**P**roject **R**eport on

**Bitcoin Stock Investment Automation (*b*-SIA)**

Submitted in fulfilment of the requirements for

**Assignment**

In

**Software Engineering**

Submitted by:

Akanksha Shrivastava Aditya

Roll: 1605006 Roll: 1605004

Under the guidance of

Prof. Arup A. Acharya

Associate Professor & Program Head (IT)

School Of Computer Engineering



**SCHOOL OF COMPUTER ENGINEERING**

**KIIT UNIVERSITY**

**Bhubaneswar- 751024**

**Bitcoin Stock Investment Automation**

Project Mentor: Prof. Arup A. Acharya

Team Members:

* Akanksha Shrivastava (1605006)
* Aditya (1605004)

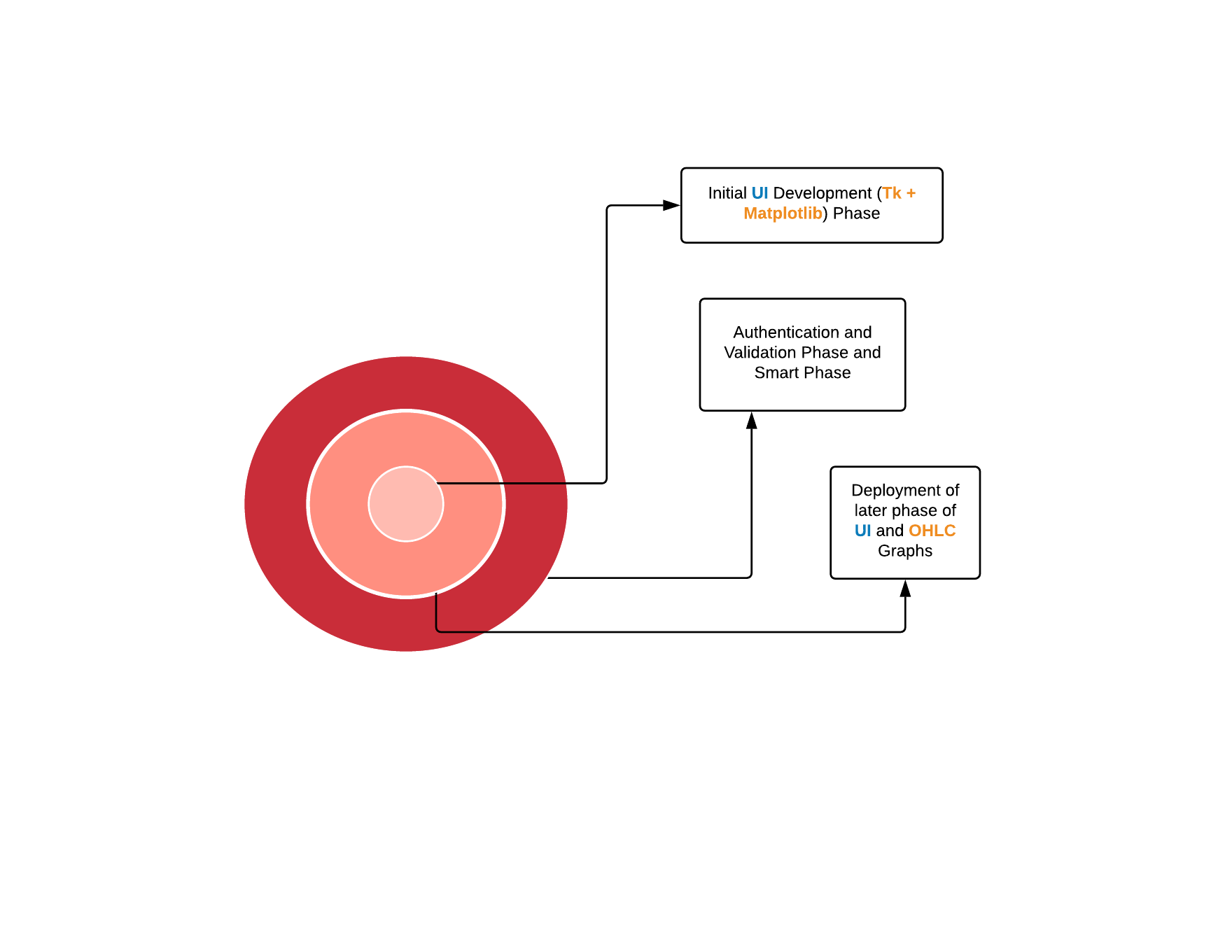
Branch: Computer Science & Engineering

# Project Abstract

Problem Definition: Design a software to automate the stock-investment process in way which is most profitable to the client.

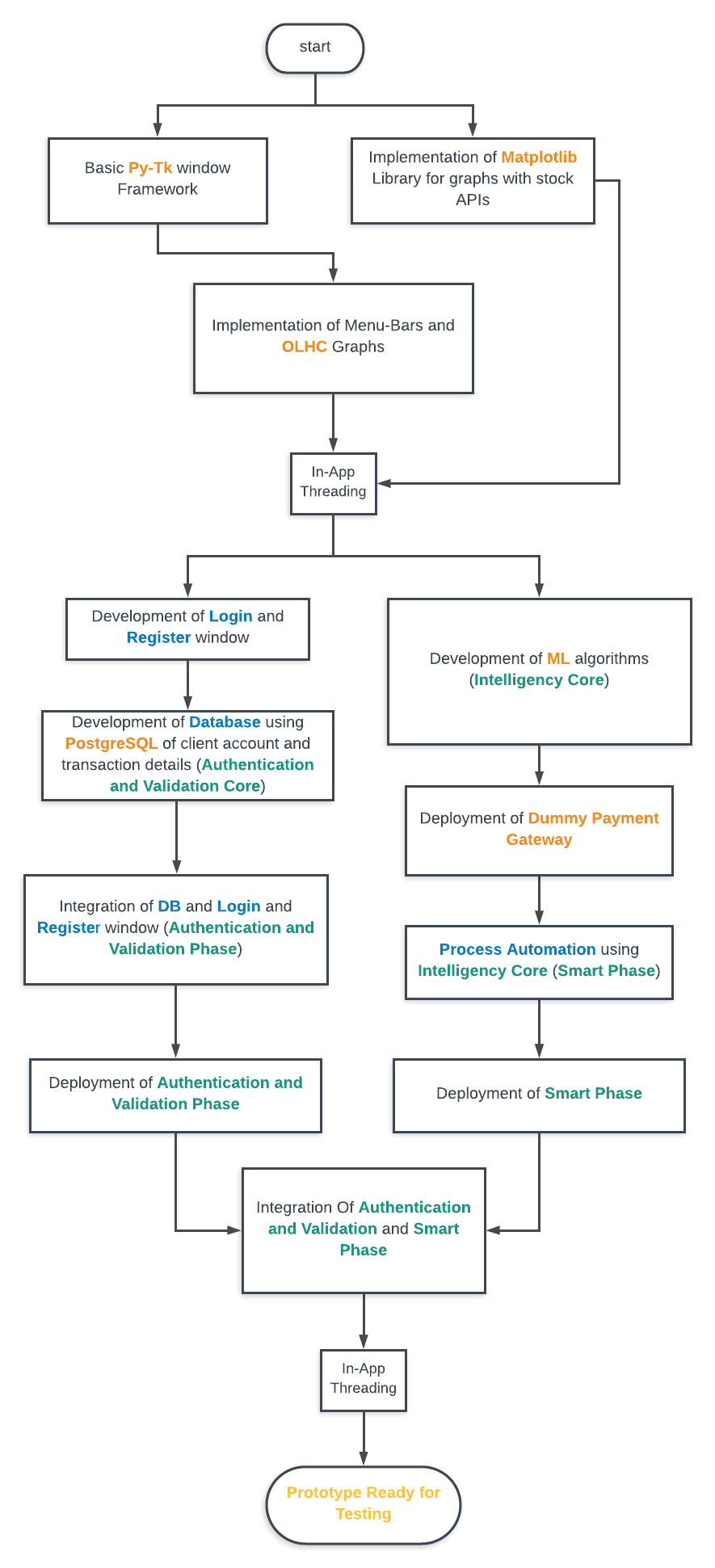
Features of Application: The project is a software based on Python-Tkinter UI framework. The user has full control to either – automate or not. If the client selects the automated process, ML algorithms take care of the rest, otherwise it is done normally on a hand-to-hand interaction basis.

