Inestment Buddy Architecture Spec

# [Contents](#_Toc199232020)

[[1. Contents 1](#_Toc199232020)](#_Toc536020092)

[[2. Document Owners 1](#_Toc199232020)](#_Toc536020093)

[[3. Overview 2](#_Toc199232020)](#_Toc536020094)

[[4. Data store 2](#_Toc199232020)](#_Toc536020095)

[[4.1. Overview 2](#_Toc199232020)](#_Toc536020096)

[[4.2. Schema 2](#_Toc199232020)](#_Toc536020097)

[[5. Data Layer 2](#_Toc199232020)](#_Toc536020098)

[[5.1. Overview 2](#_Toc199232020)](#_Toc536020099)

[[5.2. Component/Method 1 2](#_Toc199232020)](#_Toc536020100)

[[5.3. Component/Method 2 2](#_Toc199232020)](#_Toc536020101)

[[6. Application Layer 3](#_Toc199232020)](#_Toc536020102)

[[6.1. Overview 3](#_Toc199232020)](#_Toc536020103)

[[6.2. Component/Method 1 3](#_Toc199232020)](#_Toc536020104)

[[6.3. Component/Method 2 3](#_Toc199232020)](#_Toc536020105)

[[7. Presentation Layer 3](#_Toc199232020)](#_Toc536020106)

[[7.1. Overview 3](#_Toc199232020)](#_Toc536020107)

[[7.2. Component/Method 1 3](#_Toc199232020)](#_Toc536020108)

[[7.3. Component/Method 2 4](#_Toc199232020)](#_Toc536020109)

# Document Owners

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | |  |  |
|  | PM |  | Developer |  |

# Overview

*What does the product do? What are the different layers?*

This document provides the architecture details for the Investment Buddy Android App. This app is intended to be used by sophisticated users who wants to visualize what are the predictions what will be return on their investment if they invest some amount of money into a sector such as Gold, Silver etc., with the help of visualizations to take a conscious decision.

There are primarily three layers in this App.

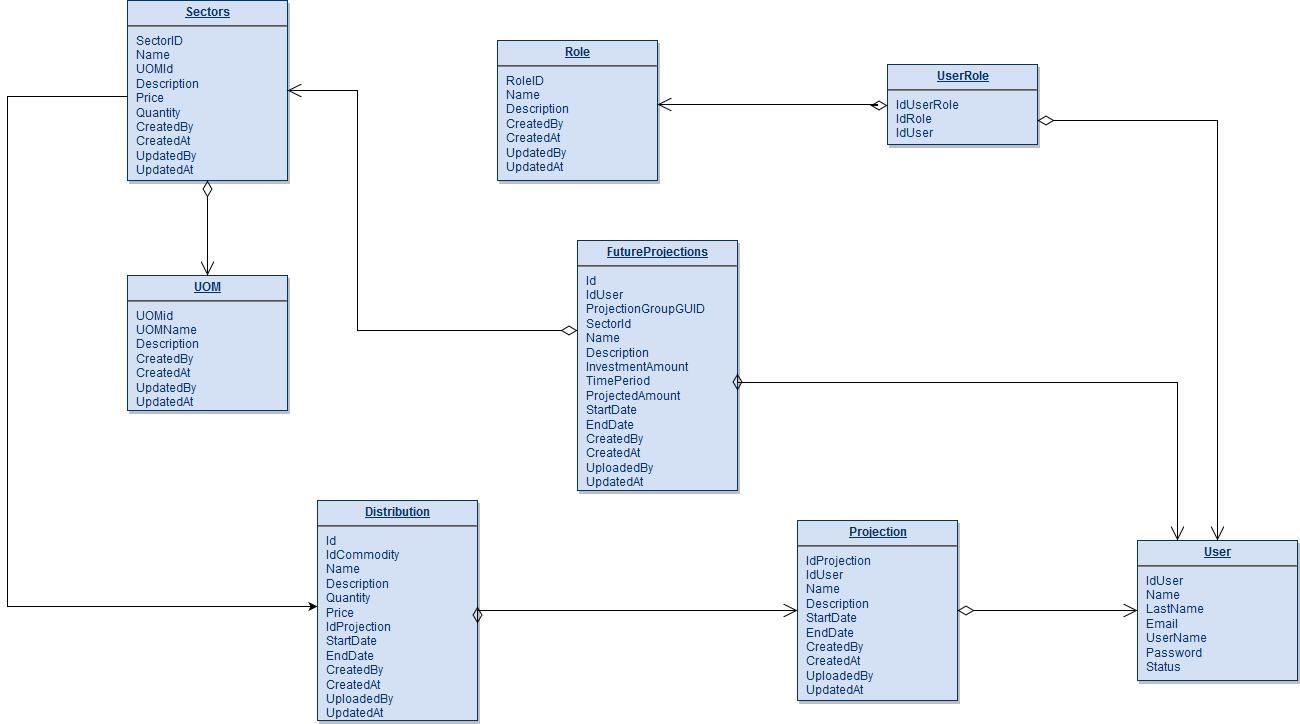
1. App (Presentation Layer) – uses Android Java implementation
2. API (Application/Integration Layer) – uses Spring REST Business Services
3. Data Services (Data Layer) – uses JPA, Hibernate and Spring REST and interacts with SQL Server database

# Data store

## Overview

*What kind of data will be in the data store? What is the platform for the data store (SQL, etc.)*

SQL Server holds the data store and stores relational data. The schema has been explained with the database diagram below along with the table details.



