



# Project Guide - Developing JavaFX Applications

# Project Guide Developing JavaFX Applications

© 2023 Aptech Limited

No part of this book may be reproduced or copied in any form or by any means – graphic, electronic or mechanical, including photocopying, recording, taping, or storing in information retrieval system or sent or transferred without the prior written permission of copyright owner Aptech Limited.

All trademarks acknowledged.

**APTECH LIMITED**

Contact E-mail: [ov-support@onlinevarsity.com](mailto:ov-support@onlinevarsity.com)

Edition 1 - 2023



## Preface

Today, the influence of IT spans over a wide area of domains including the medical field.

This Project Guide provides all the necessary information to students in developing a GUI application that can be used by patients, doctors, and hospital staff. Students have to understand the workflow and design the project accordingly.

The best way to learn something is to apply its principles and test it. Similarly, the best way to evaluate the tool knowledge of the students is to apply it through project work. The degree of success of the project depends on the strength of its Guide. This Project Guide has been prepared following the best practices in the Industry and helps you to have the experience of going through a LIVE project. It teaches you the essentials of successful development of IT projects.

The Project Guide will help you to:

- Analyze a project
- Design the specifications of the project
- Develop the solution
- Maintain disciplined documentation for the work done

This Project Guide reiterates the commitment of Aptech in keeping up its tradition of providing innovative, career oriented professional education. This ensures that modules are based on the Project Based Learning concept.

Religiously following the given systematic approaches in this book would prepare you to get the real life experience of handling projects. This is because the practices listed here have been extracted from the current industry norms. Thus, such an exercise would prepare you for joining the Software Development Industry.

The knowledge and information in this book is the result of a concerted effort of the Design Team, which is continuously striving to bring you the best and the latest in Information Technology. The process of Design has been a part of the ISO 9001 certification for Aptech-IT Division, Education Support Services. As a part of Aptech's quality drive, this team does extensive research and curriculum enrichment to keep it in line with industry trends.

We will be glad to receive your suggestions. Wishing you the very best.

Design Team

### Contents

1. How to Use this Project Guide?
2. Case Study
3. Team Building
4. Architecture of the Application
5. Phases in a Project's Life Cycle
6. Documentation Section
7. Hints and References





**LEARN**  
**@**  
**YOUR PACE**  
**YOUR**  
**DEVICE**

**Onlinevarsity**


Onlinevarsity



# Onlinevarsity App

for **Android** devices

*Download from* **Google Play Store**



# Project Guide - Developing JavaFX Applications

## 1.1 How to Use this Project Guide?

In this project-based learning approach, an entire project based on a case study has to be developed in the lab sessions. You will develop the solution for the case study throughout the labs. To do this, it is necessary to refer to the Project Guide as and when required.

Following guidelines will help to understand how to use Project Guide throughout your course:

1. Read the Case Study and understand the business scenario
2. Solve the project requirements so that different components of the application are ready
3. Document the design of the corresponding Process, Component, Interface, and Function (as applicable) as soon as you complete your lab exercise
4. Check whether your code follows common coding standards
5. Identify the areas where common coding standards can be implemented
6. Integrate the entire project that you have developed throughout lab sessions and test its functionality

## 1.2 Case Study

Philadelphia Medical Center (PMC) is one of the leading medical centers in Great Lakes Region, United States of America (USA). Most of the doctors in that area conduct their private practice in PMC. The treatment given by the medical center is believed to be one of the best. Due to its good service and fame, the number of patients visiting the doctors at the center is increasing day by day. With the constant rise in the patients, the staff of PMC find it difficult to take care of the daily transactions.

The management of PMC has decided to automate the entire system which will reduce the staff effort. The management has signed off a deal with OnTime Systems who will develop a software for PMC. The software application will help the medical center by managing doctors' appointments, patients' billing, treatment history, diagnostics information, and administrative activities.

Assume that you are a part of the team that will design and implement the application for PMC.

### **Proposed Solution**

PMC has decided to launch an application using JavaFX and Java, using which the staff will be able to enter the information regarding doctors, patients, and so forth. The application will also help to generate bills easily.

MediConnect is a software application that automates the entire work in PMC. MediConnect manages the data storage, retrieval, and billing activities. It also displays information based on specified criteria.



---

---

The staff member will have to first login to the application to benefit from the services provided by the application. To ensure complete security and authentication, maximum possible verifications and validations are provided. After successful login, the staff member can enter details pertaining to doctors, patients, checkups, and bills. All the details are stored in a database, in the form of tables.

MediConnect is beneficial in following ways:

- Stores patient information
- Stores doctor information
- Stores details about each visit of the patient
- Generates bills for amount to be paid by patients
- Enables easy retrieval of patient history information

Following are further requirements from the MediConnect software application:

1. Verification and validation of a staff member
2. A GUI should be created which displays a menu with various options depicted on it
3. The staff of PMC can add, view, update, and delete details of doctors and patients
4. The staff of PMC can enter transactions related to patient visits, diagnosis, and treatments
5. The staff of PMC can generate and print the bills for patients
6. The staff of PMC can query for different items such as number of patients visiting the hospital on a given date, number of doctors, and so forth
7. Exit from the application if the user does not want to continue



---

## 1.3 Team Building

The commission and implementation of any computerized system involves the work of a team of people. A Project Leader leads the team. A Project Leader essentially decides which tasks are to be performed by each team member and how much time should be allotted to each Project Development Phase.

A team typically consists of:

➤ **Analyst**

The Analyst studies the requirements of the system and defines the problem. The Analyst identifies the requirements of the system and determines the input, output, and processes involved in transforming those inputs into outputs.

➤ **Designer**

The Designer creates a blueprint of the system in terms of the database structure, screens, forms, and reports.

➤ **Developer/Programmer**

The Developer builds the user interface according to the specifications prepared by the Designer. Next, the developer builds a prototype of the system. After receiving client approval on the prototype, the Developer adds the necessary code to make the prototype a full-fledged system.

➤ **Tester**

The Tester tests the working of an application by first testing each module for its functionality. Test data is used to check if the module is able to process it without causing any errors. Test data may be live data extracted from existing records in the system or dummy data. The Tester then also verifies the integrated application's functionality with test data.

➤ **Implementation Engineer**

The implementation engineer ports the completed application to the Client's computers. The implementation engineer ensures that the installation process has been carried out accurately, and hands over the system to the Client.

➤ **Maintenance Engineer**

The Maintenance engineer is responsible for taking care of maintaining the system that has been built. Maintenance includes extending troubleshooting support, and performing software upgrades in case of changes in the external system.

---

---

Though each role is huge in itself, the same team member may perform more than one role in a team. For example, one person could assume the role of both Analyst and Designer.

The project team interacts with the client (customer for whom the project is being developed). For your project, the teacher takes on the role of the client. Hence, all the interactions that the project team performs with the client, you will perform with the Teacher.