For UX, the software uses Matplotlib-Live graphs to help the client visualise the flow of his liquid capital and see various economic aspects of his/her investments.

Process Model Employed: The process employed here is Evolutionary Model. We wanted to employ this model because of the project's job segregation.

As it can be seen here, the entire phases of UI/UX and logical backend are independent. So, employing this model was more rational and sensible and the others. The **Data-Flow-Diagram** would make it clearer.

Keywords: Python-Tkinter, Matplotlib-Live Graphs



ACKNOWLEDGEMENTS

We take this opportunity to express our deep gratitude to all those helping hands without whom this project would have not been what it is.

We take immense pleasure to express our thankfulness to our mentor Prof. Arup A. Acharya for his constant motivation, timely suggestions which made our sail smooth through the odds we faced in our project.

I would like to thank our Dean Prof. Samresh Mishra for giving us this opportunity to enhance our skills by giving this project and for motivating us throughout the time and also thank the complete faculty of Computer Science and Engineering who have provided us the knowledge to successfully complete the project.

Last but not the least we would like to thank our friends who not only provided their helping hands in our project as and when required but also for becoming the first user of this software and providing us with all the feedback which helped us improve our software.

Akanksha Shrivastava

(1605006)

Aditya

(1605004)

**SCHOOL OF COMPUTER ENGINEERING**

**KIIT UNIVERSITY**

**Bhubaneswar- 751024**

**CERTIFICATE**

This is to certify that the project entitled “Bitcoin Stock Investment Automation” is being carried by Akanksha Shrivastava (1605006) and Aditya (1605004) in partial fulfilment of the requirements of Software Engineering Assignment in Computer Science & Engineering at School of Computer Engineering, KIIT University, and Bhubaneswar during the academic year 2018 under my supervision. The matter embodied in this project is original and has not been submitted for the award of any other assignment.

Signature of the Mentor

(Prof. Arup A. Acharya)

Assistant Professor & Program Head (IT)

School of Computer Engineering

**Table of Contents**

1. **Introduction**
   1. FAQ Related to Bitcoin Stock Investment
   2. Purpose of the Project
   3. Scope of the Project
2. **Objective**
3. **Software Development Lifecycle**
4. **Software Requirement Specification**
   1. Questionnaires
   2. System Requirements
   3. Functional Requirements
   4. Non Functional Requirements
   5. Features of the Project
5. **Design**
   1. UML Design
      1. Use Case Diagrams
      2. Class Diagram
      3. Activity Diagram
      4. Sequence Diagram
   2. Database Design
      1. ER Diagrams
      2. Table Diagrams
   3. Test Case Design
      1. LOGIN Module
      2. BOOKKEEPING Module
      3. GRAPH Module
      4. FEEDBACK Module
      5. ADMIN Module
   4. GUI Design
      1. LOGIN/SIGNUP Window
      2. ADMIN Window
      3. GRAPH Window
      4. BOOKKEEPING Window
6. **Coding**
   * 1. Database Connection
     2. Login/Signup Process
     3. Capital Management
     4. SVM Suggestive Support
7. **Testing**
8. **Future Work**
9. **Conclusion**
10. Introduction

Bitcoin Stock Investment Automation Project (*b*-SIA) is a live stock investment software designed specifically for Bitcoin stock investors, who want to live a more profitable life due availability of Machine Learning models to help them, suggesting times when they should or should not invest their capital.

A centralized database is maintained which bookkeeps everything and the final amount is credited to/debited from the user’s account provided at the registration. A report can be generated at any point of time which might display the loss/gain within the mentioned duration.

Advantages of using this application is stock investors can have the assistance and support of their own personal computer to suggest when or when not to invest. This could probably save a lot of capital which could be spent efficiently.

The user can use this software in two ways:

* + 1. Semi-Automatic Processing (using SVM model to predict the stock prices and letting it handle bookkeeping)
    2. Fully Manual

1. FAQ Related to Bitcoin Stock Investment

* What is ‘Bitcoin Stock Investment’?

Ans. So there isn't exactly a **stock** for it, per se. However, you can treat the **bitcoins** you have as an asset that can be bought and sold, and its value as the **bitcoin stock** price. ... The **Bitcoin Investment** Trust (GBTC) is one notable option that operates similarly to an exchange-traded fund.

* How will I benefit from *b*-SIA?

Ans. If you are Bitcoin stock investor, you’d profit, and profit in easy ways. Having a suggestive software which efficiently manages your account would always be a plus for your life.

1. Purpose

The purpose of this project is to present a semi-automated solution of stock investment procedure by using this application.

This software on Stock Investment presents a very user friendly interface through which a potential investor can invest his/her capital and expect a calculated probable profit, without risking a whole lot of assets.