## Schema

*<Data-type-1> <Column-name-1>*

*<Data-type-2> <Column-name-2>*

# Data Layer

## Overview

This layer is implemented as Spring REST Data Services using JPA and Hibernate connecting to SQL Server Data Store. The Data Services provides basic CRUD operations required around different tables in the schema explained above.

## LoginDataService/validateLogin

This method takes username and password as entered by the user in the App and checks the database to see if the user exists in the database, returns user proxy object with few key fields that can be used in the App as needed.

**Input: Login**

**Type: POJO**

The JSON input represented as a POJO will have username and password for validation.

**Output: LoginResponse**

**Type: POJO**

This returns an array of user proxy objects with username along with an list of recent predecitions by this user if any and top 3 recent predictions. If there are no existing predictions both top predictions will be empty11111.

## RegistrationDataService/registerUser

This service helps registering a user in the InvestmentBuddy data store.

**Input: User**

**Type: POJO**

*<What data does this have?>*

This method takes user registration information a Java POJO with fields username, password and phone number.

**Output: RegistrationResponse**

**Type: POJO**

This holds status indicating whether the registration was successful or not along with status description. The field status description explains the details about the status.

## PredictionService/recentPredictions

This method takes userid as input and returns most recent (top 3) predictions by other users.

**Input: userName**

**Type: String**

This method takes user name.

**Output: PopularPredictions**

**Type: JSON**

This holds a list of recent (top 3) predections by other users other than the input user. If there are no predictions yet this returns an empty array

## PredictionService/predict

This method takes users input for predicting the investment projections and generates a PDF with the help of Anaconda call and then stores the PDF URL and relevant information in the data store.

**Input: Investment**

**Type: POJO**

The user investment information from the user in form of an list of investment options into different sectors along with userName.

**Output: InvestmentResponse (List<Predictions>)**

**Type: POJO**

This contains an array of predections as a list of prediction objects. The array contains each sector name and prediction url.

# Application Layer

## Overview

This layer is implemented as android that consume data services. This layer is responsible to interact between App and data services.

## AuthenticationService/validateLogin

This method consumes the validateLogin of LoginDataService by sending username and unencrypted password to the data service call.

**Input: Login**

**Type: POJO**

This input contains the user name and password base64 encoded.

**Output: LoginResponse**

**Type: POJO**

This contains the user proxy with username along with the recent predictions by this user. This also contains another array of popularPredictions by other users, the top 3 recent predictions from other users.

## RegistrationService/registerUser

This method takes users registration information as input and makes a call to RegistrationDataService to store the registration information, in case if there is an existing user with the same name it sends an error response using the status and status description fields.

**Input: User**

**Type: POJO**

The user registration information implemented as POJO with fields username, password and phone number.

**Output: RegistrationResponse**

**Type: POJO**

This object contains status indicating whether user registration is successful or not. The status description field will be determined based on the status code returned from the data service call.

## PredictionService/predict

This method takes users input for predicting the investment projections and calls the data service for predictions.

**Input: Investment**

**Type: POJO**

The user investment information from the user in form of an list of investment options into different sectors along with userName.

**Output: List<Predictions>**

**Type: POJO**

This contains an array of predections as a list of prediction objects. The array contains each sector name and prediction url.

# Presentation Layer

## Overview

This layer provides Android native application interface supporting from Android API level 15 till API level 28. This App interacts with the business/integration layer using Retrofit version 2 for REST interactions. The predictions rendered to the user will be rendered in PDF format.

