

IT Demand Manager Requests Management System

**G.K Wickramasinghe
2021**



IT Demand Manager Requests Management System

**A thesis submitted for the Degree of Master of
Information Technology**

**G.K Wickramasinghe
University of Colombo School of Computing
2021**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: G.K Wickramasinghe

Registration Number: 2018/MIT/085

Index Number: 18550859



Signature:

Date: 2021/11/28

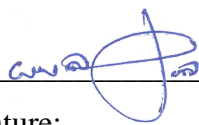
This is to certify that this thesis is based on the work of

Mr./~~Ms.~~ Gayan Kalhara Wickramasinghe

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: W.V Welgama



Signature:

Date: 2021 / 11 / 29

Abstract

This document describes the development of the web-based IT Demand Manager Requests Management System which facilitates the automation of the Demand Manager Request process in the IT department of the Bank of Ceylon in an effective way.

In Bank of Ceylon, when a new business requirement arises, it may need to be developed and implemented as a new IT system or enhance the existing IT system. In such situations, the required business division in the Bank have to submit an 'IT Demand Manager Request' with the relevant approvals from the higher management. This is currently a paper based manual process. Web based 'IT Demand Manager Requests Management System' was developed in order to automate the current manual process.

Core concept of the developed system is to track all the Demand Manager requests in the IT division with the additional features for interpreting the data. System was implemented as a fully web-based application using the PHP scripting language and the MySQL as the database storage where the users will be able to access the application through the Bank's internal network using any user preferred browser. Therefore, with the implementation of a new system, it is expected to increase the efficiency, accuracy of the information and to improve the productivity of the staff members in the IT division.

Developed application includes charts and graphs for the graphical representation of the project statuses, which will enhance the user-friendliness, and the readability of the available information. In addition, application is capable of making suggestions to the user based on the available information where application will suggest the development team which the request should be assigned based on the input data of the Demand Manager request.

Evaluation and testing were carried out by the selected core workers who were involved in the Demand Manager process including the members from the Senior Management, Development Team, QA Team and Business Teams. Then the implemented system was demonstrated to the selected team members and the opinions and viewpoints were gathered through face-to-face interviews. Positive feedbacks received from the users have proved that the system fulfills the identified requirement and it was beneficial over the previously used manual process.

Acknowledgement

First and foremost, I would like to express my grateful thanks to my supervisor Mr. Viraj Welgama for his encouragement, aspiring guidance and support gave me throughout the project. His constant supervision as well as necessary suggestions regarding the project helped me a lot to complete the implementation and conclude this thesis.

I also wish to extend my sincerest thanks to Mrs. K.P Alwis, Chief Manager IT of Bank of Ceylon and staff members of the Quality Assurance department of Bank of Ceylon for their vital support given to me at the planning, development and evaluation stages of the system. Their support paved the way towards a well-built system.

Finally, I would like to thank my parents, my loving wife, and my dear friends for their encouragement, understanding and support given in the successful completion of this project.

Table of Contents

Declaration.....	iii
Abstract.....	iv
Acknowledgement.....	v
List of Figures.....	viii
List of Tables	ix
List of Acronyms	x
Chapter 1 – Introduction	1
1.1 Project Overview.....	1
1.2 Motivation.....	2
1.3 Objectives.....	3
1.4 Background of the study	3
1.5 Scope of the study	4
1.5.1 Boundaries of the project	6
1.6 Feasibility study	7
1.7 Structure of the dissertation.....	8
Chapter 2 – Background	9
2.1 Introduction	9
2.2 Requirement Analysis	9
2.2.1 Functional Requirements	10
2.2.2 Non-Functional Requirements	10
2.3 Review of Similar Systems	11
2.3.1 Jira.....	11
2.3.2 Bugzilla – Open-source tracking system	12
2.3.3 Assembla.....	12
2.3.4 Favro	12
2.4 Related Technologies	13
2.4.1 Front End Development.....	13
2.4.2 Back End Development	14
2.4.3 Database Management	15
2.5 Related Design Strategies.....	15
2.5.1 MVC Architecture	15
Chapter 3 – Design Architecture	16
3.1 Introduction	16

3.2	System Architecture	16
3.3	UML Diagrams	18
3.3.1	Use case Diagrams	18
3.3.2	Use case narrations	21
3.3.3	Sequence Diagrams.....	24
3.3.4	Class Diagram.....	27
Chapter 4	- Implementation Details	28
4.1	Architecture of Implementation	28
4.2	Query Builder Class	30
4.3	Look and Feel of the web application	31
4.3.1	Error Message Model.....	32
4.3.2	Security Controls	33
Chapter 5	- Testing and Evaluation.....	35
5.1	Introduction	35
5.2	System Testing	35
5.2.1	Test Cases	36
5.2.2	Testing Status.....	38
5.3	System Evaluation.....	39
5.4	Analysis of the Results.....	41
Chapter 6	- Conclusion	43
6.1	Future Work	44
List of References	45
Appendix A	- Design Documentation	47
Appendix B	- MIS Reports	51
Appendix C	- Testing Plan.....	55
Appendix D	- User Manual.....	60

List of Figures

Figure 3.1 - System Architecture Diagram	16
Figure 3.2 - CodeIgniter Architecture.....	17
Figure 3.3 - Use case diagram for administrative module	18
Figure 3.4 - Use case diagram for Demand Manager requests module	19
Figure 3.5 - Use case diagram for developer module	20
Figure 3.6 - Sequence diagram for login	24
Figure 3.7 - Sequence diagram for create new user profile	25
Figure 3.8 - Sequence diagram for create new demand manager request	26
Figure 3.9 - Class diagram of the web application	27
Figure 4.1 - CodeIgniter Directory Structure.....	28
Figure 4.2 - Interface - User Maintenance	31
Figure 4.3 - Sample Error Message	32
Figure 5.1 - Screenshot of the new user creation screen.....	38
Figure 5.2 - Evaluation Form.....	40
Figure 5.3 - Graph for category wise rating.....	41
Figure A.1 - Use case diagram for Quality Assurance module	47
Figure A.2 - Use case diagram for UAT module.....	48
Figure B.1 - Report on active user by role.....	51
Figure B.2 - List of users by user status.....	52
Figure B.3 - Report on new user accounts	52
Figure B.4 - Report on demand manager - overall status	53
Figure B.5 - Report on pending DM requests.....	54
Figure D.1 - Screenshot of Login	60
Figure D.2 - Screenshot of user maintenance	61
Figure D.3 - Screenshot of the user maintenance	62
Figure D.4 - Screenshot of the user maintenance	63
Figure D.5 - Screenshot of confirmation message.....	63
Figure D.6 - Screenshot of department maintenance	64
Figure D.7 - Screenshot of user designations	65
Figure D.8 - Screenshot of user functions	66
Figure D.9 - Screenshot of role privileges	67

List of Tables

Table 3.1 - Use case narration for create new user profile	21
Table 3.2 - Use case narration for create new demand manager request.....	22
Table 3.3 - Use case narration for assign team members	23
Table 5.1 - Test cases for login.....	36
Table 5.2 - Test cases for user registration	37
Table 5.3 - Status of test cases	38
Table 5.4 - Evaluation of results	41
Table A.1 - Use case narration for login.....	49
Table A.2 - Use case narration for add new post to KM portal	50
Table C.1 - Test cases for registering department	55
Table C.2 - Test cases for registering new user designation.....	56
Table C.3 - Test cases for registering new user designation.....	57
Table C.4 - Test cases for registering new user role.....	58
Table C.5 - Status of test cases	59

List of Acronyms

AD	Active Directory
BOC	Bank of Ceylon
DM	Demand Manager
KM	Knowledge Management
LDAP	Lightweight Directory Access Protocol
MIS	Management Information System
MVC	Model-View-Control
QA	Quality Assurance
SRS	Software requirements specification
UAT	User Acceptance Testing

Chapter 1 – Introduction

1.1 Project Overview

Bank of Ceylon (BOC) is a state-owned, major commercial bank in Sri Lanka with a network of 628 branches and more than 750 Automated Teller Machines (ATMs) within the country. In Bank of Ceylon, when a new business requirement arises, it may need to be developed and implemented as a new IT system or enhance the existing IT system. In such situations, the required business division in the bank have to submit an ‘IT Demand Manager Request’ with the relevant approvals from the higher management. This is currently a paper based manual process where the Demand Manager who is responsible to coordinate the process will keep a copy of the paper-based requirement. In the IT division, there are sub teams specialized on different business areas and technologies where a Chief Manager will manage each team. For an example, Core Banking Development team will handle all the developments in the main Core Banking System and there are separate teams for Credit Cards, ATMs, and Internet Banking etc.

IT Division will receive more than 150 demand manager requests for a year where the development team will receive around 25 new requests depending on the specialized business areas and technologies. This will create a backlog in each team since most of the Demand Manager requests will require more than 3 months to complete depending on the requirement and the available resources in each team. Developments will always be processed based on the priority of the requirement. New requirements will be available frequently because of the changes in the government rules and regulations, which need to be developed immediately by holding the other ongoing developments.

As per the existing manual process, staff members are maintaining team wise excel sheets to track the status of the assigned Demand Manager requests where it will be visible only for the team members. In order to get the overall status of all the projects in the IT division it will be required to manually combine all the excel sheets which was managed by separate teams. Therefore, management will not have the visibility of the status of the ongoing developments and will not be able to identify the bottlenecks or the delays in the development process. The purpose of the project was to design and develop a web based ‘IT Demand Manager Requests Management System’ in order to automate the current manual process in the Bank. The developed solution will be mainly used by the staff members in the IT division and all the other departments in the Head Office.

1.2 Motivation

Major problems and issues identified by analyzing the existing manual process in the IT Division which was used to handle the Demand Manager requests process are shown below.

- Initial Demand Manager request is submitted as a paper-based form where it contains all the information related to the requirement. Therefore, original copy of the document is given to the assigned team and all the other teams who were involved in the project will keep a copy of the document. As a result, all the teams have to keep a physical storage space for the documents where the need of more storage space will arise as the number of documents increase.
- Maintaining the physical documents without using electronic media will raise security concerns where a misplaced document can easily get into the wrong person who do not have the authority.
- All the teams in the IT division are maintaining separate team wise excel sheets to maintain the status of each Demand Manager request where staff members have to put extra effort on maintain the details. Even though team members are updating a shared file in a shared location it will not allow the multiple users to update the file at the same time.
- In order to get the overall status of all the Demand Manager requests, it will be required to combine all the excel sheets maintained by different teams which will be time consuming and there may be frequent data entry errors.
- Corporate management of the Bank and the Senior Management of the IT division will not have the visibility on the projects progress in order to accelerate the process and avoid bottlenecks, which will delay the development process.
- Communications between the development team and the Quality Assurance team will be handled only through emails and it may delay the testing process and will require extra time and effort for the issue tracking.

In order to overcome above mentioned concerns and difficulties, there is a need of implementing a software solution which will cater the Demand Manager requests process in the Bank. Therefore, with the implementation of a new system, it is expected to increase the efficiency, accuracy of the information and to improve the productivity of the staff members in the IT division.

1.3 Objectives

Main objective of the project is to design and develop a web based 'IT Demand Manager Requests Management System' in order to automate the current manual process.

- Provide overall visibility of the project status to the Senior Management of the IT division and all other stakeholders of the project since all the information will be managed through the developed system.
- Identify the Bottlenecks or delays in the projects in the system by tracking the time taken to complete a specific activity.
- Track deviations on the overall status and completion of the projects.
- Provide an efficient and user-friendly way to track the status of issues and bugs identified during Quality Assurance and User Acceptance Testing processes which were currently communicated through email and tracked by a manual process.
- Availability of real-time reports which makes it easier to take management decisions by the corporate management.

1.4 Background of the study

In every month, the Demand Manager Steering Committee will evaluate the received requests and those requests will be assigned to IT teams based on the requirement. Then the Demand Manager will hand over a copy of the Request to the relevant teams. Afterwards, it will be the responsibility of the assigned team to coordinate with the business division and implement the solution as per the requirement. Usually, the IT division will receive 10 – 15 Demand Manager Requests in every month and most of the requests will consume more than 3 months to develop and implement in the production environment. All the teams are maintaining separate excel files to maintain the status of each request. In order to get the overall status of the all the Demand Manager Requests, it will be required to manually process all the excel sheets which will be maintained by each team in the division. Because of the manual process there may be frequent data entry errors and the employees will have to put extra effort on maintaining the team wise details and combine it whenever it is required to get the overall status of all the requests. In addition, the status of the ongoing requests will not be visible for the Senior Management of the IT division in order to decide whether the given requests can be implemented within the required target period.

Once a request is assigned to a developer, he/she will develop a project plan with tentative dates in order to develop and implement the solution. Even though the plan will be shared among all the stakeholders at the beginning, the progress of the development will not be visible to the Management and the business users. After completing the development, the solution needs to be submitted for the QA approval where a separate QA team in the IT division will do the testing of the developed solution. As per the current processes, QA team will prepare the test cases, communicate the identified bugs to the development team and the developer will reply once the issue is fixed. As per the current process, keeping a track of issues is difficult since a huge number of emails are sent between the QA team and the development teams. Therefore, with the current manual process it will not be easy to track the issues and track the time taken by the developer to fix the issues. After the QA approval, the solution will be given for the business users in order to complete the User Acceptance Testing (UAT) before implementing it in the live environment. Same as the QA process, during the UAT process as well, issues will be communicated through emails where it will be very difficult to track the issues.

1.5 Scope of the study

The developed system consists of six major modules.

1. Administrative Module

a. User Profile Management:

Allow the users with administrative privileges to add/modify and disable users in the system.

b. User Privilege Management:

Define user roles with privileges and assign it to user roles in order to control user access to each function.

c. Manage User Departments:

Add / Modify and disable User Departments in the System.

2. Demand Manager Requests Module

a. Initiate Demand Manager Requests:

Users in the other departments except the IT division will be allowed to create a demand manager request.

- b. Approval of the Higher Management through workflow:
Demand Manager Request needs to be approved by the higher management of the relevant department before submitting to the Demand Manager Committee.
 - c. Building Project Team:
Once a Demand Manager Request is assigned to an IT team, the Manager will be able to select the project team.
 - d. Visibility of the status of the request:
Status on each stage will be visible to all the users involved in the project.
3. Developers Module
- a. Task Management:
Developers will be allowed to breakdown development tasks and define the number of days required for each task. Based on that total development, the time duration will be calculated and will be tracked to identify the delays.
 - b. Knowledge Management Portal:
Portal to share knowledge and new ideas between IT Development Teams.
4. Quality Assurance Module
- a. Test Scenarios:
QA Engineers will be allowed to define test cases related to the project and QA time will be estimated based on the time mentioned for each scenario.
 - b. Issue Tracking:
For each failed test case, QA Engineers will be able to add bugs where it needs to be fixed by the developer and respond to it in order to retest the scenario.
5. User Acceptance Testing Module
- a. Workflow Management:
In Order to get the User Acceptance, developed system needs to be tested by four departments based on different aspects.
 - User Department
 - IS-Audit Department
 - Risk Department
 - Compliance Department

6. Reporting Module

a. Availability of User Reports:

Ability to generate customized reports on each module.

b. Availability Management Information System Reports:

Ability to generate management reports which will be used by the Senior Management to evaluate the projects.

1.5.1 Boundaries of the project

Boundaries of the development which will not be covered with the project development are shown below.

1. Developed system will cover only the project developments which will be handled by the IT division of the bank and all the other projects which were developed by outsourced company will not be handled through this system.
2. Even though the developed system supports team collaboration by allowing the users to add comments and messages, it will not support real-time one to one chat option since use of chat applications is blocked through the intranet due to restrictions in Bank Policy.
3. The developed web-based application will be hosted in a server with Windows Operating System and tightly coupled with windows-based computer architecture.
4. The web application will be hosted on a Virtual Server in the Bank and will be accessible only through the intranet. Therefore, users will not be allowed to access the web application through the public internet.
5. Developed system will not be integrated with any other applications used in the bank (e.g., Core Banking System) since it will not require to capture any customer or financial information related to any of the projects.
6. User password will be stored as a hash value in the database and all the other values will be stored as plain text in the database without using any of the encryption algorithms. For the protection of the data, server will be accessible through the Bank's firewall and specific ports (Port 80) will be accessible through the network.