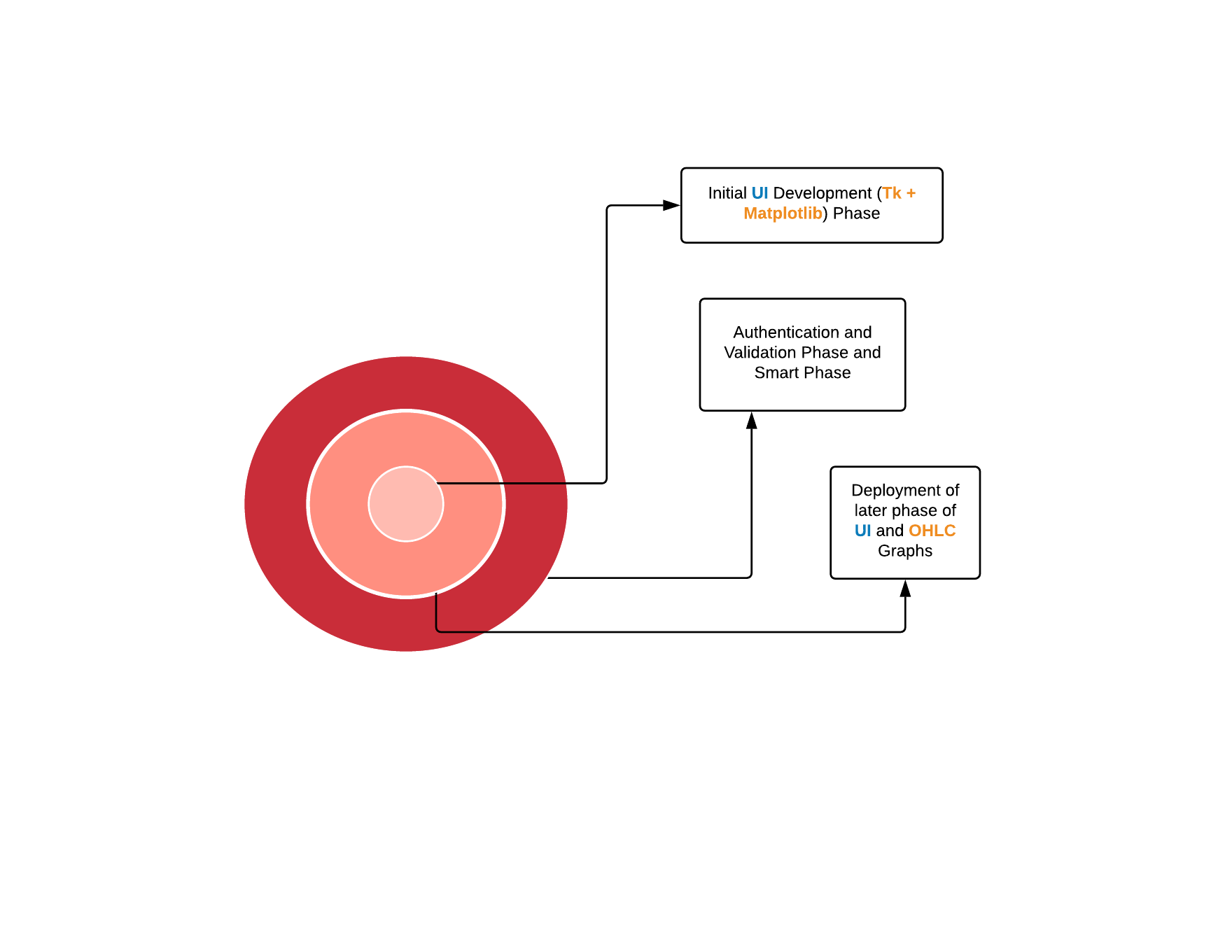
That is the main idea of Threshold Equivalent (TEQ) in this software. The user can initially only spent maximum of TEQ capital in stock he is interested in. Once getting the hold of how the software works, TEQ value can be increased.

1. Scope

* A major portion of risk in stock market can reduced, which often leads to bankruptcy.
* A live graph provide visual aid for investors to clarify any doubts about their investment decision.
* The investor can be confident in his/her decision.

1. Software Development Lifecycle

The software development life cycle used here is Evolutionary model. The process starts with an implementation of a preliminary independent module and then tested rigorously. By the time this prototype is being tested the next least dependent module is implemented and this keeps repeating over and over till the process is complete.



**Why did we use Evolutionary Model?**

* Requirements of the complete system are clearly defined and understood.
* It was useful as the project was bi, so it was beneficial to develop each phase of system efficiently.
* As major requirements were defined, however we could easily identify some more details with time.

1. Software Requirement Specification

A software requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use cases that describe interactions the users will have with the software.

1. Questionnaires
2. What is the privacy of every user?
3. What info is needed during login and registration of investors?
4. What are the features you want to manage while the login/signup has not been done?
5. Can the different users edit their profiles?
6. Will the user have a strict control over his capital investment??
7. Can the user track his transactions?
8. Can the user query?
9. Do we need any social plugins?
10. Can admin interact with the users?
11. Will the suggestive model be optional to the investor?
12. Can he ignore semi-automatic processing?
13. How do we prevent duplicity of users during registration time?
14. System Requirements

Different kinds of tools and software required to run this application:

1. Python 3.6 and above (for framework to function
2. Py-library Tkinter (for the UI of the application)
3. Py-library scikit-learn, tensorflow (for the ML module)
4. TCP/IP protocol is used communicate with the server. Client can send request to the server through TCP/IP Protocol and server responds to the client through TCP/IP Protocol.
5. 6. A centralized database is maintained to store all the data which can be queried using MYSQL.
6. Functional Requirements

R.1 Registration

*Description*: A user can register as an investor or as an admin

R.1.1 Registration as an Investor

*Description*: An investor can register himself by providing their name, email id, password, account number and *threshold equivalent*.

*Input*: Personal Details

*Output*: Investor Registration.

R.1.2 Registration as Admin:

*Description*: An admin can register themselves by providing their name and UID.

*Input*: Personal Details

*Output*: Admin Registration and UID generation.

R.2 Login

*Description*: An investor can log himself in by providing their email id and password.

R.2.1 Update Profile

*Description*: The user can update any of the details of their profile after successfully logging into their account.

R.2.1.1 Update Investor Profile

*Description*: The investor can log in into their accounts and then can do necessary changes they want in their profile.

*Input*: username, password

*Output*: Updated Voter profile

R.2.1.3 Invest Capital

*Description*: The user after successful log in can begin to invest money as he/she wishes.

*Input*: Capital TEQ confirmation

*Output*: Capital invested

R.2.1.4 Feedback

*Description*: The user can submit their feedback regarding the improvements they want in the existing system and also can report any bugs to the admin.

*Input*: feedback

*Output*: feedback submit successful.

1. Non-Functional Requirements

**Performance Requirement:** Performance of the system depends on the response time, number of transactions, graph queries and the load it can withstand. This version of the software has limited number of transactions and graph queries hence it can work in small network efficiently.

**Safety Requirement:** Application needs to have a check on:

1. No other server is running on same port in which Apache-MySQL is running otherwise it will stop Apache’s functionality.

2. If Tk-Forms with the services processing form input are consistent.

**Security Requirements:** Security is maintained by generating a unique ID for the user which he can use while login along with his password.

In case of system crash the data in the database should be back up so that no data is lost.

**Quality Requirements:** The application should have simple and user friendly GUI so that the end users find it easy to understand the functionality of the application and can carry out their task.

1. Features of the Project
2. Secure login/registration
3. Validation of investors
4. Bookkeeping System
5. Report Generator
6. Feedback
7. Invest portal
8. Profile updating
9. Design

Software design is the process of implementing software solutions to one or more set of problems. One of the important parts of software design is the software requirements analysis (SRA). It is a part of the software development process that lists specifications used in software engineering. We have thus described different forms of designs here.

1. UML Design

UML stands for Unified Modelling Language which is used in object oriented software engineering. Although typically used in software engineering it is a rich language that can be used to model an application structures, behaviour and even business processes. The different UML diagrams are as follows:

1. Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified. Now when the initial task is complete use case diagrams are modelled to present the outside view. So in brief, the purposes of use case diagrams can be as follows:

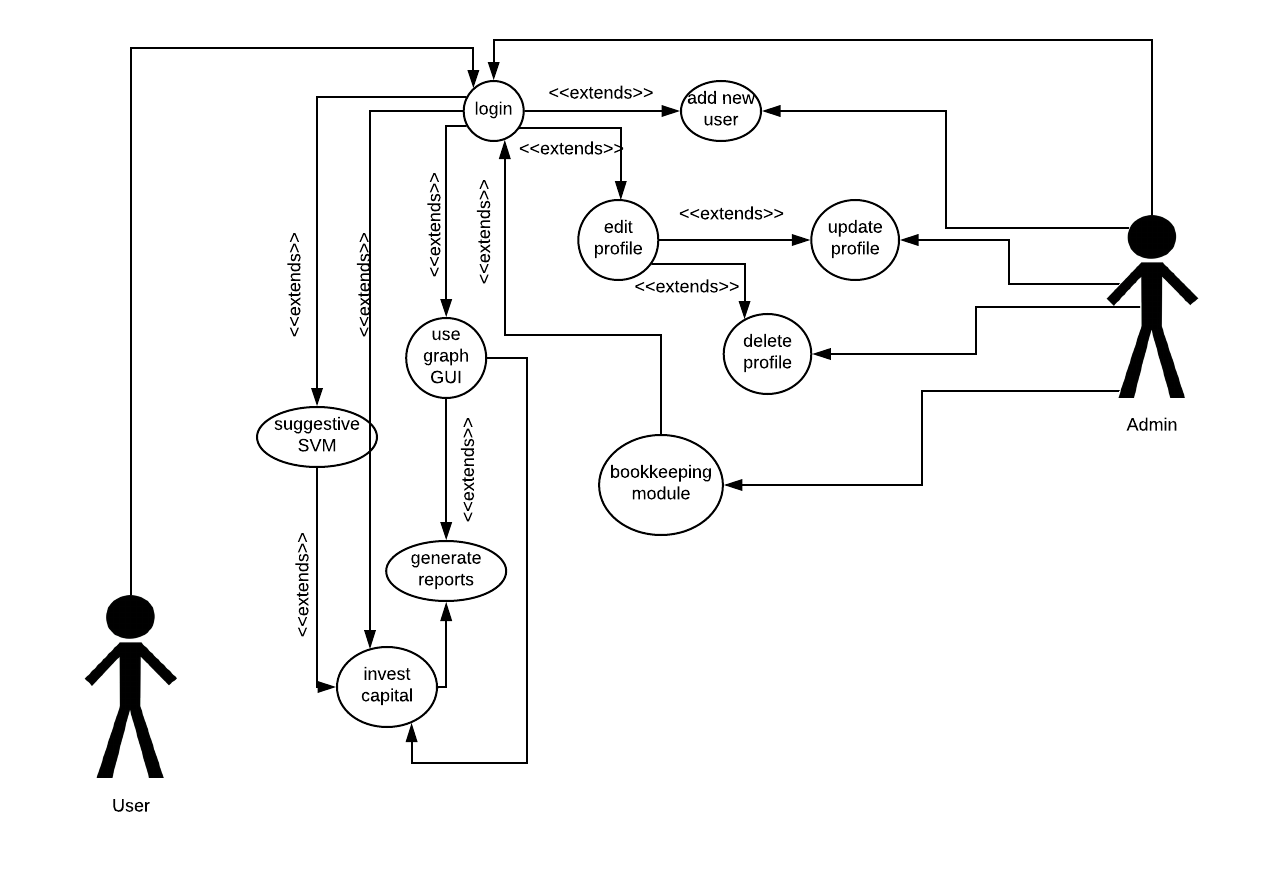
 Used to gather requirements of a system.

 Used to get an outside view of a system.

 Identify external and internal factors influencing the system.

 Show the interacting among the requirements are actors.

**Use case of different actors using the system:**



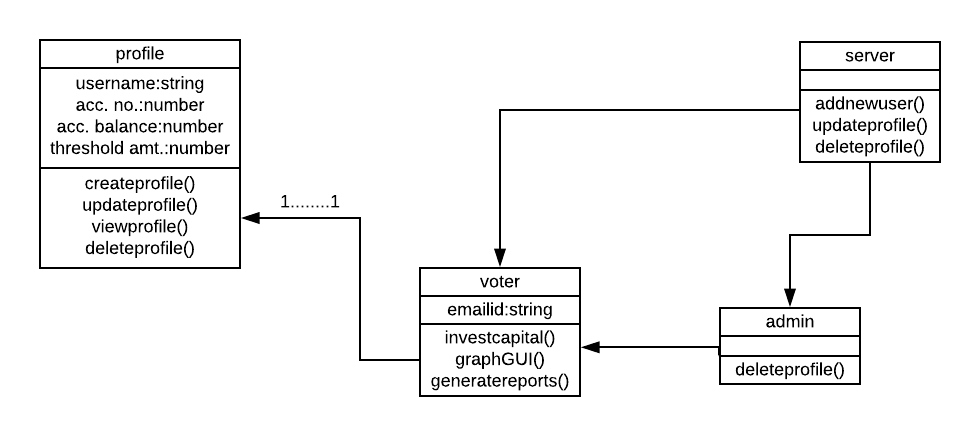
1. Class Diagram

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

• The top part contains the name of the class.

• The middle part contains the attributes of the class. They are left aligned and the first letter is lower case.

• The bottom part gives the methods or operations the class can take or undertake. They are also left aligned and the first letter is lower case.



The class diagram shows how different classes of the system interact and exchange message in order to successfully implement the system.

1. Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modelling where they are used for detailing the processes involved in business activities.

**The following sections describe the elements that constitute an activity diagram.**

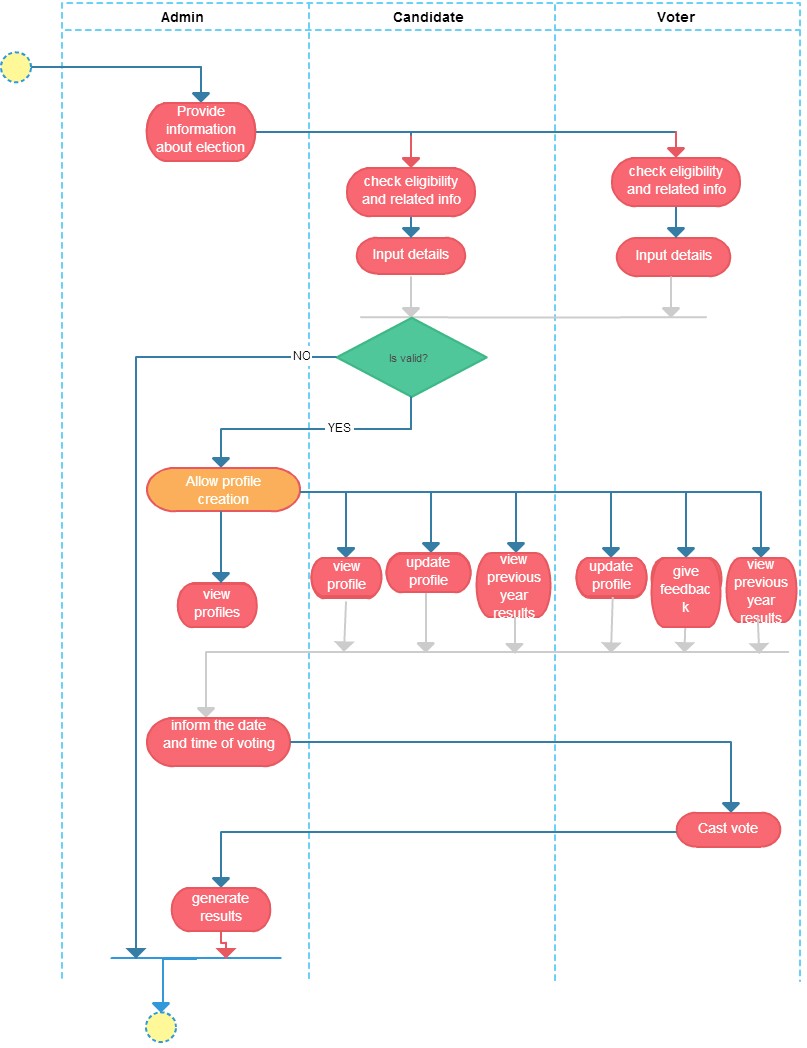
**Actions**: An action represents a single step within an activity. Actions are denoted by round-cornered rectangles.