For the **MediConnect application**, the roles of the Implementation Engineer and Maintenance Engineer are beyond the scope of this project. Since each one of you has to learn various roles, the members of the Project Group will perform all the roles of Analyst, Designer, Programmer and Tester together. Each of you will not be assigned one role and be limited to performing that role only.

## 1.4 Architecture of the Application

Since the application will have client-server architecture, it will have a two-tier architecture. You can have a combination of GUI components and console based functionality for the application.

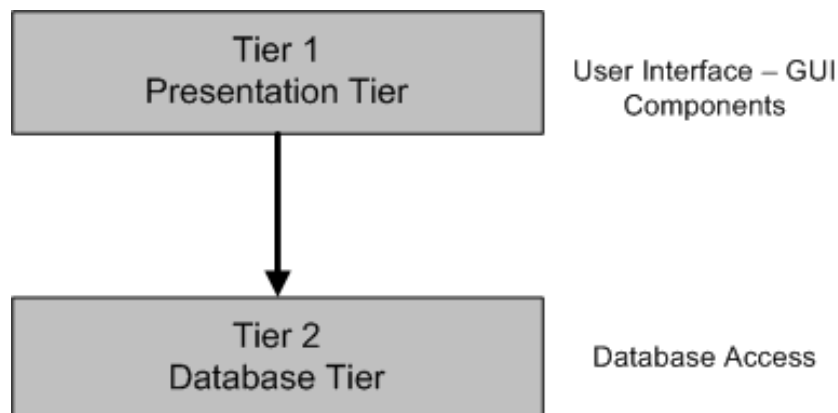


Figure 1.1: Two-Tier Architecture of the Project

## Flow Chart

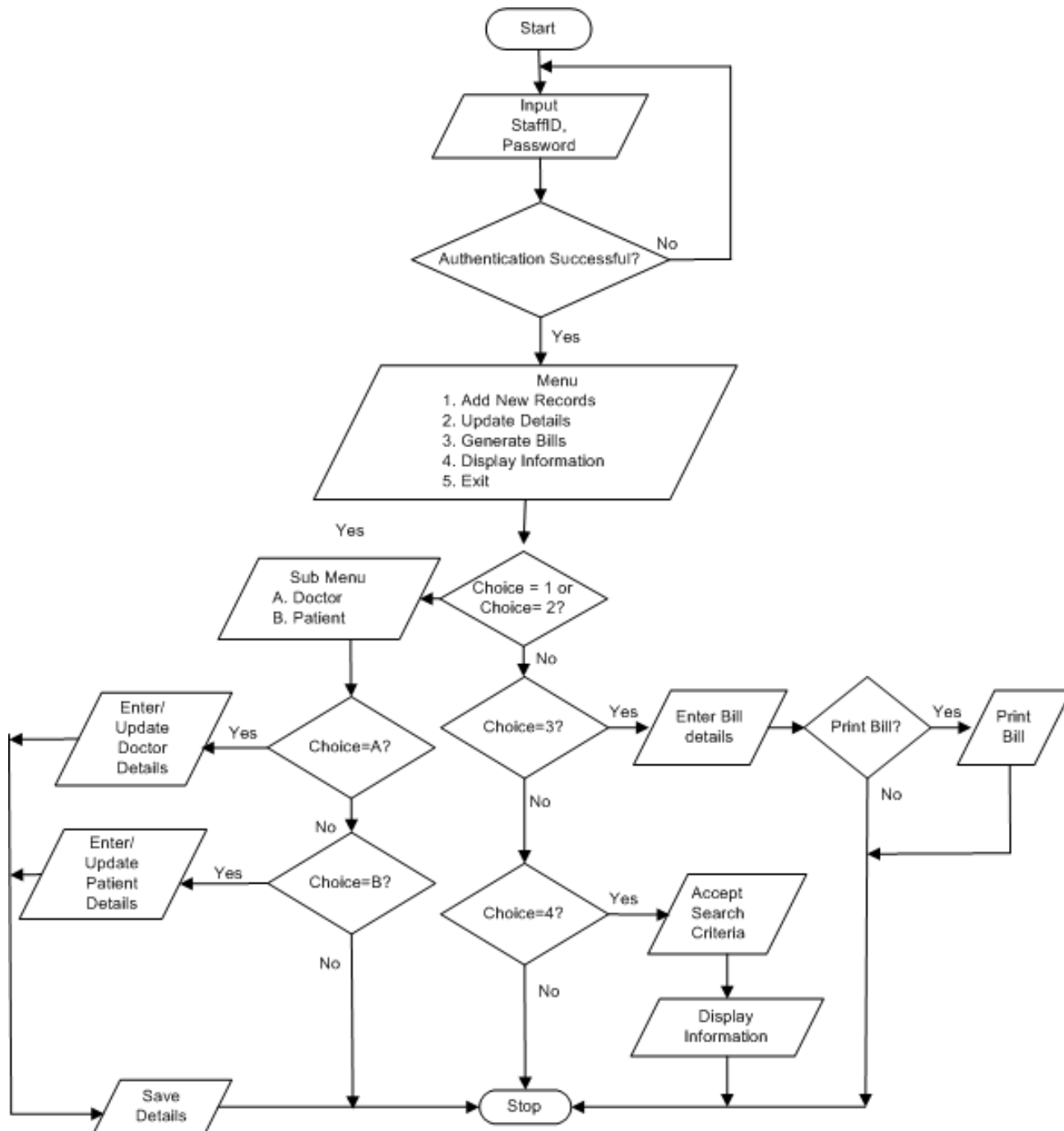
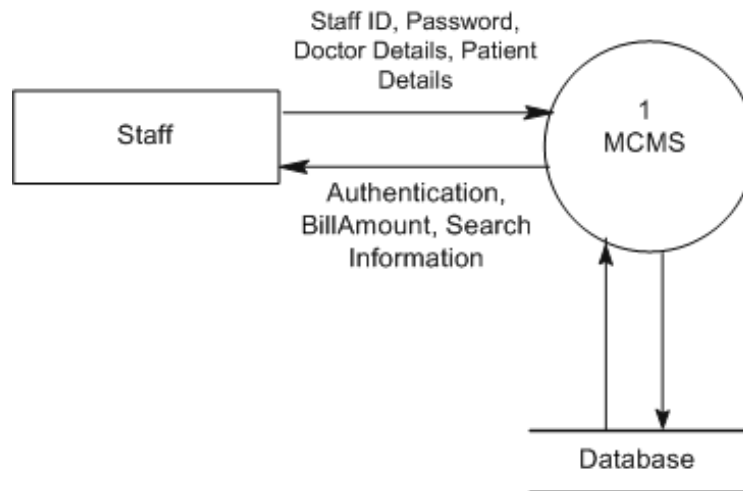


Figure 1.2: Flow Chart

---

## Data Flow Diagram

The flow of data in the application is shown by various data flow diagrams. The most basic data flow diagram is the Level 0 diagram. It shows the basic flow of data in and out of the system.