1.6 Feasibility study

“A feasibility study is an analysis that takes all of a project's relevant factors into account including economic, technical, legal, and scheduling considerations to ascertain the likelihood of completing the project successfully” (Kenton, 2020).

Five areas mentioned below are mainly considered to evaluate the feasibility of implementing an IT Demand Manager requests management system in the Bank.

1. Technical Feasibility

Project was developed as a Web-based application using PHP as an open-source server-side programming language and MySQL was used as the relational database management system which are capable of delivering highly reliable solutions. The Web application was hosted in a VM server which was located in the Bank's internal network so that it will be accessible for the entire branch network. Considering the main technologies used for the development of the web application and the hardware used for the hosting of the application it can be concluded that the project was technically feasible.

2. Economic Feasibility

All the technologies used for the development of the application were open-source technologies which were available for free of charge. VM Server which was used to host the application and the database was provided by the Bank. Therefore, it can be considered that the project was economically feasible since it did not bare any additional costs.

3. Security Feasibility

The Web application is hosted on a Virtual Server in the Bank and accessible only through the intranet. Therefore, users are not allowed to access the Web application through the public internet. Only the port 80 is opened for the internal network in order to allow the access to the Web application. Special permissions like Remote Desktop (RDP) are opened for the specific IP addresses which is used by System Administrators. Considering the above facts, it can be considered that the project is Security wise feasible.

4. Schedule Feasibility

The development of the project was expected to complete within one year time period. Project scope was developed according to the given time period and the project time line was developed accordingly. With the technical expertise and the project deadlines it can be considered that the project is schedule wise feasible.

5. Legal Feasibility

The system is not integrated with any other applications used in the bank (e.g., Core Banking System) since it does not require to capture any customer or financial information related to any of the projects. Therefore, even though the system is accessible only for the Bank staff, customer information will not be exposed to any of the staff members through the system. Therefore, it can be concluded that the project is legally feasible.

1.7 Structure of the dissertation

The remaining chapters of the dissertation are organized as follows;

First chapter mainly focuses on the introduction of the system which has been developed. It briefly mentions the problems of the current system, the solutions and the scope of the developed system. Second chapter describes the background of the system and the literature review about similar systems and technologies. It describes the analysis of the requirement with the different technologies and the design strategies which were used to develop the project. Third chapter illustrates the requirement analysis and the design of the solution and also it shows all the UML diagrams such as Deployment diagram, Component diagram, Class diagram, Sequence diagram, and the database design using the ER diagrams. Evaluation on the system with the sanction on whether the project objectives were satisfied is described in the fourth chapter. It explains the techniques that were used to analyze the data. Also mentions the failures and deficiencies in the final product with the steps taken to improve the functionalities. Suggestions for the future developments and the lessons learned are extensively stated in this chapter. Fifth chapter includes the conclusion of the project which highlights the weaknesses/limitations of the used technique and solutions for them. It describes the benefits of developing the project to the Bank.

Chapter 2 – Background

2.1 Introduction

A detailed analysis on the requirement of the system is provided in this chapter with reviews on existing systems that are similar to the developed system. When comparing the developed system with similar systems, the chapter mainly focuses on the functionality and the technologies used in both systems. Finally, the technologies and design strategies which have used to develop the system are also mentioned.

2.2 Requirement Analysis

Requirement analysis phase includes the overall requirement of the Bank that were agreed at the initial discussions, which were held in a formal manner. These requirements have then been further developed by identifying actors, creating use-cases and identifying how the actors will interact with the system. Class diagram have also been designed, including the identification of class responsibilities and collaboration between classes. All required diagrams are included in the next chapter. According to the information gathered through the discussion had with the staff members of the IT division, the overall summarized requirement as follows;

Main Requirement

- Automation of the existing IT Demand Manager Requests Management process

Sub Requirements (Performance)

- Increase productivity, efficiency, accuracy and data processing speed
- Avoid duplication of data entry
- Facilitating management of the project statuses
- Identify project bottlenecks and delays at early stage
- Reduce unnecessary printing cost
- Manage and track development issues

Basically, software requirements can be identified under two main categories: Functional requirements and Non-Functional requirements. Functional requirements describe the required software performance of the system while Non-Functional requirements discuss about the hardware and software requirements in developing, implementing and maintaining the system.

2.2.1 Functional Requirements

- Users with the administrative privileges should be able to manage other user profiles, user roles and user department details.
- Business users should be able to initiate an IT Demand Manager Request and process the required approval workflow.
- Chief Managers of the IT Division who is working as project leaders should be able to build and manage their project teams and assign resources to the Demand Manager requests.
- Developers and other team members should be able to manage their tasks and use knowledge management portal.
- Quality Assurance team members should be able to build testing scenarios and manage issues/bugs associated with the particular test scenario.
- Senior Management should be able to view the status of the each and every Demand Manager request form and generate management reports based on the customized criteria.

2.2.2 Non-Functional Requirements

- System should have user friendly graphical web pages without using heavy media content which will reduce the performance of the web application.
- In order to provide the safety and security, system shall be able to identify unauthorized users who tries to access the system and deny the access.
- All the unsuccessful attempts made by a user to access the unauthorized function or data shall be recorded on an audit trail.
- Web application should be able to handle around 300 users without affecting its performance.
- System should ensure the 99.9% availability without having any downtimes.
- The system shall sustain data integrity by keeping backups of all updates to the database for every change of the records.

2.3 Review of Similar Systems

2.3.1 Jira

“Jira is one of the most popular tools for agile project management developed by Atlassian. It is a cloud-based and subscription-based issue tracking tool which was designed to handle team coordination in agile software development” (Atlassian, 2020). Main features of the Jira application are shown below.

- Coordinate Agile and Scrum teams
- Manage user project tasks
- Detection and track software bugs

Although Jira covers most of the requirements in the developed system, there are few constraints which needs to be addressed.

- Cost

As a state-owned Bank, has to adhere to the financial regulations of the government where the product should be evaluated through a tender process and because of the per user subscription cost for Jira licenses the annual cost will be high compared to the other solutions in the market.

- Customizability

Workflows and processes in the Jira application cannot be used as it's with the Bank's development environment and may require customization where Jira will not support for client-based customizations. Since all the projects handled by the IT division are not based on Agile development methodology, Jira will not be usable for all the projects.

Therefore, developed web-based solution will address all the constraints which were raised with Jira Implementation and the functionalities will cater the exact requirements of the bank.

2.3.2 Bugzilla – Open-source tracking system

“Bugzilla is an open-source tool for managing the software development process. It allows teams of developers to keep track of outstanding bugs, problems, issues, enhancement and other change requests in their products effectively” (Bugzilla, 2020).

Main features of Bugzilla;

- Handles project bugs and changes in source code
- Create workflows of bugs
- Integrated email capabilities and reports

Even though Bugzilla is an open-source tool, it only supports the issue tracking process which was used by the Quality Assurance team. Therefore, the developed solution will cover all the main features of Bugzilla with the features to manage the Demand Manager Process in the IT Division.

2.3.3 Assembla

“Assembla provides a central location for project owners and developers to manage tasks and code using a single web application. As a code management platform with Agile development tools built in, Assembla integrates with Blame, SVN, and GIT, which makes it easy to manage repositories” (Assembla, 2021).

Assembla does not have the options for issue tracking which will be a main functionality of the developed system. But as a new feature, Assembla has the capability of integrating the SVN and GIT repositories directly through the web application which is not available in the developed solution. Since the Core Banking System of the Bank is running on IBM AS400 most of the IT developments will be developed using RPGLE programming language which is less supportive for the repositories. Most importantly, as a state bank handling with high value transactions, the developers are not allowed to share the full code with the other developers due to confidentiality and security.

2.3.4 Favro

“An all-in-one planning and collaboration app, Favro helps developers, marketers, and executives to use the same app to track the project status. Its features and functionalities are designed to scale across any business. Teams have the flexibility to plan tasks as they want while managers can have visibility to all projects and teams in one place” (Favro, 2020).

In the developed system, Marketing teams are not considered unless they provide a requirement to be developed for their use. Since IT Division of the Bank is not developing any products for third party which require the support of Marketing team to sell products it will not be required for this development. Favro does not have a module for the issue tracking where developed system has a core module for issue tracking.

2.4 Related Technologies

IT Demand Manager Requests Management System was implemented as a fully web-based application where users will be able to access it through the Bank's internal network using any user preferred browser (Internet Explorer, Chrome and Firefox). Different technologies and frameworks used for the development of the application are shown below.

2.4.1 Front End Development

This is the Client side of the developed web application with all the content seen on the user's web-browser when the web application is opened up. Responsiveness and the performance are two key objectives of the front-end where user should be able to view the content without any delay and the content should appear correctly on monitors of all sizes. Front end of the application was developed using three main languages.

1. **HTML (Hyper Text Markup Language)**

Used to design the front-end design of the web pages using markup language. Structure of the web pages were designed using the tags where all the data will be enclosed with the predefined tags.

2. **CSS (Cascading Style Sheets)**

Used to streamline the process of making web pages interactive to users where it allows to apply predefined styles to the web pages.

3. **JavaScript**

Used to enhance the functionality of the web application allowing to handle field validations and manage dynamic content.

Bootstrap 4 framework is used together with the above-mentioned front-end languages since it will provide customizable predefined design templates and classes which saves development time and effort.

Bootstrap 4 Framework

“Bootstrap is a free front-end framework for faster and easier web development and it includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins. Bootstrap is compatible with the latest versions of all modern browsers and platforms” (W3schools, 2021).

2.4.2 Back End Development

This is the Server side of the application where it stores and handles data and focuses on how the web application should work. It was used connect the web application to the database and the other API frame works which will be used for the data processing. Back end of the application was developed using PHP scripting language.

PHP Scripting Language

PHP is a widely-used open-source back-end scripting language which is specifically designed for web development and can be embedded into HTML. For the development of the web application have used PHP 7.4 as the latest version of the PHP released at the time of development.

CodeIgniter framework is used for the back-end development which includes inbuilt set of libraries and other resources that helps to create a web application with high-end capabilities and customizable features.

CodeIgniter Framework

“CodeIgniter is a fully PHP based application development framework which is used to develop dynamic web applications. Its objective is to enable the developer to develop the project faster without writing the code from the scratch. CodeIgniter provides a set of libraries for most of the commonly used tasks like accessing the database and handling user sessions” (CodeIgniter, 2021). CodeIgniter framework was developed based on the model-view-controller (MVC) development pattern which improves the overall dependency of the framework. Also, the platform offers improved security functionalities that are required to represent the overall reliability of the developed web application.

2.4.3 Database Management

MySQL database is used as the database for the web application and the database will be hosted in the same web server, which is used to host the web application.

MySQL Database

“MySQL is a freely available open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database” (Talend, 2020).

2.5 Related Design Strategies

2.5.1 MVC Architecture

Model-view-controller is a software design pattern commonly used for the development of the user interfaces which separates the related application logic into three different interconnected elements. MVC provides a way to develop an application having loosely coupling between input logic, business logic and UI logic.

“CodeIgniter is based on the Model-View-Controller development pattern. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting” (CodeIgniter, 2021).

- **Model**
Represents the data structures, mainly the classes which will be used to search, insert, delete and update the database records.
- **View**
View will be the presentation layer of the web application which displays the information to the users. Therefore, view will be a web page that is visible to the user who is browsing the web application.
- **Controller**
Controller is the intermediate layer between the Model and View. It includes all the data processing and validation algorithms which will be used to process the user requests.

Chapter 3 – Design Architecture

3.1 Introduction

This chapter provides a detailed analysis on the technologies, design methods and the best practices which have been used to design a user-friendly IT Demand Manager Requests management system. It illustrates the requirement analysis and the design of the developed solution using the UML diagrams such as Use Case diagrams, sequence diagrams and class diagrams.

3.2 System Architecture

System architecture of the Web application is a framework that contains relationships and interactions between application components, such as interfaces, middleware and databases. As per the concept of the application user will access the web application using a given URL through a general web browser like Internet Explorer or Chrome.

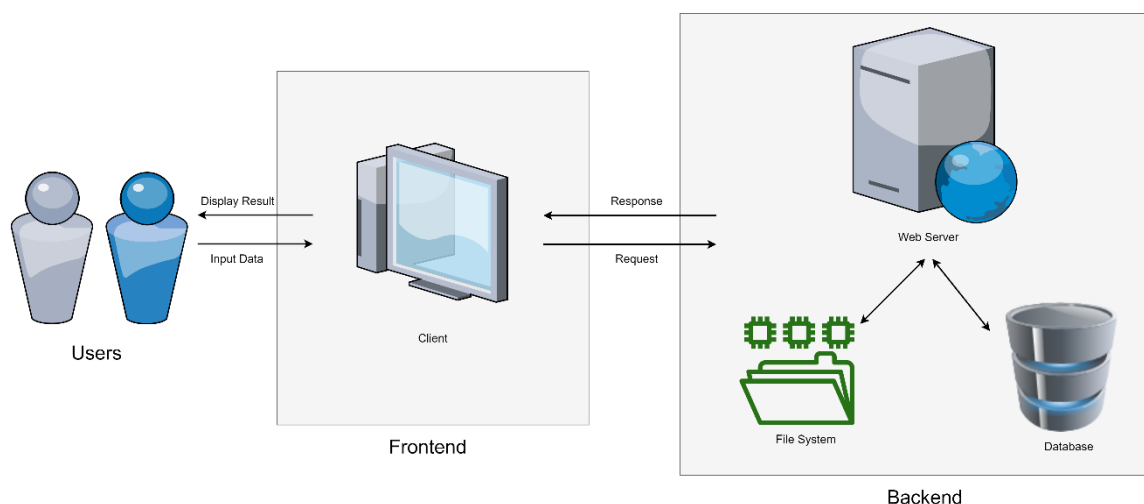


Figure 3.1 - System Architecture Diagram

As shown in figure 3.1, Developed web application consists two segments as front-end and a back-end. The frontend which is also called as client-side is the web pages which is visible to the user and interacts through the web-browser. Main objective of the client-side is to collect input data from the users and display the results. In the developed application frontend was written using HTML, CSS and JavaScript. “When you visit a website by clicking a URL, a request is sent to the server. After the request is done processing, the browser receives the files (HTML, CSS, and JavaScript) and the content of the page and then renders it. If the user decides to go to another page on the website, another request will be made” (Dabbs, 2019). Therefore, step by step process on how the web application works on the client side are shown below.

- Request will be initiated by entering the web link (URL) through the address bar in web browser.
- Then browser will identify the requested site's location from the user request and requests access.
- The server sends the relevant request data to the requested browser.
- Received information translated as a web page and will be displayed to the user by web browser.

The backend, also known as the server-side of the web application is used to store and process the user data where it will not be accessible for the users. The backend processes HTTP requests which essentially “fetch” the data (text, images, files, etc.) called for by the user. In order to increase the performance of the backend processing and enhance the security of the web application CodeIgniter framework has been used with the PHP scripting language.

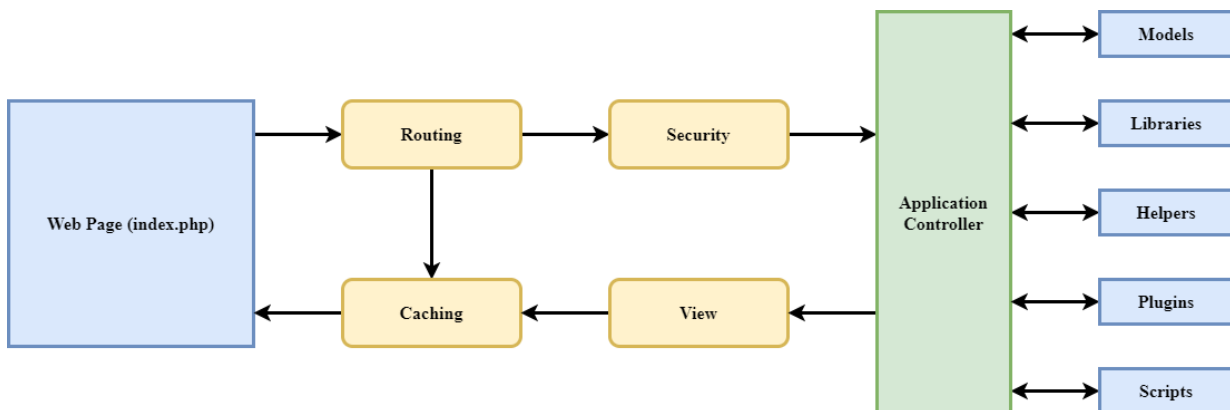


Figure 3.2 - CodeIgniter Architecture

As illustrated on the figure 3.2, main components of the CodeIgniter Architecture are;

- Index.php file will be the default file which will be called initially through the request.
- Based on the request required process will be handled by the Router.
- Information will be passed directly to the browser without processing by the Caching if the file already exists.
- Request data will be processed through the security check before the application controller.
- Based on the request application controller will load the models, libraries and other required files.
- Web page will be sent to the web browser through the View which includes the structure of the web page.