## Home/Layout Click or Tap

*<Sends some data to the application layer>*

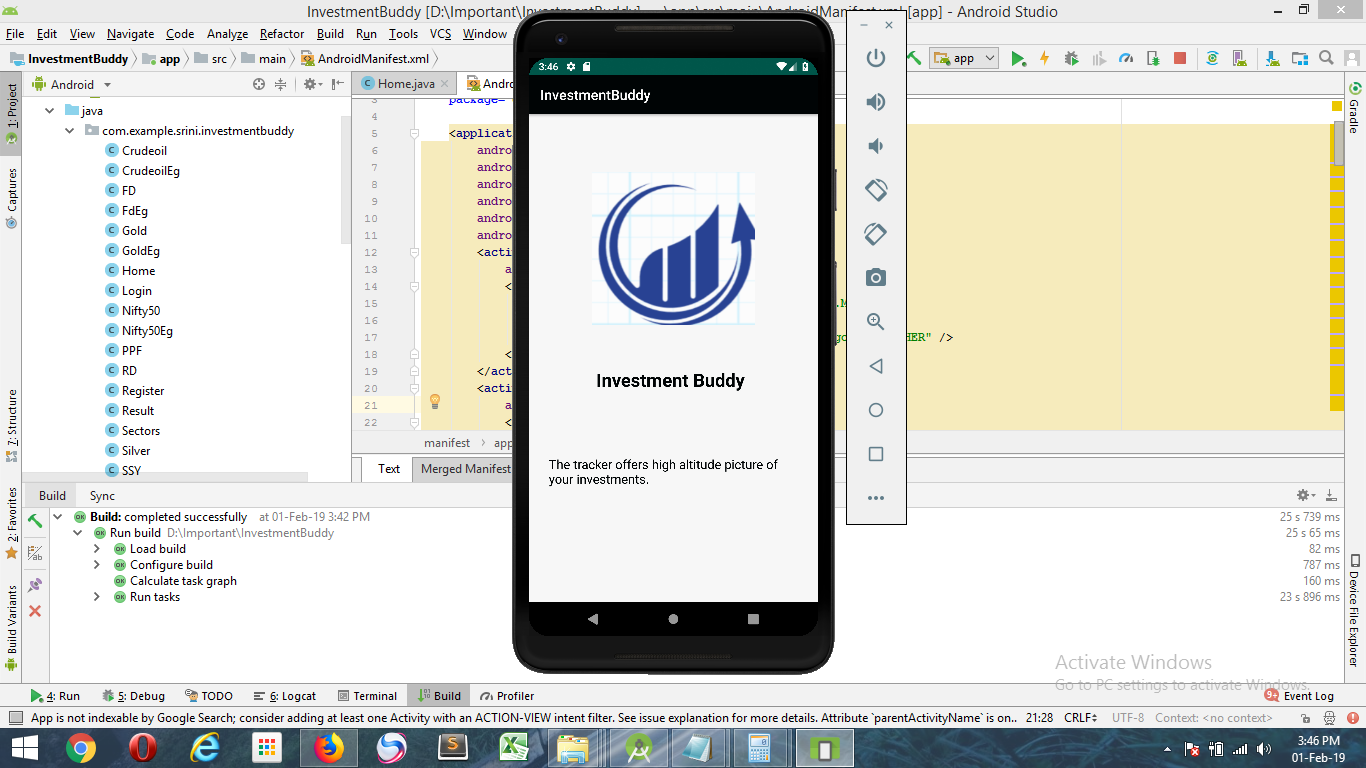
**Input: Click/Tap**

**Type: N/A**

**Output: Navigates to Trending Page**

**Type: NA**

**Touching anywhere on this screen navigates to trending page in the App.**



## Trending/Back button

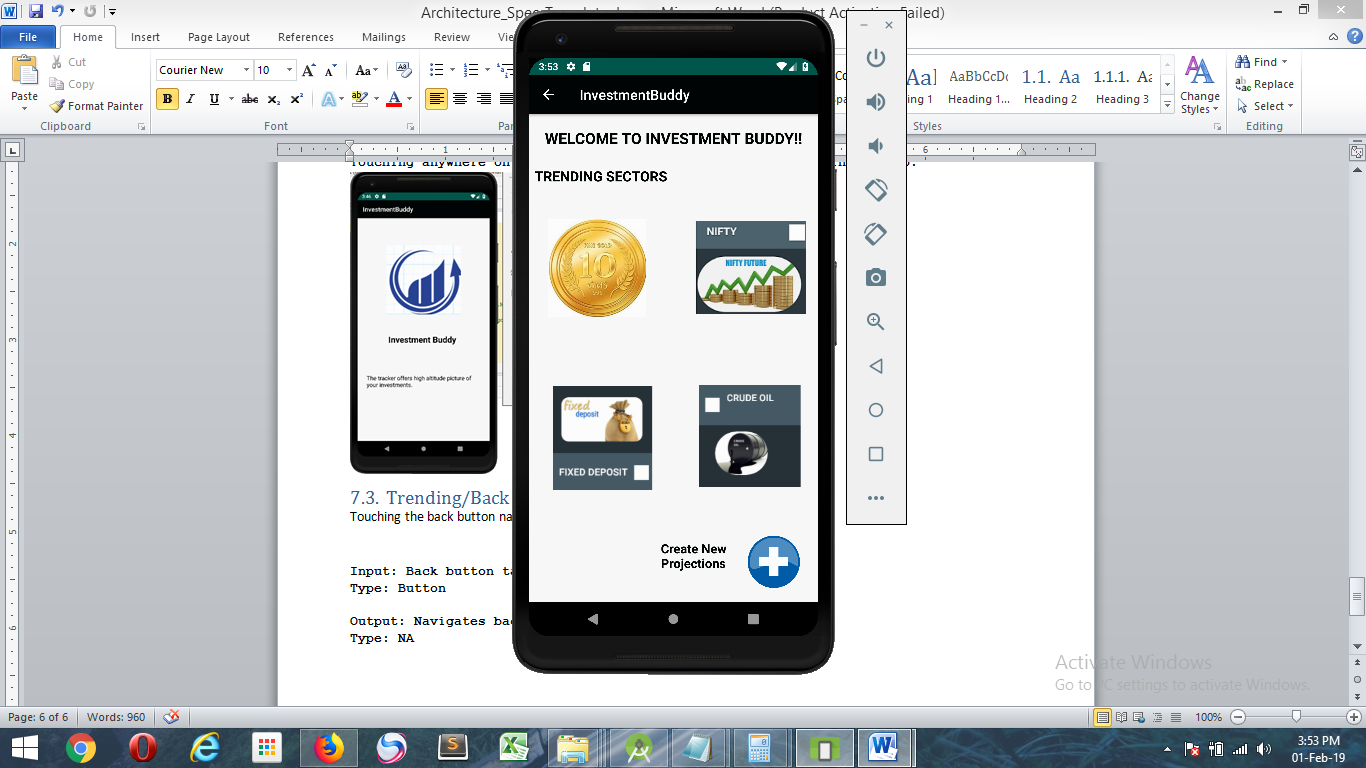
Touching the back button navigates back to the home page.