**Control Flow**: A control flow shows the flow of control from one action to the next. Its notation is a line with an arrowhead.

**Initial Node**: An initial or start node is depicted by a large black spot.

**Final Node**: There are two types of final node: activity and flow final nodes. The activity final node is depicted as a circle with a dot inside**.**

**Decision and Merge Nodes**: Decision nodes and merge nodes have the same notation: a diamond shape. They can both be named. The control flows coming away from a decision node will have guard conditions which will allow control to flow if the guard condition is met.

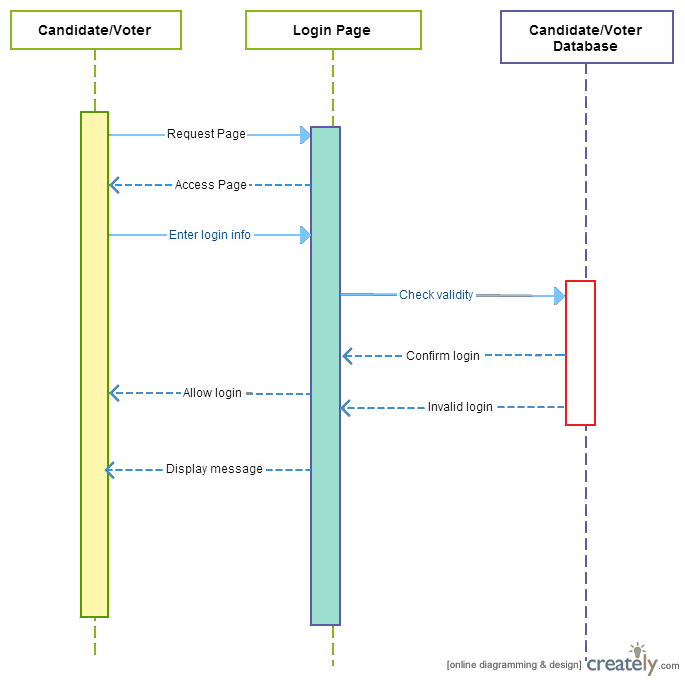
**Fork and Join Nodes**: Forks and joins have the same notation: either a horizontal or vertical bar. They indicate the start and end of concurrent threads of control.

The activity diagram shows how the activities flow in order to successfully execute the system.

1. Sequence Diagram

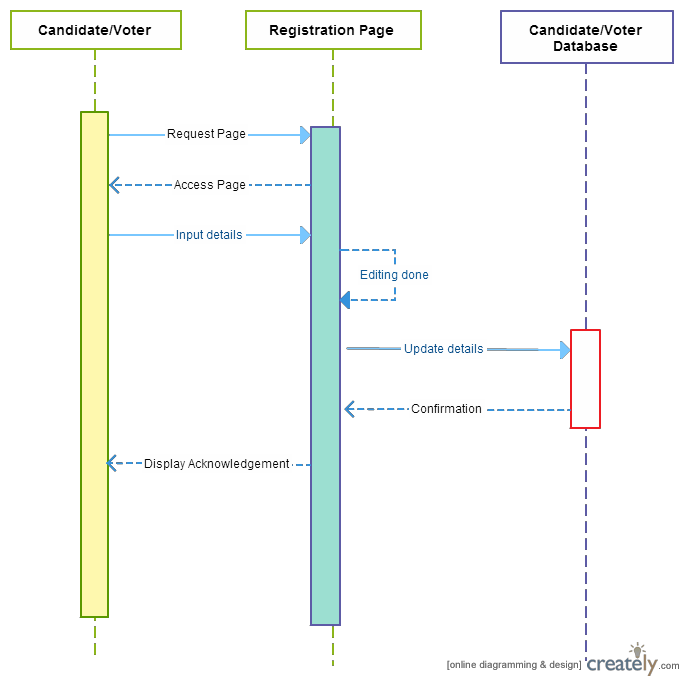
Sequence diagram shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

# LOGIN MODULE

This sequence diagram carries out the functionality of the Login Page and shows how the investor uses the login page and how the user interacts with the database for a successful or unsuccessful login. 

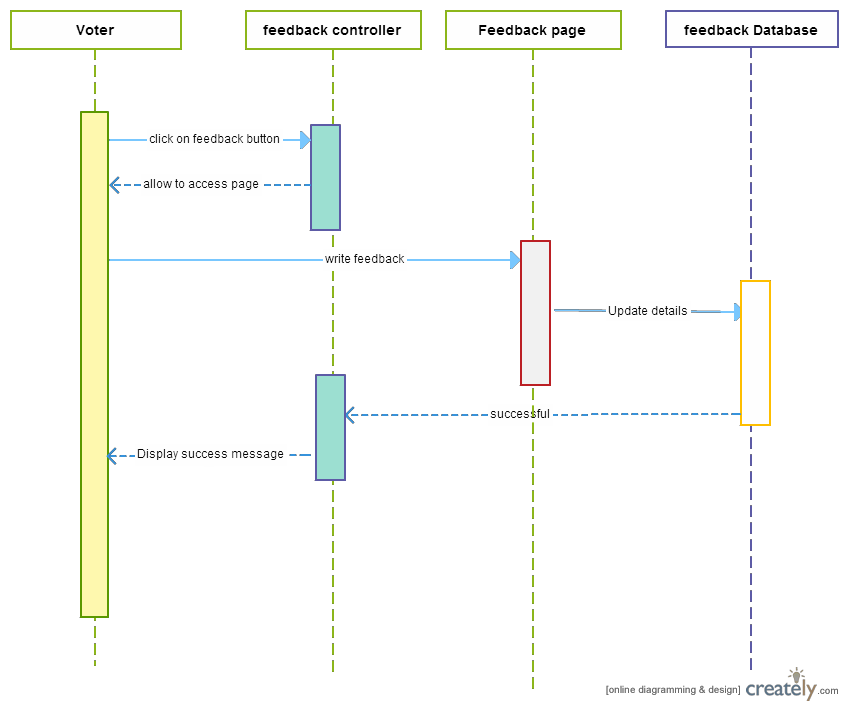
# SIGNUP/REGISTRATION MODULE

This sequence diagram carries out the functionality of the Registration System and shows how the investor uses the registration page and how he/she interacts with database to register themselves in the portal.