3.3 UML Diagrams

3.3.1 Use case Diagrams

Module wise use case diagrams for the five major modules in developed system.

- Use case diagram for administrative module



Figure 3.3 - Use case diagram for administrative module

As shown in the figure 3.3, Administrator will have access all the functions available in the system including the system configurations and user maintainence. Department manager will have access to the all the functions available for the development user and manager will be able to change the user and department details. Only the users in the IT department of the bank will have access to the administration module and business users will not have access to the administravie functions available in this module.

- Use case diagram for Demand Manager requests module

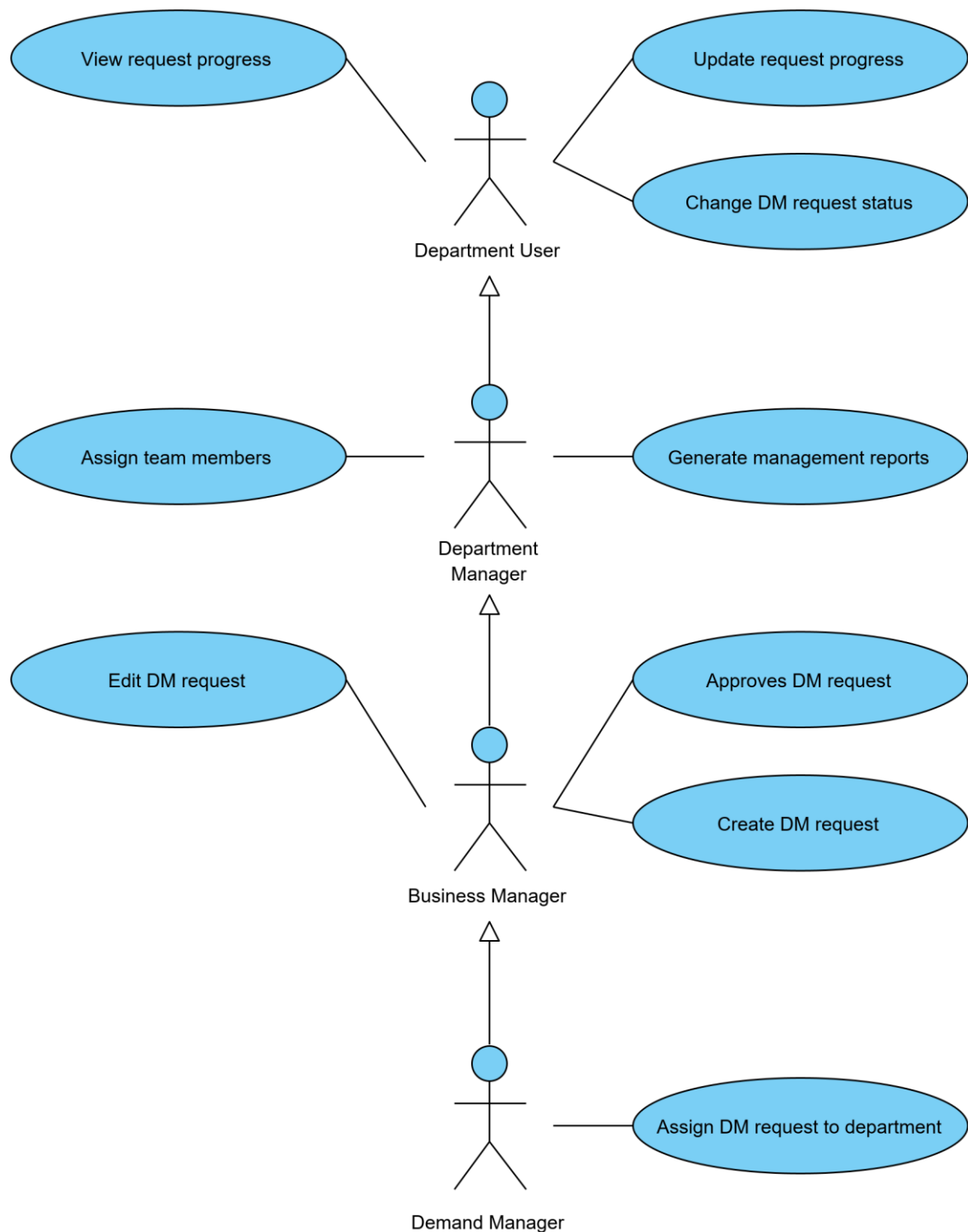


Figure 3.4 - Use case diagram for Demand Manager requests module

As shown in the figure 3.4, Demand Manager will be the user with highest privileges in the demand manager module where user will have access to the all the other functions which were available for the all the other users.

- Use case diagram for developer module

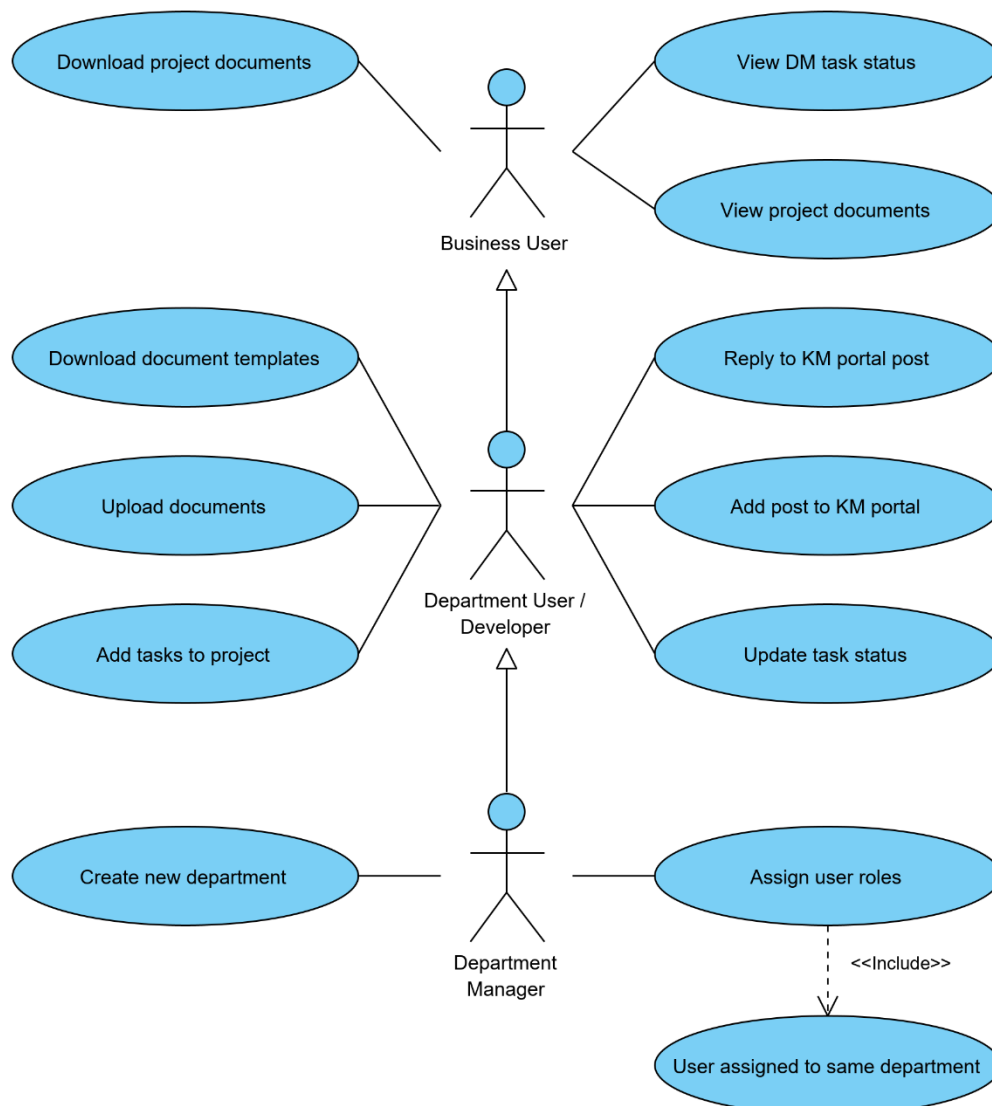


Figure 3.5 - Use case diagram for developer module

As shown in the figure 3.5, developer module will mainly focus on the progress of the project development process where developer will be able to define the tasks which need to be developed in the project. Department Manager and the developers will be able to upload documents related to the project like SRS document and other stakeholders of the project will be allowed to download and view the documents.

3.3.2 Use case narrations

Use case narrations for the main use cases in the above use case diagrams are shown below.

- Use case narration for create new user profile

Use case name	Create new user profile	<u>Use case Type</u> Business Requirements: New user will be required to access the web application
Use case id	02	
Priority	High	
Source	Web Page – User Creation DB Table - User	
Primary Business Actor	Administrator	
Other Participating Actors		
Other Interested Stakeholders		
Description	Create user profile for the new users who will required to access the web application.	
Preconditions	User has the administrative privileges to create new user profiles.	
Trigger	User select the User Management function from the menu	
Typical course of events	1) User selects new user creation option available in the page. 2) Enter the required details of the new user. 3) Click on the save button.	
Alternative course	1) Application will display an error message if user have not entered any value for required field. 2) Application will display an error message if the users provident fund number already exists in the application. 3) Application will display a success message after creating the new user profile.	
Post conditions	Default username will be user’s provident fund number and password will be the same which will be forced to change on the first login attempt.	

Table 3.1 - Use case narration for create new user profile

- Use case narration for create new demand manager request

Use case name	Create new DM request	<u>Use case Type</u> Business Requirements: Submit new DM request to IT division
Use case id	03	
Priority	High	
Source	Web Page – DM Request DB Table - DM_Request	
Primary Business Actor	Business Manager	
Other Participating Actors	Demand Manager	
Other Interested Stakeholders	Business User, Developer	
Description	Business user submit a new Demand Manager request to the IT division.	
Preconditions	User has the required manager privileges to submit a demand manager request.	
Trigger	User selects “Submit New Demand Manager Request” function from the menu items.	
Typical course of events	1) Enter the required details of the demand manager request including demand manager name, requirement, qualitative and quantitative benefits of the development. 2) User have to select the Assistant General Manager and Deputy General Manager who will approve the demand manager request before submitting to IT division.	
Alternative course	1) Application will display an error message if user have not entered any value for required field. 2) Application will display a success message after submitting new DM request.	
Post conditions	Request will be displayed for the selected Assistant General Manager and Deputy General Manager for their approval.	

Table 3.2 - Use case narration for create new demand manager request

- Use case narration for assign team members

Use case name	Assign team members	<u>Use case Type</u> Business Requirements: Assigning team members of the IT division for the DM request.
Use case id	04	
Priority	High	
Source	Web Page – DM Request DB Table – DM_Team	
Primary Business Actor	Department Manager	
Other Participating Actors	Demand Manager, Administrator	
Other Interested Stakeholders	Department User, Business User	
Description	After receiving an approved DM request to the department manager will require to assign the team members who will develop the requirement.	
Preconditions	User has the required manager privileges to assign team members.	
Trigger	User selects a Demand Manager from the list of demand managers and then selects assign team member option.	
Typical course of events	1) Select the staff member name from the list of available staff members. 2) Select the role of the staff member in the DM request. 3) Assign the staff member to the team of members allocated for DM request.	
Alternative course	System displays a success message after assigning the user to the DM request.	
Post conditions	Assigned user will get a notification mentioning that the user was added to the DM request.	

Table 3.3 - Use case narration for assign team members

3.3.3 Sequence Diagrams

Sequence diagrams for the main use cases in the above use case diagrams are shown below.

- Sequence diagram for login

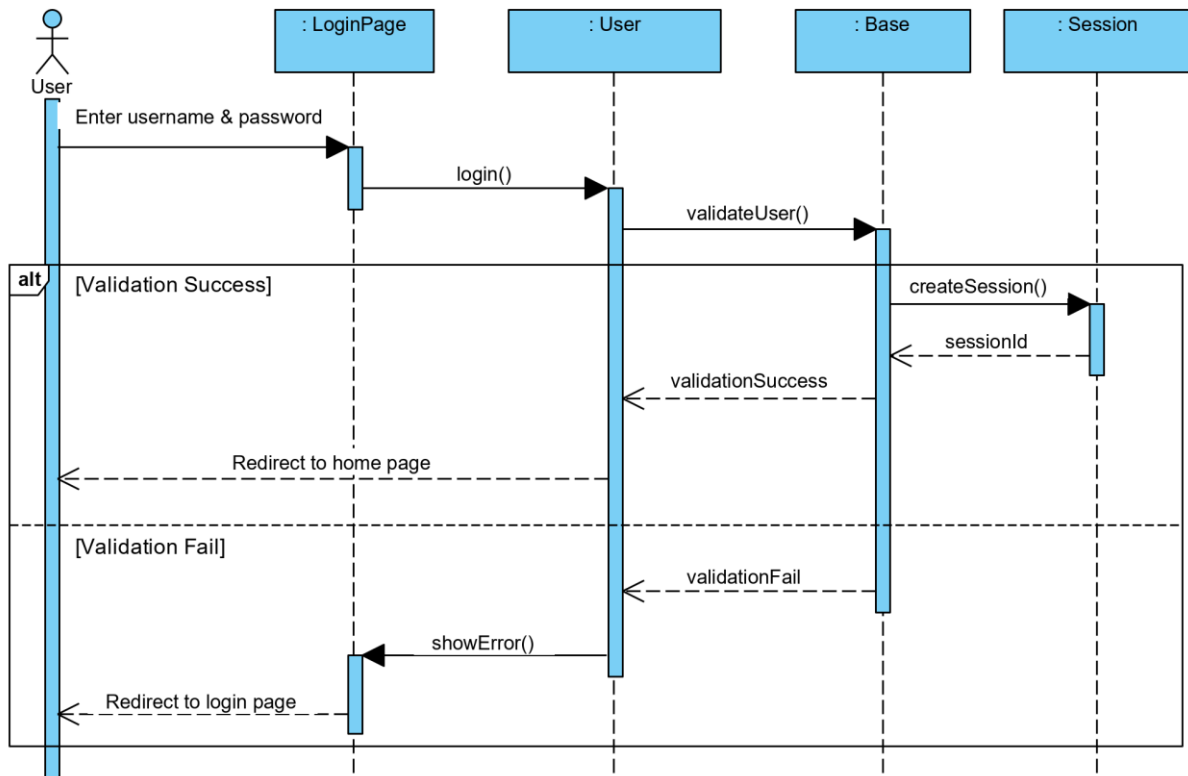


Figure 3.6 - Sequence diagram for login

Sequence diagram of the login function in the system were illustrated in the figure 3.6 which will be the entry point for the system for all the users. Once user enter the username and password it will be validated with the authentication details available in the database and if the validation fails it will display an error message and user will be allowed to enter the username and password again. If the user validation was successful initially it will create the session in the browser and user will be redirected to the home page.

