Software Design Specification Document

Employee Management System using Blockchain

Group Members
Devansh Anhal- U101116FCS031- S2
Harsha Deuri- U101116FCS043- S2

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this Software Design Specification document is to provide written description of the product “Data Storage using Blockchain” along with proper guidance about the architecture of the project. This document will assist in the development of the project through the various narrative and graphical models of representation. It will contain the specific input output functions as well the interacting classes and working of the methods

1.2 Scope of the development project

- User can insert data into the system.
- Product will provide security to the user data, through blockchain technology
- User can compare his performance to that of others, or evaluate his own performance through self-comparison.
- User can also use predictive analysis to further asses his quality of work

1.3 Definitions, acronyms, and abbreviations

- IEEE: Institute of Electrical and Electronics Engineers
- SDS: Software Design Document

1.4 References

- IEEE Software Design Document Template

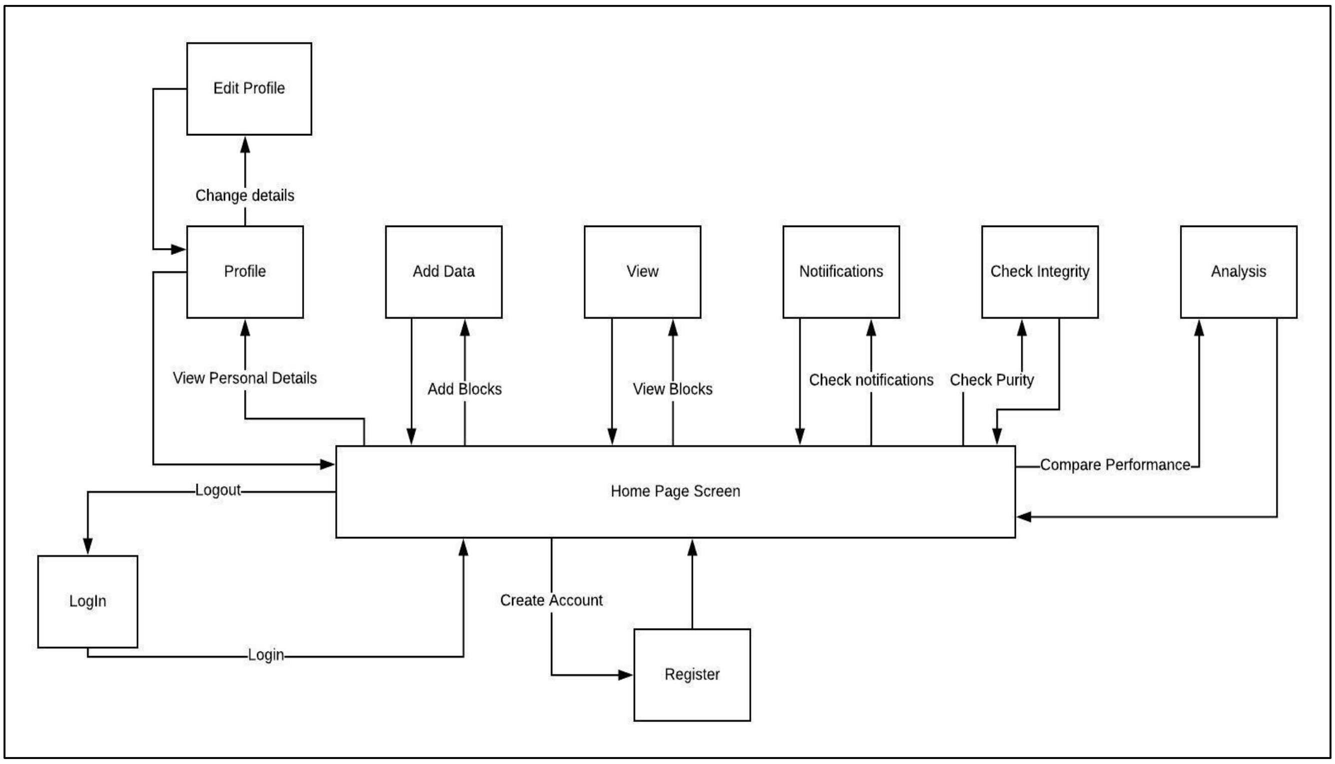
1.5 Overview of document

The SDS is divided into 7 sub categories.