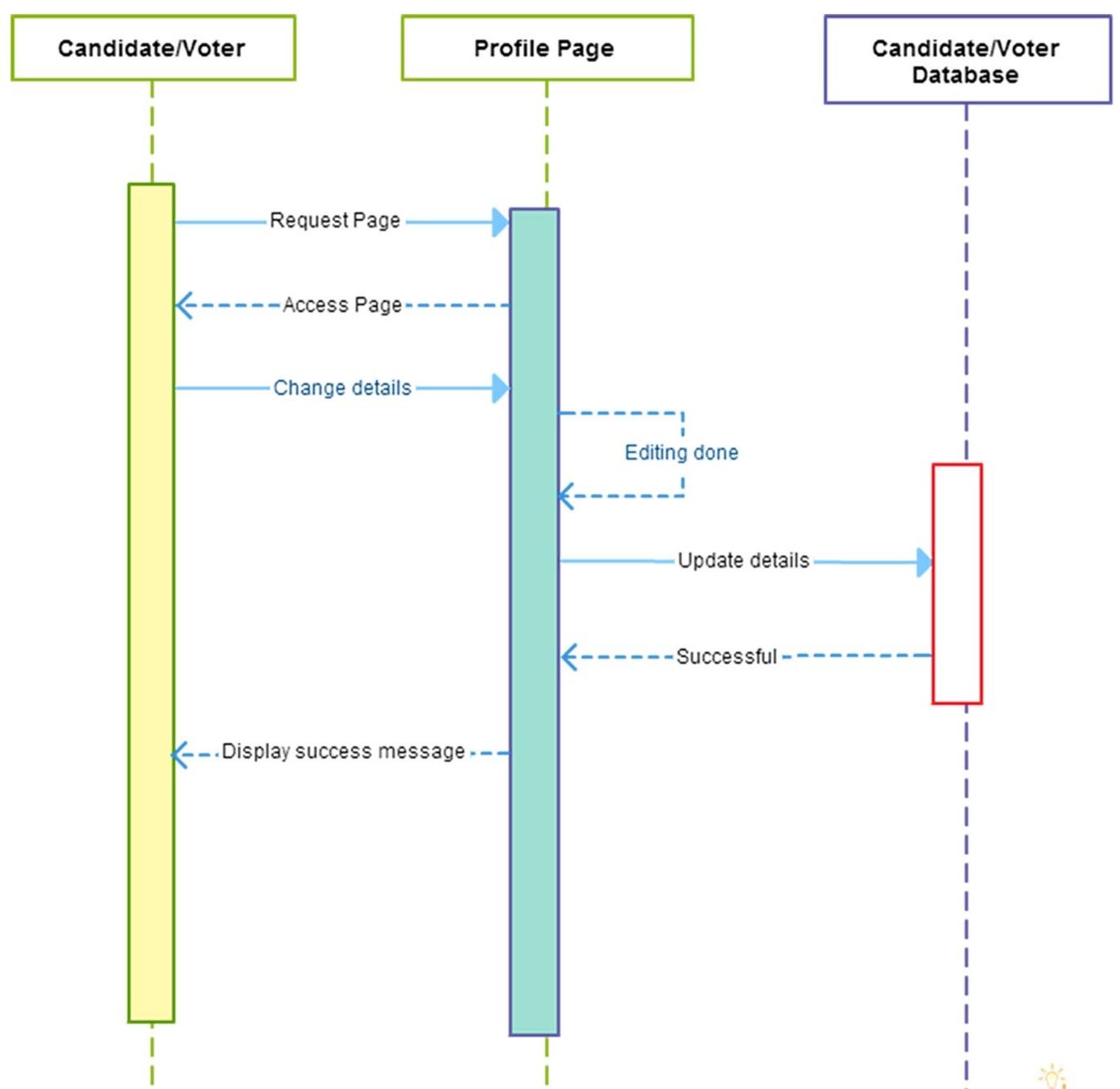
# FEEDBACK MODULE

This sequence diagram carries out the functionality of the Feedback System and shows how the vote, the feedback controller, feedback system and feedback database exchange messages in order to successfully give a feedback.



# UPDATE SYSTEM:

This sequence diagram carries out the functionality of the Update System and shows how the user, the profile page and the user database exchange messages in order to update details of their profile.

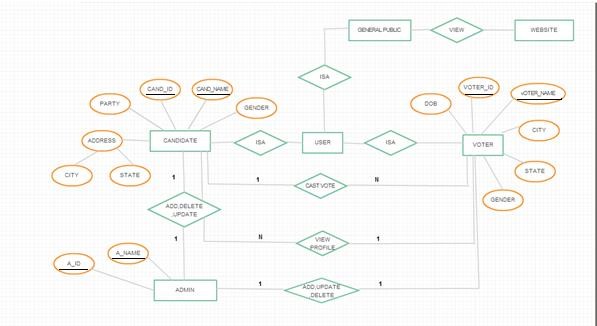


1. Database Design

Database designs the design of how persistently stored so that it can be accessed by different programs and users over a period of time.

1. ER DIAGRAM

In software engineering, an entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them. It has been reduced till second normal form (2NF) as it in 1NF and has no non-prime attribute which is dependent on any proper subset of any candidate key.



This ER Diagram shows how different entities like admin, and user have dependency upon each other like one-to-many, many-to-many etc.

1. Table Design

The table diagram allows you to visualize a database to which you are connected. It shows all the attributes of a database along with its type and constraints.

# DATABASE TABLE:

The different database tables for the Online Voting System are as follows:

# ATTRIBUTES OF ADMIN:

**TABLE 1 ATTRIBUTES OF ADMIN**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **CONSTRAINT** |
| Username | varchar(10) | Primary key,Not Null |
| Password | varchar(32) | Not Null |
| first name | varchar(10) | Not Null |
| last name | varchar(10) | Not Null |

The table diagram shows the attributes of the Admin of the system

# ATTRIBUTES OF investor:

**TABLE 2 ATTRIBUTES OF INVESTOR**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **CONSTRAINT** |
| user\_id | varchar(32) | Primary Key,Not Null |
| Name | varchar(32) | Not Null |
| City | varchar(32) | Not Null |
| Dob | Date | Not Null |
| party | varchar(32) | Not Null |
| Votes | int(11) |  |
| password | varchar(32) | Not Null |

The table diagram shows the attributes of the investor of the system.

# FEEDBACK MODULE:

**TABLE 4 FEEDBACK MODULE FOR ONLINE VOTING SYSTEM**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **CONSTRAINT** |
| s.no | Int (11) | Unique, Not Null |
| Uid | Int (11) | Unique, Not Null |
| firstname | varchar(32) | Not Null |
| lastname | varchar(32) | Not Null |
| email | varchar(32) | Not Null |
| Date | date | Not Null |
| comment | varchar(1024) | Not Null |

The table diagram shows the attributes of the feedback mechanism of the system.

1. Test Case Design

A test case, in software engineering, is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

1. LOGIN

**TABLE 5 TEST CASES FOR LOGIN MODULE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no** | **Test case name** | **Test Procedur e** | **Pre- condition** | **Expected Result** | **Reference** |
| **1.** | all three text  box empty | No  credential | Login page  is loaded | Invalid credentials  message popup | Login |
| **2.** | Username ok but password &  UID field empty | Only username | Login page is loaded | Empty password and UID field message popup | Login |
| **3.** | Password ok but username and UID  field empty | Only password | Login page is loaded | Empty username and UID field message popup | Login |
| **4.** | UID ok but username & password field empty | Only UID | Login page is loaded | Empty username and password  field message popup | Login |
| **5.** | Username & password ok but UID field empty | Username & password | Login page is loaded | Empty UID  field message popup | Login |
| **6.** | Password and UID ok but  username field empty | Password & UID | Login page is loaded | Empty username field message popup | Login |
| **7.** | Username & UID ok but password  field empty | Username & UID | Login page is loaded | Empty password field | Login |

1. FEEDBACK MODULE:

**TABLE 7 TEST CASES FOR FEEDBACK BUTTON**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Reference** |
|  |  |  |  |  |  |
| **1.** | Given feedback | Call feedback page with feedback  given | Voted page is loaded | Given feedback successfully | Feedback |
| **2.** | No feedback | Call feedback page with no  feedback | Voted page is loaded | Not given feedback | Feedback |

# 3)

3) ADMIN MODULE:

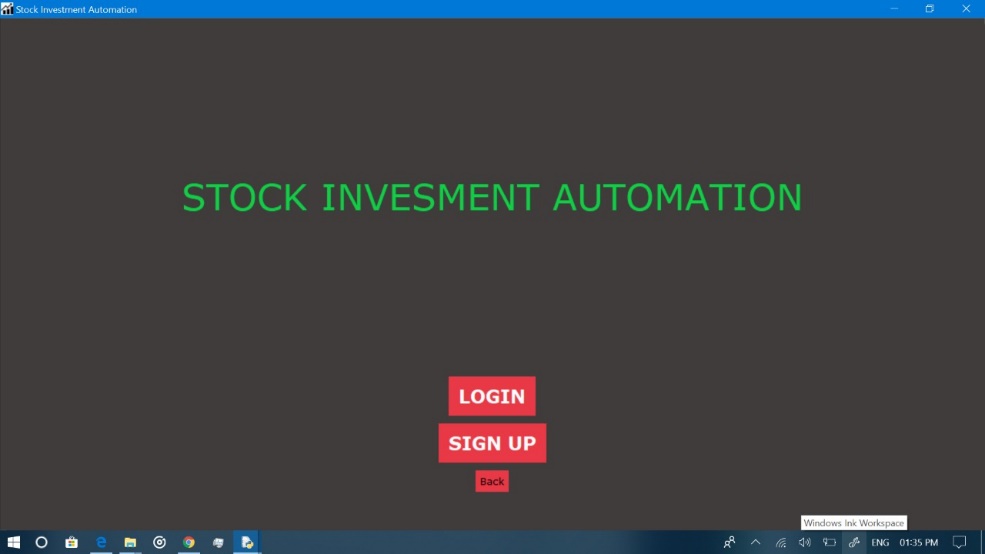
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Reference** |
|  |  |  |  |  |  |
| **1.** | Schedule | Call scheduling  page | Login page is  loaded | Scheduled successfully | Scheduling |
| **2.** | View voter’s details | Call details page for the  voter | Scheduled page is loaded | Viewed successfully | Viewing voter’s details |
| **3.** | View  candidate’s details | Call details page for the  candidate | Scheduled page is loaded | Viewed successfully | Viewing candidate’s details |
| **4.** | View live results | Call live results Page | Voter’s/Candidate’s details page  is loaded | Viewed successfully | Viewing live results |

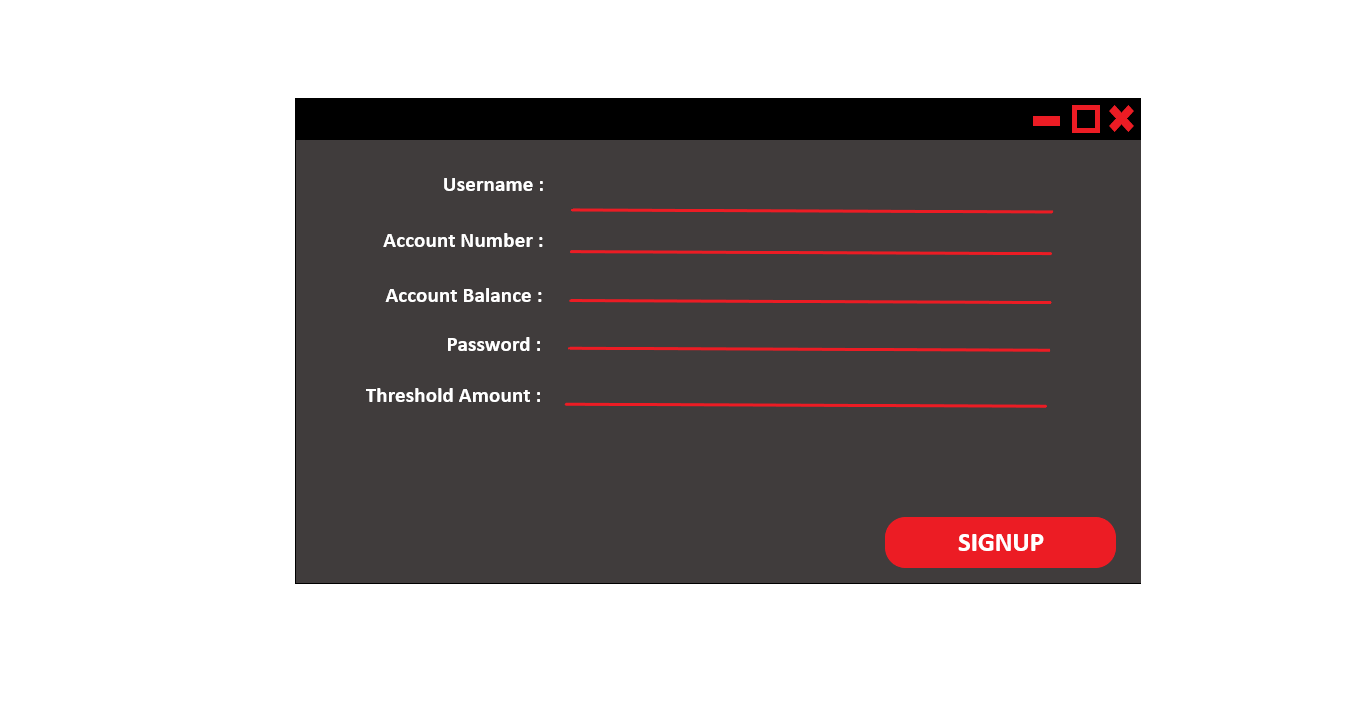
1. GUI Design

A graphical user interface or GUI, is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.

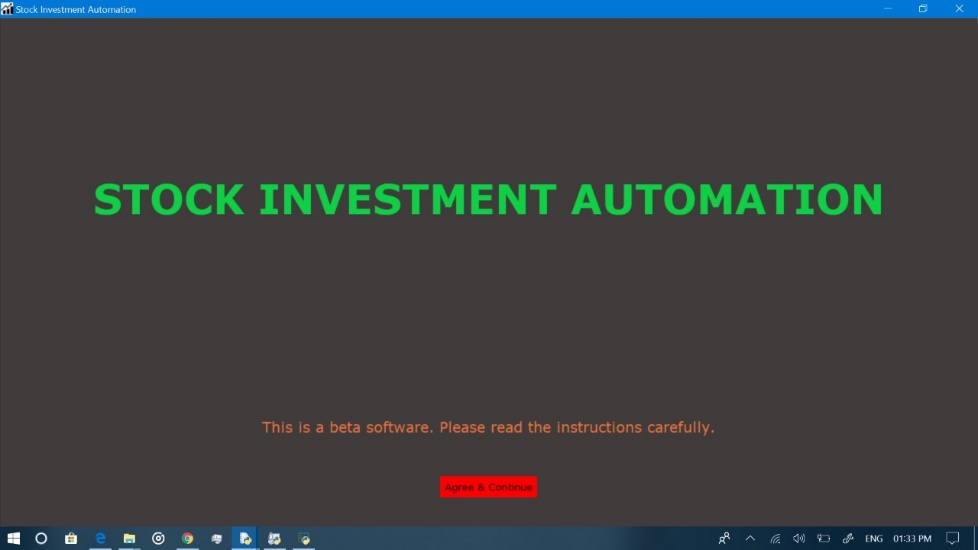
1. LOGIN/SIGNUP PAGE

This is the first window of the software where the registered user can login into the software, if not registered then they can sign up and get themselves registered to use this portal. After they have successfully signed up, each user will be provided with a unique (unique identification) profile. With the login details, each user have to provide the email and password at the time of login. If all the credentials matches with the credential saved in the database, the user is directed to the home page of the software and based on the type of user (i.e. admin and investor) they can use the features of the portal.





