**University Assets Maintenance App**

**Design Document**

**Version 1.0**



**Group Id: BC190400730**

**Supervisor Name : Safi Ullah Nasir**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 21-02-2023 | 1.0 | University assets maintenance app is an android mobile application. This application provides the easy and efficient way to maintain the university assets. The staff registers in the application and start the complaint of the item, the admin can assign the complaint to the staff member for completion. After completion of complaint the staff mark the complaint status completed, the complainant verify the complaint and close the complaint. The admin can view the complaint detail in different ways like by id, by dates etc. the admin can view the close complaint. This application provides easy and efficient way to handle all the process of the asset maintenance easily like repair or replace. | **BC190400730** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Introduction of Design Document](#One)
2. [Entity Relationship Diagram (ERD)](#ERD)
3. [Sequence Diagrams](#Six)
4. [Architecture Design Diagram](#Seven)
5. [Class Diagram](#class)
6. [Database Design](#databasedesign)
7. [Interface Design](#interfacedesign)
8. [Test Cases](#testcases)

1. Introduction of Design Document

The Software Design Document is a document to provide documentation which will be used later in software development by providing the details for how the software should be built.

Within the Software Design Document are narrative and graphical documentation of the software design for the project including Entity Relationship Diagram (ERD), Sequence Diagrams, Architecture Design Diagram, and Class diagram, Database Design, Interface Design and Test Cases.

Entity Relationship diagram shows the complete logical structure of the database. It has the components of entity set and their relationship. The entity set is a group of similar entities that have attributes. In DBMS entity is a table and attributes are the table data this diagram also shows the relationship between tables and their attributes.

Sequence diagram is an interaction diagram it describes how and in what sequential order a group of objects works together. Architecture diagrams describes what are you developing and how user interact with it. It contains the components of interface, database etc.

Class diagram shows the system structure in classes and their attributes, operation and relationship between class objects. Database diagram shows database model. It shows the various database tables of the system and their relationship with each other.

User interface design shows the appearance of the website application. It shows the design and layout of the system. Test cases are a testing scenario. It is used to test functionality with verify excepted result.

**Benefits:**

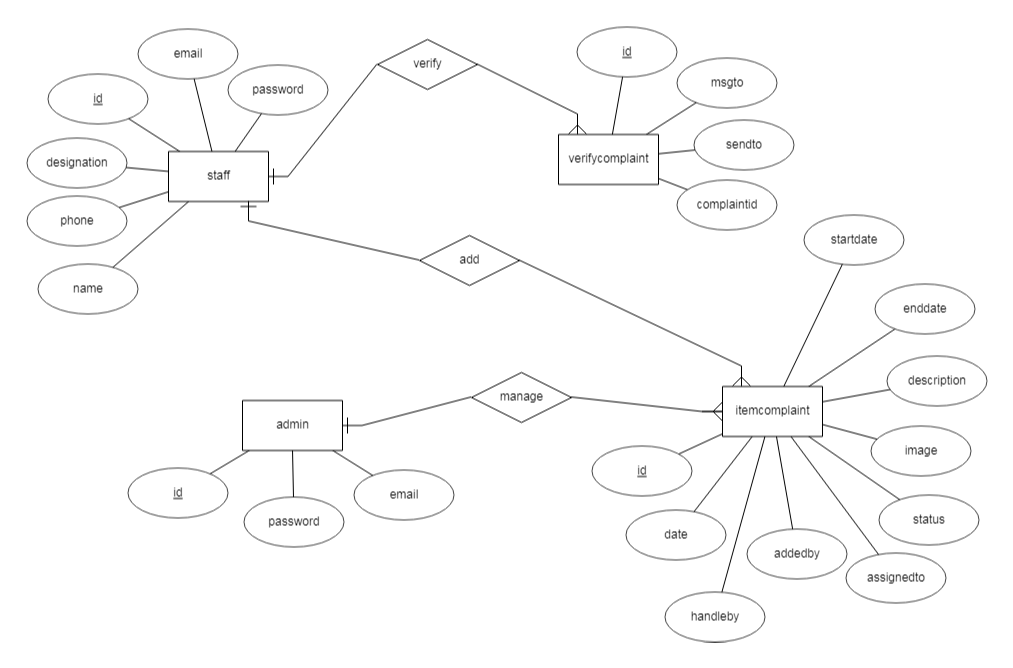
There are many benefits of a well-defined design document some of those are:

User’s expectation of the system is documented properly and helps a lot later while developing the site. All major design issues are discussed. Different analysis on the system can be performed. It saves a lot of time of the developer when all the functionality of the system is captured properly and is documented in a file.