**Figure 1.3: DFD LEVEL 0 - MediConnect**

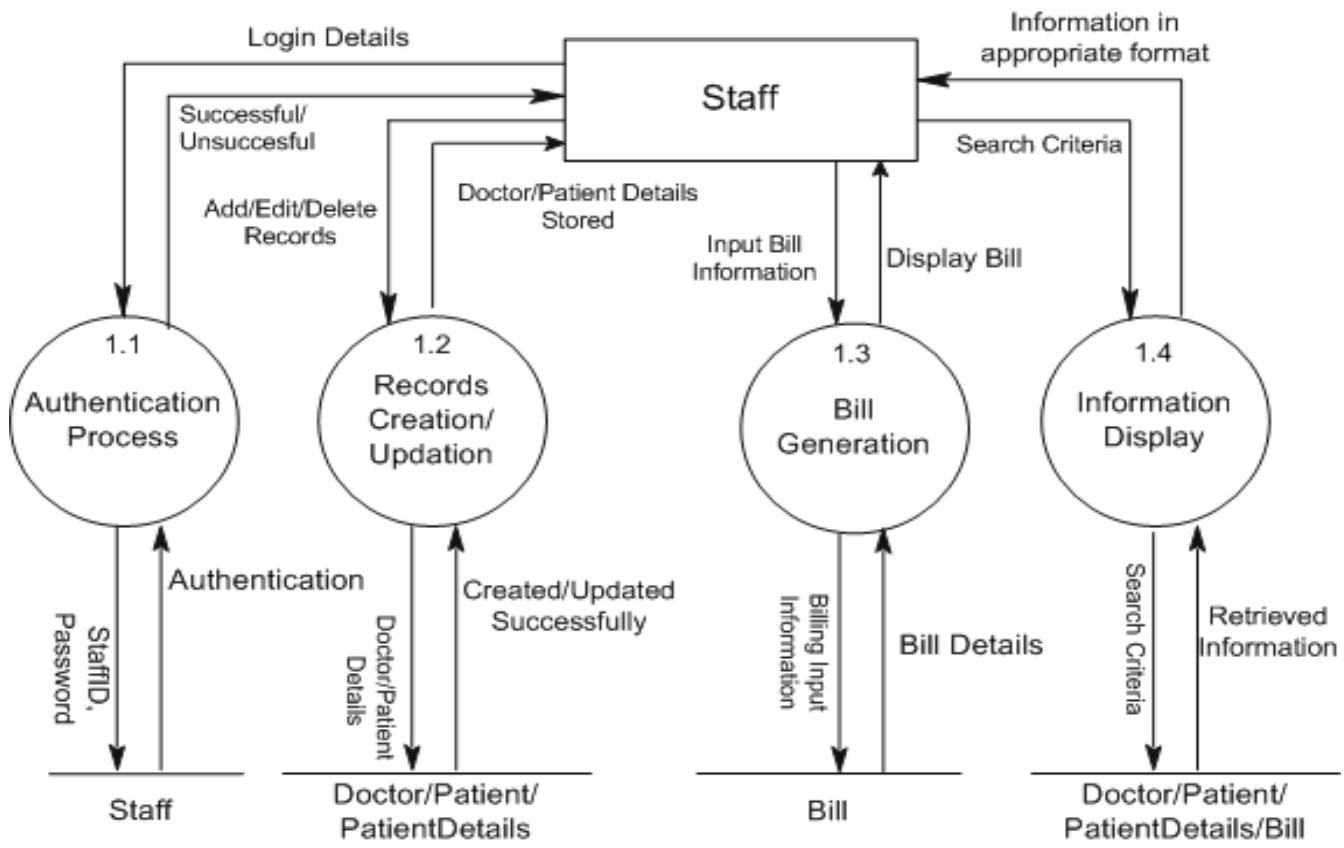
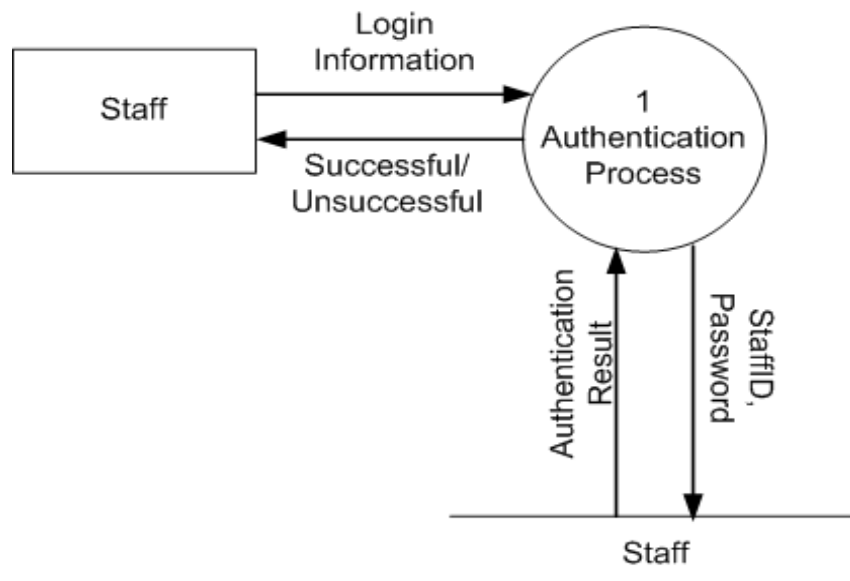