1. **Introduction**- Gives a description about what to expect in the SDS, the product scope as well as any acronyms that might have been used or external material referred to.
2. **Conceptual Architecture/Architecture Diagram:** Contains the overview of modules/components along with the structure of the product and discusses any user interface issues
3. **Logical Architecture** Contains the architectural layout of the product. A detailed working of the entire diagram is discussed through UML diagrams such as Class Diagrams, Sequence Diagrams and State Diagrams.
4. **Execution Architecture:** Discusses the reusability of the product. Also, defines the runtime environment, processes and deployment view.
5. **Design decisions and tradeoffs:** Contains details that would help anyone reading the document understand the design. It contains details about certain ideas that floated during the planning process, but did not ultimately make the cut

2. Conceptual Architecture/Architecture Diagram

2.1 Structure and relationships



Note:

- The boxes represent dedicated screens.
- The arrows represent actions to be performed in order to go to the screen.

2.2 User interface issues

The product is aimed for the employees of a company, who may be between an age range of 24-60 years. Of course, this also means they will be having varying proficiency with technology.

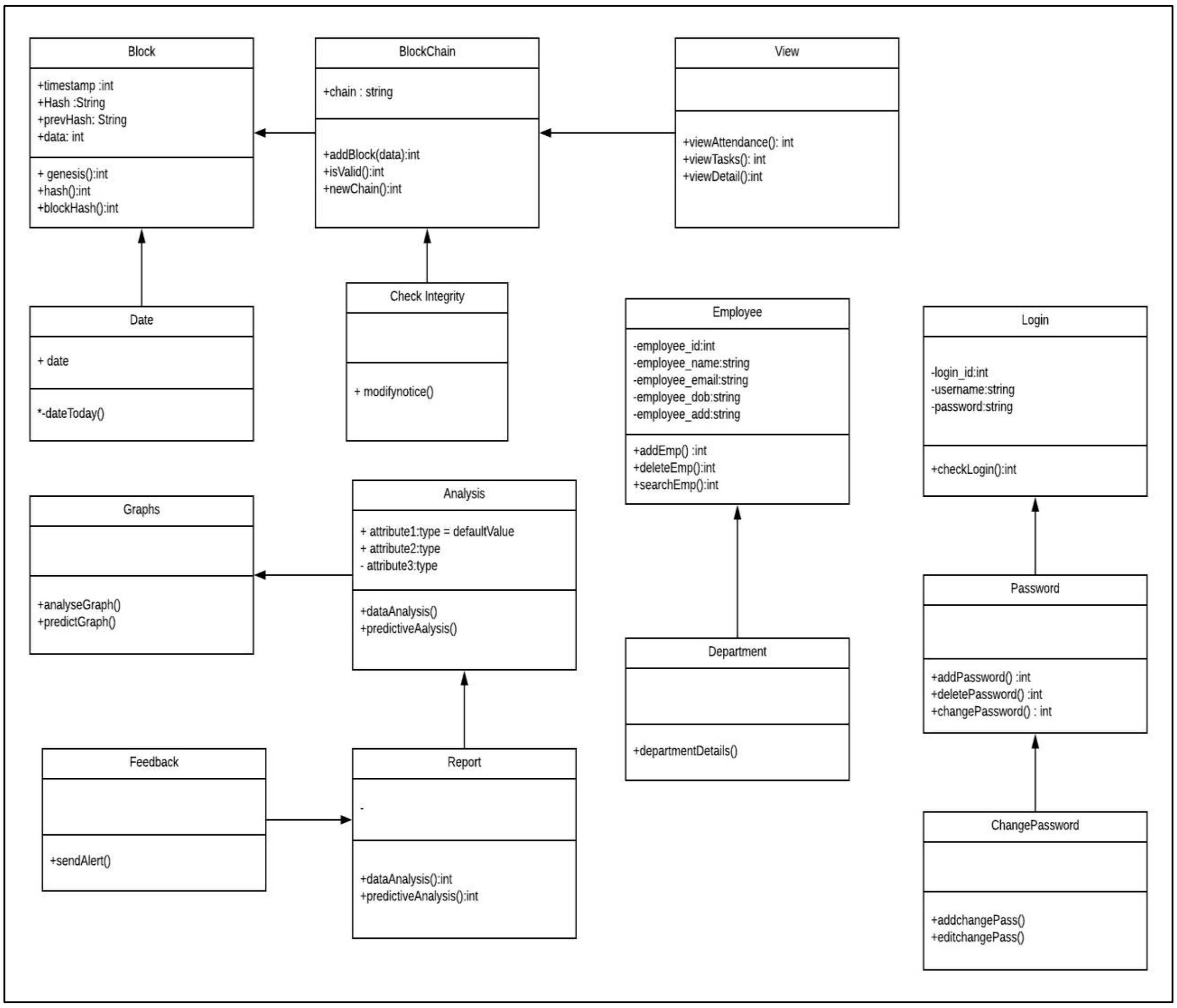
A fresher, who is just out of college may be abler in terms of understanding the new technologies used in the product whereas a person in a lower job position such as an attendant will not understand the technicality but would rather rely on the interface to get his job done.