**Purpose:**

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

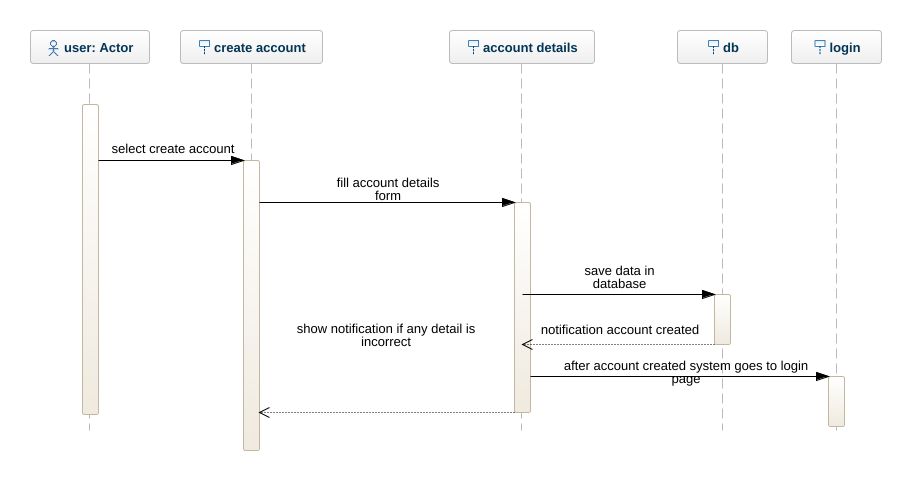
1. Entity Relationship Diagram (ERD) (To be developed using Microsoft Visio or any other drawing software of your choice)



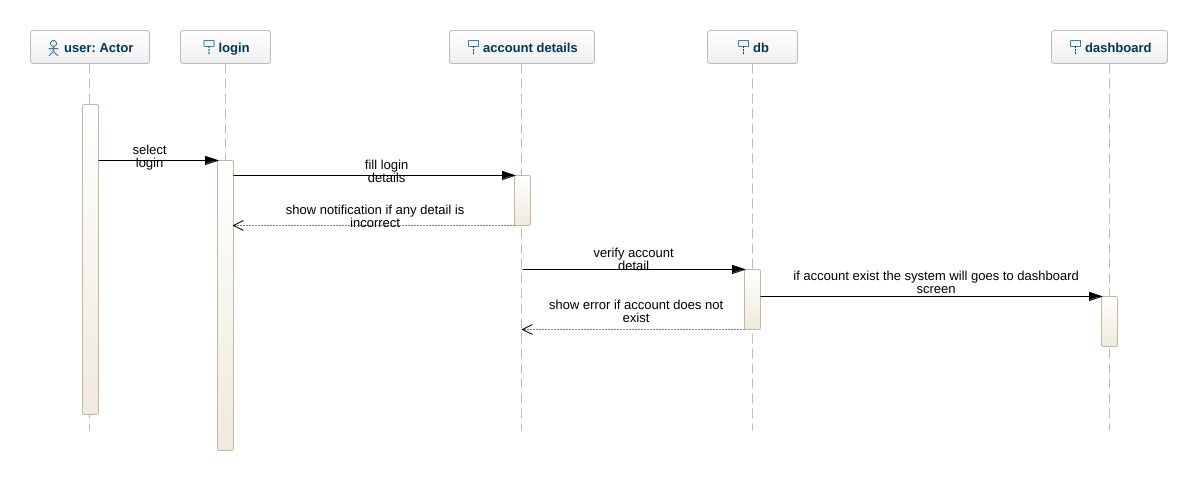
1. Sequence Diagrams (To be developed using Rational Rose or any other drawing software of your choice)