Figure 1.4: DFD LEVEL 1 - Main Processes





**Figure 1.5: DFD LEVEL 1.1.1 - Authentication Process**

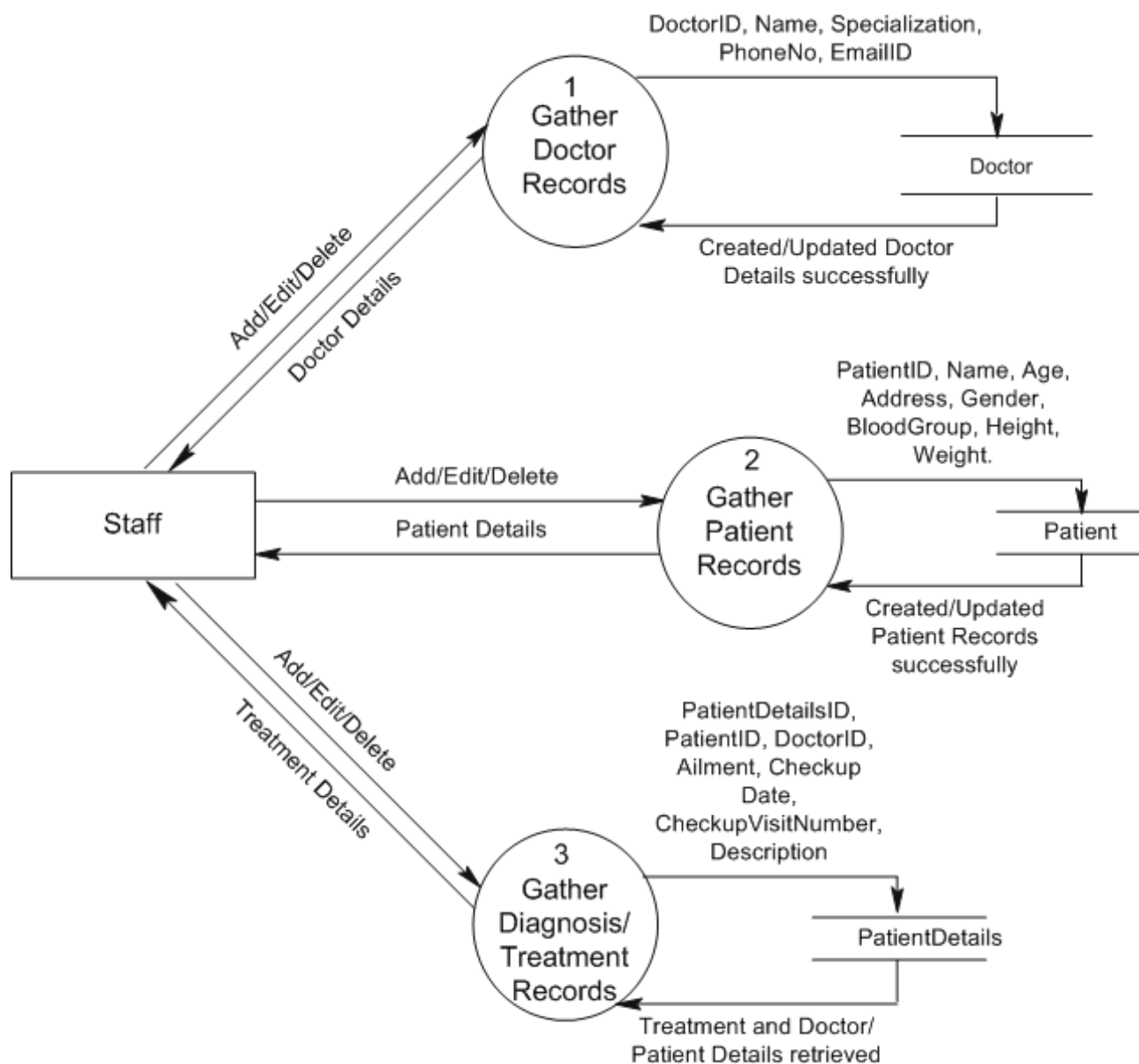


Figure 1.6: DFD LEVEL 1.2.1 - Records Creation/Updation Process

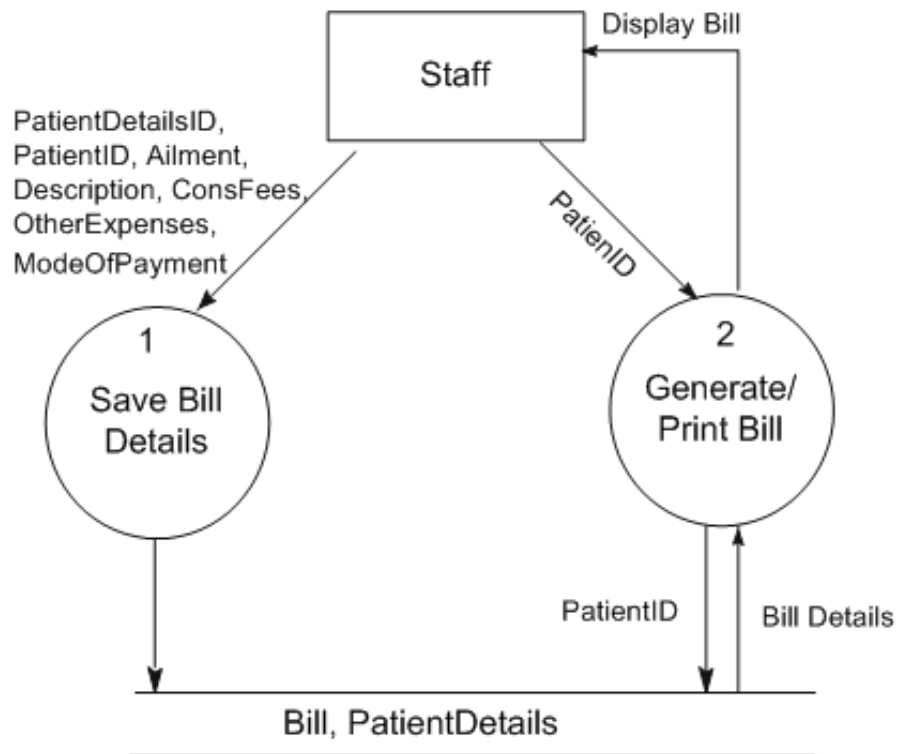
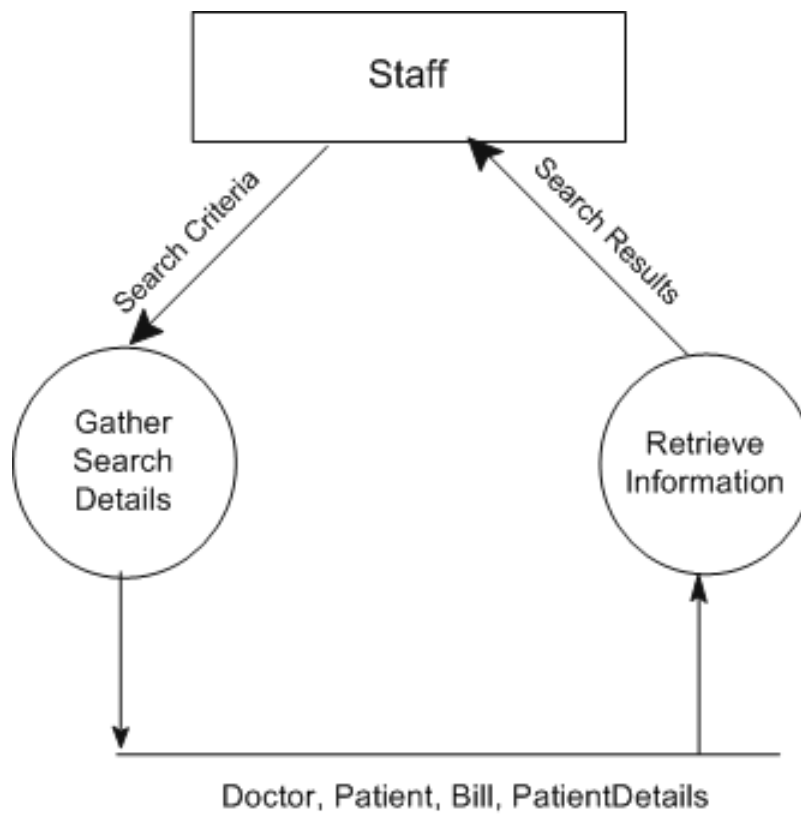