- Sequence diagram for create new user profile

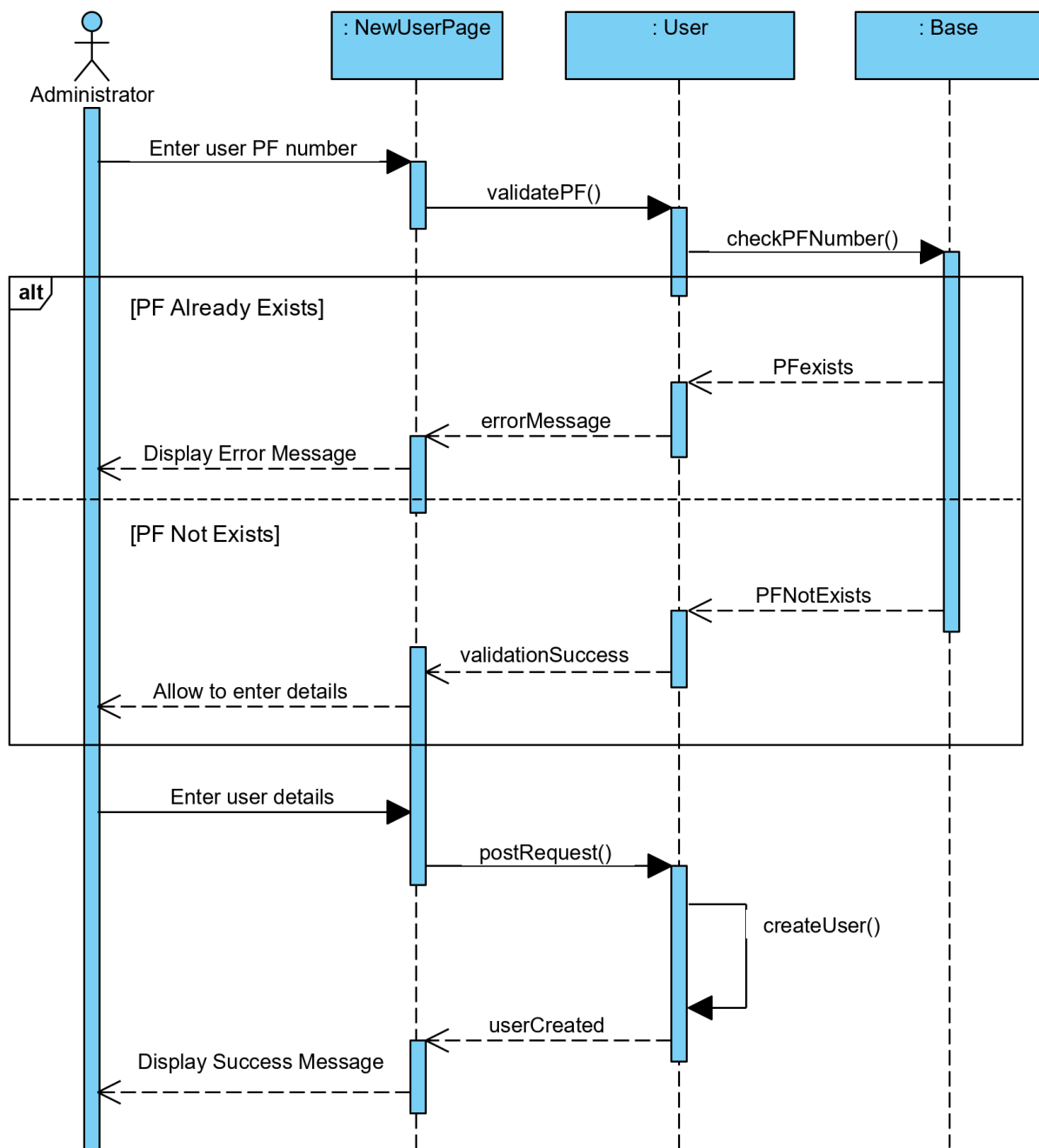


Figure 3.7 - Sequence diagram for create new user profile

Sequence diagram of the process to create a new user profile are shown in the figure 3.7 where users with administrative privileges will be able to create user profiles for the new users. Since the user's provident fund number was used as the unique identification number in the Bank special validation has been added in order to avoid using same PF Number for two users.

- Sequence diagram for create new demand manager request

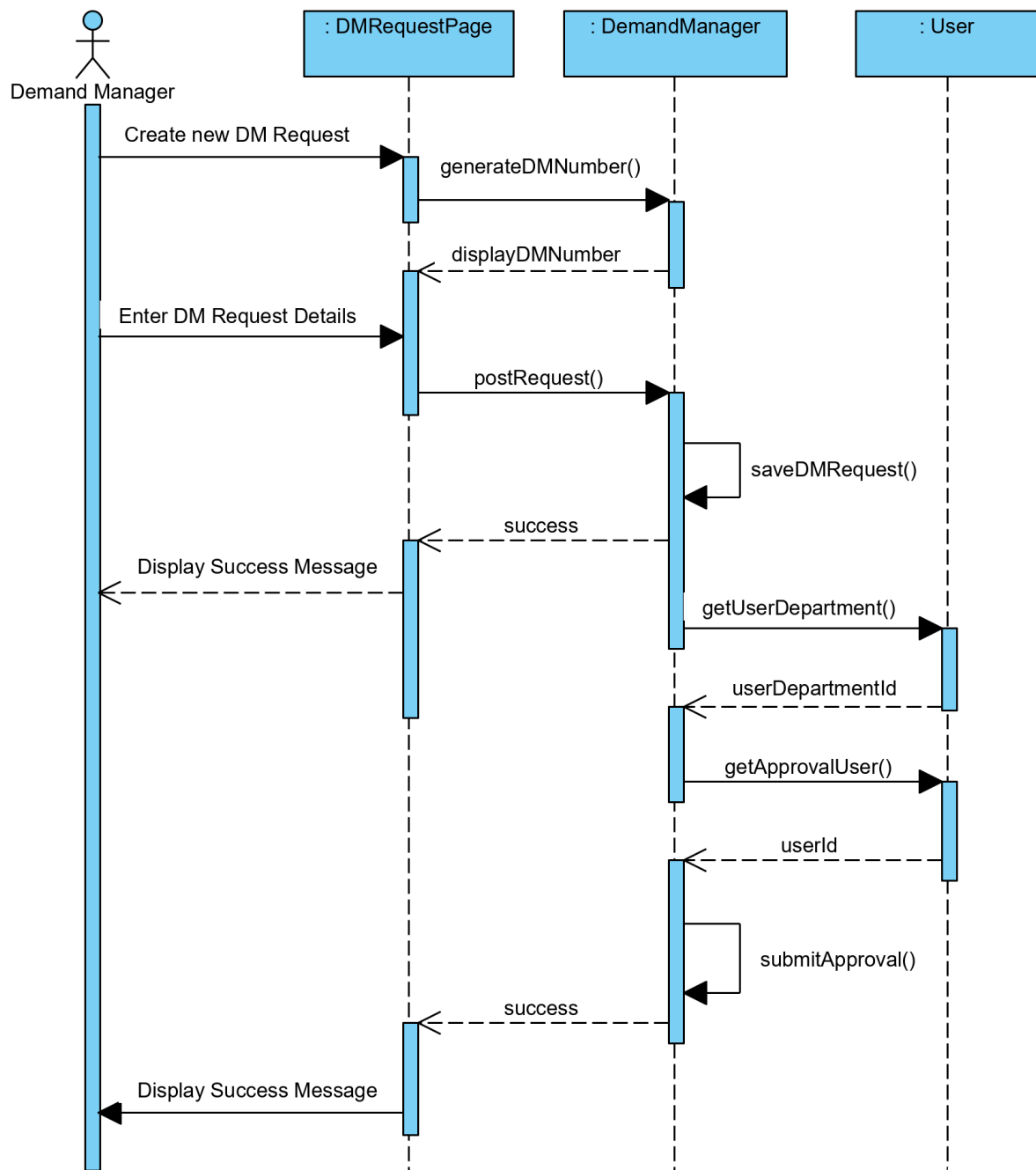


Figure 3.8 - Sequence diagram for create new demand manager request

One of the main functions in the developed application are shown in the figure 3.8 where it shows the sequence diagram for creating new demand manager request. Once user create a demand manager request it needs to be approved by the senior management in order to assign it to the IT Department.

3.3.4 Class Diagram

Class diagram for the developed web application;

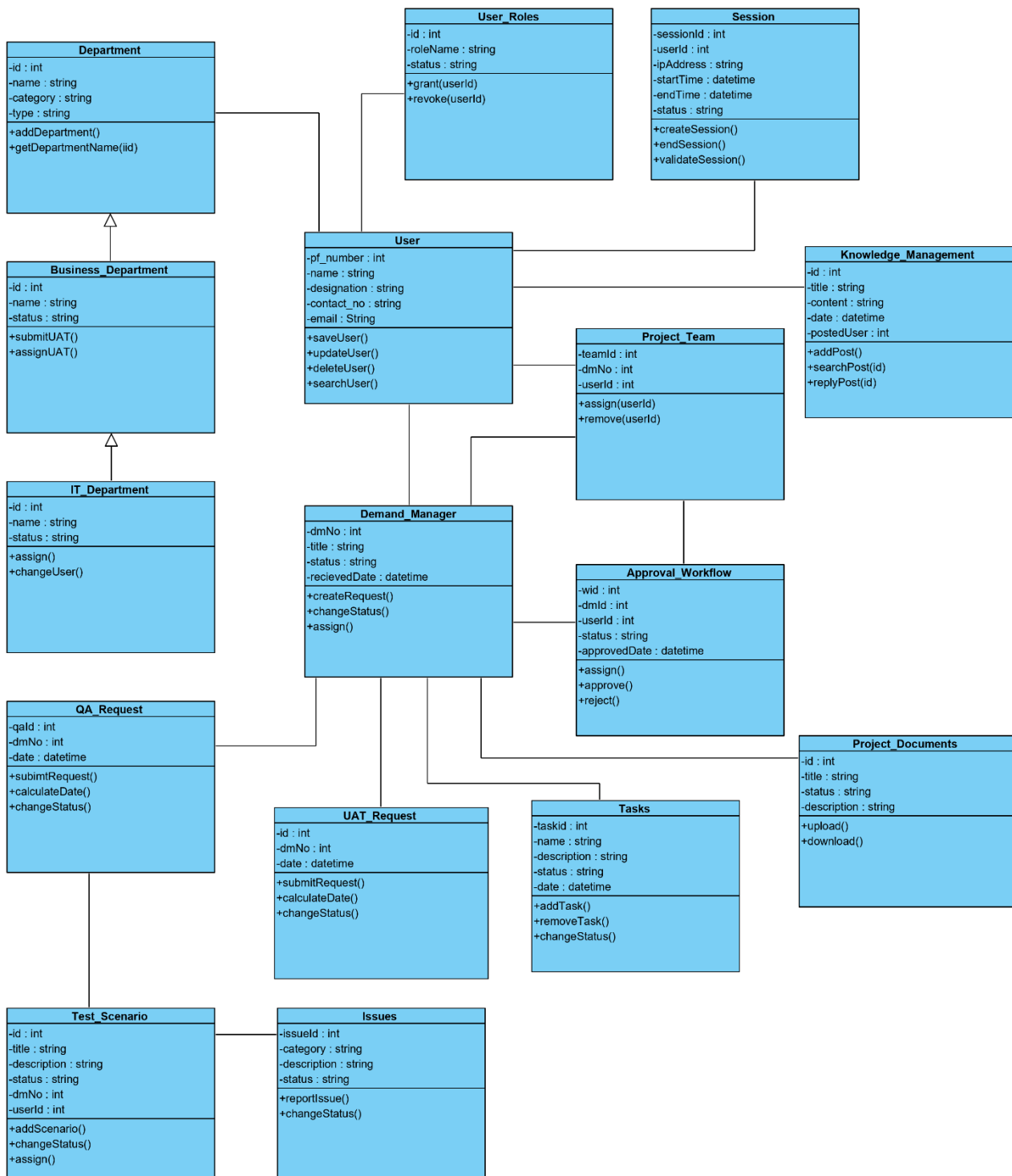


Figure 3.9 - Class diagram of the web application

Class diagram of the developed web application are shown in figure 3.9 which includes all the main classes developed in the system.

Chapter 4 - Implementation Details

In order to implement the IT Demand Manager Requests Management System, many frameworks, different technologies and third-party components have been used to enhance the user friendliness and the comprehension of the system. This section explains the functionalities and implementation details of the main technologies and components used in the system.

Sublime text editor was used for the development of the PHP based web application and MySQL Workbench IDE was used for the design, creation and the maintenance of the MySQL database.

4.1 Architecture of Implementation

As discussed in the first chapter, Model-View-Controller (MVC) architecture was implemented with the use of CodeIgniter PHP based framework. Directory structure of the CodeIgniter framework is shown in the figure 4.1.

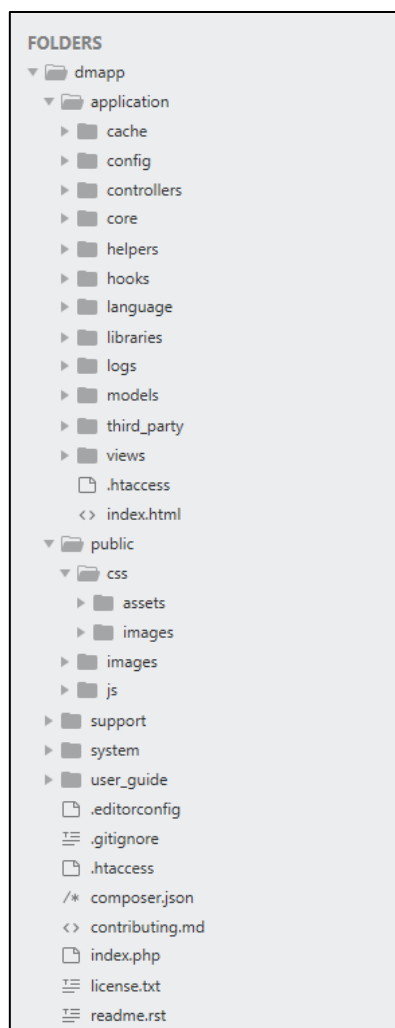


Figure 4.1 - CodeIgniter Directory Structure

Application directory includes all the code of the developed web application including all the PHP files related to MVC architecture. Application folder includes few other folders which are described below.

- Cache

This folder contains all the cached pages which was used to enhance the overall speed of accessing the pages through the application.

- Config

This folder contains all the configuration files used in the application. This folder includes all the important configuration files like config.php which includes the main configuration parameters. Using the database.php file application was connected to the MySQL database and routes.php file which includes all the routing rules used in the application. Sample routing rules used in the application are shown below.

```
$route['auth']          = 'core/Auth';  
$route['auth/login']    = 'core/Auth/login';  
$route['auth/validate'] = 'core/Auth/validate_user';
```

- Controllers

One of the main directories in the MVC architecture which includes all the controllers of the application.

- Core

This folder includes the base classes of the application.

- Helpers

This folder includes all the helper classes of the application

- Hooks

This folder includes the files which are used to modify the default files and methods used by the CodeIgniter framework.

- Language

This folder includes the files related to language.

- Libraries

This folder includes the files of the libraries developed for the application.

- Logs

This folder includes the files related to the log of the system.

- **Models**

This folder includes the model classes designed to work with the data in the MySQL database.

- **Third_party**

This folder includes all the third-party plugins used in the application.

Views

This folder includes all the HTML files which used to display the pages.

System folder contains all the CodeIgniter core files which includes codes, libraries, helpers and other related system files. These libraries and helpers will be loaded in the application and will be used without implementing it from the scratch.

Public folder contains all the CSS files, template images and the JavaScript files used for the application. Since the application has used bootstrap toolkit, all the required CSS files and JavaScript files were located in this file with the developed custom files which will be required for the additional features and styles.

4.2 Query Builder Class

CodeIgniter framework provides an inbuilt database query builder class which allows the developer to handle database queries with minimal scripting. Most of the time it will require one or two lines of code to access the data in the database without having multiple database class to handle data. Database was accessed through the model class where it was called through the controller class. Sample code of the query builder class used to retrieve the user details are shown below.

```
public function get_user_by_username($username){  
  
    $this->db->select('*');  
    $this->db->from('users');  
    $this->db->where('user_name',$username);  
  
    $query = $this->db->get();  
  
    if ( $query->num_rows() > 0 )  
    {  
        $row = $query->row_array();  
        return $row;  
    }else{ return [];}  
}
```


4.3 Look and Feel of the web application

In order to maintain the look and feel of the web pages throughout the web application, common custom build template has been used for the application without using a freely available web template. Sample web page used in the application is shown in figure 4.2.

