# Group Members:

Syed Hussnain Tahir Sherazi

Muhammad Usman Shaheen

Ans

Qasim

**Introduction**

The software engineering landscape is evolving constantly, and currently, banking

solutions have to be more efficient, robust, and user-centric than ever before. Once

having recognized this need, therefore, it became necessary for the initiation of our

project to design and develop a comprehensive banking software system that meets

the current operational demand and at the same time sets the ground for the

technological changes which will be entering the banking sector. This captures the

process involved in developing this banking software system using Mendix—one of

the reputable low-code development platforms chosen to reap its rapid development

capabilities that will allow the development team to overcome their learning curve

with respect to traditional programming skills.

This report is meant to generate a detailed account of our travel through the various

stages in the software development lifecycle, from early planning and design, to later

implementation and testing. The core of our project has been to show, even by means

of a low-code approach, how development teams with the minimum amount of coding

experience could easily and effectively implement quite complex banking

functionalities, such as: account management, financial transactions, and security

compliance. This not only made the software development process easier but assured

the developed software to be more solid, secure, and it followed standards within the

respective industry. In this report, we try to cover the problems that we faced, the

solutions that we sought, and the overall success of the project in coming up with a

fully functional software system for banking that can be used to model future similar

activities

# Technology Used

## Mendix Low Code

Mendix is a popular low-code development platform that empowers businesses to rapidly build, deploy, and manage enterprise-grade applications with minimal coding expertise. Here's a general overview along with some key benefits:

Overview:

**Drag-and-drop interface**: Mendix offers an intuitive visual development environment where users can simply drag and drop components to design and build applications.

**Model-driven development**: It follows a model-driven approach, where developers design application logic, data models, and UI elements visually, abstracting away much of the underlying complexity of coding.

**Collaborative environment**: Mendix fosters collaboration among cross-functional teams including developers, business analysts, and IT professionals, enabling them to work together seamlessly throughout the development lifecycle.

**Cloud-native architecture**: Applications built on Mendix are inherently cloud-ready, allowing for easy deployment and scalability on various cloud platforms.

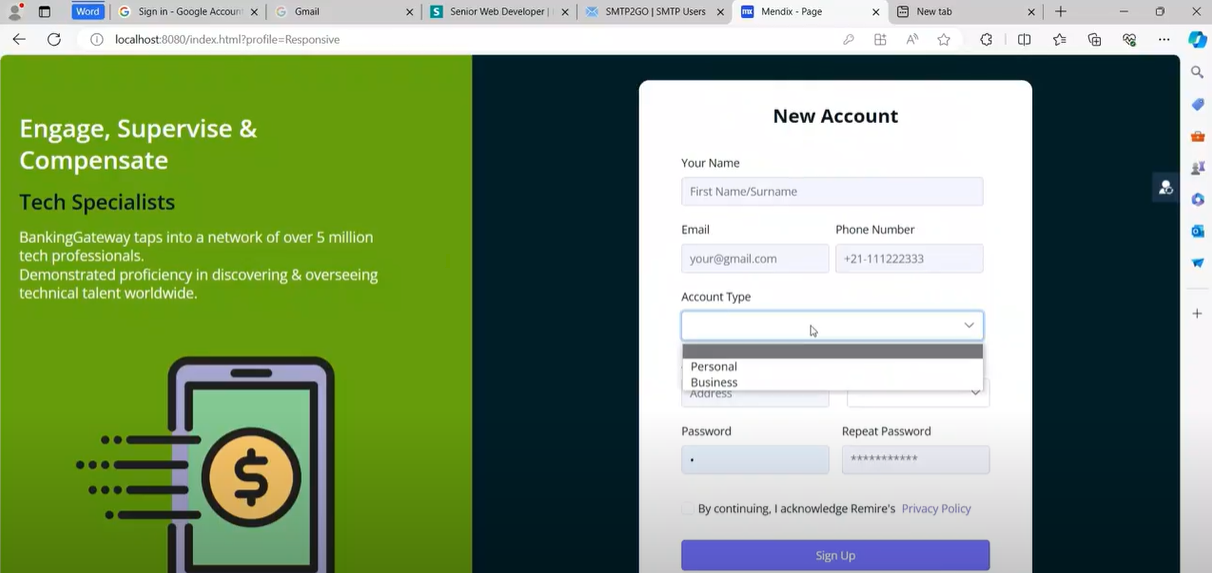
**App store ecosystem**: Mendix offers a marketplace where users can access pre-built components, connectors, and templates to accelerate development and extend the functionality of their applications.

