****

**COMSATS University Islamabad,**

**Park Road, Chak Shehzad, Islamabad Pakistan.**

Software Design Description  
(SDD DOCUMENT)

for

**Smart Construction Manager**  
Version 1.0

***By***

**Talha Ejaz CIIT/SP18-BCS-161/ISB**

**Waleed Butt CIIT/SP18-BCS-170/ISB**

***Supervisor*Dr. Ashfaq Hussain Farooqi**

*Bachelor of Science in Computer Science (2018-2022)*

**Table of Contents**

Revision History ii

Application Evaluation History iii

1. Introduction 1

1.1 Scope and Modules 1

2. Design Methodology and Software Process Model 1

2.1 Design Methodology 1

2.2 Process Model 2

3. System Overview 2

3.1 Architectural Design 2

3.2 Process Flow/Representation 2

4. Design Models 3

5. Data Design 4

5.1 Data Dictionary 4

6. Algorithm and Implementation 5

7. Software Requirement Traceability Matrix 6

8. Human Interface Design 7

8.1 Screen Images 7

8.2 Screen Objects and Action 7

9. Appendix I 8

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Application Evaluation History

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
| ‘Modules are clear and Well defined. Can Proceed for next step’ – Dr Basit Raza | N/A |
| ‘Project idea is ok. All the modules are properly explained ‘– Mr. Zulfiqar Ali | N/A |
| ‘Add some major modules other than management systems’ – Tehseen Riaz Abbasi | Modules Added |
| ‘Replace the “targeted project” with your project name in tables’ caption.’ – Mr. Tehseen Riaz Abbasi | “targeted project” replaced with project name in tables’ caption. |
| ‘There is a difference between functionality and advantages, you have written too functionality of the system in advantages section, please correct it.’ – Mr. Tehseen Riaz Abbasi | Functionality and advantages section updated |
| ‘The modules are not written with logical and technical flow, rather students tried to add modules without proper flow, understanding the system needs and boundaries.’ – Mr. Tehseen Riaz Abbasi | Modules added with the flow |
| ‘Modules needs to refine, redefine, and improved.’ – Mr. Tehseen Riaz Abbasi | Modules refined, redefined, and improved. |
| ‘Revise concept section. Explain it.’ – Mr. Tehseen Riaz Abbasi | Concept section revised. |
| ‘Milestones for the Gantt chart needs revision, so revise your Gantt chart. Format is also not correct.’ – Mr. Tehseen Riaz Abbasi | Gantt Chart revised |
| ‘Mockup images quality is not good, redesign them.’ – Mr. Tehseen Riaz Abbasi | Mockup images quality improved and redesigned. |
| ‘The screenshot of plagiarism report is missing.’ – Mr. Tehseen Riaz Abbasi | Plagiarism report added |
| ‘Limited tabular use case Add more tabular use cases.’ – Mr. Tehseen Riaz Abbasi | Tabular use cases added. |
| ‘Revise the tabular use cases and FR in proper format with all required fields to be filled as guided w.r.t to updated modules. – Mr. Tehseen Riaz Abbasi | FR tables revised |
| ‘Revise and quantify the non-functional requirements properly. Add more NFR like scalability etc. – Mr. Tehseen Riaz Abbasi | More NFRs added |
| ‘Milestones for the Gantt chart needs revision, so revise your Gantt chart. Format is also not correct. – Mr. Tehseen Riaz Abbasi | Gantt chart revised |
|  |  |

Supervised by

Dr. Ashfaq Hussain Farooqi

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 

# Introduction

Smart Construction Manager is a One Window Solution for the construction process of housing in the state. Keeping in view the incumbent government’s housing and construction policies & packages to minimize the time & effort for the construction process and ensure transparency. This solution also addresses unexpected turn of events in the fiscal policies caused by the global pandemic (COVID-19). It also ensures the safety of people regarding the SOPs. Smart Construction Manager is an all-rounder application addressing basic house construction needs.

## Scope and Modules

System will provide a one window application and act as agent of communication between construction companies, user, and government bodies. The user will provide the required information to the system the system will authenticate the user. User will submit request to respective organization (depending upon the location of project) & The construction company. The authorities and the industries will accept the request. Design and Layout will be sent by user. After review authorities and industries will generate the response.

### Modules

**First module** buys and sell, user can buy land and other construction equipment posted by other users for sale. **Second module** deals with user searching and provides functionality to search by talking. **Third module** is user chatting with other users. **Fourth module** shows user recommendations on home page. **Fifth module** helps user decide the map for construction site. And **sixth module** will be for designing and decorating house with AR. **Seventh module** deals with the user sending application requests to government bodies for various procedures. **Eight module** provides admin portal to government bodies to manage user requests.

# Design Methodology and Software Process Model

Design Methodology

The design methodology followed will be **Component Based Software Engineering (CBSE)**. CBSE emphasis the design and construction of computer based system using reusable software “components”. We adopted this methodology due to the failure of object-oriented development to support effective reuse. Single object classes are too detailed and specific. CBSE embodies the “buy, don’t built” philosophy. Our application uses MERN stack on web and React Native on mobile. React in both cases works efficiently with component based design.

CBSE increases quality, especially evolvability and maintainability. CBSE increases productivity. CBSE shortens development time. CBSE is easy to assemble and less costly to build the system constructed from discrete parts.

Process Model

**Modified Waterfall Software Process Model** will be followed in the project development. Modified Waterfall is easy to understand and easy to use. It Provides structure to inexperienced staff like us. Our milestones are well understood which suites Modified Waterfall. It sets requirements stability and is good for management control (plan, staff, track). The feedback system helps us great in going back to any previous stage.

# System Overview

Smart Construction Manager is a system for two types of users, general public and government. General public is further categorized into two sub users, buyers and sellers. Buyers are the one who will be constructing homes. They require several facilities during construction like labors, archtitectures, contractors and many more. These are the second sub category sellers. They sell their products or services. SCM will provide both of them a platform to share and explore their requirements.

SCM will further provide the general public more features to help them in construction like 3D modeling the design of house. Virtually designing the house using AR after construction.

The Second types of users i.e. government will also have two sub users. Admin and Employees. Admin will have the full control of system at their end. Admin will create accounts for employees which will interact with general public.

## Architectural Design

SCM will follow 3-tier structure. The client side, Web/IOS/Android in React and React native, will communicate with backend, handled by ExpressJS. The backend will then interact with database (MongoDB) and respond to client requests. The REST principles will be followed while making APIs. Following is the picture showing the working of Smart Construction Manager Backend.

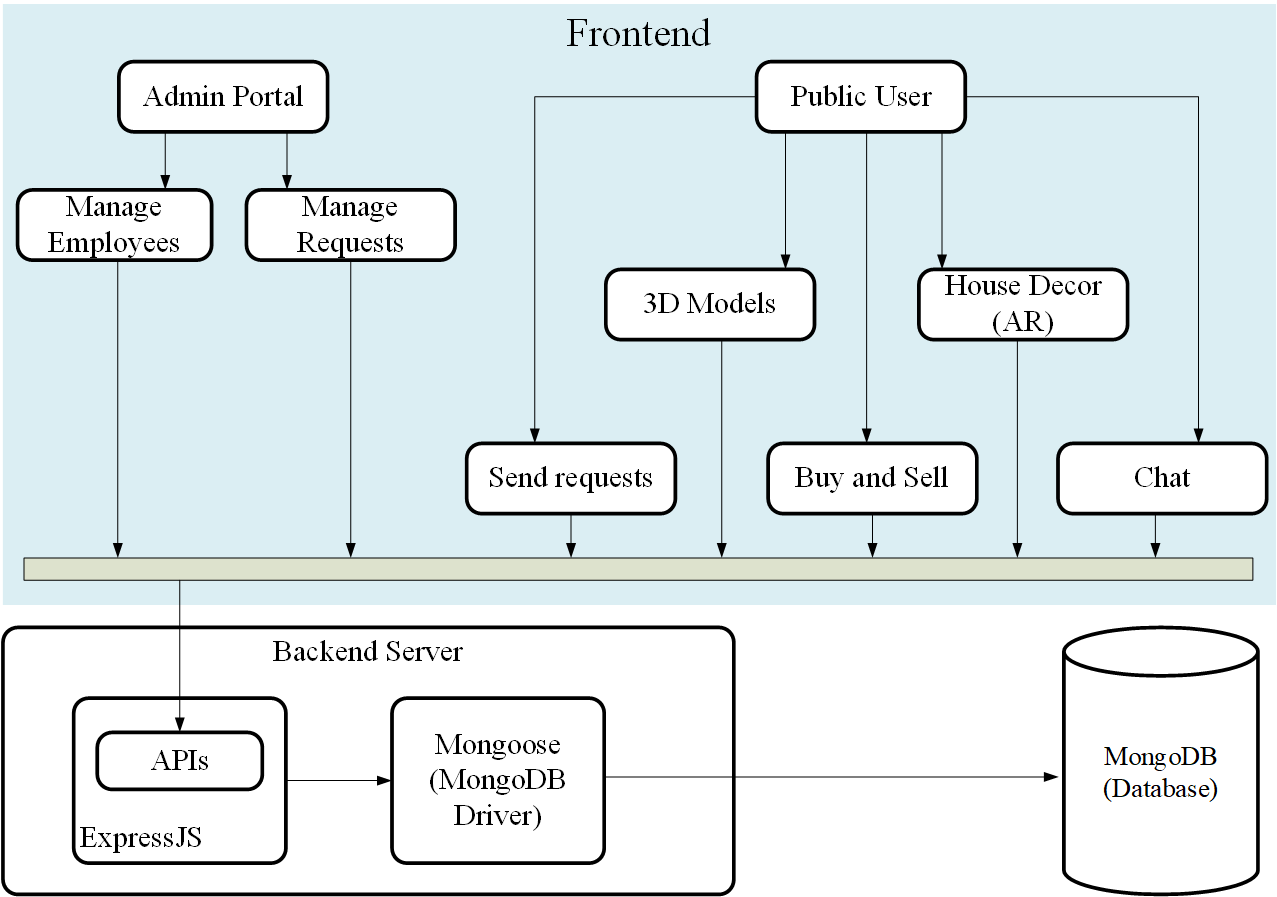


Figure 1: Architectural Design of SCM

## Process Flow/Representation

Folllowing section depicts the major process flow of Smart Construction Manager.

### Activity Diagrams

The Process flow of SCM is represented in activity diagrams. These diagrams show the general behavior of the application.

#### Basic Activity

Following Image shows the basic activities, login sign up and looking for help etc.

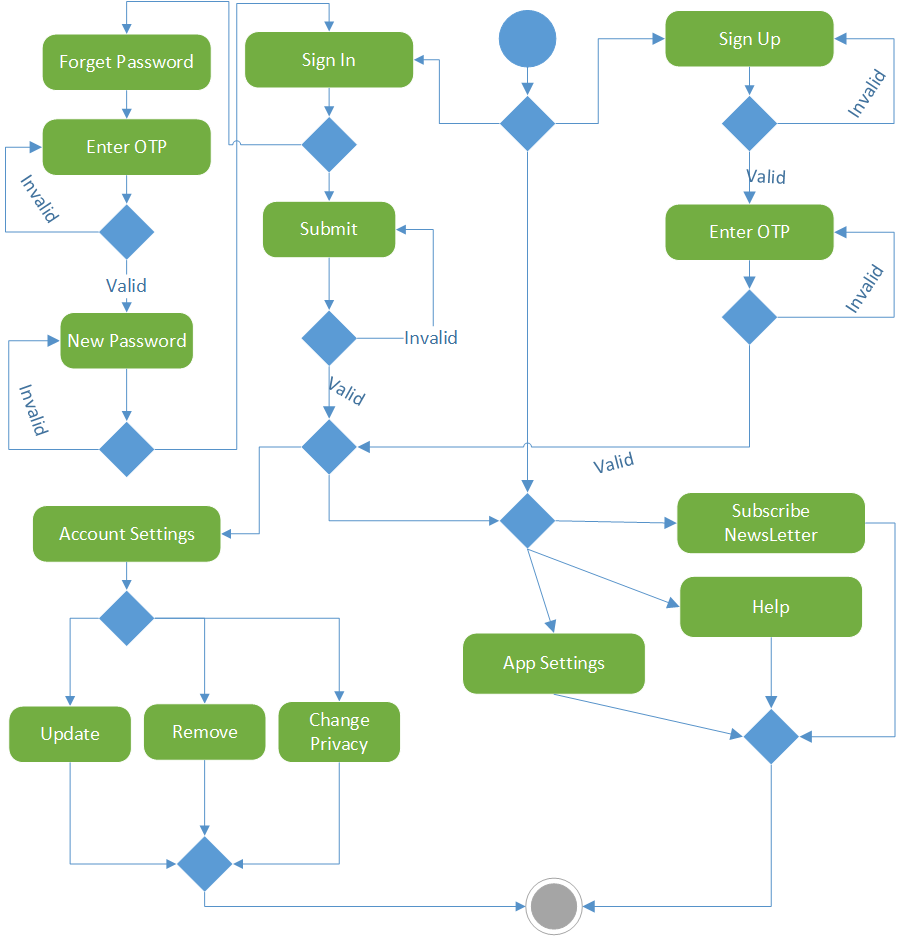


Figure 2: Basic Activities

#### Chatting

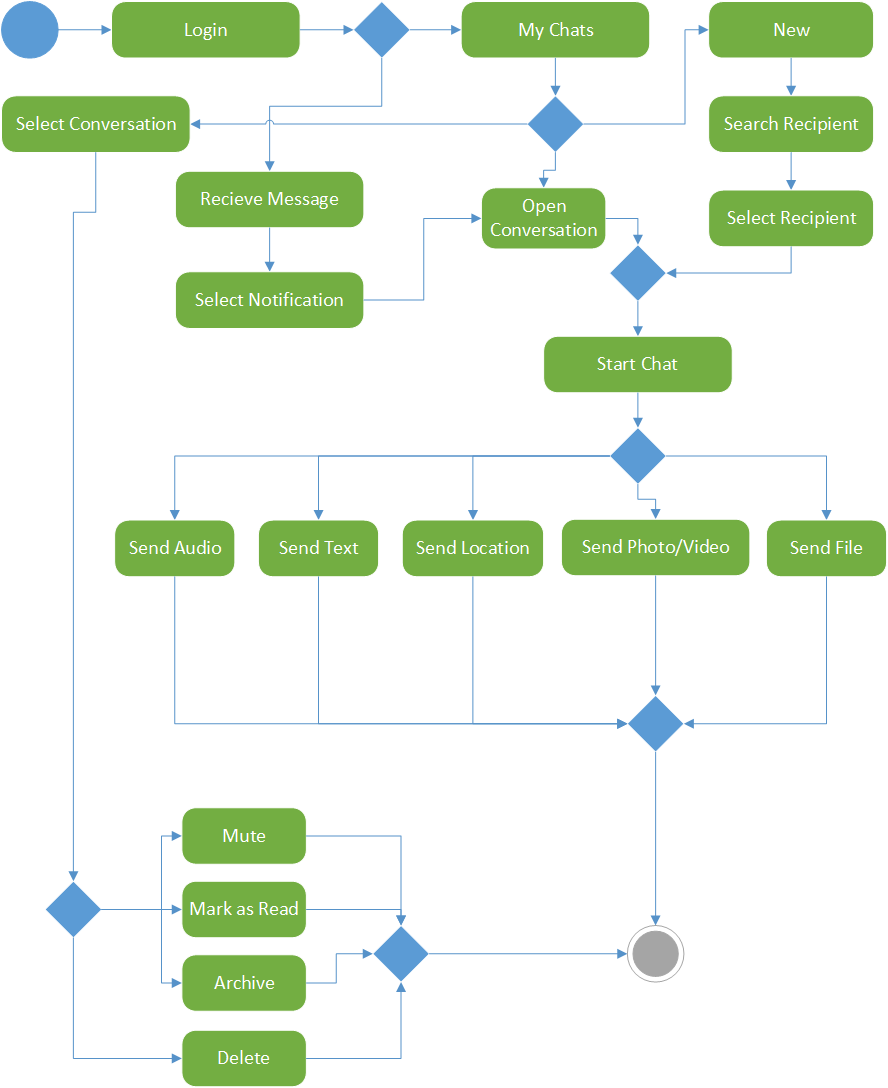


Figure 3: Conversation

#### Search

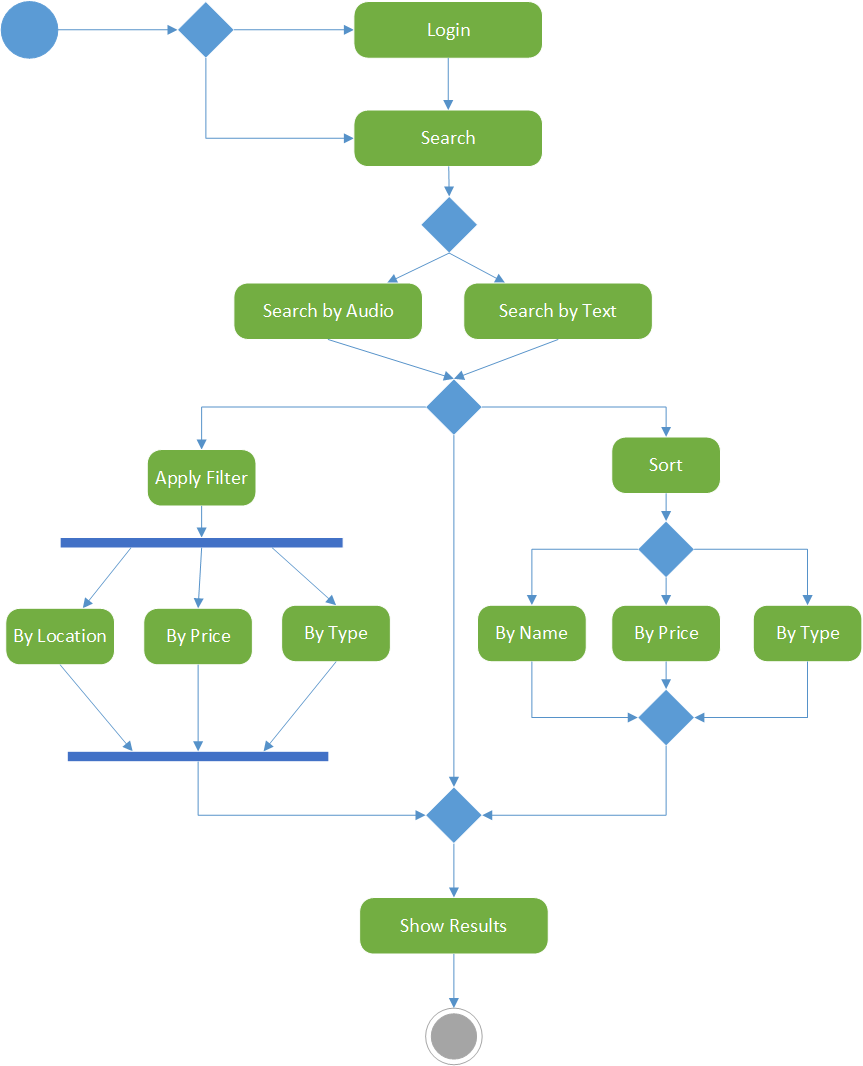


Figure 4: Searching

#### 3D Models and House Décor

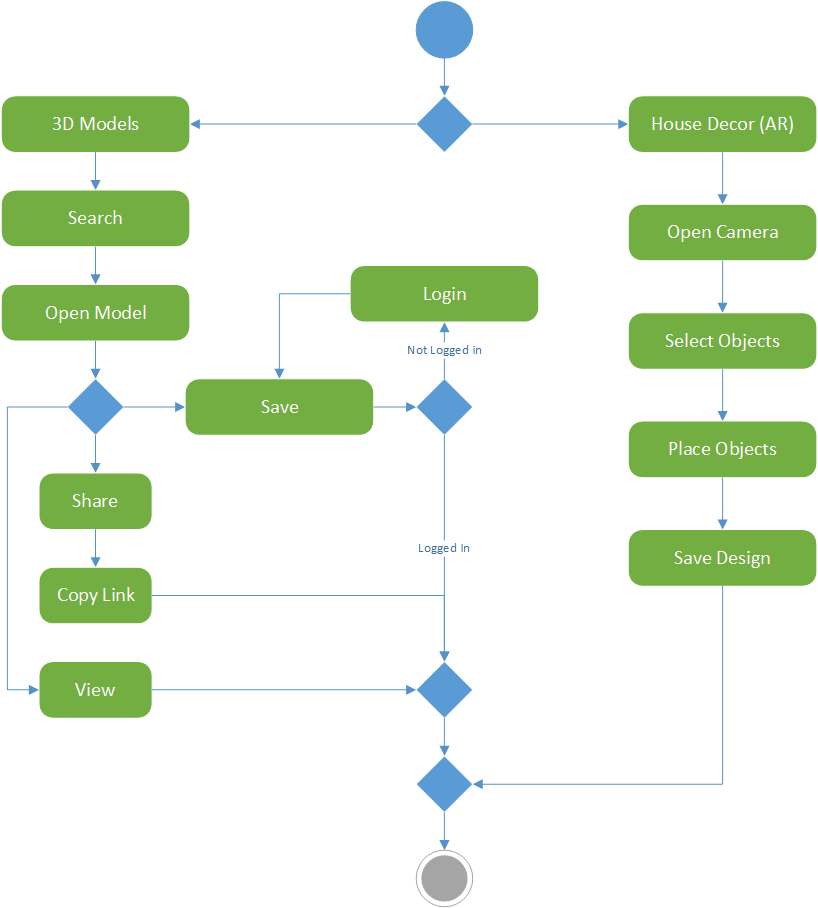


Figure 5: 3D Modeling and AR

#### Managing Ads

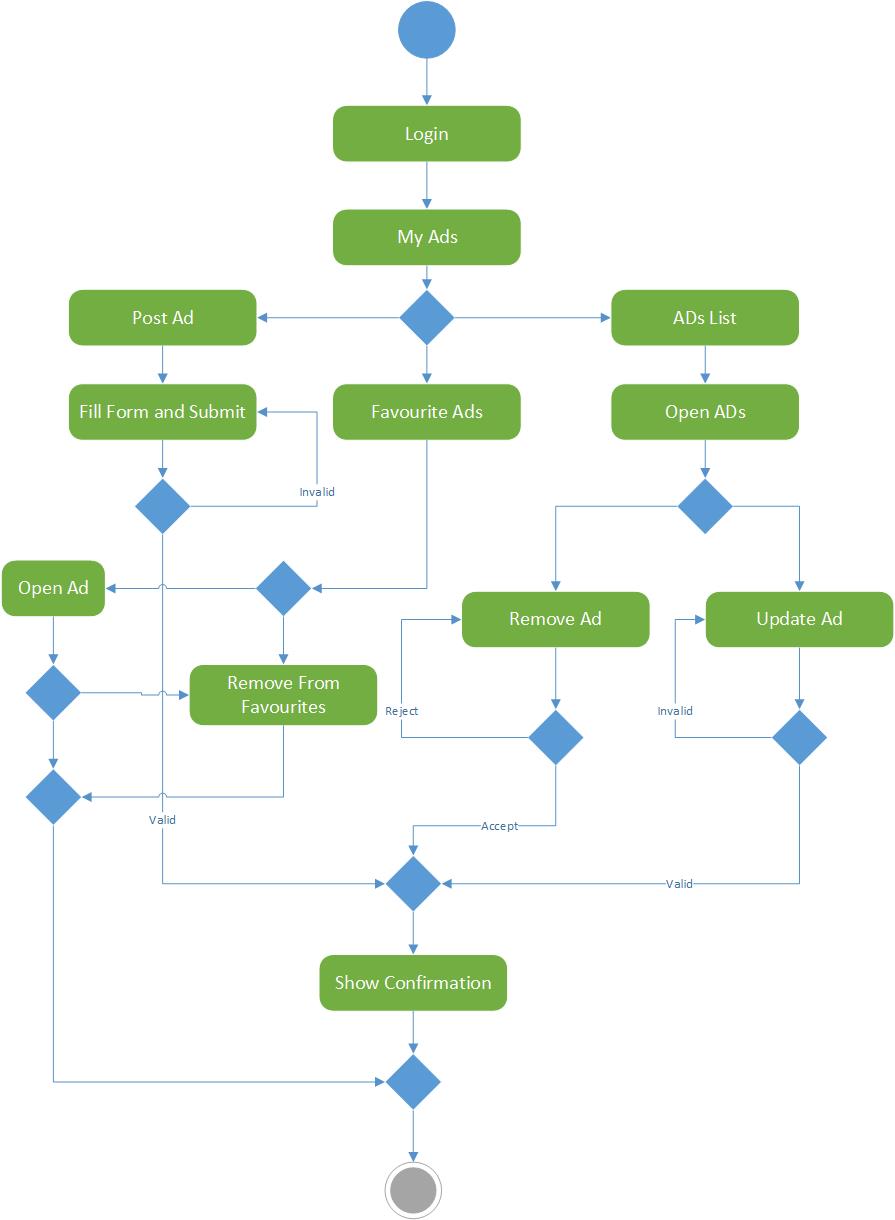


Figure 6: Managing ADs

#### User Interaction with Government

The image belows depicts the activities a user can perform to interact with government bodies.

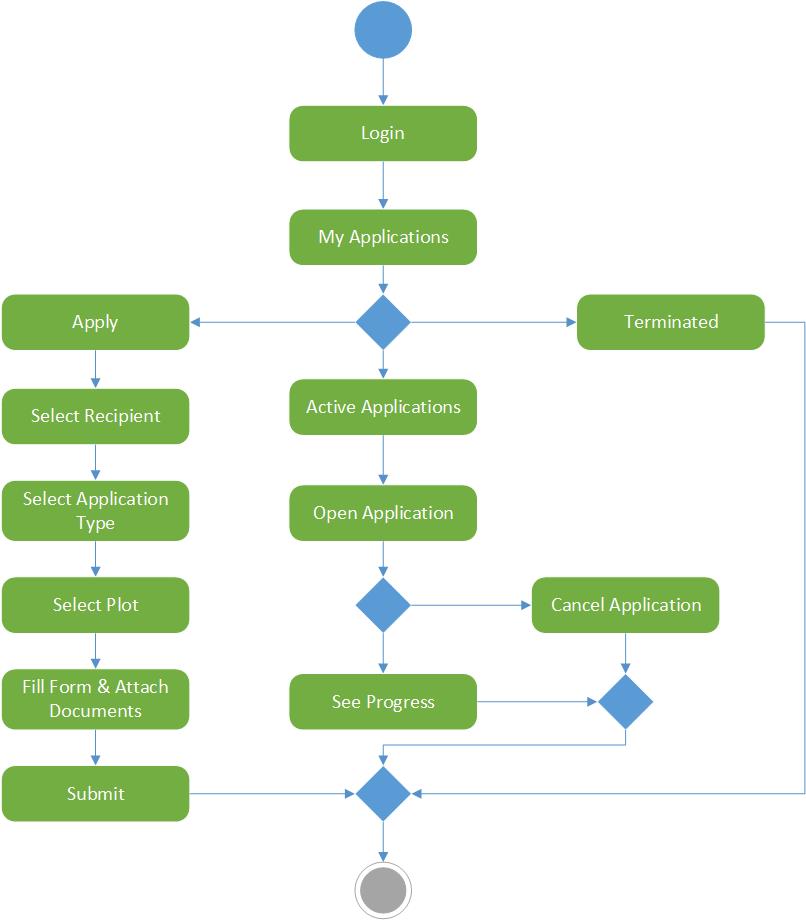


Figure 7: Interaction with Government Bodies

#### Admin Portal.

This section shows the activities that will take place at the government side of the application. Two sub actors will be using the application. The admin of organization or employee of the organization. Admin will have the ability to create and manage employees. While employees will deal with application requests seny by user.

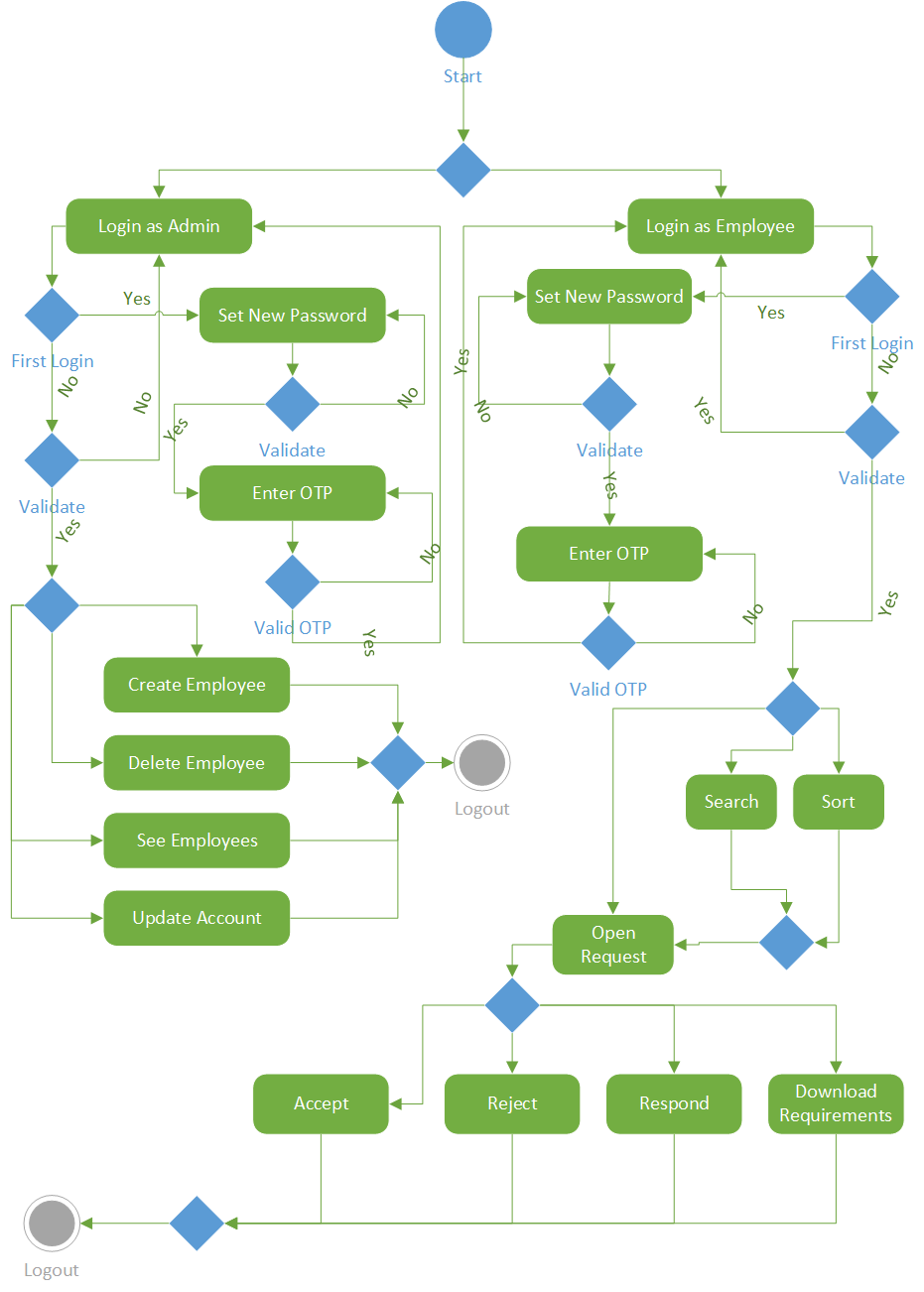


Figure 8: Admin Portal Activity

# Design Models

Following are the diagrams explaining the necessary functionalities of Smart Construction Manager.

## Class Diagram

Following is the class diagram for Smart Construction Manager. Explaining how different components of the system interact.

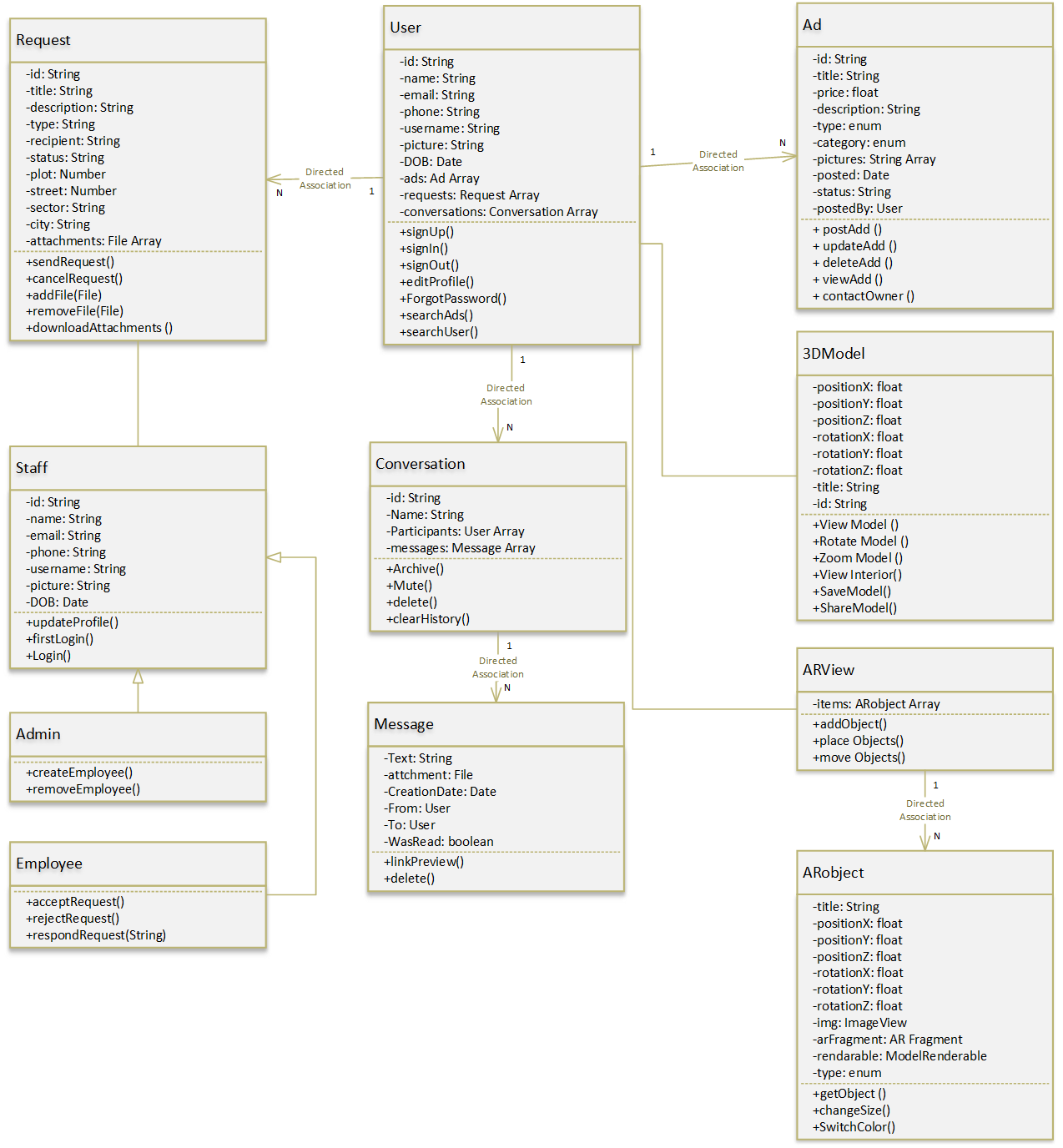


Figure 9: Class Diagram

## Sequence Diagram

### Interaction with Government Bodies

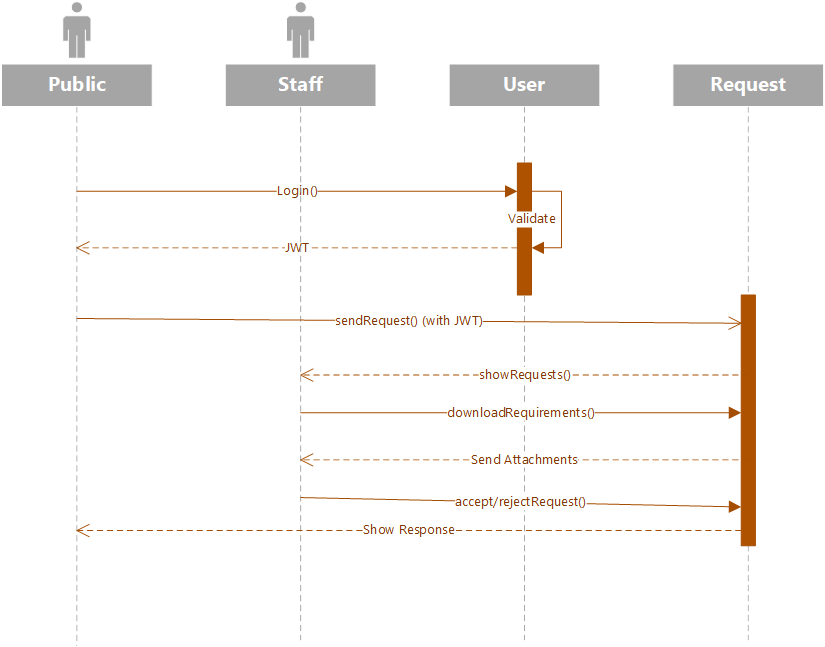


Figure 10: Sequence Diagram (Communication with Governemnt)

### Posting Ads

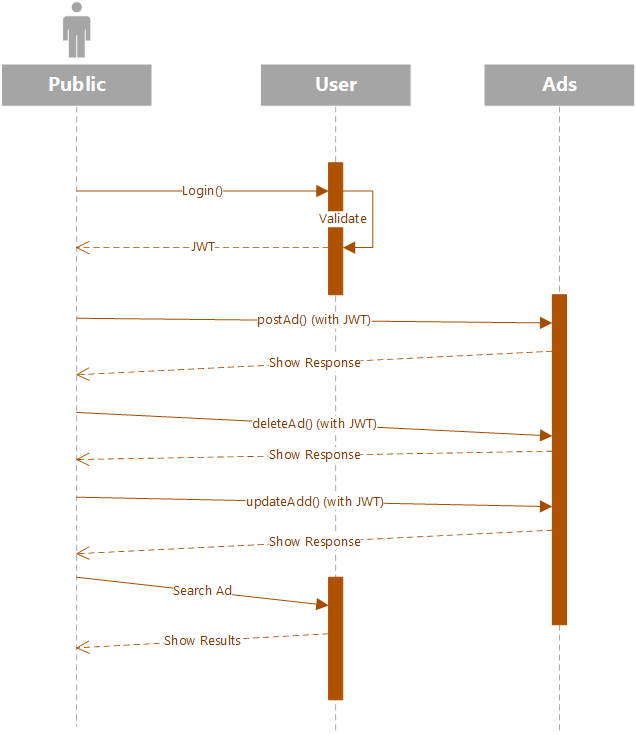


Figure 11: Sequence Diagram ( Ads)

### 3D Models

3D Models are sample house map to see before construction. These models can be shared via link with architects through chat.

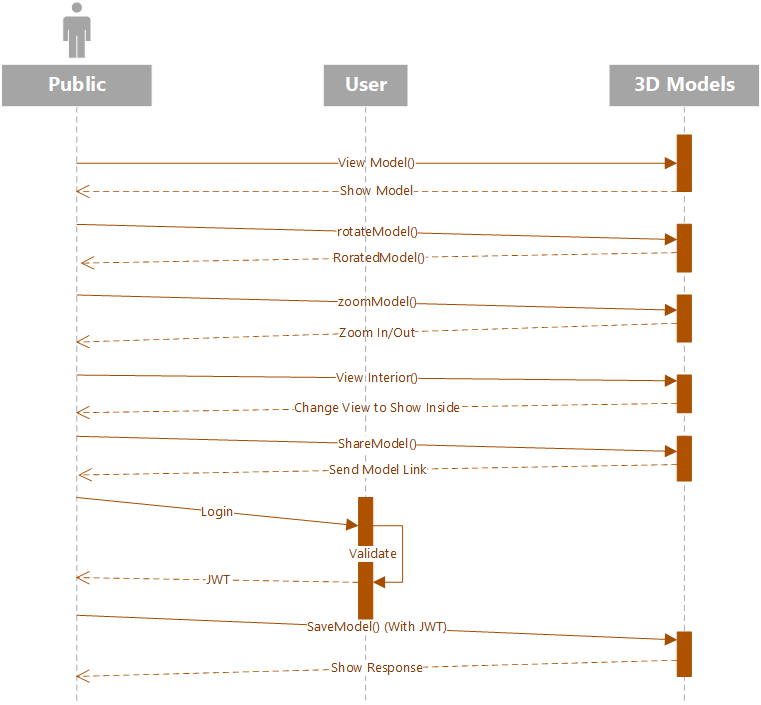


Figure 12: Sequence Diagram (3D Modeling)

### House Decor (AR)

AR helps user design house after construction by placing AR models in virtual world.

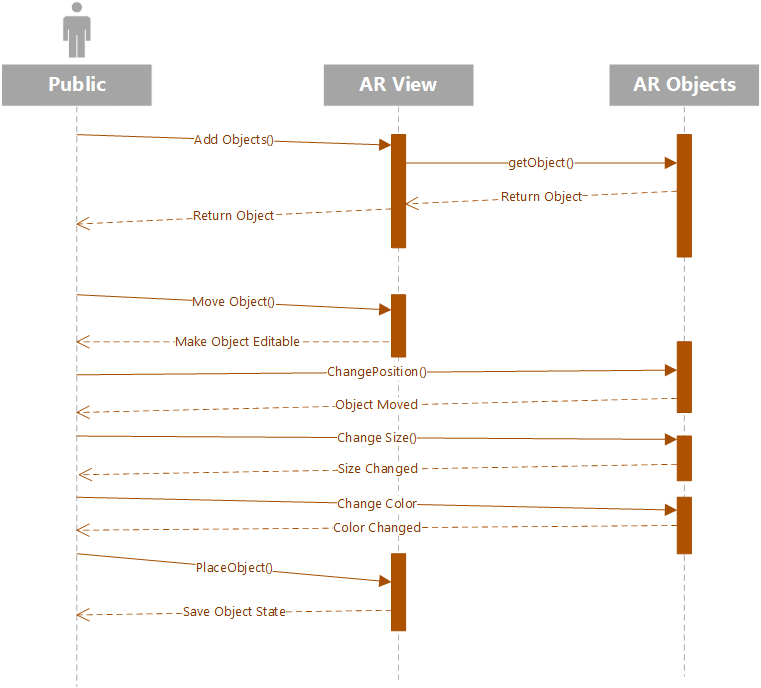


Figure 13: Sequence Diagram (AR)

### Admin Portal

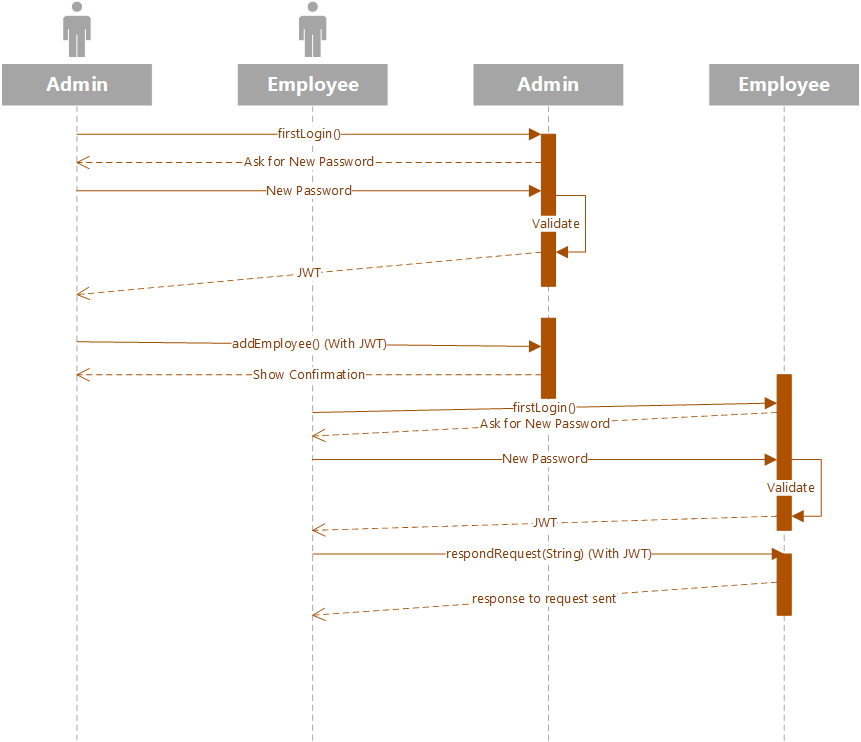


Figure 14: Sequenece Diagram (Admin Portal)

# Data Design

The data will be stored in MongoDB databases and file format in server. The paths to files will be stored in database as per requiremtns. The database will store textural data and files will be stored on the backend server. The files in the server will have random names and their directories as well. Each the purpose of directories will be judged only through databases. For example, the profile picture of user will be stored in backend server with some random generated name. This name along with random generated path will be stored in database.

## Data Dictionary

The application will follow JSON format for data storage as per required by mongoDB. External APIs used in the project will also be uisng JSON format for data.

userAccount = new schema ({

email: {

address: {type: String, default: ''},

privacy: {type: String, default: 'following', enum: ['public' , 'private']}

},

name: {type: String, default: '', required: true},

profileImage: { type: String, default: '' },

username: {type: String, required: true},

phone: {

number: {type: String, default: ''},

privacy: {type: String, default: 'following', enum: ['public', 'private']}

},

dob: { type: Date, default: Date.now },

password: { type: String , default: '', required: true},

userType: {type: String, default: 'user', enum: ['admin', 'employee', 'user']},

});

adSchema = new schema ({

title: {type: String, default: ''},

price: {type: String, default: ''},

description: {type: String, default: ''},

type: {

type: String,

default: 'Sell',

enum: ['Buy', 'Sell']

},

category: {

type: String,

default: 'Equipement',

enum: ['Property', 'Service', 'Equipement']

},

subcat: {type: String, default: ''},

status: {

type: String,

default: ' Pending ',

enum: ['Active', 'Pending', 'Removed', 'Hidden']

},

posted: {type: Date, default: Date.now},

user: {type: schema.Types.ObjectId, ref: 'userAccount'},

pictures: [{type: String, default: '', required: true}]

});

requestSchema = new schema ({

title: {type: String, default: ''},

description: {type: String, default: ''},

type: {

type: String,

default: 'Approval',

enum: ['Water', 'Electricity', 'Gas', 'Approval', 'Transfer']

},

recipient: {

type: String,

default: 'CDA',

enum: ['WAPDA', 'Sui', 'CDA']

},

status: {

type: String,

default: ‘Pending',

enum: ['Active', 'Accepted', 'Rejected', 'Pending']

},

plot: {type: String, default: '', required: true},

street: {type: String, default: '', required: true},

sector: {type: String, default: '', required: true},

city: {type: String, default: '', required: true},

user: {type: schema.Types.ObjectId, ref: 'userAccount'},

attachments: [{type: String, default: '', required: true}]

});

# Algorithm and Implementation

In this section, we take a closer look at what each component does in a more systematic way. Provide a summary of your algorithm for each function listed in procedural description language (PDL) or pseudo code.

If you gave an OO description, summarize each object member function for all the objects listed in PDL or pseudocode. Describe any local data when necessary

# Software Requirement Traceability Matrix

This section should contain a table that summarises how each software requirement has been met in this document. The tabular format permits one-to-one and one-to-many relationships to be shown.

Table 1 Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. Number** | **Ref. Item** | **Design Component** | **Component Items** |
| FR01 | Class Diagram | ClassName | FunctionName(s) |

# Human Interface Design

Describe the functionality of the system from the user’s perspective. Explain how the user will be able  to use  your system to complete  all the  expected  features and  the  feedback  information that will be displayed for the user.

## Screen Images

Display screenshots showing the interface from the user’s perspective. These can be hand-drawn, or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## Screen Objects and Action

A discussion of screen objects and actions associated with those objects

# Appendix I

SCM: Smart Construction Manager (Project Title)

CBSE: Component Based Software Engineering