Sequence Diagram:

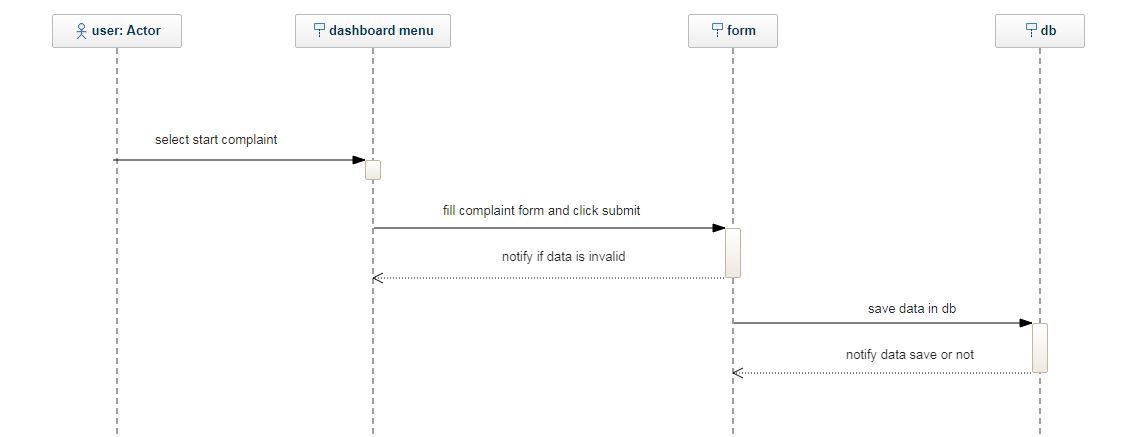
1. Register



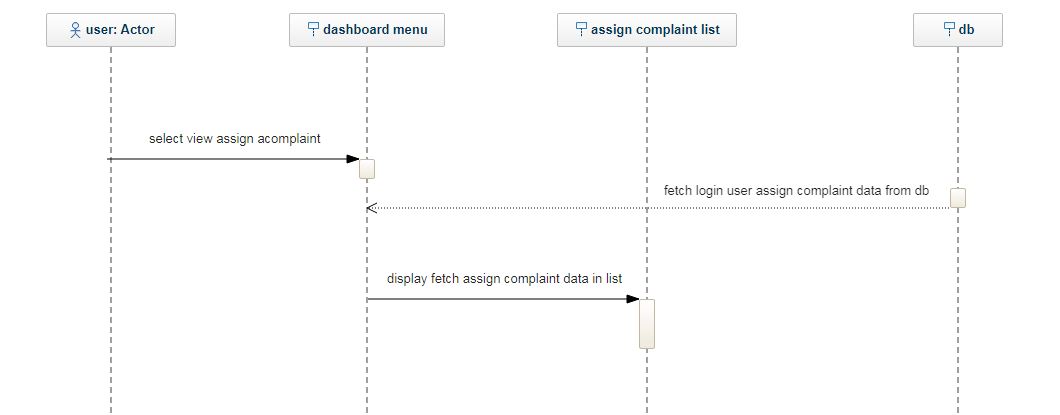
2. login



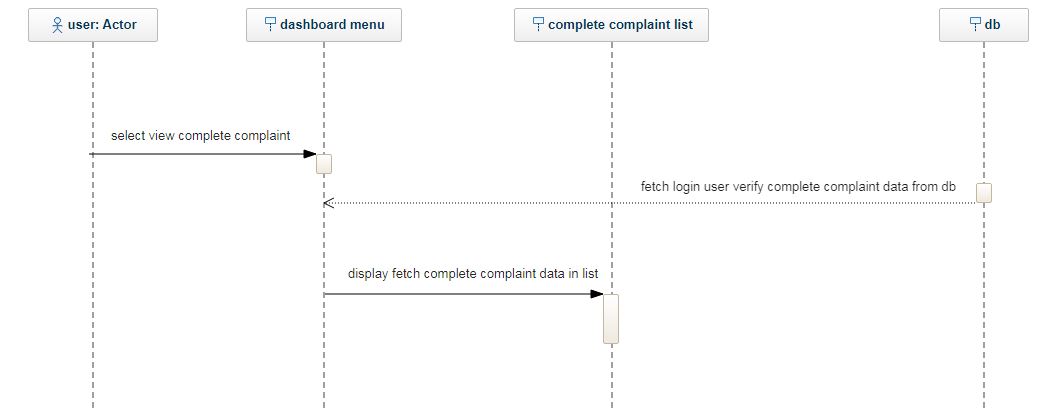
3. start complaint



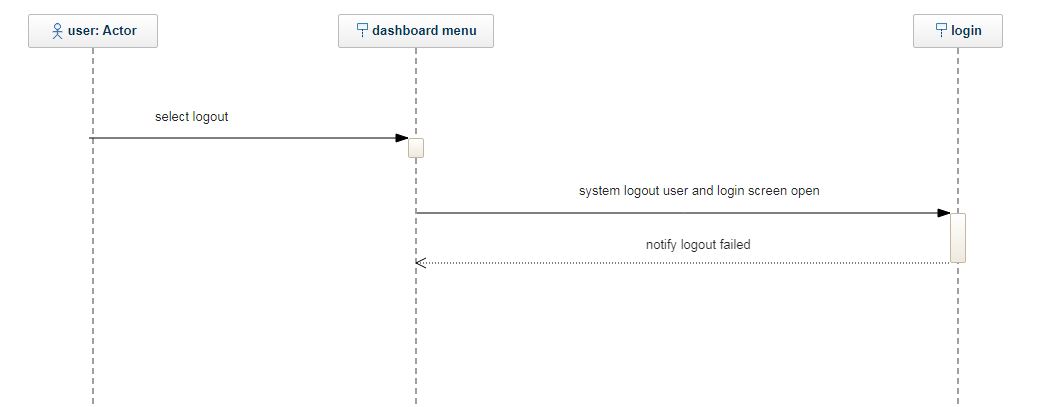
4. view assign complaint



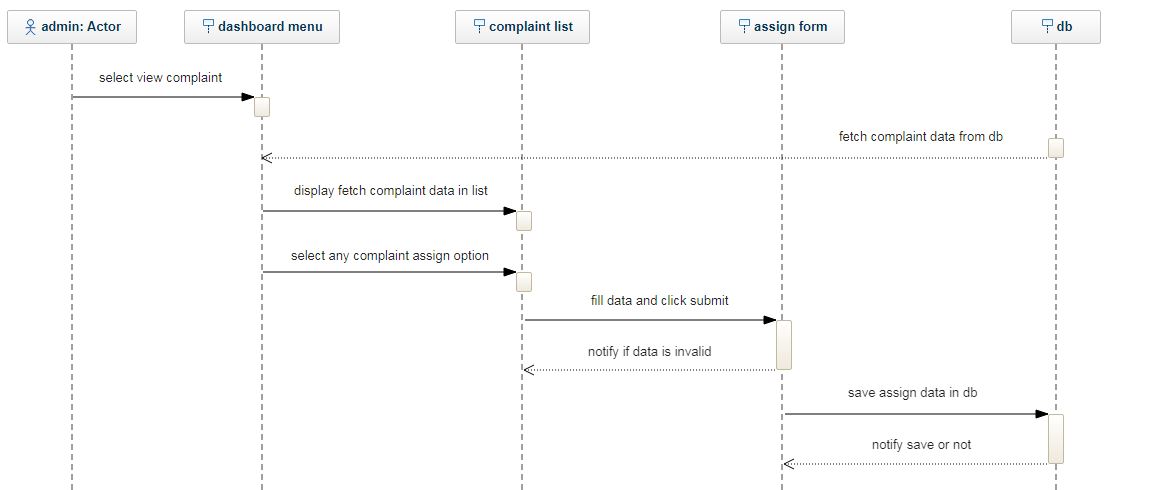