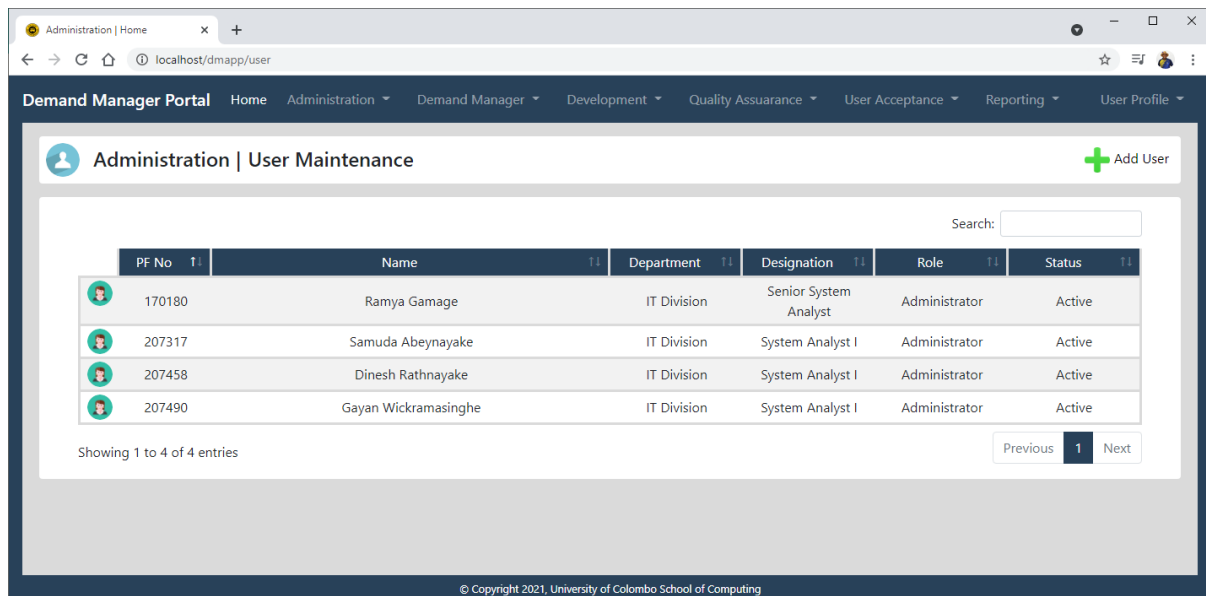


Figure 4.2 - Interface - User Maintenance

Web page consists of three main sections as header, footer and the content section. Since the header and footer sections were common for the each and every page in the web application it was created as two separate PHP files as header.php and footer.php. Only the content section was developed as a separate page and was loaded with the header and footer as shown in the below code segment.

```
$data['page_title'] = "Administration | Home";  
$data['main_section'] = "Administration";  
$data['sub_section'] = "User Maintenance";  
$data['icon_image'] = "icon-user";  
$data['display_add_new_btn'] = true;  
$data['add_btn_title'] = 'Add User';  
$this->load->view('template/header', $data);  
$this->load->view('admin/user', $data);  
$this->load->view('template/footer', $data);
```

Header page includes the variables for the page title and other sub section headings which needs to be passed before loading the page.

4.3.1 Error Message Model

Every error message displayed in the application includes a unique error id for the easy reference of the users. Sample of the error messages displayed in the application is shown in figure 4.3.

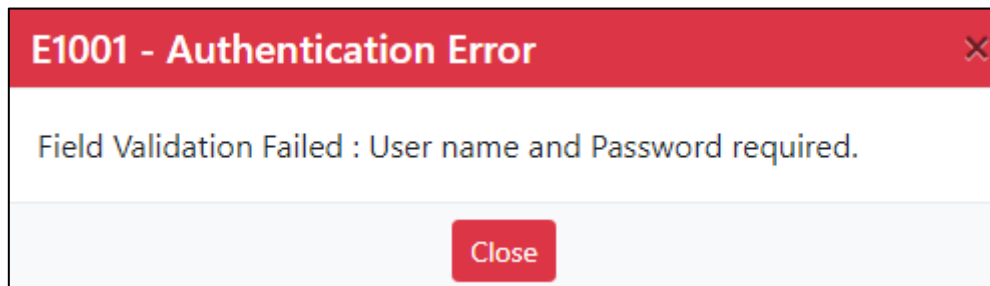


Figure 4.3 - Sample Error Message

Without hardcoding the error message text in the controller class, separate model has been used to define the error messages as a multidimensional array. Error message will be displayed by passing the error category and the error code. Sample code used to define the error message model are shown below.

```
class MessageModel extends CI_Model{

    protected $message_log = array(
        'auth' => array(

            'E1001' => array(
                'type'  => 'Error',
                'title'  => 'Authentication Error',
                'code'   => 'E1001',
                'message' => "Field Validation Failed : User name and Password required.",
                'status' => 0,),

            'E1002' => array(
                'type'  => 'Error',
                'title'  => 'Authentication Error',
                'code'   => 'E1002',
                'message' => "Login Failed: Invalid Credentials.",
                'status' => 0,),

            'E1003' => array(
                'type'  => 'Error',
                'title'  => 'Authentication Error',
                'code'   => 'E1003',
                'message'  => "Login Failed: User access has been disabled.",
                'status' => 0,),
        ),); }
```

Sample code of the controller class which was used to call and retrieve the error message details are shown below.

```
public function base_show_error_message($errorcategory, $errorcode){  
  
    $data['alert_message'] = true;  
    $data['message_color'] = 'danger';  
    $errorMessage = $this->messagemodel->get_message($errorcategory, $errorcode);  
  
    $data['message_type'] = $errorMessage['type'];  
    $data['message_code'] = $errorMessage['code'];  
    $data['message_title'] = $errorMessage['title'];  
    $data['message_text_line1'] = $errorMessage['message'];  
    $data['message_text_line2'] = "\n"; //Will be used only if required  
  
    return $data;  
}
```

4.3.2 Security Controls

In order to prevent unauthorized access and other security threats to the developed Demand Manager Requests system, it was essential to use security controls for the development of web-based application.

- Password Hashing

For security reasons, it was required to store passwords in hashed format without storing it as plain text in the database. It was decided to use password_hash() function available in the PHP without using the MD5 or SHA hashing methods which uses a weak algorithm and non-salted hashes are vulnerable to the rainbow tables and dictionary attacks. The hash result generated from the password_hash is more secure due to the strong hashing algorithm and also it will add a random salt to prevent rainbow tables and dictionary attacks.

- Audit Logs

Each and every user action handle through the developed web application will be logged in the application for the auditing purposes of user actions. As per the standard audit requirement of the Bank, below mentioned details will be captured for the each and every user action performed in the application.

- User Name
- Host Name
- IP Address
- OS Name
- Browser Name
- Browser Version
- Action Module
- Action Title
- Action Description
- Action Status

Chapter 5 – Testing and Evaluation

5.1 Introduction

Testing is a process used to assure the quality of the developed application and the functionality meets the customers requirement. Also testing will be used to validate the application for the issues which will affect the performance and quality of the developed application. This chapter includes the complete testing plan for the developed web application in detail and the evaluation on the system with the confirmation on whether the project objectives were satisfied by the development.

5.2 System Testing

For the development and implementation of high-quality software application it is important have a proper testing plan in order to identify the issues and fix it before implementing the application in production environment. It was decided to do the unit testing during the development phase where all the major components have been separated from the other components and have tested for the inputs and outputs individually.

During the testing phase of waterfall model, it was decided to use the integration testing which allows to combine individual components and test it as a group. Main objective of the integration testing was to test the interfaces between the units and modules after combining it together.

Therefore, as per the test plan all the individual components and modules will be tested initially and once unit testing was completed integration testing will be started where it will be tested by integrating one by one together until all the modules are integrated.

5.2.1 Test Cases

Test cases used for testing of the developed Demand Manager Requests Management System are shown below.

- Test Cases for Login

ID	Activity	Test Case Steps	Expected Result
1.1	Login with valid user account	Visit the localhost/dmapp/auth	Should be able to view the login page
		Enter the username and password	Should be able to enter both fields
		Click on Login button	User login should be successful
1.2	Login with valid username and empty password	Visit the localhost/dmapp/auth	Should be able to view the login page
		Enter the username only	Should be able to enter fields
		Click on Login button	Error Message should be displayed.
1.3	Login with empty username and password	Visit the localhost/dmapp/auth	Should be able to view the login page
		Enter the password only	Should be able to enter fields
		Click on Login button	Error Message should be displayed.
1.4	Login with valid empty username and empty password	Visit the localhost/dmapp/auth	Should be able to view the login page
		Click on Login button	Error Message should be displayed.
1.5	Login with valid username and password with status as “Disabled”	Visit the localhost/dmapp/auth	Should be able to view the login page
		Enter the password only	Should be able to enter fields
		Click on Login button	Error Message should be displayed.
1.6	Login with valid username and password with status as “Locked”	Visit the localhost/dmapp/auth	Should be able to view the login page
		Enter the password only	Should be able to enter fields
		Click on Login button	Error Message should be displayed.

Table 5.1 - Test cases for login

- Test Cases for User Registration

ID	Activity	Test Case Steps	Expected Result
2.1	Create new user screen	Click Add New User option	Create New User Profile screen should open.
2.2	Check Validations	PF Number Field <ul style="list-style-type: none"> - Cannot be blank - Characters not allowed - Already registered PF should not allow Name Field <ul style="list-style-type: none"> - Cannot be blank Grade Field <ul style="list-style-type: none"> - Cannot be blank User Role Field <ul style="list-style-type: none"> - Cannot be blank Mobile Number Field <ul style="list-style-type: none"> - Cannot be blank - Characters not allowed Email Address <ul style="list-style-type: none"> - Should be valid email format - Cannot be blank Extension Number <ul style="list-style-type: none"> - Cannot be blank User Department <ul style="list-style-type: none"> - Cannot be blank 	Should display an error message if validation fails.
2.3	Create New User Profile– Success	Enter valid values for all the fields Click on create user button	Request should be successfully submitted.
2.4	Create New User Profile– Failed	Enter valid values for all the fields avoiding any mandatory field Click on create user button	Error Message should be displayed
2.5	Create new user profile with already registered PF number	Enter a PF number which is already registered. Enter all the necessary details accurately	Should display an error message

Table 5.2 - Test cases for user registration

5.2.2 Testing Status

Test ID	Description	Pass / Fail
1.1	Login with valid user account	Pass
1.2	Login with valid username and empty password	Pass
1.3	Login with empty username and password	Pass
1.4	Login with valid empty username and empty password	Pass
1.5	Login with valid username and password with status Disabled	Pass
1.6	Login with valid username and password with status Locked	Pass
2.1	Create new user screen	Pass
2.2	Check Validations	Pass
2.3	Create New User Profile– Success	Pass
2.4	Create New User Profile– Failed	Pass
2.5	Create new user profile with already registered PF number	Pass

Table 5.3 - Status of test cases

Detail test plan is mentioned in the Appendix C with all the other test cases and test results.

Register New User

PF Number * The PF Number field is required.

User Name *

First Name *

Last Name *

Name with Initials * The Name with initials field is required.

Gender * ☐ Male ☒ Female

Department * The User Department field is required.

Grade * The User Grade field is required.

User Role * The User Role field is required.

Email Address * The Email Address field is required.

Mobile Number * The Mobile Number field is required.

Extension Number *

IP Phone Number

Direct Number

IP Address

Status Code *

Notes

Save

Figure 5.1 - Screenshot of the new user creation screen

Figure 5.1 shows the screenshot of the new user profile creation screen which will be used to register a new user. As per the test id 2.2 in the table 5.3 test cases, once user click on the save button application will display all the error messages after validating the user input fields. Without showing the error messages one by one all the errors will be shown in the screen which helps the user to correct all the fields at once.

5.3 System Evaluation

In the software industry there are different types of methods and techniques available to evaluate the system. For the evaluation of the developed IT Demand Manager Request Management System, it was decided to process criteria-based assessment which is a quantitative assessment for the software system based on the terms like look and feel of the system and functionality. In order to collect the user feedback, questioner has been prepared and given to the users who have used the system for the testing purposes.

UAT environment was configured using a Virtual Server Machine in the Bank's network and shared the URL with the users who will access the system for the testing. Only the port 80 on the server was opened for the IT division located at the 6th floor of the Bank of Ceylon head office building and few other business users were granted access through specific IP addresses. After allowing selected users from both IT and business teams to use the system for a time period of one week in order to get familiar with the system, evaluation form is given to 15 staff members.

Questioner was designed with multiple questions regarding the developed web application and allows users to provide rating for each question. Because of the numerical rating used for each question it will be easy to analyze data and evaluate the user rating. Questions are scaled from Strongly Disagree to Strongly Agree which are having five levels as 1 to 5.

Evaluation Form of IT Demand Manager Requests Management System for IT Department - Bank of Ceylon								
Level of experience with regard to this subject					Years			
Are you a staff member of the IT team ?					<div style="display: flex; justify-content: space-around;"> Yes No </div>			
				Very dissatisfied	Not satisfied	Neutral	Satisfied	Very satisfied
				1	2	3	4	5
1	Attractiveness of the User Interface Design (Colors etc.)							
2	Smooth navigation through the web pages							
3	Error messages are meaningful and can be understood							
4	Learning the workflow of the system is easy							
5	Clarity of the main menu functions by it's name							
6	The system is more efficient than the manual process							
7	Usefulness of the real time reports generated by the system							
8	System is capable of providing the overall project status to stakeholders							
9	Speed and Response time of the system							
10	User Friendliness of the product							
11	Available functions in the system							
12	Quality of the content provided in the application							
13	Ability to perform intended tasks successfully							
14	Speed and Response time of the web application							
15	Use of terms throughout the system							
16	Security features provided with the login							
17	Controls used to prevent human errors while entering data							
18	Your information which will be visible for other users							
19	Visibility of the project details to the other stakeholders in the project							
20	Overall protection of your data available in the application							
Other Comments <div style="border: 1px solid black; height: 60px; margin-top: 5px;"></div>								

Figure 5.2 - Evaluation Form

Sample evaluation form which was given to the users are shown in the figure 5.2. Printed copies of this evaluation form were given to the users who was involved in the testing process of the web application.

5.4 Analysis of the Results

	Look & Feel					Usefulness					Functionality					Security				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
User 01	4	4	5	4	5	4	4	5	4	5	4	4	4	5	5	5	5	4	5	4
User 02	5	5	5	5	5	4	4	4	5	5	4	4	5	5	5	5	5	5	4	5
User 03	4	5	5	4	4	5	5	5	4	4	5	4	4	4	5	5	4	4	5	4
User 04	4	5	5	4	4	5	5	5	5	5	5	4	4	5	4	5	4	5	4	4
User 05	4	4	5	4	5	5	5	5	4	5	5	5	4	4	3	5	5	4	5	4
User 06	5	4	5	4	4	5	5	5	5	4	5	4	5	4	4	5	4	4	5	5
User 07	4	5	5	4	5	5	4	5	5	4	5	5	5	5	5	5	5	4	4	5
User 08	5	5	5	4	5	4	5	5	4	5	5	5	5	5	5	5	5	4	5	4
User 09	5	4	5	4	4	5	5	5	5	4	5	4	5	4	5	5	4	4	5	5
User 10	5	5	5	5	4	5	5	5	5	4	5	4	4	4	5	4	4	5	5	4
User 11	4	5	5	4	4	5	5	5	4	5	5	4	4	4	5	5	4	4	5	5
User 12	5	5	5	4	5	4	4	5	4	5	4	4	4	5	5	5	5	4	5	4
User 13	4	4	5	5	5	4	4	4	5	5	4	5	5	5	5	5	4	5	4	5
User 14	5	4	5	4	4	5	5	5	4	4	5	4	4	4	5	5	4	4	5	4
User 15	5	4	5	5	4	5	5	5	5	4	5	4	4	4	4	5	5	4	4	5
	4.53	4.53	5.00	4.27	4.47	4.67	4.67	4.87	4.53	4.53	4.73	4.27	4.40	4.47	4.67	4.93	4.47	4.27	4.67	4.47
	4.56					4.65					4.51					4.56				

Table 5.4 - Evaluation of results

After collecting the filled evaluation forms from the users, data grid has been prepared as shown with all the results given by each user for each question in order to analyze the data. The results were used to compute the mean score for the all-individual questions and individual categories as shown in the table 5.4.