**Integration capabilities**: It provides robust integration capabilities, allowing developers to seamlessly connect with external systems, databases, and APIs.

# Interface

## Sign Up

This is the Sign Up page of the application on which a user lands. The user can put in all their information and the system will generate a user account and a bank account for the user. For Personal Account there will only be one login account, for Business Account User can add more than one user accounts so all of them can access the same account.

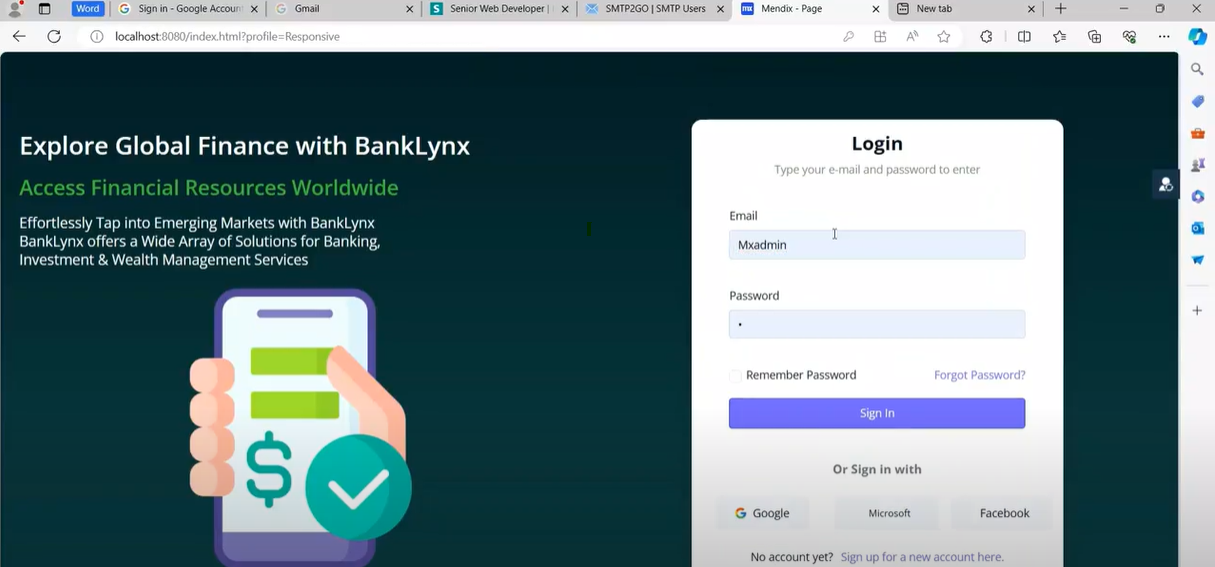


A screenshot of a computer

Description automatically generated

## Login Page

This is the login page which the user can use to login into the application.



## 2FA Popup

This popup is shown when the user clicks on Sign In button and has entered valid credentials. The system sends an email to the registered email of the user with the OTP.

A screenshot of a computer

Description automatically generated

The OTP is received in an email as shown in the below screenshot.

A screenshot of a computer

Description automatically generated

## HomePage

When a user logs in after entering the OTP he is redirected to this homepage. All the available options are listed in the sidebar, from here the user can use different banking functionalities that are implemented.

A screenshot of a computer

Description automatically generated

## Transaction Popup

On Clicking the Make a Transaction popup, this below popup is shown where user can enter details related to a Transaction. The transaction happens in real time and balance updates are done.

A screenshot of a computer

Description automatically generated

## Generate Transaction Reciept

When the User performs a successful transaction a system generated receipt is shown which can be downloaded, printed. The user can also access this receipt in the transactions page.

A screenshot of a computer

Description automatically generated

## Account Information Page

This page shows all the information related to bank account of the user. Current balance, and all other necessary information is shown and recent transactions are also listed at this page.