Hence, the user interface will be simple with easy to understand navigations. Some of these can be:

- To log in to the system, the user will have to simply click on the button “Log In”.
- To add his data user will click on the button “Add My Data”.
  Thus, the simple interface will be accommodating employees of different strata.
  The user interface is also set to have some special features.
- For better interpretation, the performance comparisons or analysis performed will be represented graphically along with figures.
- Customer will receive notifications and alerts for certain conditions as mentioned previously in the SRS.

3. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

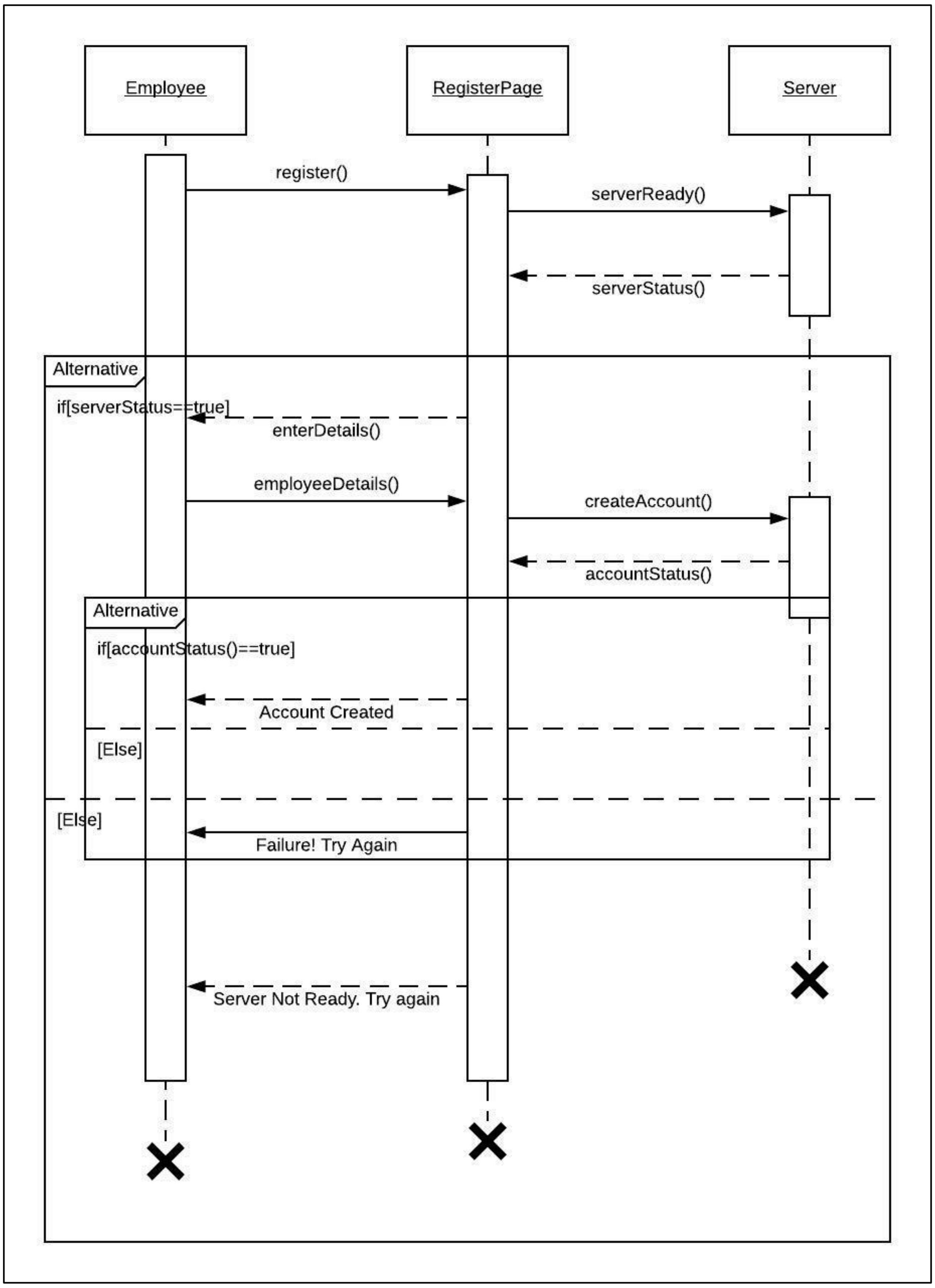
3.0.1 Class Diagram.