**Input: Back button tap**

**Type: Button (Circled in Red below)**

**Output: Navigates back to home page**

**Type: NA**



## Trending/Gold Tap

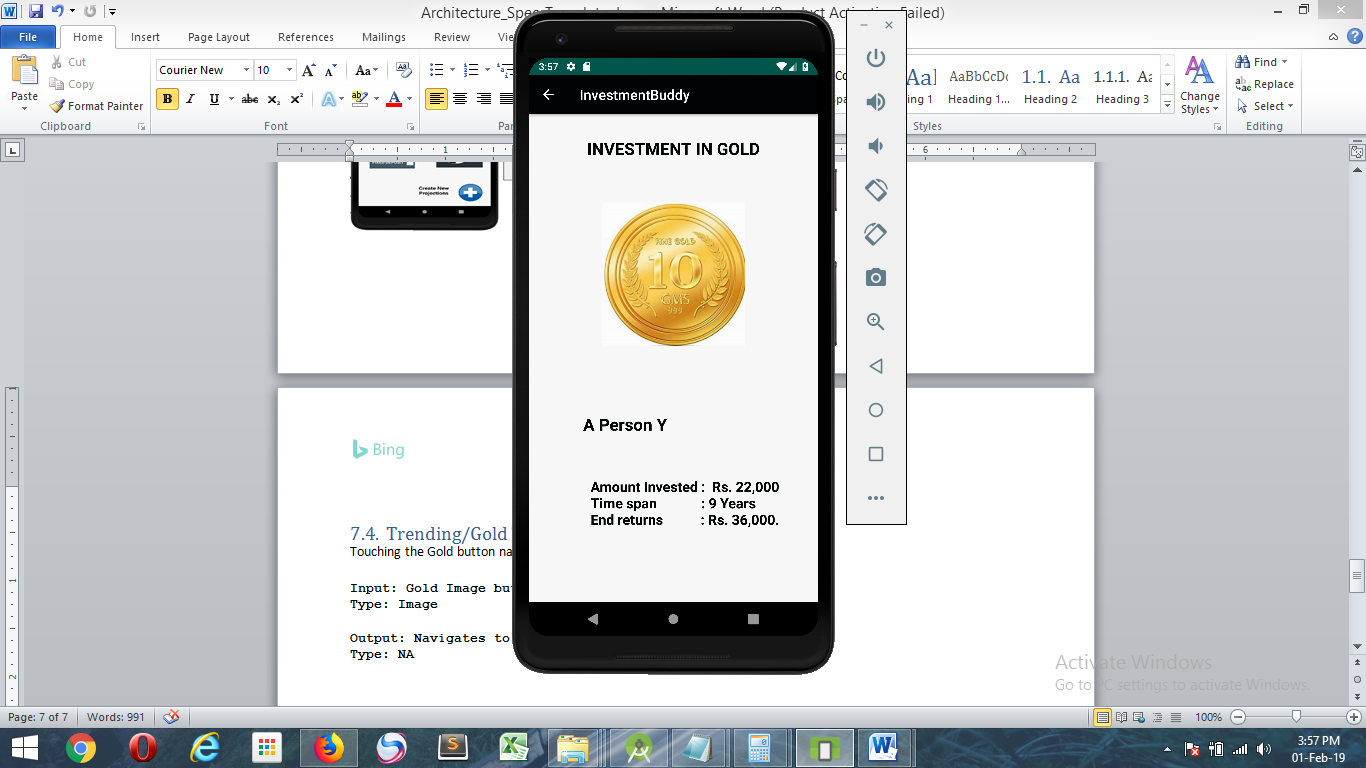
Touching the Gold button navigates to the Gold description page.