A screenshot of a computer

Description automatically generated

## Transaction History Page

This page shows the transaction history of the user. All the transactions done by user are listed and user can also see the transaction receipt by clicking the view receipt button against each transaction.

A screenshot of a computer

Description automatically generated

## Recurring Payments/Direct Debits

This page is used to display the recurring payments or direct debits. The User can do a new transaction by clicking new button or can see the history.

A screenshot of a computer

Description automatically generated

## Services Request

From this page the user can request services like Checkbook, Credit/Debit Card or Bank statement etc.

A screenshot of a computer

Description automatically generated

## Admin Home Page

On the home page, the admin can see all the accounts in the app. For bank account information we have added a separate menu item. This page only lists the user accounts, the admin can block users from logging into the application if there is anything suspicious.

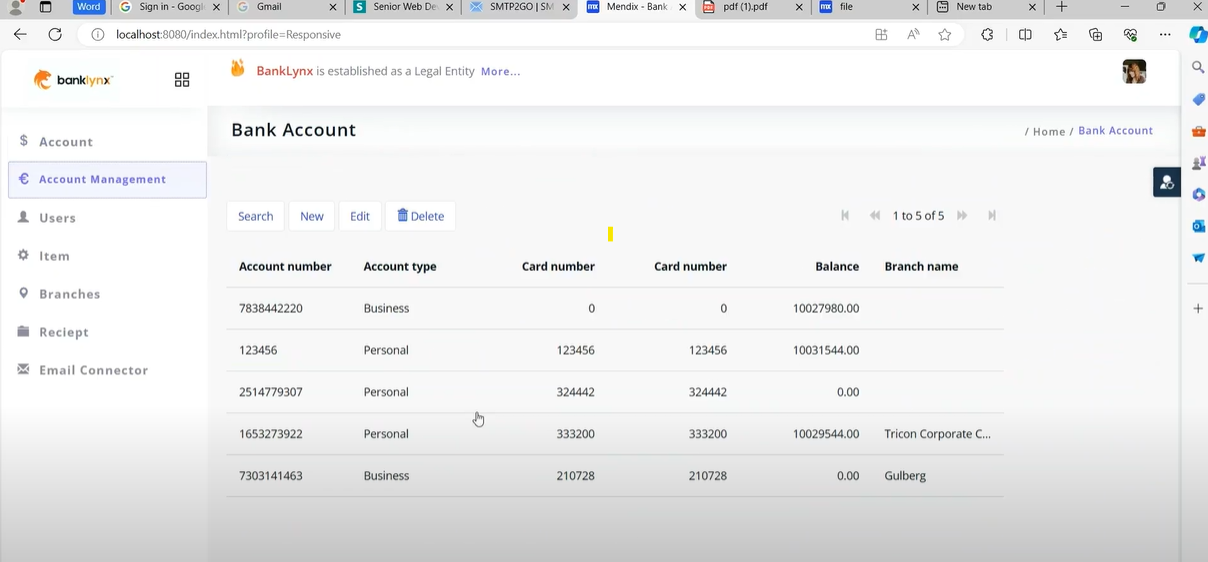
Admin has different routes and settings that he can access, so all information shown to admin cannot be accessed by Users and vice versa.

A screenshot of a computer

Description automatically generated

## Bank Account Management

This page is shown to admin where he can see all the bank accounts in the app and can see the account specific information.



## User Login Stats

On this page the admin can see that which users logged in to the system by verifying the 2FA.

A screenshot of a computer

Description automatically generated

## Branches Management

On this page the admin can add branches and their location, if the bank has physical branches.

A screenshot of a computer

Description automatically generated

## Email Settings

This page is connected to show Email conifiguration and the admin can test that emails are being sent successfully.