5. view completed complaint



6. logout



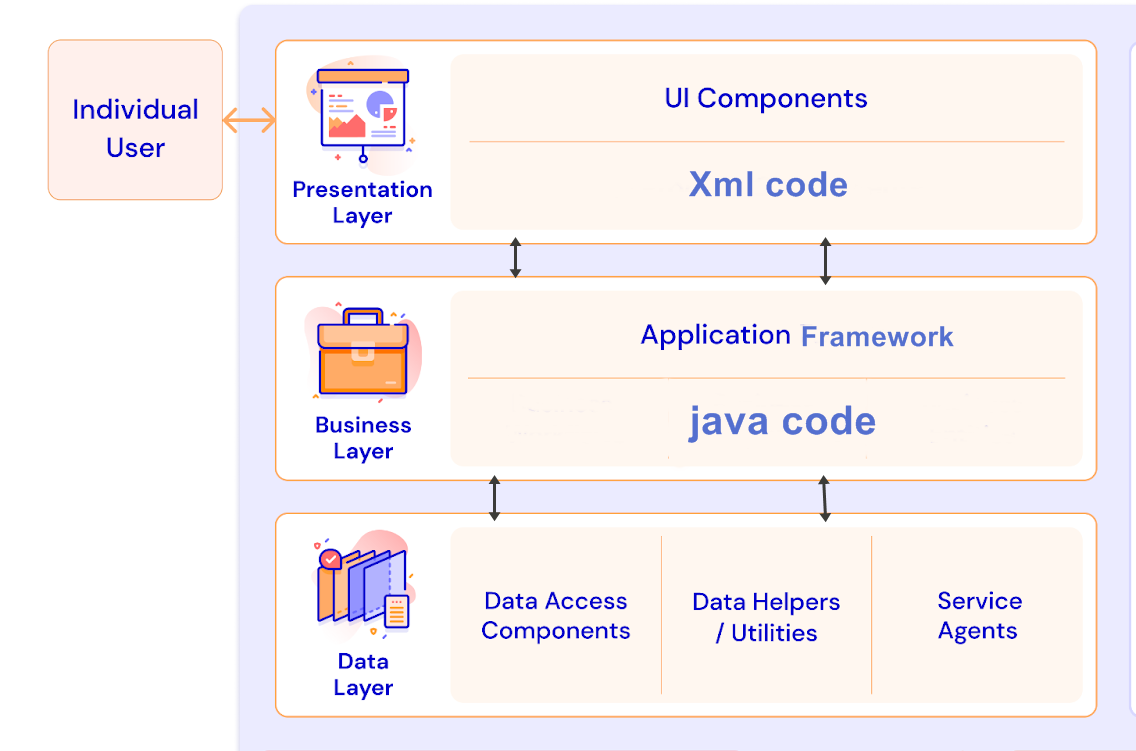
7. view complaints and assign complaints