**Input: Gold Image button tap**

**Type: Image**

**Output: Navigates to Gold Description**

**Type: NA**



**Touching the back button in the action bar navigates back to Trending page.**

## Trending/Nifty Tap

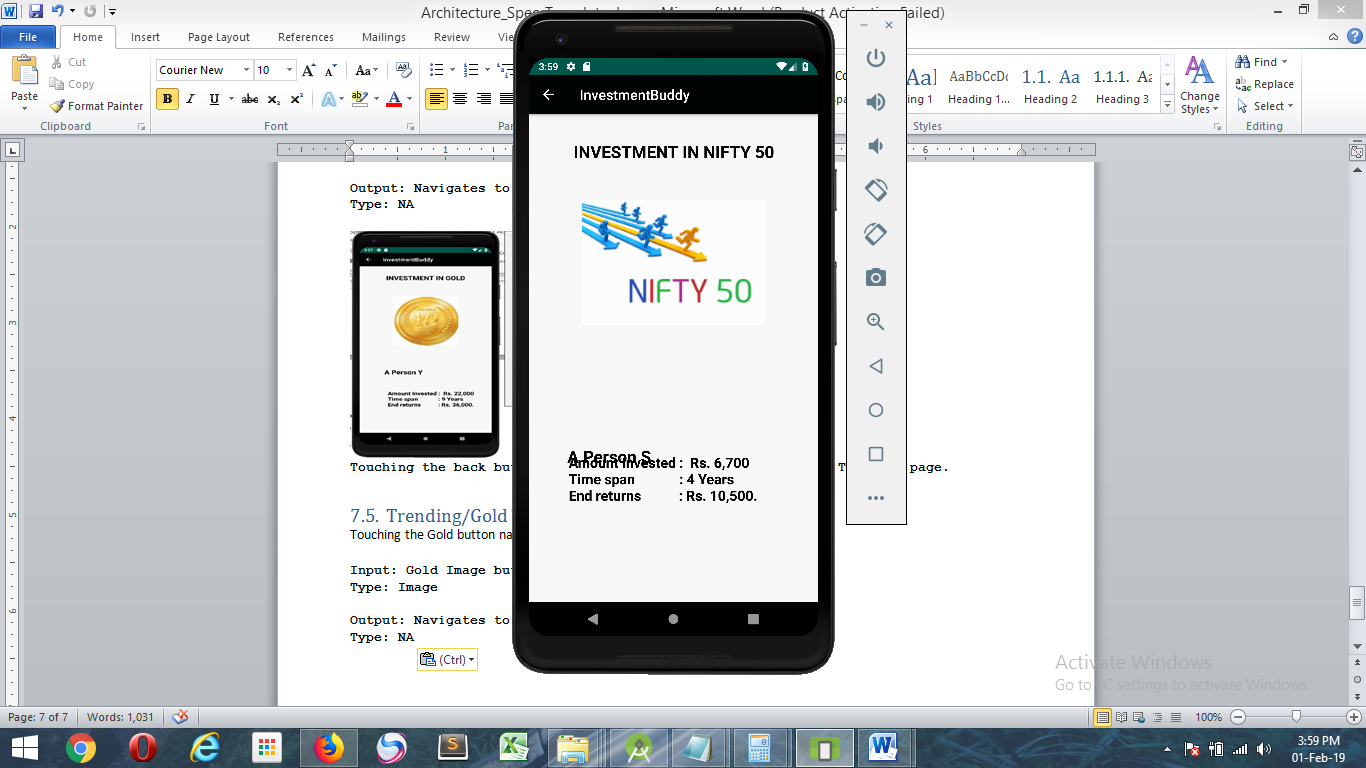
Touching the Nifty Image navigates to the Nifty description page.