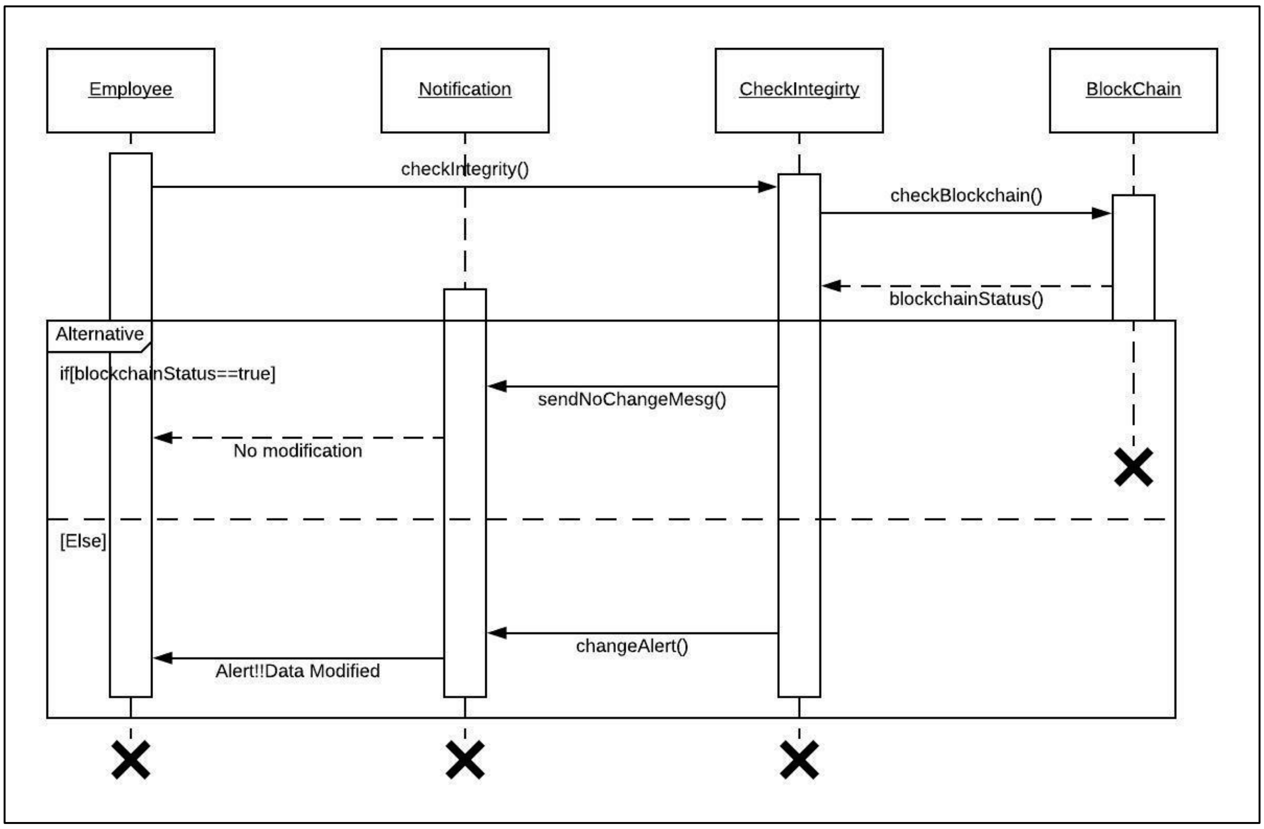
Note:

- The boxes represent classes.
- The arrows represent inheritance.