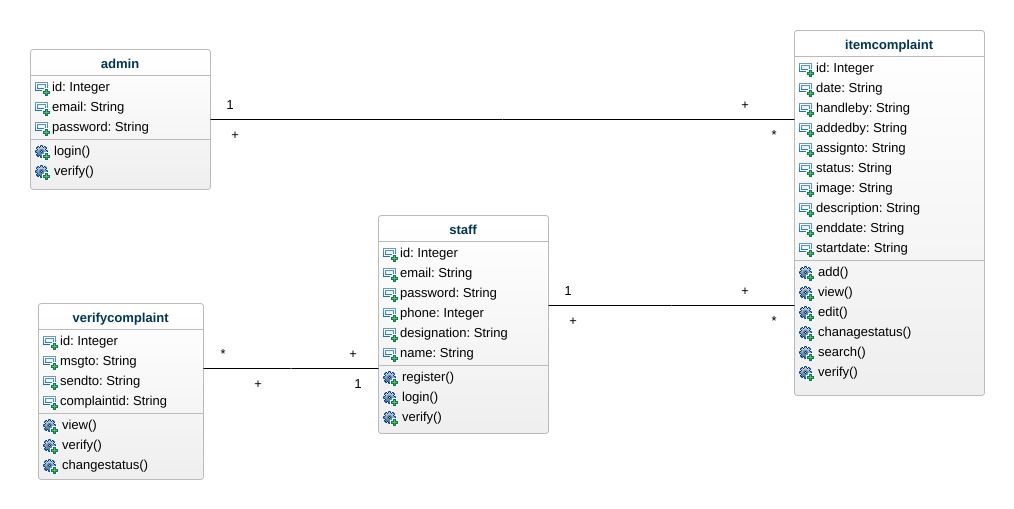
8. view notification of close complaint



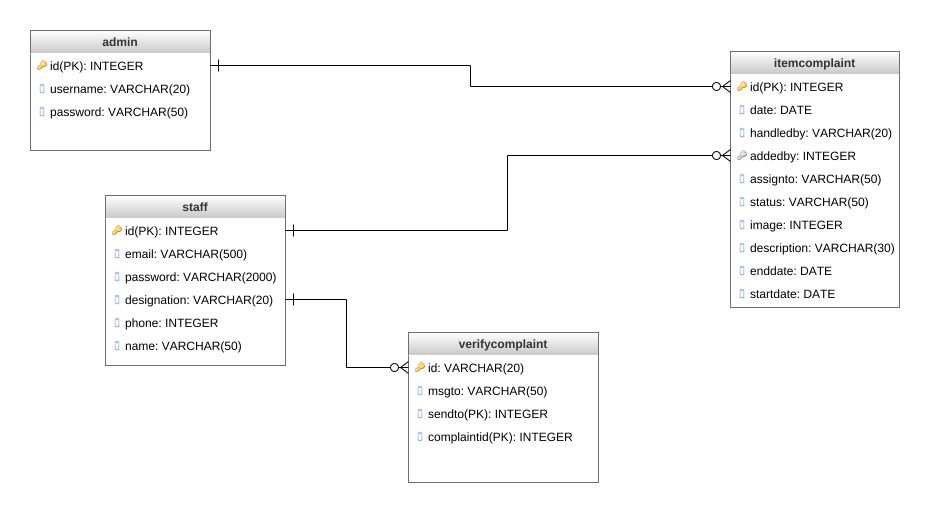
1. Architecture Design Diagram



1. Class Diagram

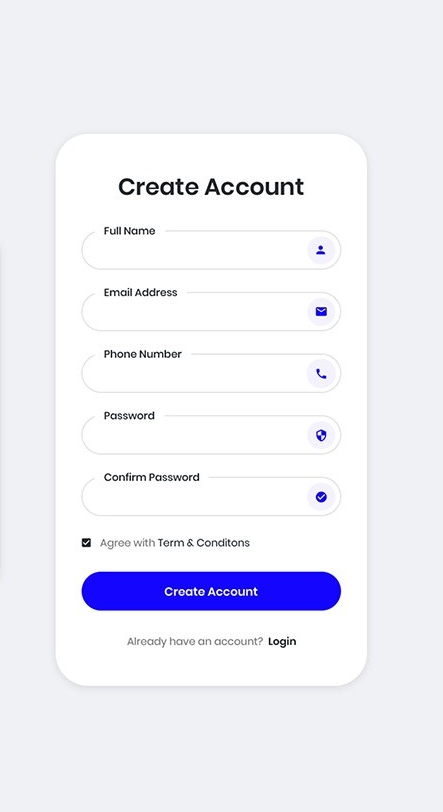


1. Database Design

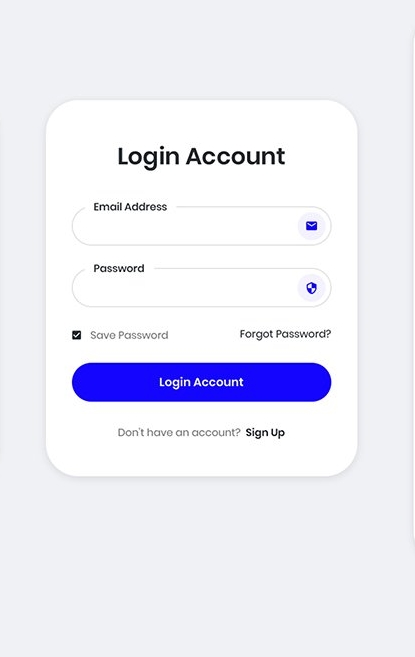


1. Interface Design

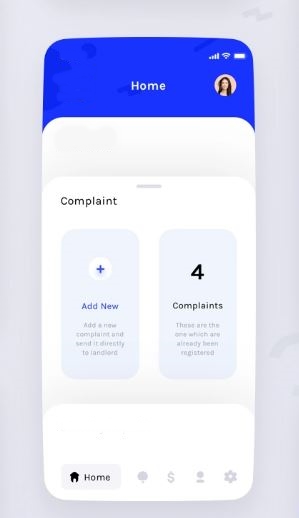
Register



Login



Dashboard



1. Test Cases

**Test Case # 1**

**Test Case Title:** register.

|  |  |
| --- | --- |
| Preconditions | User open the app and select register option |
| Actions | Create account form will open and user add its detail to create account form and select the button register. |
| Expected Results | System creates the user account and save account detail in the database. |
| Tested By: |  |
| Result | Pass/Fail |

**Test Case # 2**

**Test Case Title:** login

|  |  |
| --- | --- |
| Preconditions | User opens the app and select login option and previously created account. |
| Actions | Login form will open and user adds its account detail for login and selects the button login. |
| Expected Results | System login the user in its account if user account details exist in the database. After login the system open the dashboard page. |
| Tested By: |  |
| Result | Pass/Fail |

**Test Case # 3**

**Test Case Title:** start complaint

|  |  |
| --- | --- |
| Preconditions | User must login in his account |
| Actions | The user select the start complaint option from the menu |
| Expected Results | System open the start add complaint form and the user enter the detail of the complaint in the form and click submit the system save the complaint detail in the database and notify complaint added successfully. |
| Tested By: |  |
| Result | Pass/Fail |

**Test Case # 4**

**Test Case Title:** view assign complaint

|  |  |
| --- | --- |
| Preconditions | User must login in his account. |
| Actions | The user select the view assign complaint option from the menu |