**Input: Nifty Image button tap**

**Type: Image**

**Output: Navigates to Nifty Description**

**Type: NA**



**Touching back button in the action bar navigates back to Trending page**

## Trending/Fixed Deposit Tap

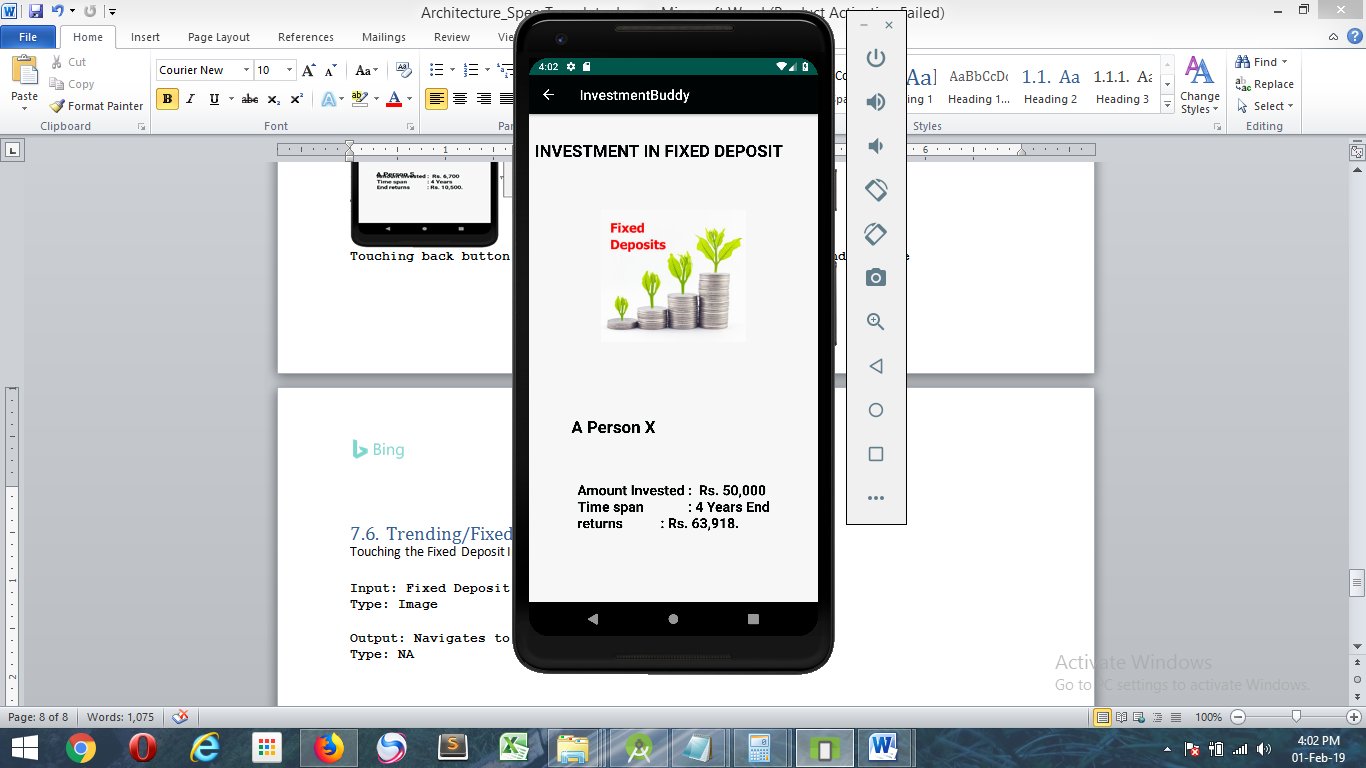
Touching the Fixed Deposit Image navigates to the Fixed Deposit description page.

**Input: Fixed Deposit Image button tap**

**Type: Image**

**Output: Navigates to Fixed Deposit Description**

**Type: NA**



**Touching back button in the action bar navigates back to Trending page**

## Trending/Crude Oil Tap

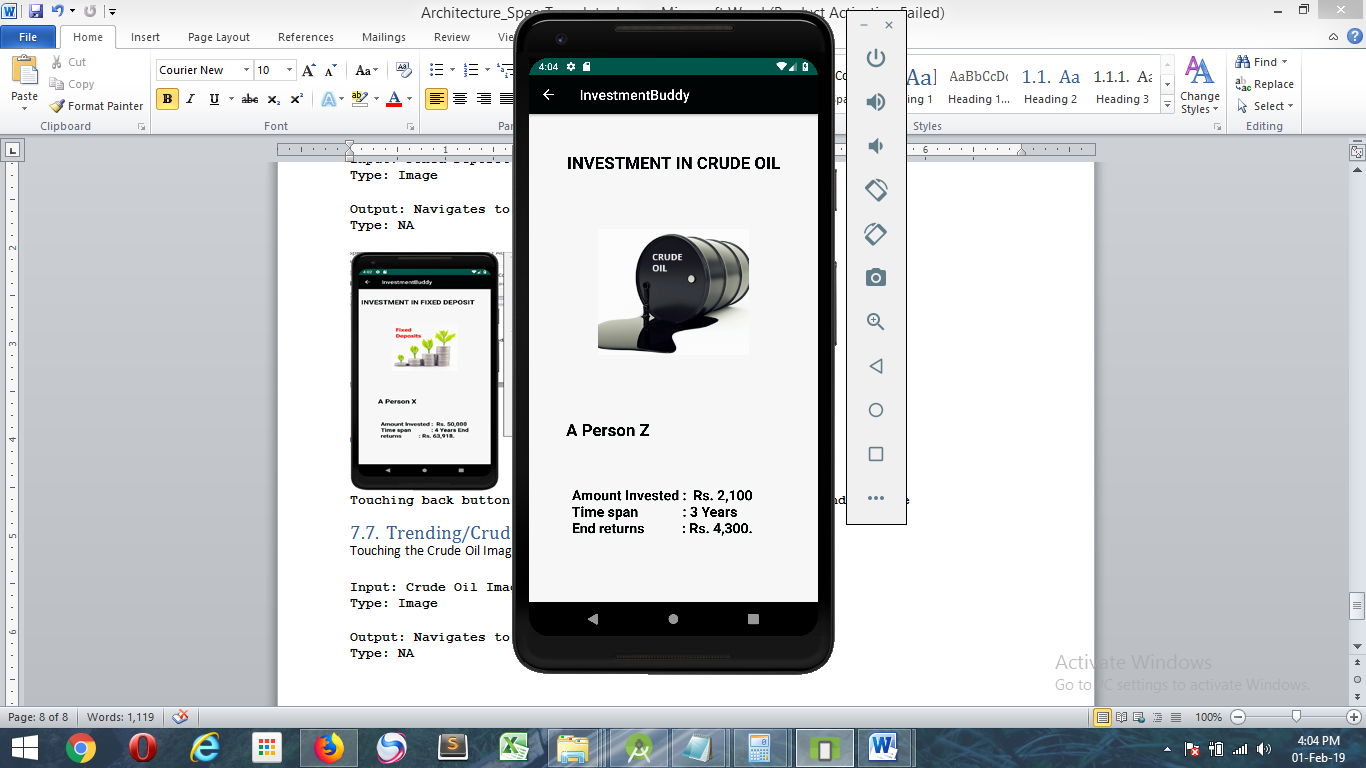
Touching the Crude Oil Image navigates to the Crude Oil description page.

**Input: Crude Oil Image button tap**

**Type: Image**

**Output: Navigates to Crude Oil Description**

**Type: NA**



**Touching back button in action bar navigates back to Trending page.**

## Trending/Create New Projection button tap

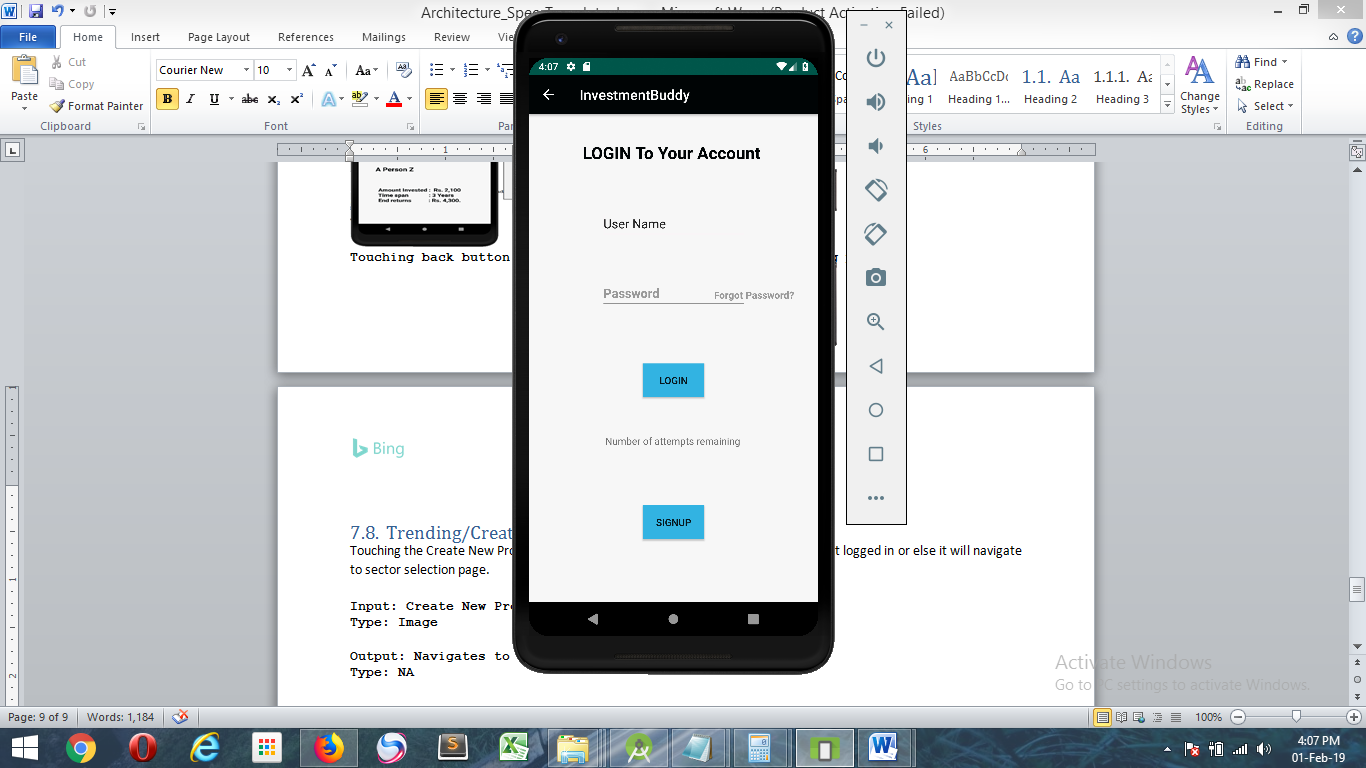
Touching the Create New Projection button navigates to the Login page if the user not logged in or else it will navigate to sector selection page.

**Input: Create New Projection Image button tap (Plus image)**

**Type: Image**

**Output: Navigates to Login page or sector selection page**

**Type: NA**



**This page is where a new user can signup using Signup button or Login by entering User name and password followed by Login button click.**

## Login/Login button tap

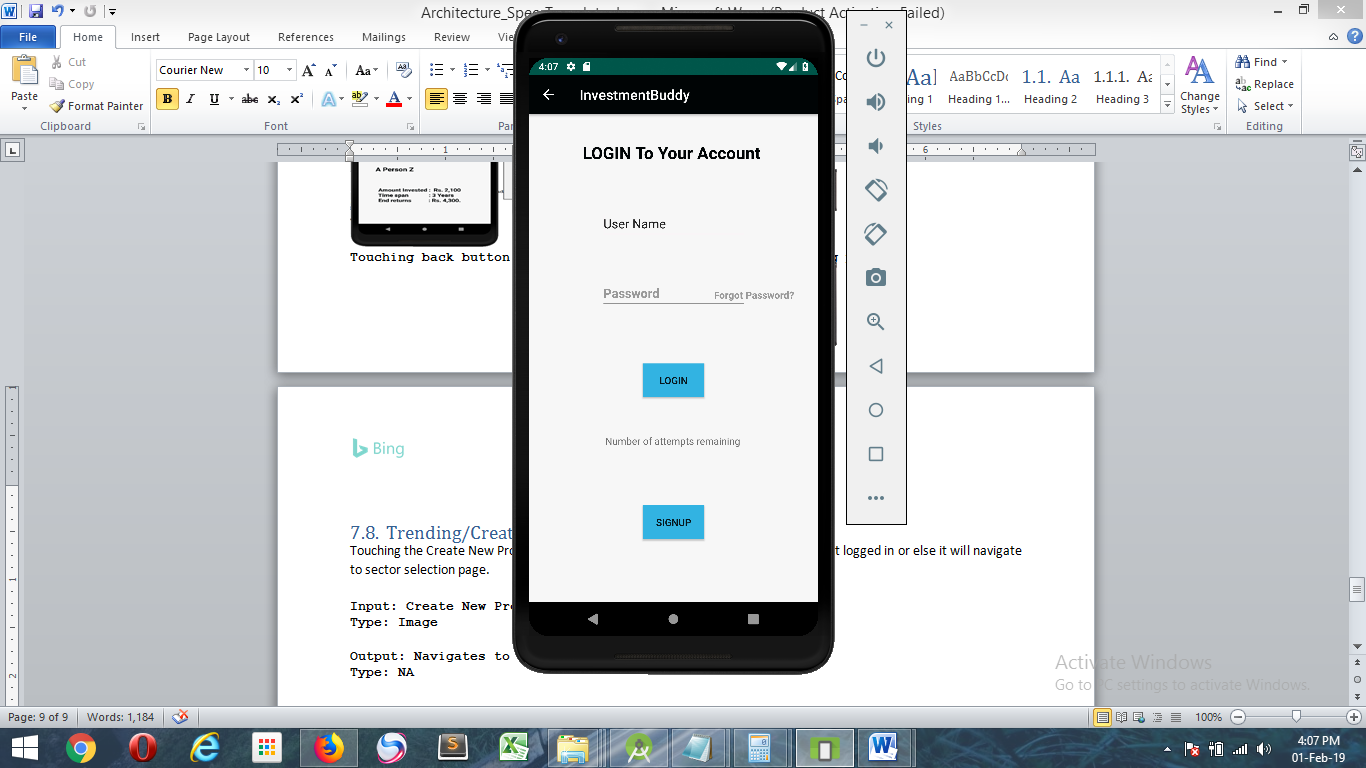
Touching the Login button after providing username and password logs in the user if already registered. This will also validate if uername and password are entered or not before clicking Login button. In case if the password does not match with the username or user does not exist appropriate validation messages displayed.

**Input: Login button tap**

**Type: Button**

**Output: Logs the user into the InvestmentBuddy app or else displays appropriate validation messages.**

**Type: NA**



## Login/Signup button tap

Touching the Signup button navigates to new user registration page.

**Input: Signup button tap**

**Type: Button**

**Output: Navigates the user to signup page.**

**Type: NA**