Figure 1.7: DFD LEVEL 1.3.1 - Bill Generation Process



**Figure 1.8: DFD LEVEL 1.4.1 - Information Display Process**

Entity Relationship (ER) Diagram

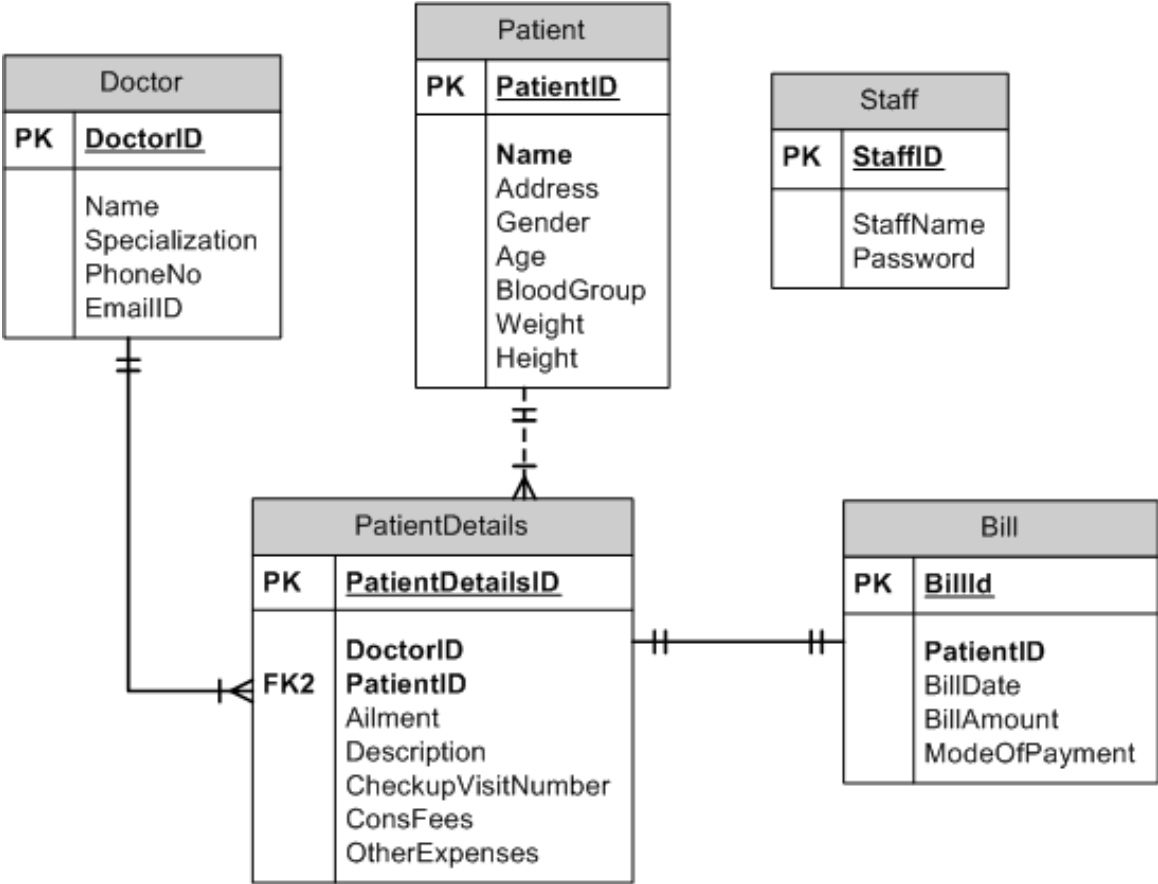


Figure 1.9: E-R Diagram



---

## Database Design/Structure

### Table Design

Table: Staff				
Field Name	Data Type	Null	Key	Description
StaffID	varchar(4)	No	PK	Stores the identification code of the staff
StaffName	varchar(50)			Stores the name of the staff
Password	varchar(10)			Stores the password

Table: Doctor				
Field Name	Data Type	Null	Key	Description
DoctorID	varchar(4)	No	PK	Stores the identification code of the doctor
Name	varchar(50)			Stores the name of the doctor
Specialization	varchar(50)			Stores the specialization of the doctor
PhoneNo	int			Stores the phone number of the doctor
EmailID	varchar(30)			Stores the e-mail id of the doctor

Table: Patient				
Field Name	Data Type	Null	Key	Description
PatientID	varchar(4)	No	PK	Stores the identification number of the patient
Name	varchar(50)			Stores the patient name
Address	varchar(40)			Stores the patient address
Gender	varchar(1)			Stores the patient gender
Age	int			Stores the age
BloodGroup	varchar(4)			Stores the blood group
Weight	decimal			Stores the weight
Height	decimal			Stores the height

---

## Database Design/Structure

### Table Design

Table: PatientDetails				
Field Name	Data Type	Null	Key	Description
PatientDetailsID	varchar(4)	No	PK	Stores identification code of each patient with the corresponding doctor
DoctorID	varchar(4)	No	FK	Stores identification number of the doctor
PatientID	varchar(4)	No	FK	Stores identification number of the patient
Ailment	varchar(50)			Stores ailment of the patient
Description	varchar(150)			Stores description in detail
CheckupVisitNumber	int			Stores current visit number. This is useful for tracking follow-ups
ConsFees	int			Stores the consultation charges
OtherExpenses	int			Stores any other expenses incurred such as X-ray charges and so on

Table: Bill				
Field Name	Data Type	Null	Key	Description
BillID	int	No	PK	Stores identification number of the bill
PatientID	varchar(4)	No	FK	Stores identification number of the patient
BillDate	datetime			Stores bill date
BillAmount	decimal			Stores the total amount
ModeOfPayment	varchar(6)			Stores the details of the mode of payment such as cheque, cash, or other mechanisms

---

## 1.5 *Phases in a Project's Life Cycle*

Every software development activity can be distinguished into clear phases. Each phase has precise starting and ending points, with clearly identifiable deliverables to the next phase. Each phase may have certain documents, which help to keep track of various activities, processes, procedures, inputs and outputs associated with that phase of the Project.

The project development life cycle consist of following phases:

- Phase 1: Definition of the Problem
- Phase 2: Requirement analysis
- Phase 3: Design
- Phase 4: Development
- Phase 5: Evaluation/Testing
- Phase 6: Implementation
- Phase 7: Maintenance

Each of these phases is discussed.

### 1.5.1 *Definition of the Problem*

Correct and accurate deduction of the client requirements and expectations from the system is the key to the development of the system. This phase assumes more significance as it has cost implications attached and these will affect the whole project. When the requirement is wrongly identified, the whole system will be developed on a wrong premise.