| Expected Results | System fetch the login user assign complaint detail data from the database and display it on the screen. |
| Tested By: |  |
| Result | Pass/Fail |

**Test Case # 5**

**Test Case Title:** view completed complaint

|  |  |
| --- | --- |
| Preconditions | User must login in his account |
| Actions | The user select the view completed complaint verification option from the menu |
| Expected Results | System fetch the complaint that has the verification status in the database and show it the user can mark the complaint as completed by selecting complete button the system change the status of selected complaint in the database and notify user complaint marked as completed successfully. |
| Tested By: |  |
| Result | Pass/Fail |

**Test Case # 6**

**Test Case Title:** logout

|  |  |
| --- | --- |
| Preconditions | User must login in his account |
| Actions | The user select logout option from menu |
| Expected Results | system logout the login user from his account and open the login screen |
| Tested By: |  |
| Result | Pass/Fail |

**Test Case # 7**

**Test Case Title:** View Compliant and assign complaint

|  |  |
| --- | --- |
| Preconditions | admin must login in his account. |
| Actions | The user selects the complaint option from dashboard menu. |
| Expected Results | Systems fetch the complaint detail data from the database and show it in the list. The admin can select any complaint and click assign to the system open the assign form and admin enter the assign detail the form and click submit. System save the assign detail of the complaint in the database and notify complaint assign successfully. |
| Tested By: |  |
| Result | Pass/Fail |

**Test Case # 8**

**Test Case Title:** view notification of close complaint

|  |  |
| --- | --- |
| Preconditions | Admin must login in his account |
| Actions | The admin select the options view notification |
| Expected Results | System fetches all the complaint from the databases that has the status closed and display it in the list. |
| Tested By: |  |
| Result | Pass/Fail |