1. HOME PAGE



# TESTING

1. Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

We have done two phases of testing, in the first phase some of the functionalities were not working correctly but in second phase all the functionalities were successfully implemented.

# PHASE-1 LOGIN:

**TABLE 9 TEST CASES FOR LOGIN MODULE FOR PHASE-1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no** | **Test case name** | **Test Procedure** | **Pre- conditio**  **n** | **Expected Result** | **Refer ence** | **Actual Result** | **Status** |
| **1.** | all three text box empty | No credential | Login page is loaded | Invalid credentials message  popup | Login | Invalid credentials message  popup | Successful |
| **2.** | Username ok but password & UID  field empty | Only username | Login page is loaded | Empty password and UID field message  popup | Login | Empty password and UID field message  popup | Unsuccess ful |
| **3.** | Password ok but username and UID field empty | Only password | Login page is loaded | Empty username and UID field message  popup | Login | Empty username and UID field message  popup | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **4.** | UID ok but | Only UID | Login | Empty | Login | Empty | Successful |
|  | username |  | page is | username |  | username |  |
|  | & |  | loaded | and |  | and |  |
|  | password |  |  | password |  | password |  |
|  | field empty |  |  | field |  | field |  |
|  |  |  |  | message |  | message |  |
|  |  |  |  | popup |  | popup |  |
| **5.** | Username | Username | Login | Empty UID | Login | Empty UID | Unsuccess |
|  | & | & | page is | field |  | field | ful |
|  | password | password | loaded | message |  | message |  |
|  | ok but UID |  |  | popup |  | popup |  |
|  | field empty |  |  |  |  |  |  |
| **6.** | Password | Password | Login | Empty | Login | Empty | Successful |
|  | and UID | & UID | page is | username |  | username |  |
|  | ok but |  | loaded | field |  | field |  |
|  | username |  |  | message |  | message |  |
|  | field empty |  |  | popup |  | popup |  |
| **7.** | Username | Username | Login | Empty | Login | Empty | Successful |
|  | & UID ok | & UID | page is | password |  | password |  |
|  | but |  | loaded | field |  | field |  |
|  | password |  |  |  |  |  |  |
|  | field empty |  |  |  |  |  |  |

# FEEDBACK MODULE:

**TABLE 11 TEST CASES FOR FEEDBACK BUTTON FOR PHASE-1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- conditio n** | **Expected Result** | **Refere nce** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Given feedback | Call feedback page with feedback  given | Voted page is loaded | Given feedback successfully | Feedback | Given feedback successfully | Successful |
| **2.** | No feedback | Call feedback page with no  feedback | Voted page is loaded | Not given feedback | Feedback | Not given feedback | Unsuccessf ul |

# ADMIN PORTAL:

**TABLE 12 TEST CASES FOR ADMIN PORTAL FOR PHASE-1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- condition** | **Expected Result** | **Referen ce** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Schedule | Call scheduling  page | Login page is  loaded | Scheduled  successfully | Scheduling | Scheduled  successfully | Successf  ul |
| **2.** | View voter’s details | Call details page for the  voter | Scheduled page is loaded | Viewed successfully | Viewing voter’s  details | Viewed successfully | Successf ul |
| **3.** | View  candidate’s  details | Call details page for the  candidate | Scheduled page is loaded | Viewed successfully | Viewing  candidate’s  details | Viewed successfully | Unsucce ssful |
| **4.** | View live results | Call live results Page | Voter’s/Candidat  e’s details page is loaded | Viewed successfully | Viewing live results | Viewed successfully | Successf ul |

## PHASE-2: LOGIN:

**TABLE 13 TEST CASES FOR LOGIN MODULE FOR PHASE-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no** | **Test case name** | **Test Procedure** | **Pre- conditio**  **n** | **Expected Result** | **Refer ence** | **Actual Result** | **Status** |
| **1.** | all three text box empty | No credential | Login page is loaded | Invalid credentials message  popup | Login | Invalid credentials message  popup | Successful |
| **2.** | Username ok but password & UID  field empty | Only username | Login page is loaded | Empty password and UID field message  popup | Login | Empty password and UID field message  popup | Successful |
| **3.** | Password ok but username and UID field empty | Only password | Login page is loaded | Empty username and UID field message  popup | Login | Empty username and UID field message  popup | Successful |
| **4.** | UID ok but username & password field empty | Only UID | Login page is loaded | Empty username and password field  message popup | Login | Empty username and password field  message popup | Successful |
| **5.** | Username & password ok but UID field empty | Username & password | Login page is loaded | Empty UID field message popup | Login | Empty UID field message popup | Successful |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **6.** | Password and UID ok but username  field empty | Password & UID | Login page is loaded | Empty username field message  popup | Login | Empty username field message  popup | Successful |
| **7.** | Username | Username | Login | Empty | Login | Empty | Successful |
|  | & UID ok | & UID | page is | password |  | password |  |
|  | but |  | loaded | field |  | field |  |
|  | password |  |  |  |  |  |  |
|  | field empty |  |  |  |  |  |  |

## FEEDBACK MODULE:

**TABLE 15 TEST CASES FOR FEEDBACK BUTTON FOR PHASE-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl no**  **.** | **Test case name** | **Test Procedure** | **Pre- conditio n** | **Expected Result** | **Refere nce** | **Actual Result** | **Status** |
|  |  |  |  |  |  |  |  |
| **1.** | Given feedback | Call feedback page with feedback  given | Voted page is loaded | Given feedback successfully | Feedback | Given feedback successfully | Successful |
| **2.** | No feedback | Call feedback page with no  feedback | Voted page is loaded | Not given feedback | Feedback | Not given feedback | Successful |

# ADMIN PORTAL:

**TABLE 16 TEST CASES FOR ADMIN PORTAL FOR PHASE-2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sl no  . | Test case name | Test Procedure | Pre- condition | Expected Result | Referen ce | Actual Result | Status |
|  |  |  |  |  |  |  |  |
| 1. | Schedule | Call scheduling  page | Login page is  loaded | Scheduled  successfully | Scheduling | Scheduled  successfully | Successf  ul |
| 2. | View investor’s details | Call details page for the  investor’s | Scheduled page is loaded | Viewed successfully | Viewing investor’s  details | Viewed successfully | Successf ul |
| 3. | View  investor’s details | Call details page for the  investor | Scheduled page is loaded | Viewed successfully | Viewing  investor’s details | Viewed successfully | Successful |
| 4. | View live results | Call live results Page | Investor’s details page is loaded | Viewed successfully | Viewing live results | Viewed successfully | Successful |

1. FUTURE WORK:

This web software on automated stock investment so far allows only one way authentication of the user i.e. through their username, password etc. Right now only basic functionalities has been provided i.e. user registration, updating user profile, session check in order to avoid duplication of accounts, direct and simple Bitcoin investment.

As an extension of this application we plan to incorporate a two way authentication system of the user. First through a usual way of authentication i.e. username and password.

Second with the help of real time phone PIN which will act as an intermediate layer between the first authentication system and home page so that even if a person with someone else username and password tries to login would not be able to do so. This would minimize the forgery.

Other feature that we like to add subsequently is providing the database with an extra security so that no tampering with the transactions can be done in case someone tries to hack into this application.

# CONCLUSION: