**Design Document**

**for**

**Software Design Project**

**Version 2.0**

**Prepared by: Gonul Kilic**

**Pooja Khare**

**Edward Ai**

**Surabhi Trivedi**

**Table of Contents**

1. **INTRODUCTION**
   1. **Purpose…………………………………………………………………………………………………………………3**
   2. **Scope…………………………………………………………………………………………………………………….3**
   3. **Overview……………………………………………………………………………………………………………….3**
   4. **Reference Material………………………………………………………………………………………………..3**
   5. **Definitions and Acronyms………………………………………………………………………………………3**
2. **SYSTEM OVERVIEW………………………………………………………………………………………………………3**
3. **SYSTEM ARCHITECTURE………………………………………………………………………………………………..4**
   1. **Architectural Design……………………………………………………………………………………………….4**
   2. **Decomposition Description…………………………………………………………………………………….5**
   3. **Design Rationale…………………………………………………………………………………………………….6**
4. **DATA DESIGN……………………………………………………………………………………………………………….7**
   1. **Data Description……………………………………………………………………………………………………7**
   2. **Data Dictionary……………………………………………………………………………………………………..7**
5. **COMPONENT DESIGN………………………………………………………………………………………………….10**
6. **HUMAN INTERFACE DESIGN…………………………………………………………………………………………11**
   1. **Overview of User Interface……………………………………………………………………………………11**
   2. **Screen Images……………………………………………………………………………………………………….11**
   3. **Screen Objects and Actions…………………………………………………………………………………...12**
7. **REQUIREMENT MATRIX……………………………………………………………………………………………….13**
8. **APPENDICES…………………………………………………………………………………………………………………14**
9. **INTRODUCTION**
   1. **Purpose**

The purpose of this document is to describe how architecture and implementation of Software Development Project. The Software Requirement Specification for Software Development project is the main source for this document. However, the intended audiences for this document are mainly software developers if there is any need to maintenance project also Building Administrator can use this document.

* 1. **Scope**

The scope of this project is mainly help with the users with web application while they are doing the maintenance of buildings. User needs this application to follow the forms and control probable issues with the buildings easily. Basically , technician will check the buildings and will keep record with filling the forms and Manager and Administrator will be able to view and edit the form. Also. Whenever needed the Administrator will be able to create a new type of the form.

* 1. **Overview**

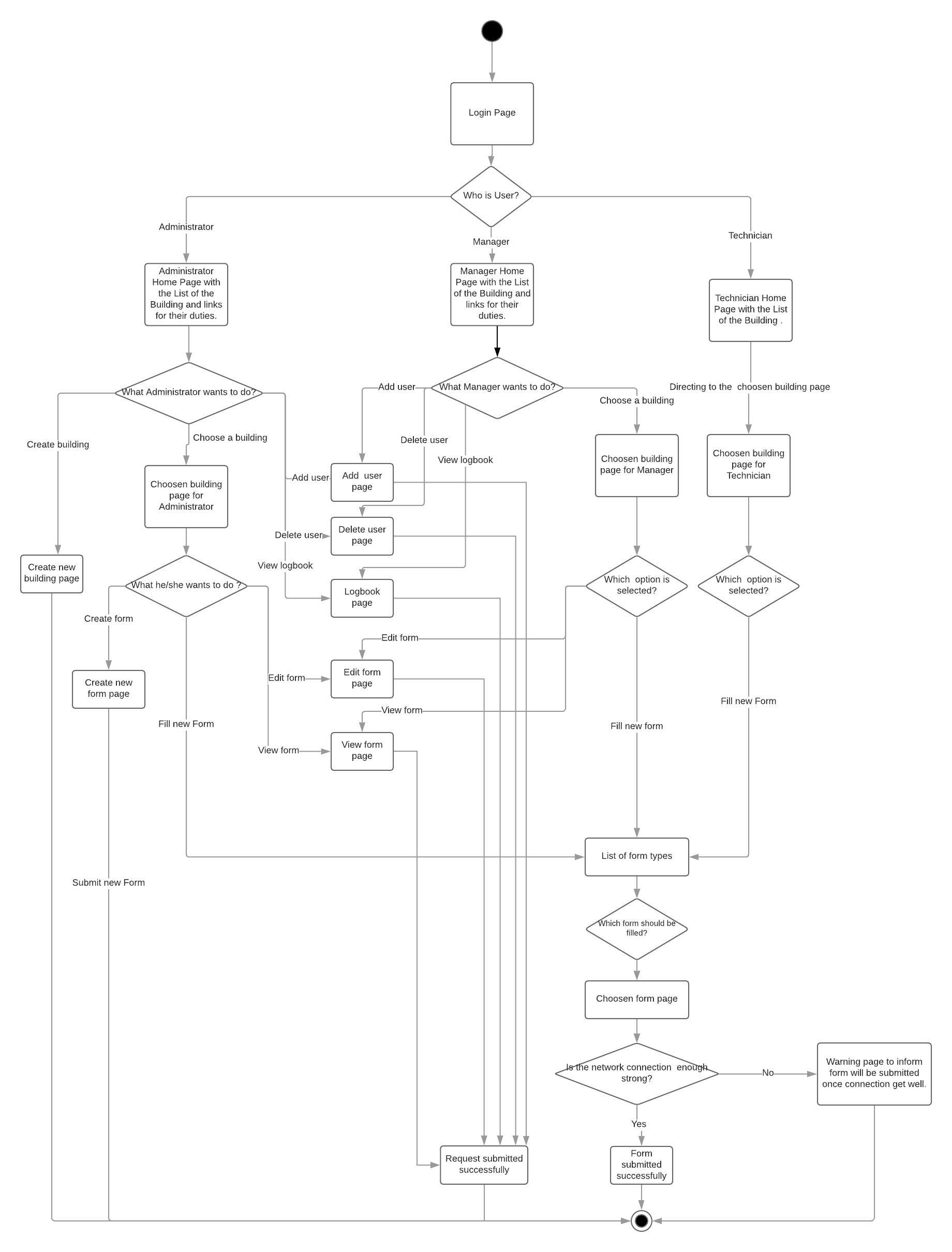
This document has been created according to the IEEE Recommended Practice for Software Design Descriptions. Document will have 8 sections as can be seen on Table of Contents.

* 1. **Reference Material**

1. Software Development Project / Software Requirement Specification Document
2. IEEE Recommended Practice for Software Design Descriptions
   1. **Definitions and Acronyms**
3. **SYSTEM OVERVIEW**

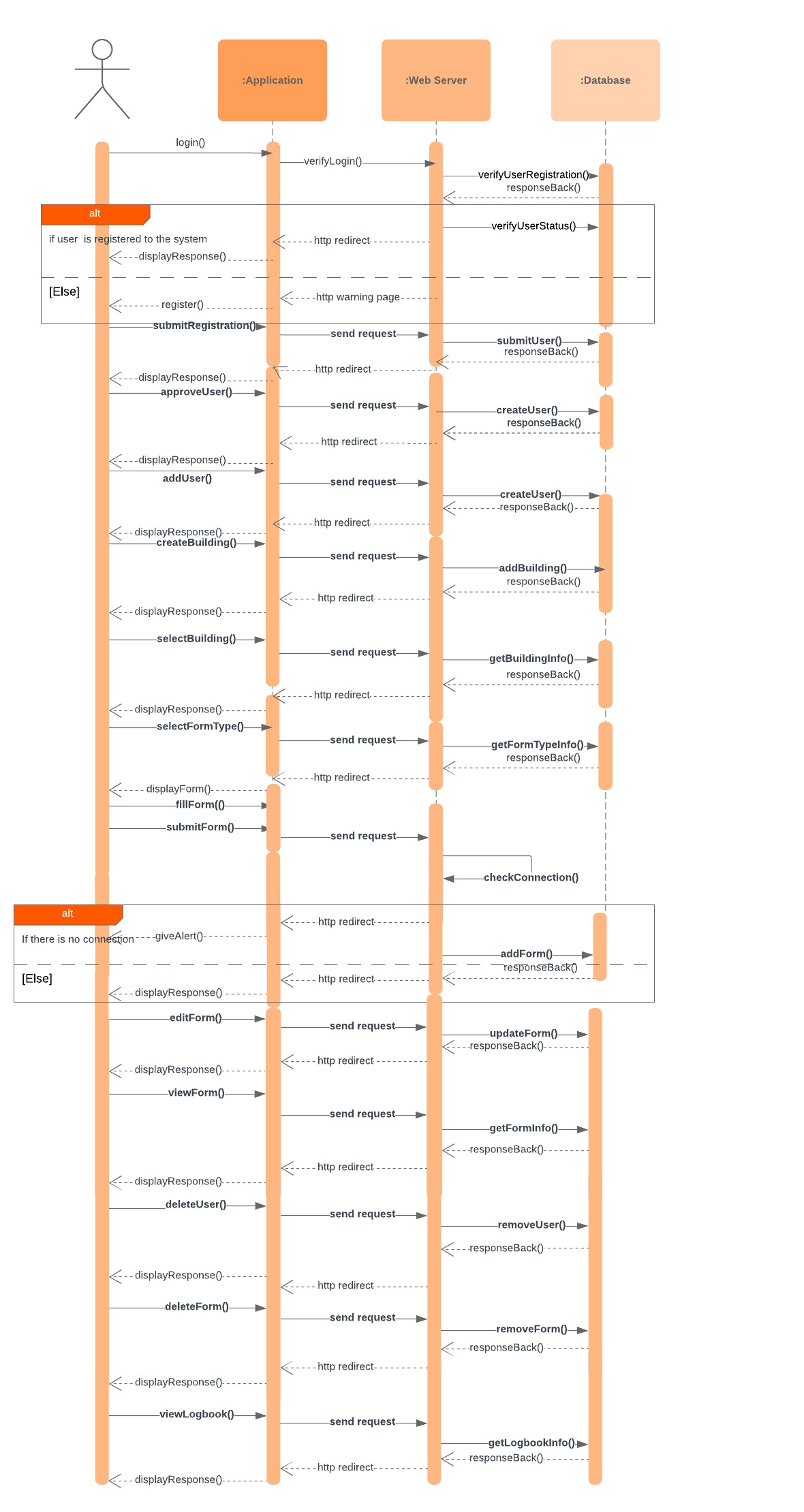
The project is database backed website project. The system is accessible through any of the devices and on any other Operating System. The system will run on Server and whenever user wants to use it he/she needs an access to the Internet. The application has designed based on user needs and aimed to come out with user friendly website. Users who can use a computer and website also will be able to use this software product easily. In addition to these functionalities, the security of website and work process will be handled with giving a different level of privilege to each kind of users.

1. **SYSTEM ARCHITECTURE**
   1. **Architectural Design**

****

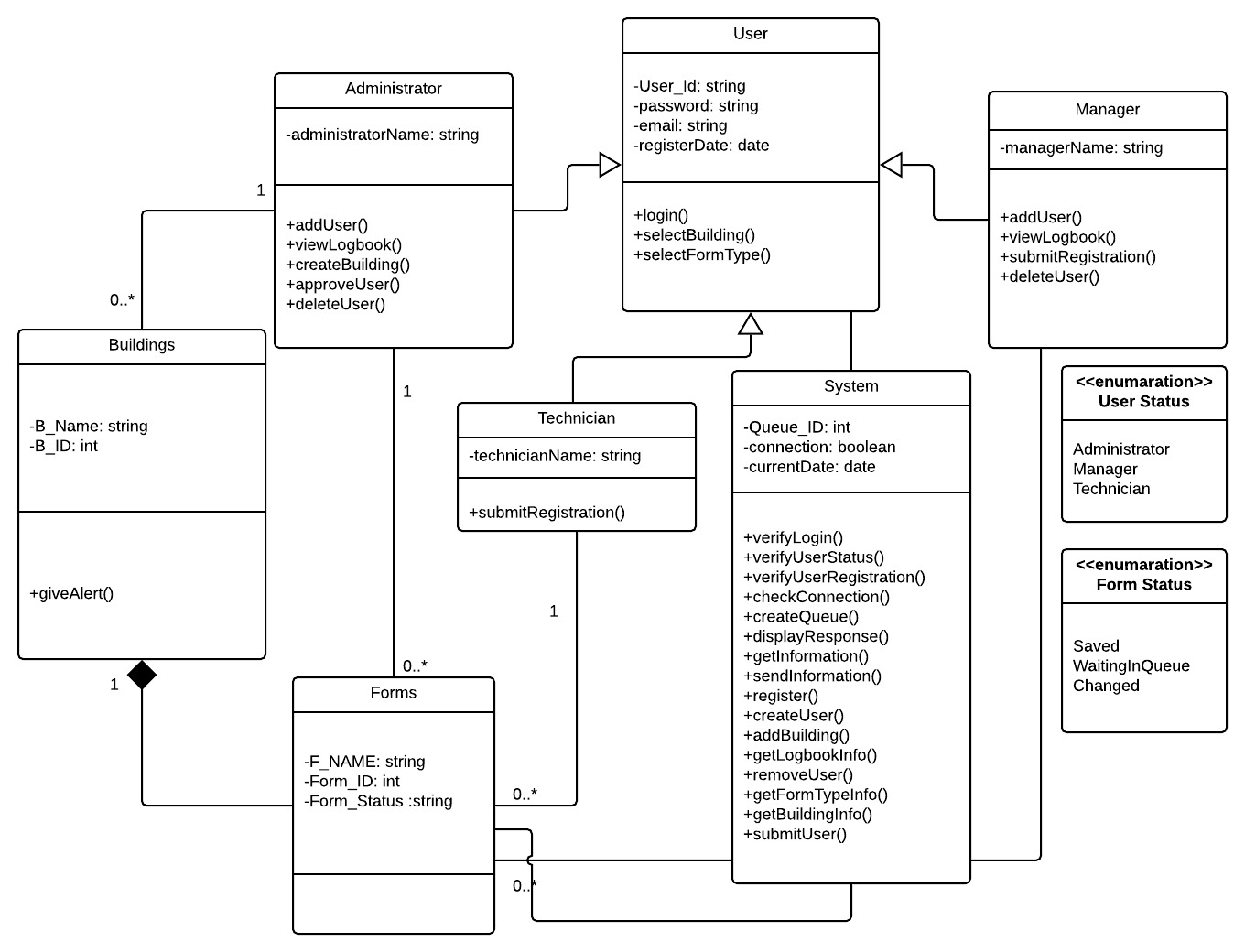
The flow diagram above shows;

* + The functionalities of the system based on user authorization.
  + Workflow of the system.
  + System control from backend.
  1. **Decomposition Description**
     + - **Sequence Diagram**

****

The sequence diagram above shows;

* + The relationship between the objects of the system.
  + The process of the functionalities from the backend part of the system.

****

The Class diagram above shows;

* + Description of the static structure for the system.
  + The relationship between classes.
  + The functionality of each classes.
  1. **Design Rationale**

The design rationale can be defined as below;

1. User Authorization: Back and front end structure of the system mainly should be focused on the user discrimination. Each user has his/her own capability over the workflow. For instance, if the user doesn’t have authorization on doing some works, this user shouldn’t be able to see related screen or links. Even if he/she wants to do something they are not authorized the system should put restriction and give a warning to the user.
2. Workflow: The structure and design of the project should be parallel with the workflow. For example, as we know that main job for user is filling the forms and record them for possible issues and future records. Every form is included a specific building so that it is important if the users be able to choose the building before the choosing the form type they want to fill out. Similar constraints, which come out base on workflow, can be considered during the time for designing.
3. Network: Software Development Project aimed to supply database backed website to the customer which needs network connection to serve properly. The locations where user wants to fill the form probably will have a connection problem. For this reason, the design of the project is decided with considering connection problem aswell.
4. **DATA DESIGN**