A screenshot of a computer

Description automatically generated

# Functionality

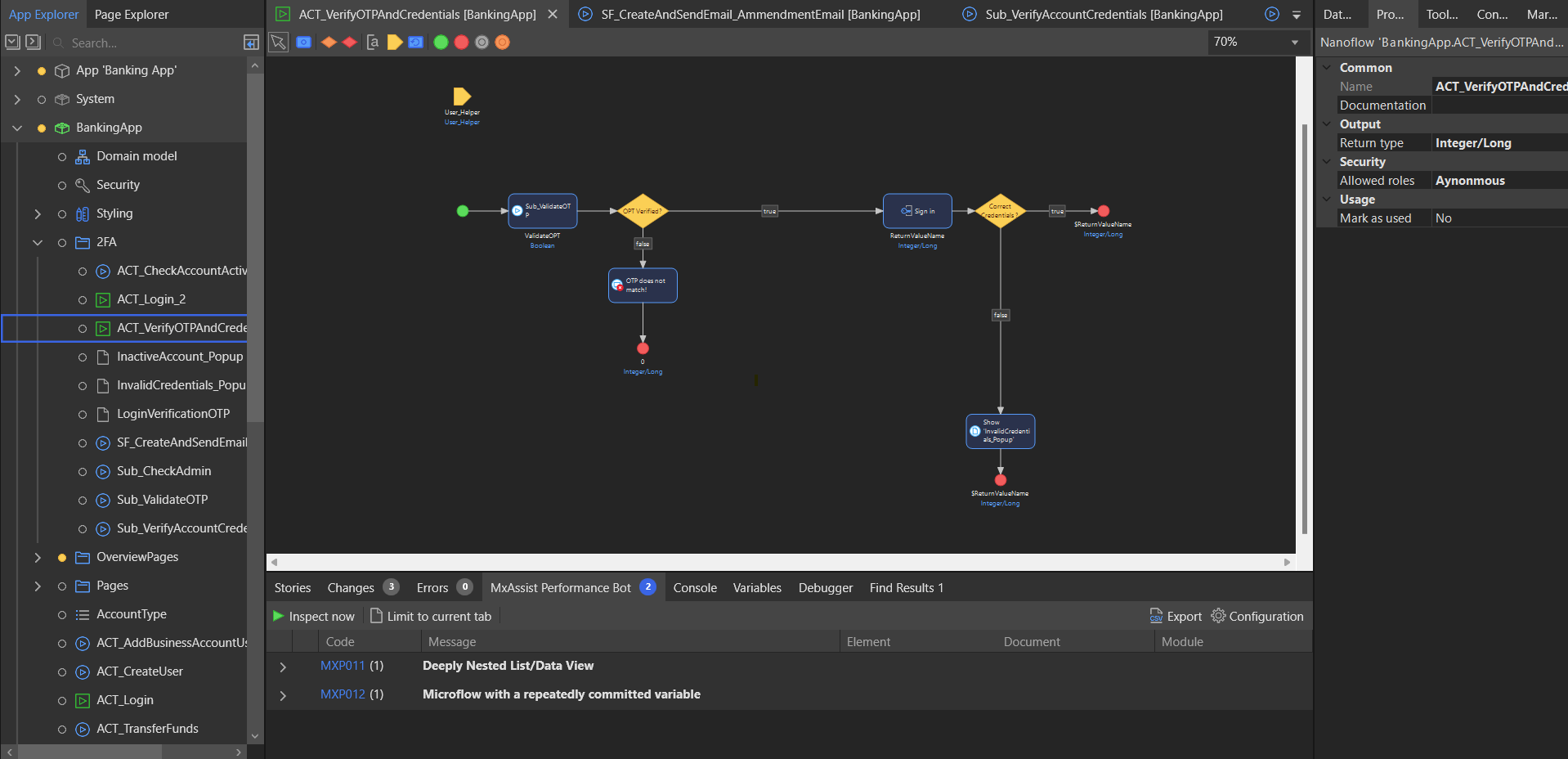
## Data Model

This is the schema of the Database that we have designed to implement banking functionalities in the application.



## 2FA

This is the logic for OTP verification and login.



## Email

This is the logic for sending OTP in the emails.

A computer screen shot of a computer

Description automatically generated

## Adding Users to Business Account

This is the logic for adding multiple users to a Business Account

A diagram of a network

Description automatically generated with medium confidence

## Transfer Funds

This logic for transferring funds to accounts in the bank is shown in the picture below.

A computer screen shot of a diagram

Description automatically generated

## Get Recent Transactions

This is the logic for retrieving recent transactions from the database. The database has constraints implemented which only let us retrieve the information that is related to the current logged in user.