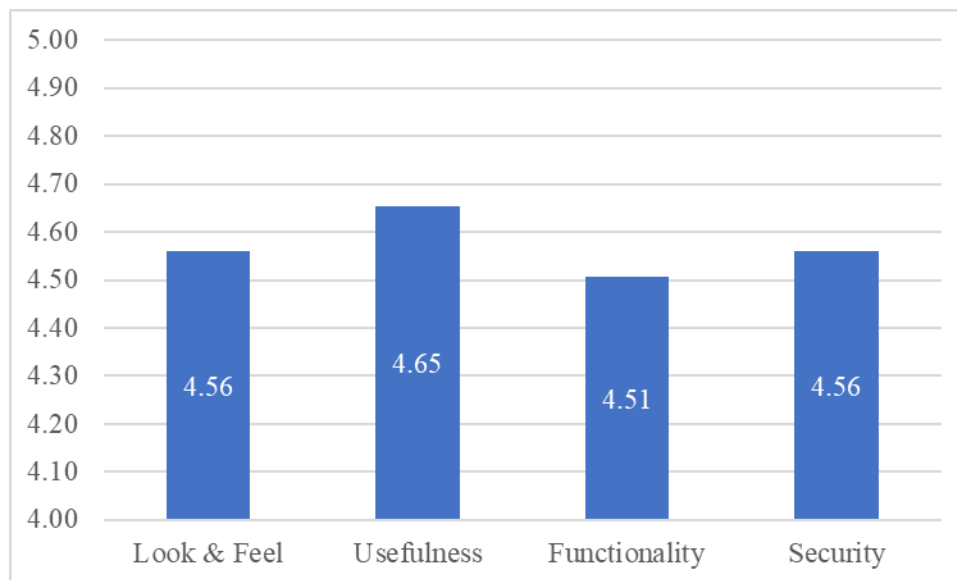


Figure 5.3 - Graph for category wise rating

Graphical representation of the category wise mean value which was calculated based on the user evaluation are shown in the figure 5.3. Key highlights which were identified after analyzing the evaluation results are described below.

- Look and feel

For the look and feel category, it has achieved 4.56 as the mean score where it gives an indication that users are satisfied with the colors used, layout of the web pages and design of the overall web application. The 3rd question which has received a rating of 5 from all the users highlights that the error messages are meaningful and clear confirming that the users are able to learn and understand the system easily

- Usefulness

For the usefulness category it has received a rating of 4.65 as the mean score which was the highest among 4 categories. Therefore, it can be concluded that the system is useful and convenient for all the users who has involved in the UAT process.

- Functionality

Since Functionality category has received a rating of a good 4.51 score it can be concluded that the system is able to perform the intended tasks successfully. The users are satisfied with the quality of content and the available functions of the system. It seems that the users are happy with the speed and response time of the system as well.

- Security

As security wise, it has achieved a score of 4.56 which shows that the users are content with the security features of the system. It can be concluded that the overall protection of the data and information stored in the system is very well secured.

Therefore, it can be concluded that the system has achieved the customer satisfaction by fulfilling the objectives of the development and system can be implemented in the production environment.

Chapter 6 – Conclusion

As stated in the first chapter, main objective of the developed project was to implement a web based ‘IT Demand Manager Requests Management System’ in order to automate the current manual paper-based process in the IT department of the Bank of Ceylon. Therefore, when considering the successfulness of the system it mainly depends on two aspects:

1. Whether the developed solution is capable to replace the existing manual process and continue the process on a digital platform.
2. Whether the transformation of the existing process and users into the web-based system is handled successfully.

Based on the detailed analysis described in the previous chapter, it can be concluded that the developed solution was capable to replace the existing manual process since all the users who have been involved in the testing of the application has confirmed that the functionality of the developed application was sufficient to continue the existing process in digital platform.

When developing a software solution to automate a manual paper-based process to web-based system, the biggest difficulty is the transformation to the digital process because most of the users may not have the required technical knowledge to handle an online system. But the advantage of the developed web-based Demand Manager Requests Management System was that it will be mostly used by the staff members of the IT department in the Bank who are having technical skills and knowledge to work with an online system. Therefore, it can be concluded that the transformation can be successfully implemented once the system is deployed in the production environment.

Therefore, it can be determined that the development and implementation of the IT Demand Manager Request Management System was successful and have achieved the objective within given tight time period.

6.1 Future Work

Even though the application contains all the required functionalities, there are few enhancements and new functionalities which will be developed in future after deploying the application in the live environment.

Developed application will be using its own authentication username and password for the users to access the system. Since it's very difficult for the users to maintain different credentials for different system, in future application will be connected to the Banks LDAP server allowing the user to login to the application using Active Directory (AD) credentials. With the LDAP integration it will be required to create a separate function as self-registration process where users will be allowed to get registered with the system by their own.

As the first phase application and database will be hosted in VM server in the production environment and as later server will be replicated to the DR server. Realtime data replication mechanism will be implemented to transfer data from the live database to DR database.

List of References

- Assembla, 2021. *Project Management in the Cloud*. [Online]
Available at: <https://www.assembla.com/projects> [Accessed 1 February 2021].
- Atlassian, 2020. *Jira Overview / Projects and Hosting*. [Online]
Available at: <https://www.atlassian.com/software/jira/guides/getting-started/overview>
[Accessed 1 February 2021].
- BOC, 2021. *About Us*. [Online] Available at:
https://www.boc.lk/index.php?route=information/information&information_id=4
[Accessed 1 February 2021].
- Bugzilla, 2020. *About - Bugzilla*. [Online] Available at: <https://www.bugzilla.org/about>
[Accessed 2 February 2021].
- Burke, R., 2010. *Fundamentals of Project Management*. 2nd ed. United Kingdom: Burke Publishing.
- Codeigniter, 2021. *Model-View-Controller - CodeIgniter 3.1.11 documentation*. [Online]
Available at: <https://codeigniter.com/userguide3/overview/mvc.html>
[Accessed 1 February 2021].
- Codeigniter, 2021. *Welcome to CodeIgniter4 - CodeIgniter 4.1.1 documentation*. [Online]
Available at: https://www.w3schools.com/bootstrap/bootstrap_get_started.asp
[Accessed 1 February 2021].
- Dabbs, M., 2019. *The Fundamentals of Web Application Architecture*. [Online]
Available at: <https://reinvently.com/blog/fundamentals-web-application-architecture>
[Accessed 06 March 2021].
- Favro, 2020. *The Most Agile Planning and Collaboration Tool For Organizational Flow*. [Online] Available at: <https://www.favro.com/product> [Accessed 04 February 2021].
- GeeksforGeeks, 2021. *Frontend vs Backend*. [Online]
Available at: <https://www.geeksforgeeks.org/frontend-vs-backend>
[Accessed 4 February 2021].
- Intellectsoft, 2021. *Web Application Architecture: Components*. [Online]
Available at: <https://www.intellectsoft.net/blog/web-application-architecture>
[Accessed 1 March 2021].

- Javatpoint, 2021. *CodeIgniter Architecture*. [Online]
Available at: <https://www.javatpoint.com/codeigniter-architecture> [Accessed 7 March 2021].
- Kenton, W., 2020. *Feasibility Study*. *Investopedia*. [Online]
Available at: <https://www.investopedia.com/terms/f/feasibility-study.asp>
[Accessed 2 February 2021].
- Kerzner, H., 2017. *Project Management*. 12th ed. United Kingdom: Wiley.
- Talend, 2020. *What is MySQL? Everything You Need to Know*. [Online]
Available at: <https://www.talend.com/resources/what-is-mysql> [Accessed 5 February 2021].
- W3schools, 2021. *Bootstrap Get Started*. [Online]
Available at: https://www.w3schools.com/bootstrap/bootstrap_get_started.asp
[Accessed 5 February 2021].
- Workzone, 2020. *18 Best JIRA Alternatives for Agile Project Management In 2019*. [Online]
Available at: <https://www.workzone.com/blog/jira-alternatives>
[Accessed 5 February 2021].

Appendix A – Design Documentation

Use Case Diagrams

- Use case diagram for Quality Assurance module

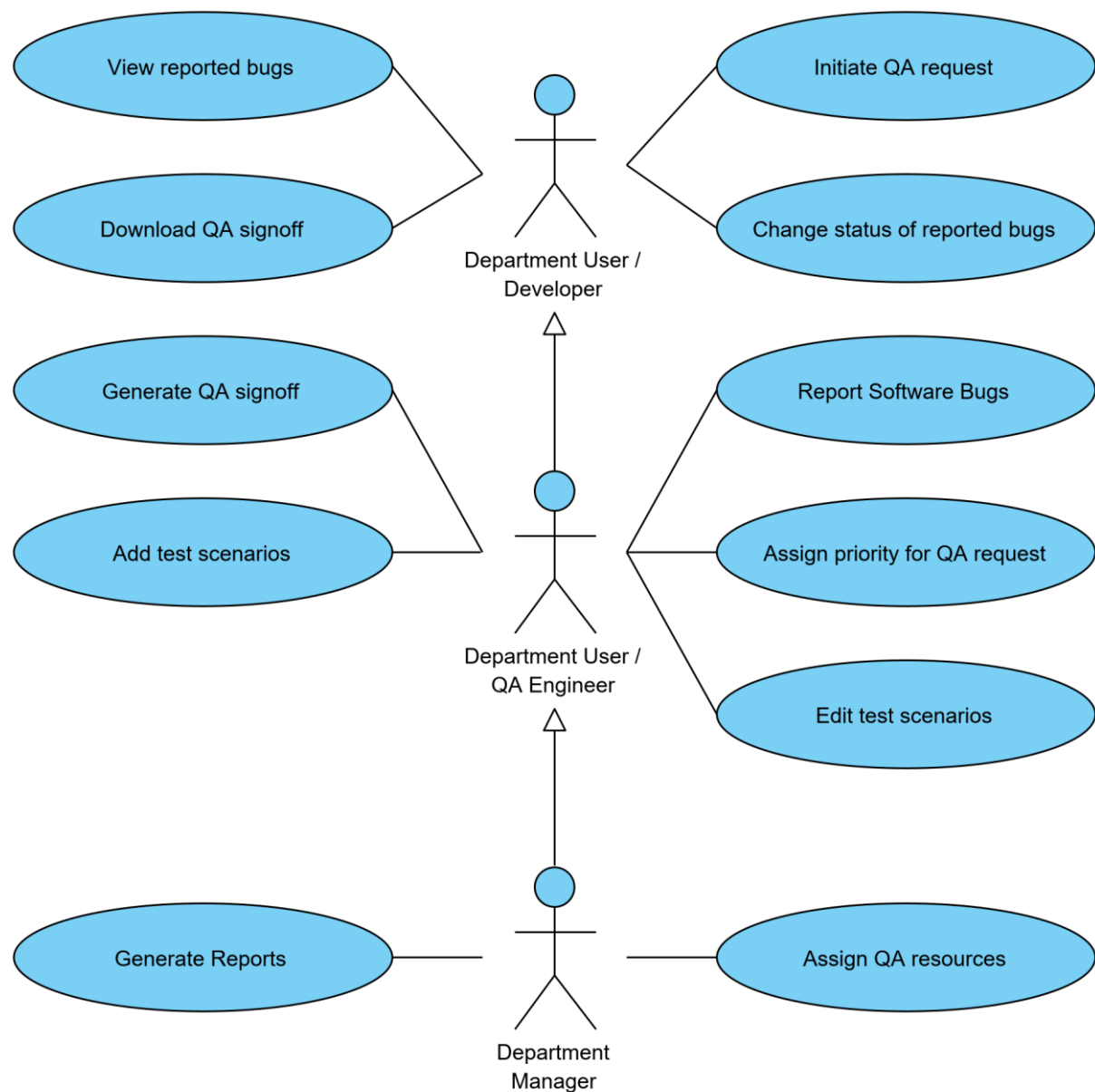


Figure A.1 - Use case diagram for Quality Assurance module

Use case diagram for the Quality assurance module in the developed application are shown on figure A.1, where department manager has the highest authority for the available functions. Mainly functions related to the testing are available in this module including creating test scenarios and managing bugs.

- Use case diagram for UAT module

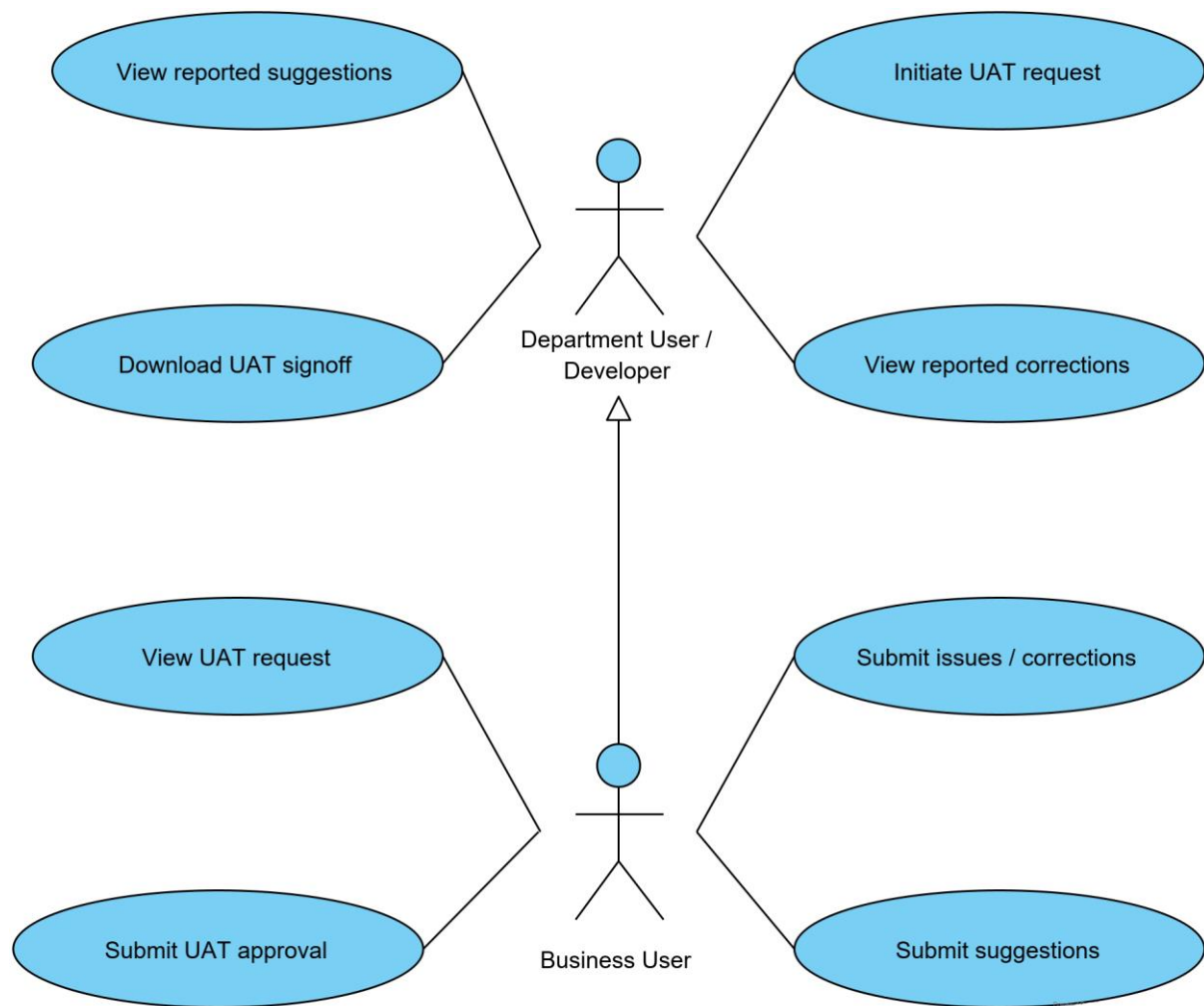


Figure A.2 - Use case diagram for UAT module

Use case diagram for the user acceptance testing module in the developed application are shown on figure A.2, where business user has the highest authority for the available functions. This was the only module in the developed application which have more functionalities for the business users than the users in the IT department.