* 1. **Data Description**

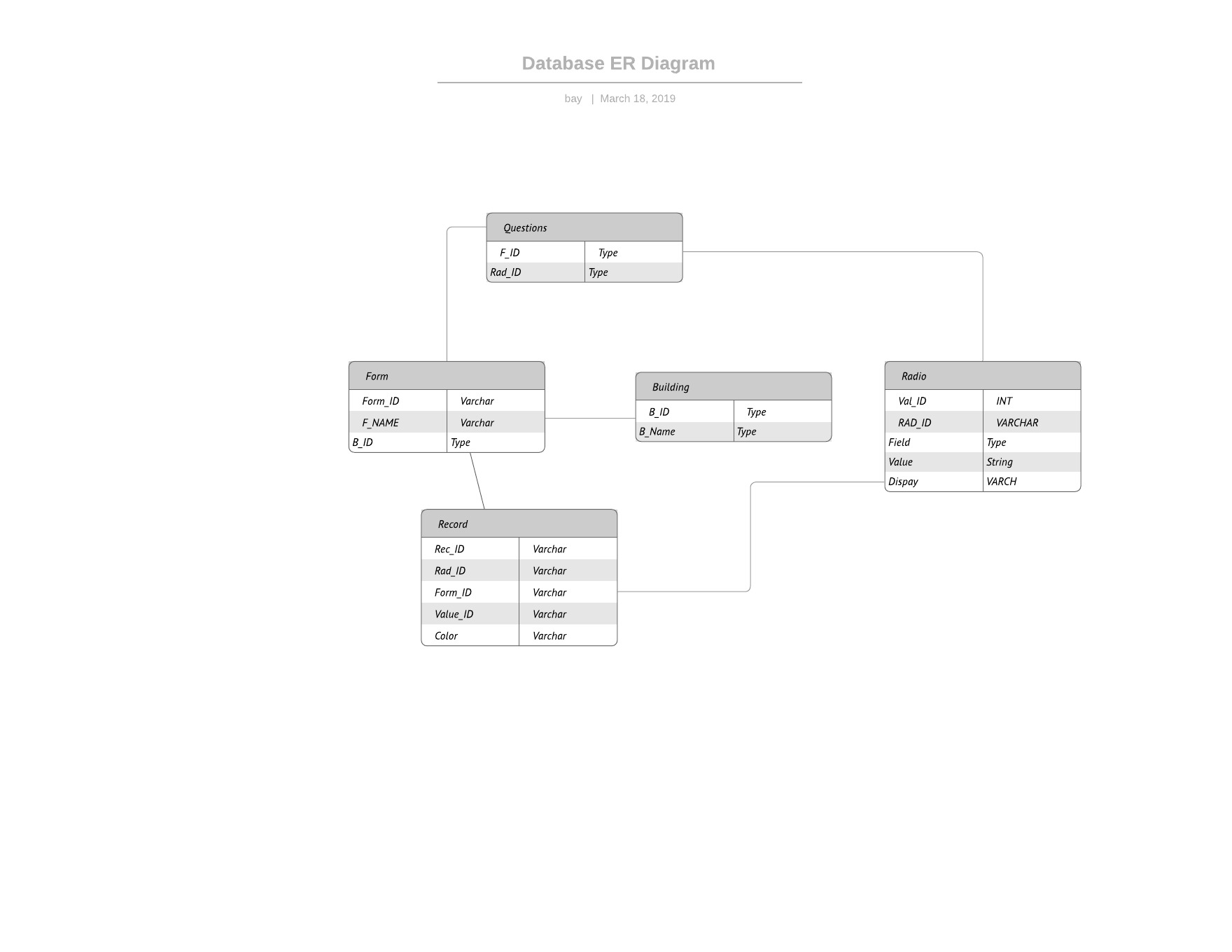
XAMPP and PhpMyAdmin Control Panel database communicate with the database which is installed locally on the server.

* 1. **Data Dictionary**

The table at the below is consisting of the list of the objects, their attributes, their methods and parameters of the methods based on class diagram which is given in the section 3.2 .

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Object Attributes** | **Object Methods** | **Parameters of Methods** |
| User | User\_ID :string  password: string  email: string  registerDate: date | login | (User\_ID, password) |
| selectBuilding | ( B\_ID ) |
| selectFormType | ( Form\_ID) |
| Manager | managerName: string | addUser | (U\_Name, password,email) |
| viewLogbook | (date, User\_ID, B\_ID , Form\_ID ) |
| submitRegistration | (Request\_ID) |
| deleteUser | (U\_Name, password,email) |
| Administrator | administratorName: string | addUser | ( U\_Name , password,email) |
| viewLogbook | ( date, User\_ID, B\_ID , Form\_ID ) |
| createBuilding | (B\_Name, B\_ID ) |
| approveUser | (User\_ID,email,password) |
| deleteUser | ( U\_Name , password,email) |
| Technician | technicianName: string | submitRegistration | (Request\_ID) |
| Buildings | -B\_Name: string  -B\_ID: int | giveAlert | ( Form\_ID , B\_ID) |
| Forms | F\_Name: string  Form\_ID: int  F\_Status: string | editForm | ( Form\_ID B\_ID) |
| deleteForm | ( Form\_ID , B\_ID) |
| viewForm | ( Form\_ID , B\_ID |
| displayForm | ( Form\_ID , B\_ID) |
| fillForm | ( Form\_ID , B\_ID) |
| submitForm | ( Form\_ID , B\_ID) |
| removeForm | ( F\_NAME , B\_ID) |
| System | Queue\_ID: int  connection: boolean currentDate: date | verifyLogin | (User\_ID, password) |
| verifyUserStatus | (User\_ID, User\_Status) |
| verifyUserRegistration | (User\_ID, User\_Status) |
| checkConnection | (connection) |
| createQueue | ( Form\_ID ,date) |
| displayResponse | () |
| getInformation | () |
| sendInformation | () |
| register | (User\_ID,email,password) |
| createUser | ( User\_ID,email,password) |
| addBuilding | (B\_Name, User\_ID) |
| getLogbookInfo | (date, B\_ID , Form\_ID) |
| removeUser | (User\_ID,email,password) |
| getFormTypeInfo | ( Form\_ID ) |
| getBuildingInfo | ( B\_ID ) |
| submitUser | (User\_ID,email,password) |

The diagram below shows the relationship and design of the tables in database.

****

|  |
| --- |
| login(): |
| selectBuilding(): |
| selectFormType(): |
| addUser(): |
| viewLogbook(): |
| submitRegistration(): |
| deleteUser(): |
| createBuilding(): |
| approveUser(): |
| giveAlert(): |
| editForm(): |
| deleteForm(): |
| viewForm(): |
| displayForm(): |
| fillForm(): |
| submitForm(): |
| removeForm(): |
| verifyLogin(): |
| verifyUserStatus(): |
| verifyUserRegistration(): |
| checkConnection(): |
| createQueue(): |
| displayResponse(): |
| getInformation(): |
| sendInformation(): |
| Register(): |
| createUser(): This function sends user information to database to add user in to the database. |
| addBuilding(): |
| getLogbookInfo(): |
| removeUser(): |
| getFormTypeInfo(): |
| getBuildingInfo(): |
| submitUser(): |

1. **COMPONENT DESIGN**
2. **HUMAN INTERFACE DESIGN**
   1. **Overview of User Interface**

The functionality of the system from user’s perspective listed as below:

* + **Administrator**
    - Administrator can use login screen for login to the system.
    - Administrator can use Welcome Page for being able to do whatever he/she wants like Choose a Building, Add/Delete User, Create Building, Create New Type of Form, View Logbook.
    - Administrator can choose a building from Building List Page.
    - Administrator can choose a new form type from the Form List Page.
    - Administrator can fill/submit/edit form from the Form Page.
    - Administrator can Create new building from Create Building Page.
    - Administrator can Create new type of form from the Create New Form Page.
    - Administrator can see logbook Form the View Logbook Page.
    - Administrator can list saved list from Building Main Page.
  + **Manager**
    - Manager can use login screen for login to the system.
    - Manager can use Welcome Page for being able to do whatever he/she wants like Choose a Building, Add/Delete User, View Logbook.
    - Manager can choose a building from Building List Page.
    - Manager can choose a new form type from the Form List Page.
    - Manager can fill/submit/edit form from the Form Page.
    - Manager can see logbook Form the View Logbook page.
    - Administrator can list saved list from Building Main Page.
  + **Technician**
    - Technician can use login screen for login to the system.
    - Technician can use Welcome Page for being able to Choose a Building.
    - Technician can choose a building from Building List Page.
    - Technician can choose a new form type from the Form List Page.
    - Technician can submit form from the Form Page.
    - Administrator can list saved list from Building Main Page.
  1. **Screen Images**

Login page is as below;

****

Registration page is as below;

**A screenshot of a cell phone

Description automatically generated**

Add User page is as below;

**A screenshot of a computer screen

Description automatically generated**

Assign Manager or Technician page is as below;

**A screenshot of a computer screen

Description automatically generated**

List of Building page is as below;

**A screen shot of a computer

Description automatically generated**

Home page for admin is as below;

**A picture containing monitor, black, indoor, screen

Description automatically generated**

* 1. **Screen Objects and Actions**

There are three dashboards in the system and the general structure of the user interface is as follows: There exist 2 main tabs on the top of the dashboard 1 is the home button which would get the user to the home button of the user’s page. And the second is the sign-out here the session would be ended and the user who is signed in is signed out of the system.

When the Admin logs in he can see the option of creating the new building. He can create multiple building and the buildings would be displayed on the same page in the synchronized manner.

When the Admin in homepage, he would have following options such as Add User, Delete User, Create Form , Create Building, View Logbook. Add User and Delete User would allow to create Technician or Manager. Create form would allow the Admin to create the form the way he wants to collect the data. And View Logbook would allow the admin to view the data uploaded by the technician. Once he/she choose building he/she will be show the list of the buildings. Then whatever building he/she is going to choose he would have options Fill the Form, View Form. If he/she choose Fill the Form he would have a page with the list of the form to choose and fill it out with Submit button. If he/she choose the View Form he/she would be able to see al the forms have been submitted about this building with Edit button.

When the Manager in homepage, he would have following options such as Add User, Delete User, View Logbook. Add User and Delete User would allow to create Technician. Create form would allow the Manager to create the form the way he wants to collect the data. And View Logbook would allow the admin to view the data uploaded by the technician. Once he/she choose building he/she will be show the list of the buildings. Then whatever building he/she is going to choose he would have options Fill the Form, View Form. If he/she choose Fill the Form he would have a page with the list of the form to choose and fill it out with Submit button. If he/she choose the View Form he/she would be able to see al the forms have been submitted about this building with Edit button.

When the Technician logs in the system he/she would see Welcome Home Page with the link called choose building. When he/she click-on the link he/she will see the list of Building Page to choose whatever building he/she wants. Once he/she choose the building he/she will see list of the forms. After choosing the type of the form the related Form Page will show up to the Technician to fill it up with the Submit button.

1. **REQUIREMENT MATRIX**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Module Name** | **Applicable Roles** | **Description** |
| **4.1** | **Login Feature** | **Administrator, Manager,Technician** | **Administrator, Manager,Technician login to system through here.** |
| **4.2** | **View List of Building** | **Administrator, Manager,Technician** | **Administrator, Manager,Technician can see list of the buildings through here.** |
| **4.3** | **Assigning Roles** | **Administrator, Manager** | **Administrator, Manager can add user through here.** |
| **4.4** | **Create a New Building** | **Administrator** | **Administrator can create a new building through here.** |
| **4.5** | **View Data** | **Administrator, Manager, Technician** | **Administrator, Manager, Technician can view data from here** |
| **4.6** | **Edit Form Content** | **Administrator, Manager** | **Administrator and Manager can edit forms through here.** |
| **4.7** | **Create a New Form** | **Administrator** | **Administrator can create new type of form through here** |
| **4.8** | **Fill the Data** | **Administrator,Manager,Technician** | **Administrator,Manager,Technician can fill the form through here** |
| **4.9** | **Create and Export the Report** | **Administrator,Manager,Technician** | **Administrator,Manager,Technician can create report and export them into the their devices through here.** |
| **4.10** | **Keeping a Record in a Cache** | **System** | **System Can keep a recored in a cache if there is no network connection through here.** |
| **4.11** | **Keeping a Logbook** | **System** | **System can keep logbook through here.** |
| **4.12** | **Determine Repeated Problems** | **System** | **System can determine the problems those are repated through here.** |

1. **APPENDICES**

This section is optional.