A computer screen shot of a diagram

Description automatically generated

## Data Encryption and Validations

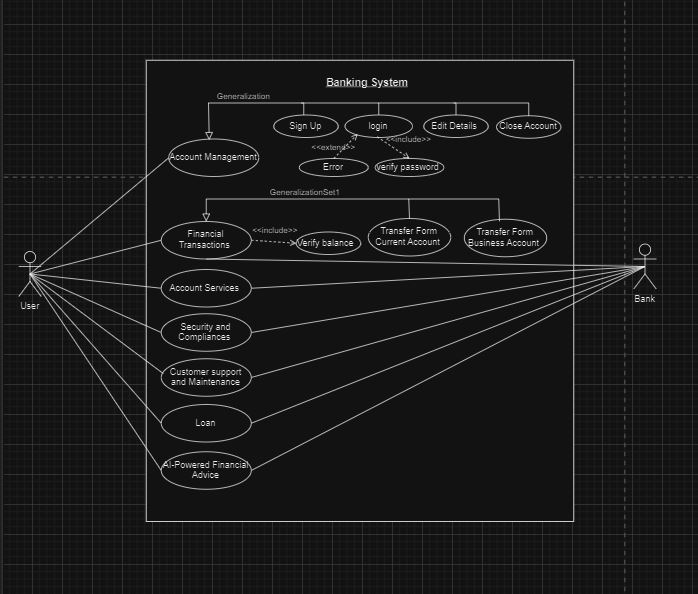
Mendix prioritizes security and all the sensitive information is handled on the database level by adding XPATH constraints, Mendix automatically manages data encryption and comply to all industry standards for data security.

All client side validations are handled on the client side to prevent user from entering wrong information.

# Behavioral Diagram

## Use Case Diagram

This is our use case diagram that help us to understand the system functionality, identify our actors and clarify our system boundaries. It provide foundation for our system, help use to priorities the features of our system.

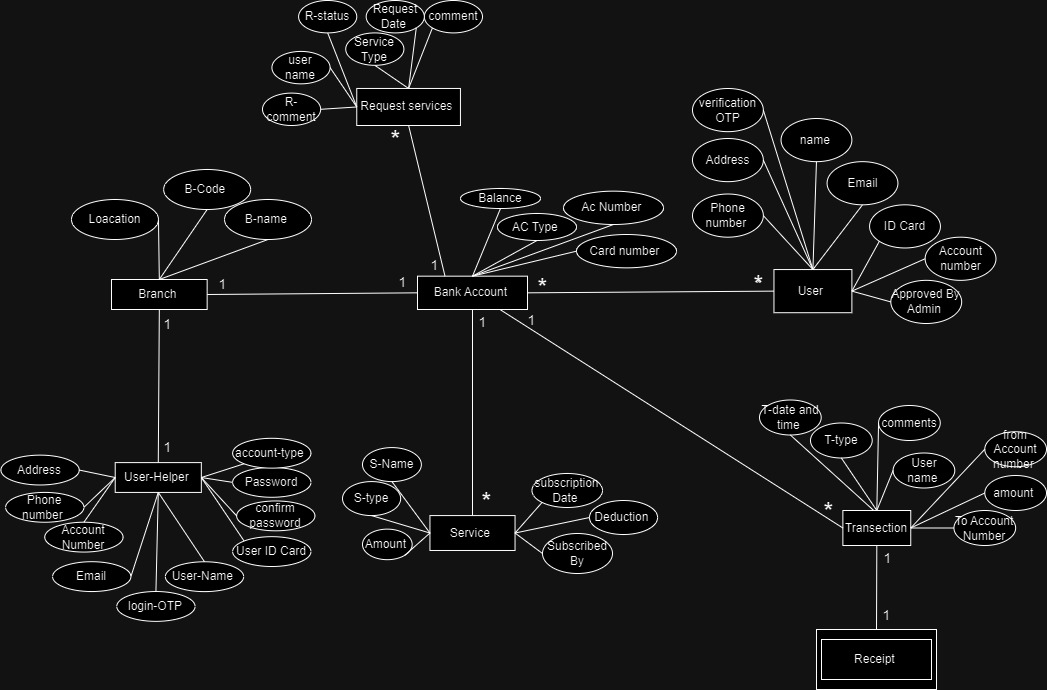


## Sequence Diagram

# Structural Diagram

## ER Diagram

ER diagram provides us overall graphical structure of over all database, it present the relationship between different entities in database. It is used to models the real-world objects.



