the string more efficiently and perfectly.

The maintenance can be done in more detail that have been mentioned in the report could be done more efficiently i.e. the software developer has mentioned like few lines he could mention about a paragraph or more than a paragraph so it could be understandable for the reader more efficiently.

The topic of relationship between objects and indentation could be more explained than it is mentioned in the report by the software developer. The research papers according to it could be in more detail so that it would be easy for the reader to understand the respective topics.

The topics mentioned in the report could be more efficient i.e. explained with more detail so every person who reads the report could understand the topics.

## Self-Evaluation

The topics mentioned in the report could be explained more efficiently. The software developer has done a good work in writing the report like mentioning the details about the topics according to the assignment brief. The software developer has done all the work according to scenario given to him.

The tools and techniques used for the report are very effective in writing the report according to the scenario given to him. According to the researcher the work for distinction and merit is done very effectively so he is sure about the distinction and merit. For merit different types of working is done. The indentation screenshots are given which helps in understanding the indentation. The interface screenshots are given in the report which helps in understanding the interface.