Use Case Narrations

- Use case narration for login

Use case name	Login	<u>Use case Type</u> Business Requirements: Prevent unauthorized access and authenticate user.
Use case id	01	
Priority	High	
Source	Web Page – Login DB Table – User, Session	
Primary Business Actor	All Users	
Other Participating Actors		
Other Interested Stakeholders		
Description	Validate the username and the password entered by the user and grant access to the web application.	
Preconditions	User already has a user account created in the web application.	
Trigger	User send request to login by pressing the login button	
Typical course of events	1) User enter the username and password 2) Click on the Login button.	
Alternative course	Error message will be displayed for the invalid username or password entered by the user.	
Post conditions	Application will create a session and redirect the user to the home page.	

Table A.1 - Use case narration for login

- Use case narration for add new post to KM portal

Use case name	Add new post to KM portal	<u>Use case Type</u> Business Requirements: User will be allowed to share their knowledge on digital platform.
Use case id	05	
Priority	High	
Source	Web Page – KM Portal DB Table – KM_Posts	
Primary Business Actor	Department User	
Other Participating Actors	Department Manager, Business Manager	
Other Interested Stakeholders	Business User	
Description	Adding a new post to the Knowledge Management portal where other users also can view and reply to it.	
Preconditions	User have required access privileges to post on the knowledge management portal.	
Trigger	User selects the Knowledge Management function from the menu.	
Typical course of events	1) User select the category of the Knowledge Management portal. 2) Click on the add new post button to get the screen to submit new post. 3) Enter the post title and content. 4) Click on the post button to submit the post.	
Alternative course	If the content is too long than the accepted length, error message will be displayed for the user to amend the content.	
Post conditions	Once user post the message it will be displayed on the home screen of the knowledge management portal.	

Table A.2 - Use case narration for add new post to KM portal

Appendix B – MIS Reports


MIS Reports stands for the Management Information System reports which will be generated by the developed application automatically based on the criteria provided by the user. MIS reports will be critical for the smooth operation of the Bank since those reports will be referred by the senior management of the Bank in order to view the current status of the on-going projects. In this chapter it will demonstrate the MIS reports types and layouts used in the system with sample reports.

Administrative Reports

Administrative reports will include all the MIS reports regarding the user management and user actions handled through the system.

- Active users by user role

This report was designed to get the active users list by selecting the user role. For an example if the management needs a report on list of users with the administrative privileges have to execute this report.

 IT Demand Manager Requests Management System Users having Administrative Privileges					
#	PF Number	Name	Department	Assigned Role	Assigned Date
1	207490	G.K Wickramasinghe	IT Department	Administrator	2021-05-14
2	207317	S.N Abeynayake	IT Department	Administrator	2021-06-01
3	179180	R.M Gamage	IT Department	Administrator	2021-06-12


Report ID - A10010101 / Generated User - G.K Wickramasinghe (207490) / Date - 2021-07-01

Figure B.1 - Report on active user by role

If the user selects the user role as Administrator and generate the report it will display the output as shown in the figure B.1.

- List of users by user status

This report was designed to get the list of users by a specific user status. Since administrators will required to get a list of active users list at any time using this available report.

 IT Demand Manager Requests Management System List of users with status - Locked					
#	PF Number	Name	Department	User Status	Last Login
1	207205	R.M.L Perera	IT Department	Locked	2021-06-14
2	188190	T.M Nirupamali	IT Department	Locked	2021-06-14


Report ID - A10010201 / Generated User - G.K Wickramasinghe (207490) / Date - 2021-07-01

Figure B.2 - List of users by user status

Once user selects the user status as “Locked”, application will generate a report as shown in the figure B.2.

- New user accounts created last 30 days

Report generated for user accounts created within last 30 days is shown in figure B.3.

 IT Demand Manager Requests Management System New User Accounts - Last 30 Days					
#	PF Number	Name	Department	Created User	Created Date
1	200240	K. W Priyadarshana	Central Back Office	Gayan Wickramasinghe	2021-06-28
2	200245	L.S Weerakoon	Central Back Office	Gayan Wickramasinghe	2021-06-28

Report ID - A10010301 / Generated User - G.K Wickramasinghe (207490) / Date - 2021-07-01

Figure B.3 - Report on new user accounts

Demand Manager Module

Demand Manager module consists of multiple reports which will highlight the overall and the project wise status of the completed and ongoing Demand Manager requests handled by the IT division. Most of the reports available in this module will be accessed by the members in the Senior Management of the Bank.

- **Demand Manager – Overall Status**

Users will be able to generate a report on overall status of the demand manager requests attached to the particular department.


		
IT Demand Manager Requests Management System		
Demand Manager - Overall Status		
Department	Core Banking Development	
Time Period	2021	
Demand Manager Status	Count	Percentage
New	10	18%
Completed	25	44%
On Hold	3	5%
Development	6	11%
QA	8	14%
UAT	5	9%
	57	
Report ID - A20020101 / Generated User - G.K Wickramasinghe (207490) / Date - 2021-07-01		


Figure B.4 - Report on demand manager - overall status

In figure B.4, it shows a report which was generated by a team member in the Core Banking Development team on the status of all requests which were initiated on year 2021.

- Users were allowed to generate the report only for the user's department which was mentioned in the user profile.
- All the AGMs and DGMs were allowed to select a department and generate the report.
- All the AGMs and DGMs were allowed to generate the report for all the departments at once in order to view the status of all the departments.

- Demand Manager – Pending with Development Teams

By selecting a Development team in the IT Department, it will allow to generate a report on all the pending demand manager requests in the selected team.

 <h2 style="text-align: center;">IT Demand Manager Requests Management System</h2> <h3 style="text-align: center;">Demand Manager - Development Pending</h3>				
DM Number	DM Name	Assigned Date	Current Status	Requested Department
202100010	Inward Clearing - EIM Reject	2021-06-20	Not Started	Central Back Office
202100015	GoAML Reporting Module	2021-06-25	Not Started	Compliance Department

Report ID - A20020201 / Generated User - G.K Wickramasinghe (207490) / Date - 2021-07-01

Figure B.5 - Report on pending DM requests

Sample report on all the pending Demand Manager requests which needs to be developed by the IT Division are shown in the figure B.5

Appendix C – Testing Plan

Additional test cases used for testing of the developed Demand Manager Requests

Management System are shown below.

- Test Cases for Department Registration

ID	Activity	Test Case Steps	Expected Result
3.1	Create New Department screen	Click Add New Department option	Create New Department screen should open.
3.2	Check Validations	Name Field - Must be unique - Cannot be blank	Should display an error message if validation fails
		Location Field - Cannot be blank	
		Manager Field - Cannot be blank	
		Assistant General Manager Field - Cannot be blank - Characters not allowed	
		Deputy General Manager Field - Cannot be blank	
3.3	Create New Department – Success	Enter valid values for all the fields. Click on create user button.	Request should be successfully submitted.
3.4	Create New Department – Failed	Enter valid values for all the fields avoiding any mandatory field Click on create user button	Error Message should be displayed
3.5	Create New Department with already registered department name	Enter a department name which is already registered.	Should display an error message
		Enter all the necessary details accurately	
3.6	Update New Department – Success	Enter values for all the fields. Click on update button.	Request should be successfully submitted.
3.7	Update New Department – Failed	Enter valid values for all the fields avoiding any mandatory field Click on update button	Error Message should be displayed

Table C.1 - Test cases for registering department

- Test Cases for registering user designations

ID	Activity	Test Case Steps	Expected Result
4.1	Create New Designation	Click Add New Designation option	Create New User Designation screen should open.
4.2	Check Validations	Designation Field - Must be unique - Cannot be blank	Should display an error message if validation fails
		Category Field - Cannot be blank	
		User Field - Cannot be blank	
		Priority Field - Cannot be blank - Must be numeric value	
4.3	Create New Designation – Success	Enter valid values for all the fields. Click on create user button.	Request should be successfully submitted.
4.4	Create New Designation – Failed	Enter valid values for all the fields avoiding any mandatory field Click on create user button	Error Message should be displayed
4.5	Create New Designation with already registered designation name	Enter a department name which is already registered.	Should display an error message
		Enter all the necessary details accurately	
4.6	Update New Designation – Success	Enter values for all the fields. Click on update button.	Request should be successfully submitted.
4.7	Update New Designation – Failed	Enter valid values for all the fields avoiding any mandatory field Click on update button	Error Message should be displayed

Table C.2 - Test cases for registering new user designation

- Test Cases for registering user function

ID	Activity	Test Case Steps	Expected Result
5.1	Create New Function	Click Add New Designation option	Create New User Functions screen should open.
5.2	Check Validations	Function Code - Must be unique - Cannot be blank Function Name - Cannot be blank Function Path - Cannot be blank Page Title - Cannot be blank Main Section - Cannot be blank Sub Section - Cannot be blank Function Status - Cannot be blank	Should display an error message if validation fails
5.3	Create New Function – Success	Enter valid values for all the fields. Click on create user button.	Request should be successfully submitted.
5.4	Create New Function – Failed	Enter valid values for all the fields avoiding any mandatory field Click on create user button	Error Message should be displayed
5.5	Create New Function with already registered function name	Enter a department name which is already registered. Enter all the necessary details accurately	Should display an error message
5.6	Update New Function – Success	Enter values for all the fields. Click on update button.	Request should be successfully submitted.
5.7	Update New Function – Failed	Enter valid values for all the fields avoiding any mandatory field Click on update button	Error Message should be displayed

Table C.3 - Test cases for registering new user designation

- Test Cases for registering new user role

ID	Activity	Test Case Steps	Expected Result
6.1	Create New Role	Click Add New Role option	Create New Role screen should open.
6.2	Check Validations	Role ID - Must be unique - Cannot be blank	Should display an error message if validation fails
		Role Name - Cannot be blank	
		Role Status - Cannot be blank	
6.3	Create New Role – Success	Enter valid values for all the fields. Click on create user button.	Request should be successfully submitted.
6.4	Create New Role – Failed	Enter valid values for all the fields avoiding any mandatory field Click on create user button	Error Message should be displayed
5.5	Create New Role with already registered role name	Enter a department name which is already registered.	Should display an error message
		Enter all the necessary details accurately	
6.6	Set Role Privileges	Click on edit privileges button	Role Privileges screen should open.
6.7	Add Privilege to role	Activate the function from the list of functions	Function access should be granted to the user and should be able to access it.
6.8	Remove Privilege from role	Deactivate the function from the list of functions	Function access should be removed from the user and should not be able to access it.
6.9	Update New Role – Success	Enter values for all the fields. Click on update button.	Request should be successfully submitted.
6.10	Update New Role – Failed	Enter valid values for all the fields avoiding any mandatory field Click on update button	Error Message should be displayed

Table C.4 - Test cases for registering new user role

- Testing Status

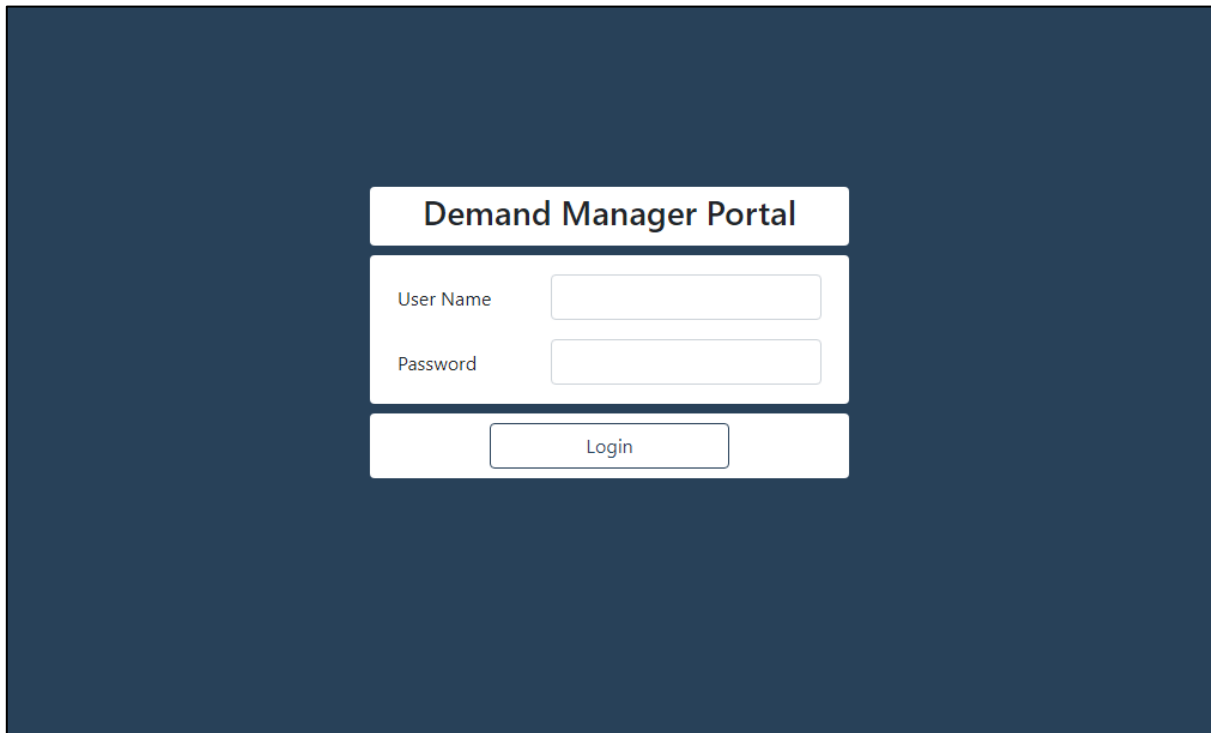
Test ID	Description	Pass / Fail
3.1	Create New Department screen	Pass
3.2	Check Validations	Pass
3.3	Create New Department – Success	Pass
3.4	Create New Department – Failed	Pass
3.5	Create New Department with already registered dept name	Pass
3.6	Update New Department – Success	Pass
3.7	Update New Department – Failed	Pass
4.1	Create New Designation screen	Pass
4.2	Check Validations	Pass
4.3	Create New Designation – Success	Pass
4.4	Create New Designation – Failed	Pass
4.5	Create New Designation with already registered name	Pass
4.6	Update New Designation – Success	Pass
4.7	Update New Designation – Failed	Pass
5.1	Create New Function screen	Pass
5.2	Check Validations	Pass
5.3	Create New Function – Success	Pass
5.4	Create New Function – Failed	Pass
5.5	Create New Function with already registered function name	Pass
5.6	Update New Function – Success	Pass
5.7	Update New Function – Failed	Pass
6.1	Create New Role	Pass
6.2	Check Validations	Pass
6.3	Create New Role – Success	Pass
6.4	Create New Role – Failed	Pass
6.5	Create New Role with already registered role name	Pass
6.6	Set Role Privileges	Pass
6.7	Add Privilege to role	Pass
6.8	Remove Privilege from role	Pass
6.9	Update New Role – Success	Pass
6.10	Update New Role – Failed	Pass

Table C.5 - Status of test cases

Appendix D – User Manual

User manual of the developed application for the main functionalities are described in this section including the screenshots and instructions on how to use the function.

Login



The screenshot shows a login interface for the 'Demand Manager Portal'. The interface is centered on a dark blue background. It consists of a white box with the title 'Demand Manager Portal' at the top. Below the title, there are two input fields: 'User Name' and 'Password'. Below these fields is a 'Login' button.

Figure D.1 - Screenshot of Login

- Initial Login screen of the application shown in figure D.1 and user needs to enter already registered username and password in order to login to the web application.
- Upon the successful authentication, user will be redirected to the home page and error message will be displayed if the authentication fails.
- If the user enters the incorrect password for more than 3 times, account will be automatically locked. Only the users with administrative privileges will be able to activate the account again.

User Maintenance

The screenshot displays the 'User Maintenance' interface within the 'Demand Manager Portal'. The top navigation bar includes links for Home, Administration, Demand Manager, Development, Quality Assurance, User Acceptance, Reporting, and User Profile. The main header shows 'Administration | User Maintenance' and an 'Add New User' button. A search bar is located above the table. The table lists four users with the following data:

PF No	Name	Department	Grade	Role	Status
170180	Ramya Gamage	IT - Core Banking Development	Branch Manager	Administrator	Active
207317	Samuda Abeynayake	IT - Core Banking Development	Executive Officer	Administrator	Active
207458	Dinesh Rathnayake	IT - Core Banking Development	Executive Officer	Developer	Active
207490	Gayan Wickramasinghe	IT - Core Banking Development	Executive Officer	Administrator	Active

Below the table, it indicates 'Showing 1 to 4 of 4 entries' and includes 'Previous', '1', and 'Next' pagination controls. The footer contains the copyright notice: '© Copyright 2021, University of Colombo School of Computing'.

Figure D.2 - Screenshot of user maintenance

- As shown in the figure D.2, initial user maintenance screen will display the list of users registered in the system.
- User Maintenance function will be available for the users with the administrative privileges.
- User's provident fund number, name, department, grade, role and user status will be shown in the table with the facility to sort and search.
- By clicking on the Add New User button users will be allowed to create a new user profile.
- Users are allowed to search by any column value through the search text box on top of the table.

- By clicking on the image at each row it will display all the details related to the selected user profile as shown in figure D.3.

User Details | PF207490

User ID	1
PF Number	207490
User Name	it207490
First Name	Gayan
Last Name	Wickramasinghe
Name with Initials	GK Wickramasinghe
Gender	M
Department	IT - Core Banking Development
Grade	Executive Officer
User Role	Administrator
Email Address	gayanw@boc.lk
Mobile Number	0773333848
Extension Number	3569
IP Phone Number	13569
Direct Number	
IP Address	172.20.106.142
Status Code	Active
Notes	
Added Date	2021-07-12 20:06:14
Modified Date	2021-07-21 17:18:28

[Edit User Data](#)

© Copyright 2021, University of Colombo School of Computing

Figure D.3 - Screenshot of the user maintenance

- All the user details available for the selected user will be displayed in this form and users are allowed to edit the data by clicking on the Edit User Data button.
- Also, this form will display the record added timestamp and the last modified timestamp for the reference of the user.

- By Clicking on the Edit User Data, it will allow the user to edit the details as shown in figure D.4

User Details | User ID - 1

PF Number *	207490
User Name *	it207490
First Name *	Gayan
Last Name *	Wickramasinghe
Name with Initials *	GK Wickramasinghe
Gender *	<input checked="" type="radio"/> Male <input type="radio"/> Female
Department *	IT - Core Banking Development
Grade *	Executive Officer
User Role *	Administrator
Email Address *	gayanw@boc.lk
Mobile Number *	0773333848
Extension Number *	3569
IP Phone Number	13569
Direct Number	
IP Address	172.20.106.142
Status Code *	Active
Notes	

Update

© Copyright 2021, University of Colombo School of Computing

Figure D.4 - Screenshot of the user maintenance

- List of all the active departments and list of all the active user roles registered in the application will be shown as a dropdown list for the user.
- Confirmation message will be displayed upon the successful update of the record as shown in figure D.5.

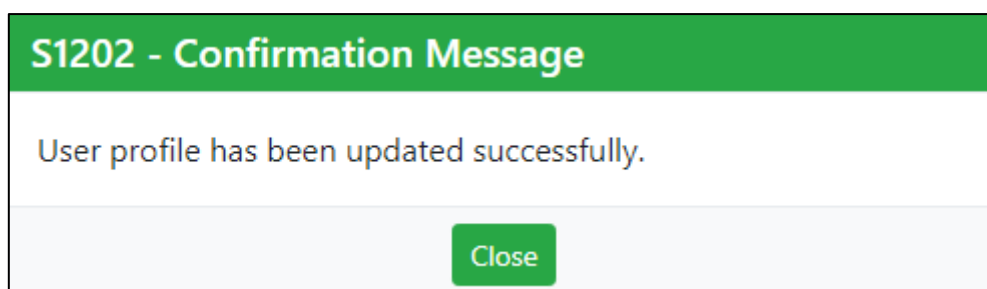


Figure D.5 - Screenshot of confirmation message

Department Maintenance

The screenshot displays the 'Administration | Department Maintenance' page. At the top, there is a navigation bar with links: Home, Administration, Demand Manager, Development, Quality Assurance, User Acceptance, Reporting, and User Profile. Below this, a sub-header shows 'Administration | Department Maintenance' and a '+ Add New Department' button. A search bar is located above the table. The table has columns: ID, Name, Manager, AGM, Location, and Status. It contains 4 entries, all with a status of 'Active'. Below the table, it says 'Showing 1 to 4 of 4 entries' and has pagination controls for 'Previous', '1', and 'Next'.

ID	Name	Manager	AGM	Location	Status
1	IT - Core Banking Development	RP Gamage	AGM - IT (Application Systems)	Head Office - 6th Floor	Active
2	IT - Core Banking Maintenance	GK Wickramasinghe	AGM - IT (Application Systems)	Head Office - 6th Floor	Active
3	IT - Trade Finance	SN Abeynayake	AGM - IT (Application Systems)	Head Office - 11th Floor	Active
4	IT - General Applications	RMLD Rathnayake	AGM - IT (Delivery Channels)	Head Office - 6th Floor	Active

Showing 1 to 4 of 4 entries

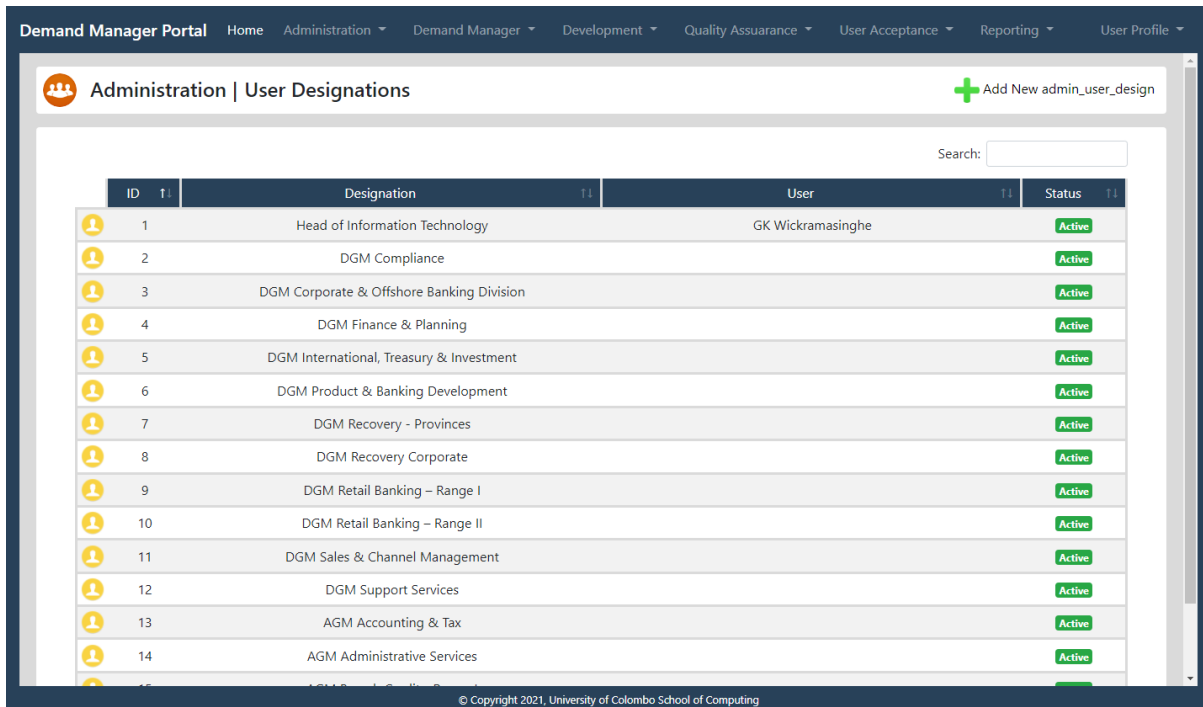
Previous 1 Next

© Copyright 2021, University of Colombo School of Computing

Figure D.6 - Screenshot of department maintenance

- Once user select the department maintenance option through the main menu it will display the function as shown in figure D.6
- Department Maintenance function will be available for the users with the administrative privileges.
- Department Id, name, manager, Assistant General Manager, Location and status will be shown in the table with the facility to sort and search.
- Users are allowed to search by any column value through the search text box on top of the table.
- By clicking on the Add New Department button users will be allowed to create a new user department.
- Same as all the other functions user will be allowed to view and edit the data by clicking on the row icon.

User Designations

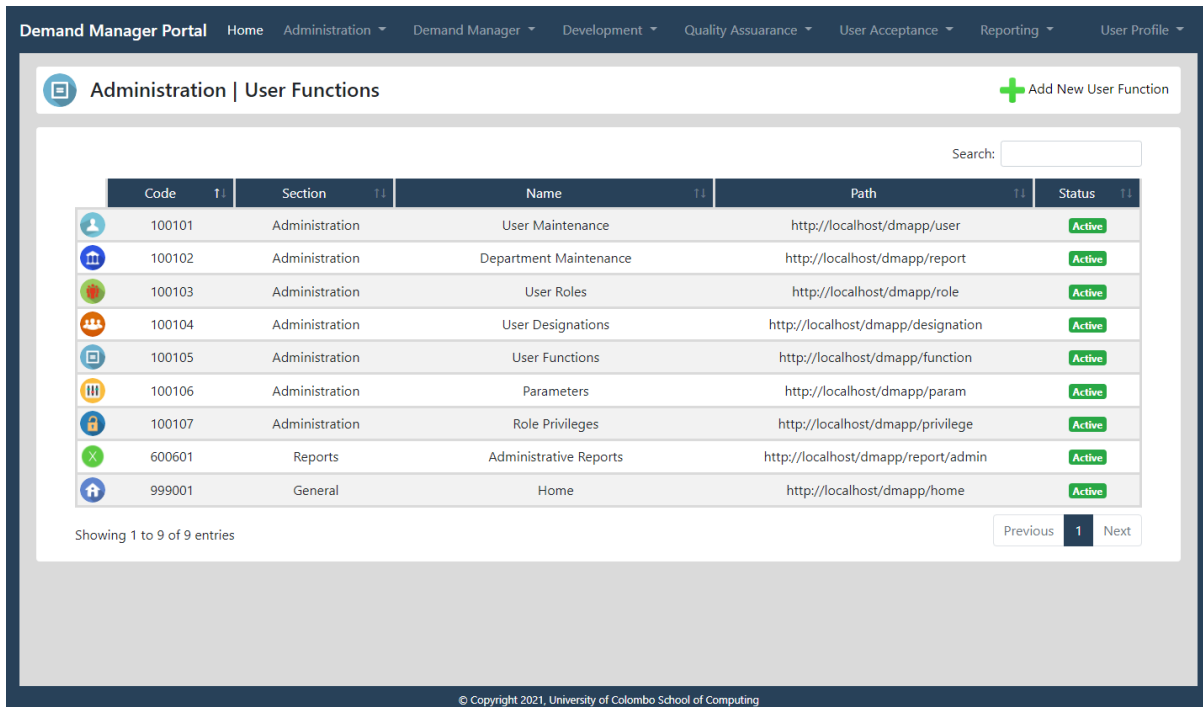


ID	Designation	User	Status
1	Head of Information Technology	GK Wickramasinghe	Active
2	DGM Compliance		Active
3	DGM Corporate & Offshore Banking Division		Active
4	DGM Finance & Planning		Active
5	DGM International, Treasury & Investment		Active
6	DGM Product & Banking Development		Active
7	DGM Recovery - Provinces		Active
8	DGM Recovery Corporate		Active
9	DGM Retail Banking – Range I		Active
10	DGM Retail Banking – Range II		Active
11	DGM Sales & Channel Management		Active
12	DGM Support Services		Active
13	AGM Accounting & Tax		Active
14	AGM Administrative Services		Active

Figure D.7 - Screenshot of user designations

- Once user select the User Designations option through the main menu it will display the function as shown in figure D.7
- User Designations function will be available for the users with the administrative privileges.
- Id, designation, user and status will be shown in the table with the facility to sort and search.
- Users are allowed to search by any column value through the search text box on top of the table.
- By clicking on the Add New Designation button users will be allowed to create a new user designation.
- Same as all the other functions user will be allowed to view and edit the data by clicking on the row icon.

User Functions



Demand Manager Portal Home Administration Demand Manager Development Quality Assurance User Acceptance Reporting User Profile

Administration | User Functions + Add New User Function

Search:

	Code	Section	Name	Path	Status
	100101	Administration	User Maintenance	http://localhost/dmapp/user	Active
	100102	Administration	Department Maintenance	http://localhost/dmapp/report	Active
	100103	Administration	User Roles	http://localhost/dmapp/role	Active
	100104	Administration	User Designations	http://localhost/dmapp/designation	Active
	100105	Administration	User Functions	http://localhost/dmapp/function	Active
	100106	Administration	Parameters	http://localhost/dmapp/param	Active
	100107	Administration	Role Privileges	http://localhost/dmapp/privilege	Active
	600601	Reports	Administrative Reports	http://localhost/dmapp/report/admin	Active
	999001	General	Home	http://localhost/dmapp/home	Active

Showing 1 to 9 of 9 entries Previous 1 Next

© Copyright 2021, University of Colombo School of Computing

Figure D.8 - Screenshot of user functions

- Once user select the User Functions option through the main menu it will display the function as shown in figure D.8
- This option will be used to define the user functions available in the system.
- User Functions option will be available for the users with the administrative privileges.
- Function Code, section, name, path and status will be shown in the table with the facility to sort and search.
- Users are allowed to search by any column value through the search text box on top of the table.
- By clicking on the Add New Function button users will be allowed to create a new user function.
- Same as all the other functions user will be allowed to view and edit the data by clicking on the row icon.

Role Privileges

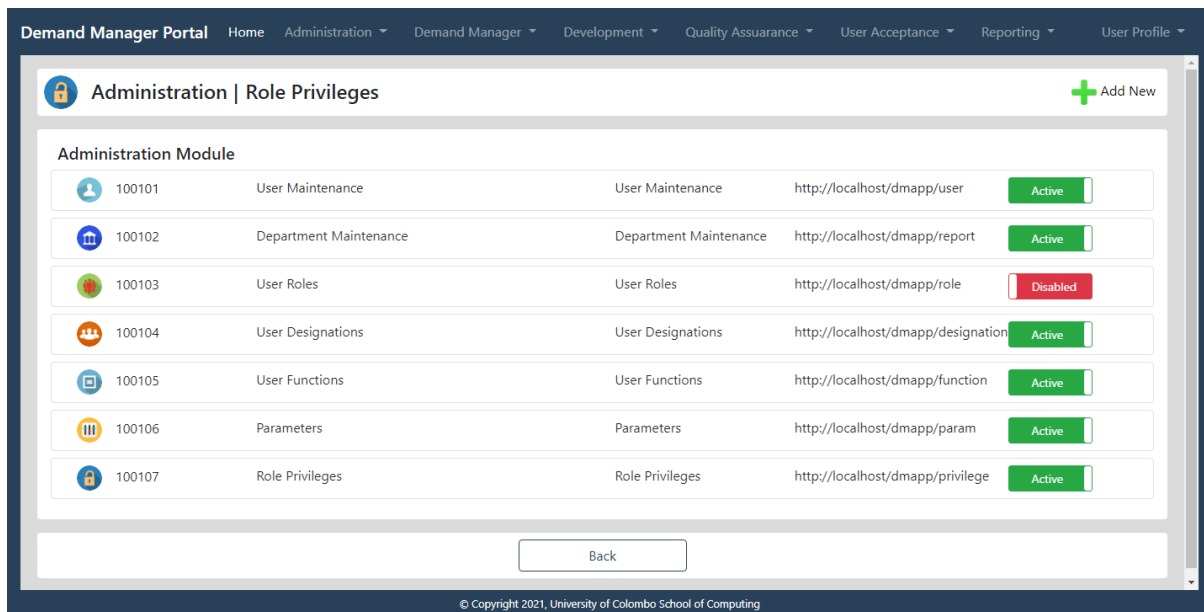


Figure D.9 - Screenshot of role privileges

- Once user select the edit privileges option through the user roles function it will display the function as shown in figure D.9
- This option will be used to define the user functions available in the system.
- Role Privileges Functions option will be available for the users with the administrative privileges.
- By enabling the option, it will be available for the users with the selected privilege.
- By disabling the option, user access will be removed and users will not be allowed to access the function.