Therefore, this phase involves defining the problem and fixing up its boundaries. The requirements and problems faced by the client are recorded in this phase. At the end of this phase, the team is clear with the project objectives and their work purview. Inputs to this phase are always unstructured. These inputs are gathered from interactions with the client.

The activities involved in this phase are as follows:

- Meeting the Client
- Understanding the Client's requirements
- Identifying the probable solution
- Defining the scope of the project

---

---

You must identify the Problem Definition and record it in the ***Problem Definition Document*** in 'SWD/ Form No.1'

Once a client's problem has been identified, the next step would be to generate the **Customer Requirement Specification** as a part of the Requirement Analysis phase.

Now, carefully study and understand the Case Study specifications in the ***MediConnect*** application.

## 1.5.2 Requirement Analysis Phase

During this phase, the Analyst identifies the processes of the current system, the inputs and outputs for those processes. The processes, inputs, and outputs are recorded in a document called the Customer Requirement Specification. Actually, the **Customer Requirement Specification** is not a single document; it is a docket of many documents. It consists of:

➤ **List of inputs to the system**

These refer to the inputs required for the system to work on and thus, produce the desired output.

**For example**, in the ***MediConnect*** case, some inputs would be patient details or staff details.

➤ **List of outputs expected from the system**

These refer to various reports that the system would produce.

**For example**, in the ***MediConnect*** case, some of the outputs would include patient record or bills.

➤ **Overview of processes involved in the system**

Once the inputs and outputs are listed down, the list of processes that convert the inputs into desired outputs is prepared.

**For example**, in the ***MediConnect*** case, this list would include the following:

- Login
- Accepting and saving Patient data
- Displaying doctor data

➤ **Hardware and software required for implementing the project**

Here, the team gives a list of software and hardware required to implement the system. The Client uses this information to get an approximation of the required setup and training requirements.

**For example**, in the ***MediConnect*** application, the software required could be **Eclipse IDE 2022, SQL Server 2022, Java SDK 21, JavaFX 20, and related libraries**.

---

For **MediConnect**, you must record the same in the document '**SWD/Form No.2A**'.

➤ **Customer's acceptance criteria for the project**

The software product developed must meet the Client's requirements and expectations as specified in the problem definition. Apart from this, it should also fulfill the performance, speed, and reliability parameters specified by the Client.

**For example**, in the **MediConnect** application, the acceptance criteria could be the following:

- Create an administrator account
- Login in to the system with the created administrator ID
- Add Patient
- Delete Patient
- Modify Patient
- Display all Patients
- Search for Patient

For the **MediConnect** application, you must record the **Customer Acceptance Criteria** in the document '**SWD/ Form No.2B**'. This Customer Requirement Specification Document requires client approval, which is termed as **sign-off**. Once the Client has given a sign off on the **Customer Requirement Specification** Document, the **Project Leader** creates a '**Project Plan**'. A Project Plan is a brief of all the phases, deliverables, milestones, implementation issues related to a project.

For the **MediConnect** application, you must create a '**Project Plan**' and submit to the teacher for review. You can fill your **Project Plan** details in the document '**SWD/Form No. 3**' given in the **Documentation Section** of this Project Guide. Your teacher will prepare the '**Project Plan**' for the **MediConnect** and demonstrate it to you.

### 1.5.3 Design Phase

Design Phase involves the preparation of the blueprint of the proposed system, which involves the following:

➤ **Designing GUI Standards**

When an application is designed, it must follow standards with respect to flow, appearance and look of an application. Standards are used to bring about consistency throughout the application.

**GUI Standards** are related to the appearance of an application. It is mandatory for the Project Leader and team members to visualize the entire look of the application before it is actually developed. This visualization has to be defined in terms of GUI Standards so that each screen being



---

developed maintains consistency in look and flow. The color, font style, size of titles and labels, appearance of header and footer, theme, position and size of controls on various screens are defined here.

**For example**, in the case of **MediConnect** application, multiple forms can be created that are linked to each other. Each of these forms should have a consistent look with respect to appearance, theme, and the color schemes used. The font styles, color of the labels, design and appearance of the command buttons, the appearance of header and footer, and the design and size of controls such as check boxes or text boxes should be consistent through all the forms.

For the **MediConnect** application, set your GUI Standards using the document '**SWD/ Form No. 4**' given in the **Documentation Section** of this Project Guide.

### ➤ **Designing the Interface**

Here, the layouts of the screens are designed in line with the **GUI Standards** set. Either these screens can be input forms, which accept user inputs, or reports that display information to the user. In this phase, the content and appearance of the input forms and reports are decided. The number of forms and purposes of each form are decided here.

The navigation details of the entire application are also specified during this phase if the application has a browser interface.

The Interface Design is recorded in the **Interface Design Document**.

For the **MediConnect** application, you must record your **Interface Design** details in the document '**SWD/Form No.5**' given in the **Documentation Section** of this Project Guide.

### ➤ **Designing the Database**

In this phase, the tables have to be created according to the Database design given in the project guide.

For the **MediConnect** application, you must record your **Table Design** details in the document '**SWD/ Form No.6**' given in the **Documentation Section** of this Project Guide.

### ➤ **Designing the Process Modules**

Process design involves translating the process definitions arrived at in the analysis phase into code modules. This module design is then expanded into program specifications. The types of validations required to verify the functionality of each process in the project, are also specified in this phase.

**For example**, in the case of **MediConnect** application, the modules could be for:

- Patient Registration
- Patient Management
- Bill Management

---

For the **MediConnect** application, you must record your **Process Design** details in the document '**SWD/Form No.7**' given in the **Documentation Section** of this Project Guide.

## 5. Designing Coding Standards

The process modules designed will be considered good only if they are standardized. Standardization involves setting up of naming conventions of program entities and database referencing, to name a few. Standardization helps to bring about better readability and easy maintenance of the code. Standard naming conventions make it easy to refer to program entities such as forms or modules. When standard naming conventions are used, the names of variables, and modules will denote their purpose even to a person other than the developer.

The Teacher will show you how to set the Coding Standards using the **MediConnect** application.

For the **MediConnect**, application you must define and state your **Coding Standards** in the document '**SWD/Form No.8**' given in the **Documentation Section** of this Project Guide.

## 6. Building the Prototype

Next, a Prototype of the application is created and shown to the client for approval. The Designer delegates the development of the prototype to the developer. The Prototype is a model of what the application would look like. The Prototype's screens allow the Client to view the User Interface of the application. In case a browser interface is used, the Client gets an idea of the navigation sequence of the application. The Prototype also helps the Client to understand the functionalities that will be achieved in the completed application. Once a Prototype is created, it requires Client approval. The Client has to give approval for:

- The navigation sequence of the application
- The look and appearance of the application
- The functions that will be performed by the application when fully developed.

There is no document associated with the Prototype since the client is shown the prototype on the computer for approval.

Once all the **Design Specification Documents** have been prepared, they require Client approval, which is called **sign-off**.

➤ After the sign-off, the system is taken up for development. Now, if the client requests massive new additions or changes to be made, then a document known as **IMPACT ANALYSIS** is generated.

➤ **Impact Analysis** is a document which informs the client about the additional cost to be incurred as a result of the changes suggested by him and also the time delays involved due to the requested changes.

After a Client gives a sign off on Impact Analysis, the requested changes are made.

---

During the Design Phase, the Project Leader also carries out the task of **Assigning and Monitoring Tasks**.

➤ **Assigning and Monitoring Tasks**

When the project development starts, the Project Leader must budget the number of person-hours required to complete the project. In order to do this, the Project Leader creates a '**Task Sheet**'.

A task sheet is used for recording the following:

- The number of man-hours required to complete the project
- The planned start date of the project
- The planned end date of the project
- The number of members required in the project team
- The Module name and specification allocated to each project member
- The time required to complete each module
- The progress status

With a Task sheet, the Project Leader will be able to monitor and track various phases of project development. It helps the Project Leader to anticipate delays, and therefore become proactive and take preventive action.

For the **MediConnect** application, you are expected to fill the **Task sheet** as shown in the document '**SWD/Form No.9**' given in the **Documentation Section** of this Project Guide. The teacher will demonstrate how to fill the **Task sheet** using the **MediConnect** application.

## 1.5.4 Development Phase

This phase involves the following:

- Giving the finishing touches to the User Interface built during the prototype
- Building the Database with the RDBMS tool
- Making the screens functional by adding code to them

For the **MediConnect** application, you must now implement the User Interface using JavaFX. You are then, required to create database in SQL Server 2022.

## 1.5.5 Evaluation/Testing Phase

This is the most crucial phase where each unit is tested for its functionality. Test data is used to check if the module is able to process it without causing any errors. Test data may be live data extracted from existing records in the system, or dummy data. Then, the individual tested modules are integrated and

---

tested as a whole through its various paths.

During this phase, the Project Leader reviews the developed system against each of the Customer Requirement Specifications and thus ensures that the developed system is able to resolve the Problem Definition completely.

After finishing the project, give it to your peers for testing the entire project. The project should be verified for the data entered by the customer in various forms. The findings can be records in the document '**SWD/Form No. 10**'.

### ***1.5.6 Implementation Phase***

In this phase, the developed system is ported to the client's computers. The implementation engineer ensures that the installation procedure has been performed accurately. At the end of this phase, a final sign off is taken from the client.

### ***1.5.7 Maintenance Phase***

In this phase, troubleshooting support is given to the Client. Depending upon external changes in the system, any software upgrades required by the application are performed.

### ***1.5.8 Project Tracking and Monitoring Activities***

During the Project Development Life Cycle, the Project Leader and other team members follow certain monitoring procedures and practices which help to streamline projects and achieve the project objectives. These activities happen in parallel to the process of project development. The Project's successful completion depends on these activities. These are known as project reviews.

A Review is a procedure used to check the progress of a project. It helps to understand and resolve constraints and to ensure that the project is progressing as per the Project Plan and will meet all Project Specifications, Guidelines, and deadlines.

**Reviews** can be:

- Internal Reviews- reviews conducted within the team members
- External Reviews – reviews conducted with the Client
- On a weekly or a monthly basis and can be conducted either through e-mails or in person

---

---

Review Minutes are recorded in the Project Review and Monitoring Reports document.
--

For the MediConnect Project, your teacher will conduct a review to understand the project status and record the findings of the Project Review in the document '**SWD/Form No.11**', given in the **Documentation Section** of this Project Guide.

Finally, your project would be considered complete when you give a final demonstration of the project to the teacher. During the demonstration, the teacher will fill and sign the '**Final Check List**' (document –'**SWD/Form No.12**') given in the Documentation Section. After this, furnish a spiral bound book consisting of following documents:

- Certificate of Completion
- Table of Contents
- Definition of the Problem
- Customer Requirement Specification Document
- Project Plan
- GUI Standards Document
- Interface Design Document
- Coding Standards Document
- Process Description Document
- Task Sheet
- Integration Testing Report
- Review Report
- Final checklist



---

## 1.6 Documentation Section

➤ Certificate of Completion

	<p>This is to certify that</p> <p><b>Mr./ Ms</b> .....</p> <p>Has successfully Designed &amp; Developed</p> <p>.....</p> <p>Submitted by:</p> <p>.....</p> <p>Date of Issue:.....</p> <p>Authorized Signature:.....</p> <p>.....</p>	
		

---

➤ **Various forms required in the project**

<b>Design Plan:</b>	<b>Document Name: Problem Definition Document</b>	<b>SWD/Form No. 1</b>
<b>Effective Date:</b>	<b>Version : 1</b>	<b>Page No.:</b>

*Problem definition of MediConnect –*

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

---

<b>Design Plan:</b>	<b>Document Name: CRS</b>	<b>SWD/Form No. 2A</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

**Client/Project Undertaken:** \_\_\_\_\_

**1. List of inputs to the system**

---



---



---

**2. List of outputs expected from the system**

---



---



---

**3. Overview of processes involved in the system**

---



---



---

**4. Hardware and software required for implementing the project**

---



---



---



---

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

<b>Design Plan:</b>	<b>Document Name: CRS/ Customer Acceptance Criteria</b>	<b>SWD/Form No. 2B</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

<b>S.No</b>	<b>Customer's Acceptance Criteria</b>
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

---

<b>Design Plan:</b>	<b>Document Name: Project Plan</b>	<b>SWD/Form No. 3</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

## Project Plan

### 1. Project Details

- **Name of the Client:**

\_\_\_\_\_

- **Date of Project Plan:** \_\_\_\_\_

- **Project Vision/Objectives:**

(Define the project vision/objectives as stated by the client)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

- **Scope :**

(Mention the scope of the project giving the locations that will be covered, processes, and range of services, and so on)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

- **Our understanding of the client organization**

(Give the range of services, functions, overview of processes, and so on)

\_\_\_\_\_

\_\_\_\_\_

- **Project Organization with Responsibilities and Authorities:**

(Give the name of Project team members their roles and responsibilities)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

---

---

2. **Project Initiation/Requirement Documents:**

(Information required from the Client as inputs regarding the system; could be the information about the services, processes, and so on)

---

---

---

---

3. **Deliverables**

(The documents to be handed over to the client – such as CRS, Design Document, Installation Manual, User Manual, Maintenance Manual, and Code documents)

---

---

---

---

4. **Project Dependencies**

Any event or task outside the scope of the project, which will affect the success of the project

---

---

---

---

5. **Major Project Milestones**

(Generating CRS, Building a Prototype, and so on)

---

---

---

---

6. **Quality Plan**

- **Review Activities** (Review meeting participants, frequency, and so on)

---

---

- 
- 
- 
- **Testing Activities** (Final Test)

- 
- 
- 
- **Backup and recovery strategies** (in case of disk crash, network failures, and so on)

---

---

---

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

<b>Design Plan:</b>	<b>Document Name:</b> GUI Standards Document	<b>SWD/Form No. 4</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

## Document Design

Property	Value
Document theme and color scheme	
Form - Background color	
Title - Font Size	
Title - Font Color	
Title - Font Style	
Title - Alignment	
Background color of Controls on the form	
Foreground color of Controls on the form	
Control Caption -- Font size	
Control Caption - Font Color	
Control Caption - Font Style	
Control caption and controls - Alignment	
Command button - Alignment	

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		



<b>Design Plan:</b>	<b>Document Name: Interface Design Document</b>	<b>SWD/Form No. 5</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

1. List of forms to be created

Document Name	Description	Controls on the Document

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

<b>Design Plan:</b>	<b>Document Name: Table Design Document</b>	<b>SWD/Form No. 6</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

<b>Name of the Table</b>	<b>Table Description</b>	<b>Number of Fields</b>	<b>Primary Key</b>	<b>Related Tables</b>	<b>Foreign Key</b>
				1. 2. 3.	

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		



---

<b>Design Plan:</b>	<b>Document Name: Coding Standards Document</b>	<b>SWD/Form No. 8</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

1. **Programming Standards**

---



---

2. **Standards for code writing style**

---



---

3. **Standards for declaring Variables**

---



---

4. **Standards for Function Declarations**

---



---

5. **Other Standards**

---



---

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

<b>Design Plan:</b>	<b>Document Name: Task Sheet</b>	<b>SWD/Form No. 9</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

<b>Project Ref. No.:</b>	<b>Project Title:</b>	<b>Activity Plan Prepared By:</b>		<b>Date of Preparation of Activity Plan:</b>		
<b>Task Sub division</b>	<b>Description</b>	<b>Planned Start Date</b>	<b>Actual Start Date</b>	<b>Actual Days</b>	<b>Team Member Names</b>	<b>Status</b>
1.						
2.						

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

<b>Test Document:</b>	<b>Document Name: Testing Document</b>	<b>SWD/Form No. 10</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

<b>S.No.</b>	<b>Features Tested</b>	<b>Remarks</b>

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

<b>Review Report:</b>	<b>Document Name: Project Review</b>	<b>SWD/Form No. 11</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

<b>Date:</b>	<b>Project Plan Activity/ Milestone</b>	<b>Work Specification</b>	<b>Status of the Activity</b>	<b>Remarks</b>	<b>Responsibility</b>

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		

---

## Final Check List

(This document has to be filled by the teacher only. The teacher will do a complete functionality testing of the application and add his/her findings and suggestions to this form.)

<b>Test Document:</b>	<b>Document Name: Final Check List</b>	<b>SWD/Form No. 12</b>
<b>Effective Date:</b>	<b>Version: 1</b>	<b>Page No.</b>

<b>S.No.</b>	<b>Aspect tested</b>	<b>Suggestions/ Remarks</b>
1.	Have all the functionalities been properly integrated and are they completely functional?	
2.	Does each unit meet its objective and purpose?  Are all the validations happening as specified in Process Design?	
3.	Have all Design and Coding standards been followed and implemented?	
4.	Is the GUI design consistent all over?	
5.	Are the codes working as per the specification?	
6.	Does the application's functionality resolve the client problem, and satisfy his/her requirements completely?	
7.	Have the hardware and software been correctly chosen?	
8.	Additional features and utilities that give value addition to the entire project.	

	<b>Prepared By (Student)</b>	<b>Approved By (Teacher)</b>
<b>Date</b>		



---

## 1.7 Hints and References

You will use Java, JavaFX, JDBC, and related libraries to accomplish the functionality for the project.

You can use basic and advanced concepts of Java and JDBC (such as stored procedures, transactions, and more) wherever necessary. Ensure all relevant libraries are added to the Java classpath in Eclipse by using **Configure → Build Path** option.

A simple example of a JavaFX login form that authenticates user credentials using JDBC to connect to an SQL Server database. is shown here. Note that this is a basic example and in your project, you should use more secure practices, such as prepared statements, to enhance security. Replace placeholders in the code such as `your_server`, `your_port`, `your_database`, `your_username`, and `your_password` with your actual SQL Server connection details. You may also have to modify the string as per authentication mode used in SQL Server. Also, remember to handle exceptions properly.

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class LoginApp extends Application {
    private static final String DB_URL = "jdbc:sqlserver://your_server:
        your_port;databaseName=your_database";

    private static final String DB_USER = "your_username";
    private static final String DB_PASSWORD = "your_password";

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Login Form");
        GridPane grid = new GridPane();
        grid.setHgap(10);
        grid.setVgap(10);
        grid.setPadding(new Insets(25, 25, 25, 25));
        Label usernameLabel = new Label("Username:");
        TextField usernameTextField = new TextField();
        Label passwordLabel = new Label("Password:");
        PasswordField passwordField = new PasswordField();
```

---

```

        Button loginButton = new Button("Login");
        grid.add(usernameLabel, 0, 0);
        grid.add(usernameTextField, 1, 0);
        grid.add(passwordLabel, 0, 1);
        grid.add(passwordField, 1, 1);
        grid.add(loginButton, 1, 2);
        loginButton.setOnAction(event -> {
            String username = usernameTextField.getText();
            String password = passwordField.getText();
            if (authenticateUser(username, password)) {
                // Authentication successful, you can perform further
                //actions

                System.out.println("Login successful");
            } else {
                // Authentication failed
                System.out.println("Login failed");
            }
        });

        Scene scene = new Scene(grid, 300, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private boolean authenticateUser(String username, String password) {
        try (Connection connection = DriverManager.getConnection(DB_URL,
            DB_USER, DB_PASSWORD)) {

            String query = "SELECT * FROM Users WHERE username = ? AND
password = ?";
            try (PreparedStatement preparedStatement = connection.
                prepareStatement(query)) {
                preparedStatement.setString(1, username);
                preparedStatement.setString(2, password);
                try (ResultSet resultSet = preparedStatement.executeQuery()) {
                    return resultSet.next(); // If there
                    //is a match, the user is authenticated
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}

```

---

## Helpful Links:

For Java coding conventions, refer to following link:

<https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

For more tips on creating JavaFX applications along with sample source code, refer to following link and look under the right pane:

[https://docs.oracle.com/javafx/2/get\\_started/css.htm#sthref18](https://docs.oracle.com/javafx/2/get_started/css.htm#sthref18)

For building connection string, refer to following link:

<https://learn.microsoft.com/en-us/sql/connect/jdbc/building-the-connection-url?view=sql-server-ver16>