## Class Diagram

Class Diagram provides us high level information of our system that will help us to communicate and document. In a class diagram objects are instance of classes and each class have its own attributes and methods.

A computer screen shot of a computer

Description automatically generated

## Project Architecture

**A picture containing object, device

Description automatically generated**

## Chart Of Activities(PERT)

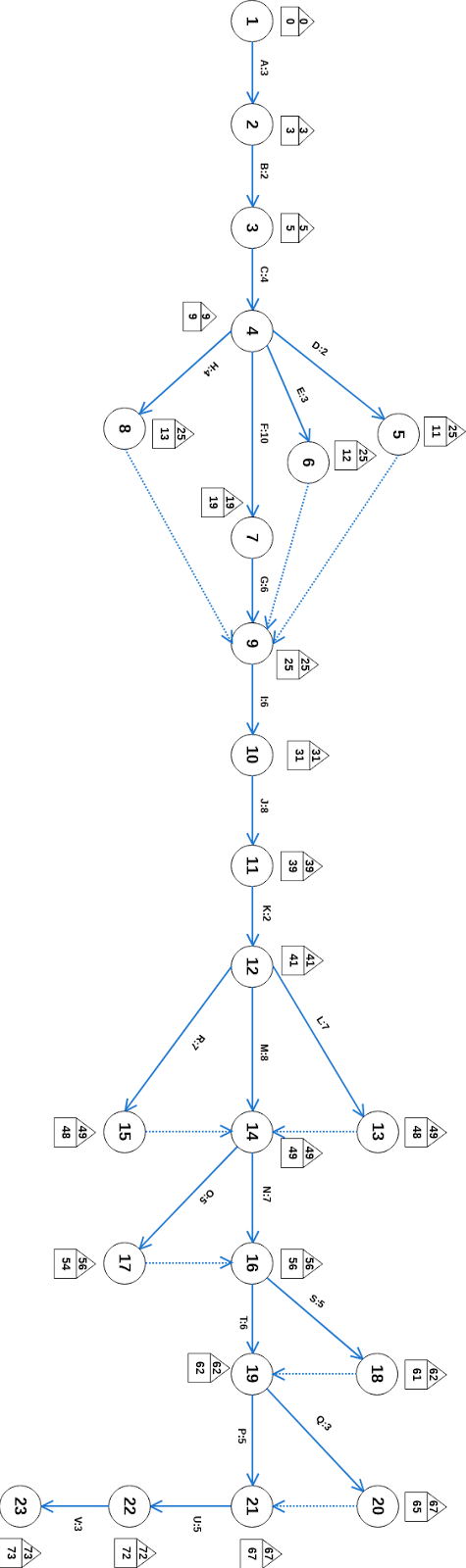
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity Description** | **Immediate Predecessor** | **Time Estimate** | | | **Expected Time**  D=(to+tm+tp) /6 |
| **to** | **tm** | **tp** |
| **A:** Project planning | None | 2 | 3 | 4 | **3** |
| **B:** Meetings | A | 1 | 2 | 3 | **2** |
| **C:** Project scheduling | B | 2 | 3 | 6 | **4** |
| **D:** Scope management | C | 1 | 2 | 3 | **2** |
| **E:** Risk management | C | 1 | 2 | 5 | **3** |
| **F:** Requirement gathering | C | 8 | 10 | 12 | **10** |
| **G:** Requirement specification | F | 3 | 5 | 9 | **6** |
| **H:** Designing prototypes | C | 2 | 4 | 6 | **4** |
| **I:** Designing database | H | 4 | 5 | 8 | **6** |
| **J:** Designing document | I | 6 | 7 | 10 | **8** |
| **K:** Development planning | J | 1 | 2 | 3 | **2** |
| **L:** Develop World Directory module | K | 5 | 6 | 9 | **7** |
| **M:** Develop Tourist Guide module | K | 5 | 8 | 10 | **8** |
| **N:** Develop Map Styling module | L,M,R | 4 | 7 | 10 | **7** |
| **O:** Develop Routes and Directions module | L,M,R | 3 | 4 | 6 | **5** |
| **P:** Develop Augmented Reality module | S,T | 3 | 5 | 7 | **5** |
| **Q:** Develop Favorite Destination module | S,T | 2 | 3 | 4 | **3** |
| **R:** Develop Places Details module | K | 5 | 6 | 9 | **7** |
| **S:** Develop Push Notification module | N,O | 4 | 5 | 6 | **5** |
| **T:** Develop Search Nearby Places AR | N,O | 2 | 6 | 7 | **6** |
| **U:** Testing | L,M,N,O,P,  Q,R,S,T | 2 | 5 | 8 | **5** |
| **V:** Deployment | U | 1 | 3 | 5 | **3** |

## Activity chart

|  |  |  |
| --- | --- | --- |
| **Activity** | **Immediate predecessor** | **Duration (Weeks)** |
| A | None | 3 |
| B | A | 2 |
| C | B | 4 |
| D | C | 2 |
| E | C | 3 |
| F | C | 10 |
| G | F | 6 |
| H | C | 4 |
| I | H | 6 |
| J | I | 8 |
| K | J | 2 |
| L | K | 7 |
| M | K | 8 |
| N | L,M,R | 7 |
| O | L,M,R | 5 |
| P | S,T | 5 |
| Q | S,T | 3 |
| R | K | 7 |
| S | N,O | 5 |
| T | N,O | 6 |
| U | L,M,N,O,P,Q,R,S,T | 5 |
| V | U | 3 |

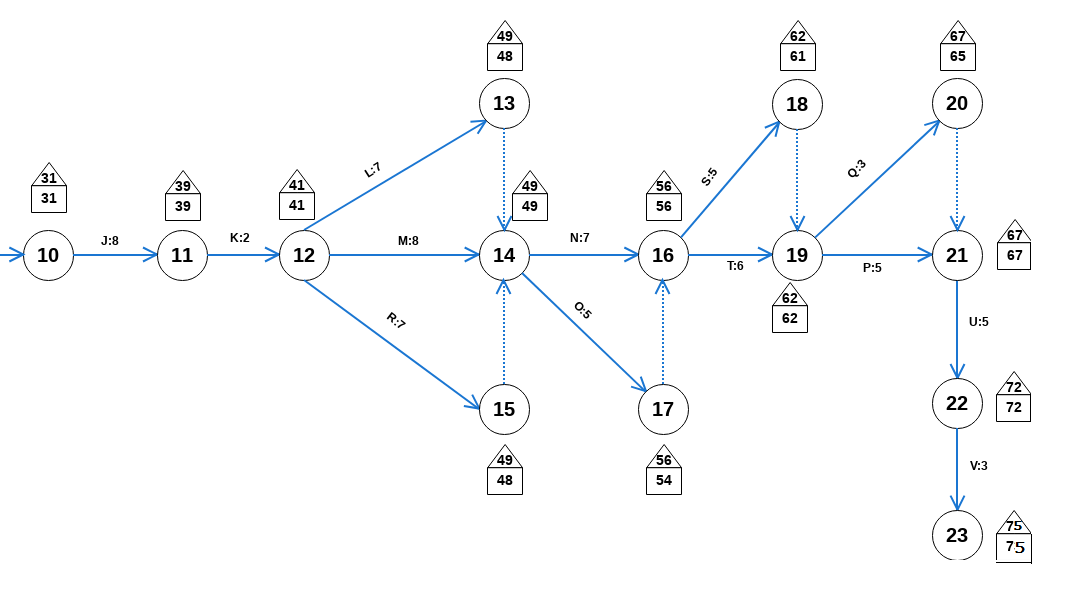
## Project schedule

**AOA Network**



A diagram of a diagram

Description automatically generated



## Critical Path

A:3 B:2 C:4 F:10 G:6 I:6 J:8 K:2 M:8 N:7 T:6 P:5 U:5 V:3

**Expected project Duration** = 75 days