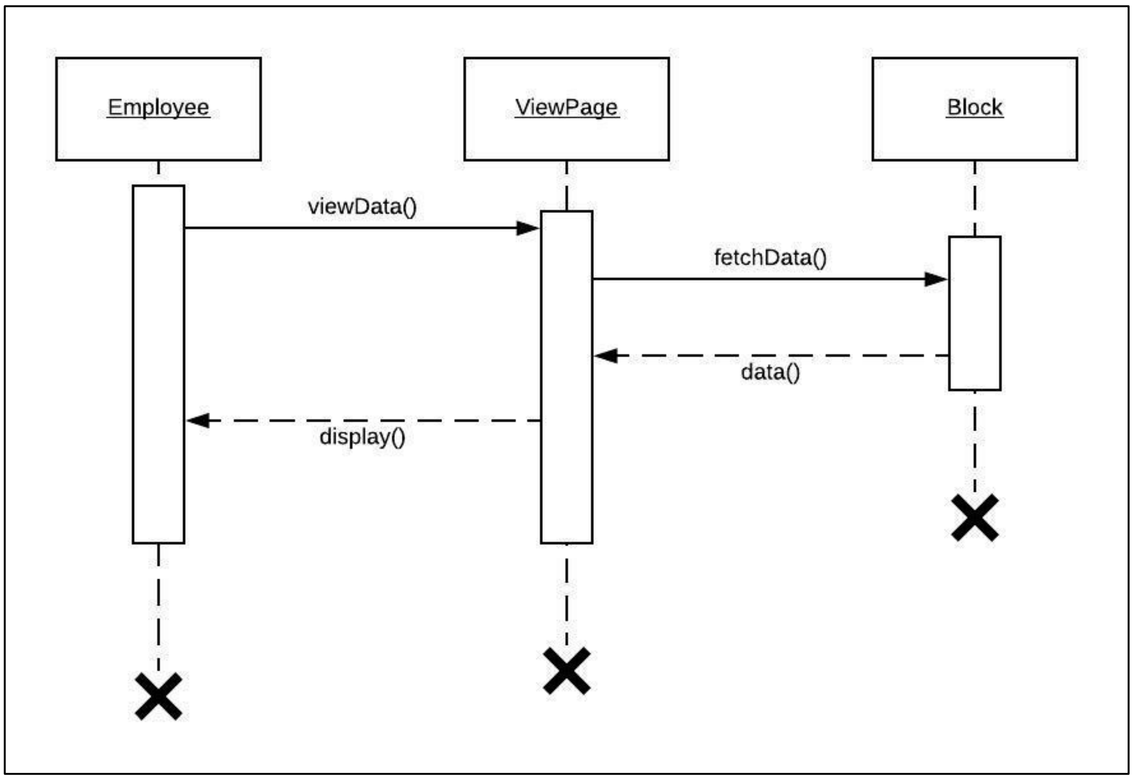
3.0.2 Sequence Diagram: Register Page



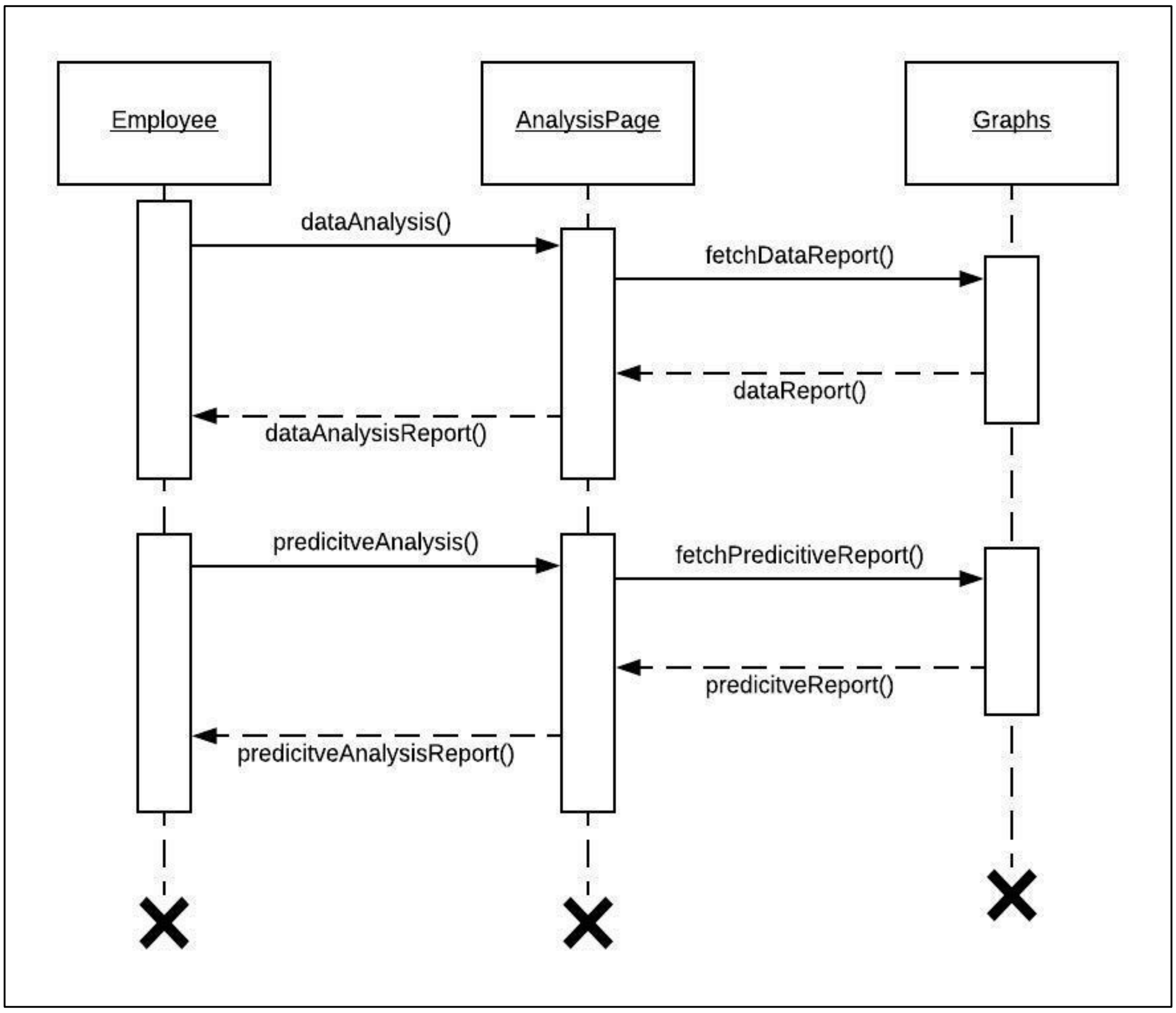
3.0.3 Sequence Diagram: CheckIntegrity Page



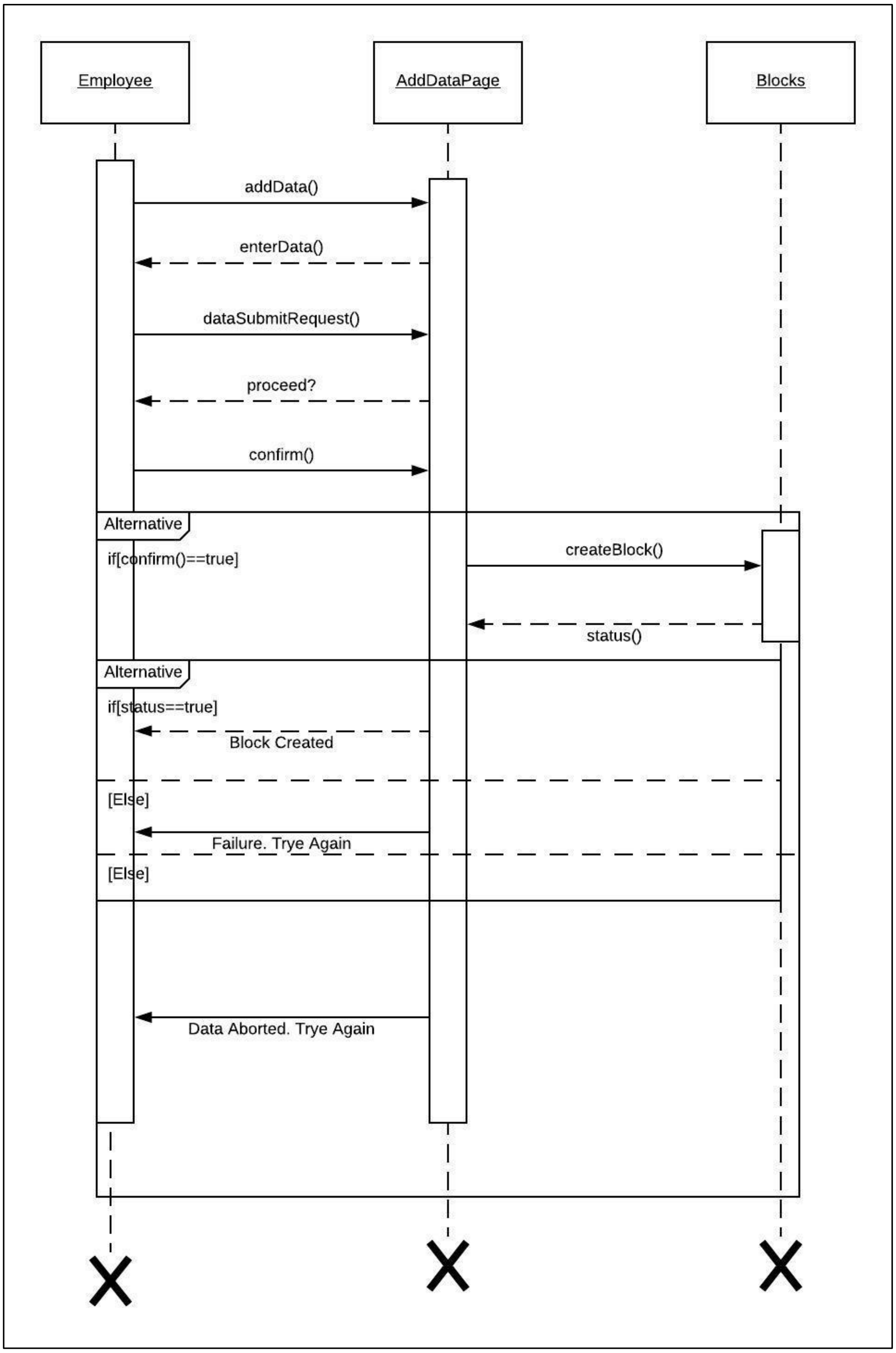
3.0.4 Sequence Diagram: View Page



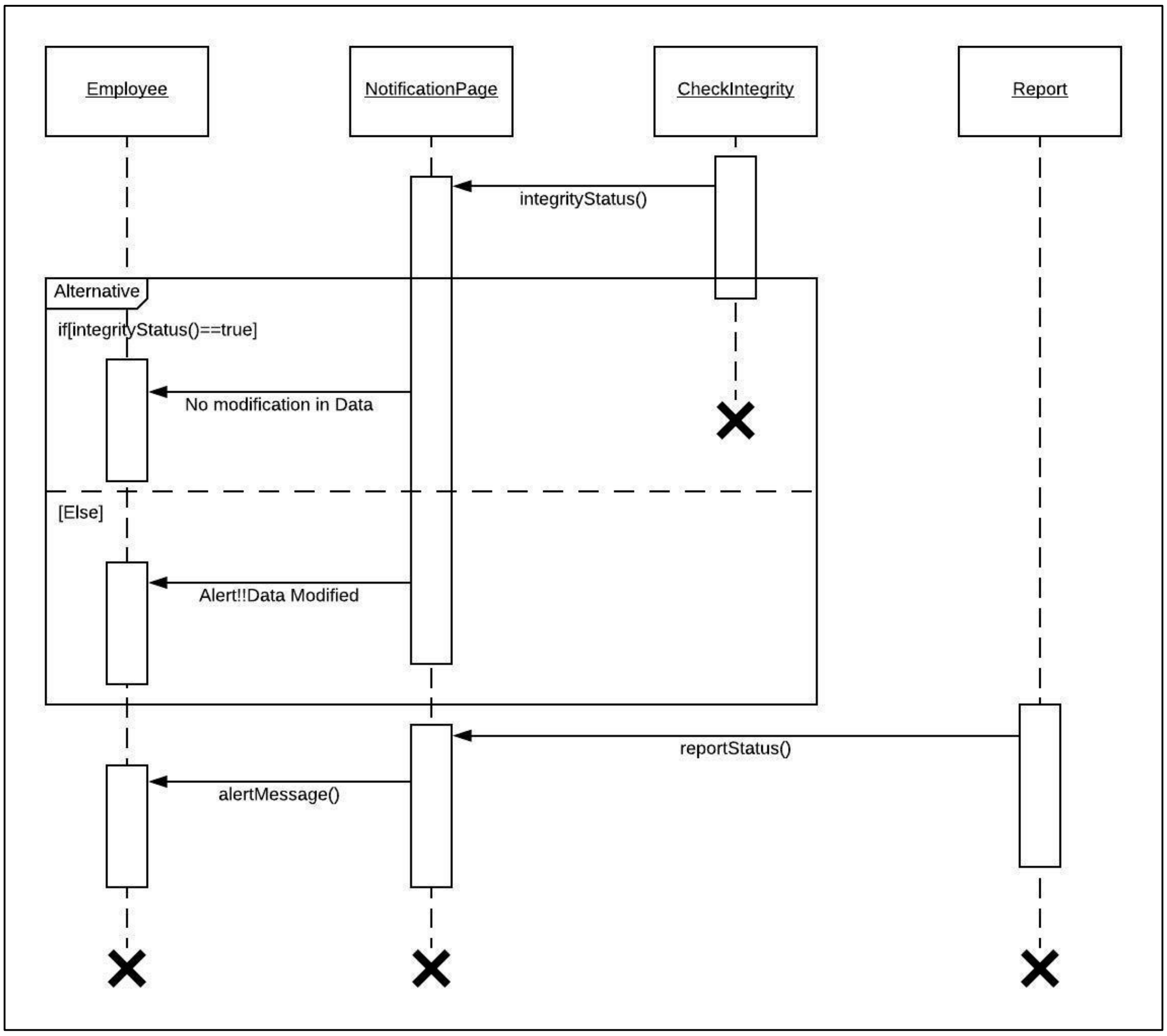
3.0.5 Sequence Diagram: Analysis Page



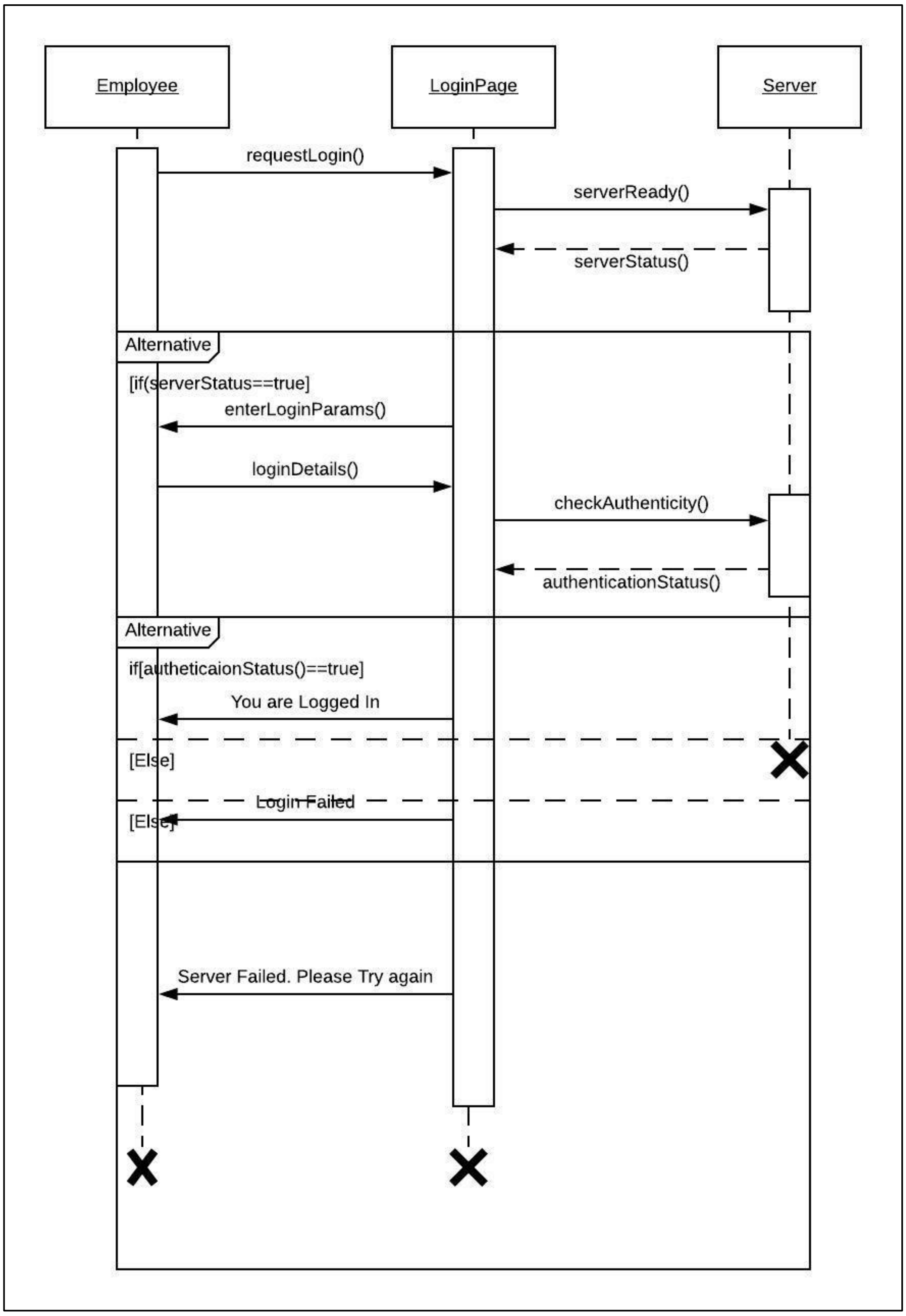
3.0.6 Sequence Diagram: AddData Page



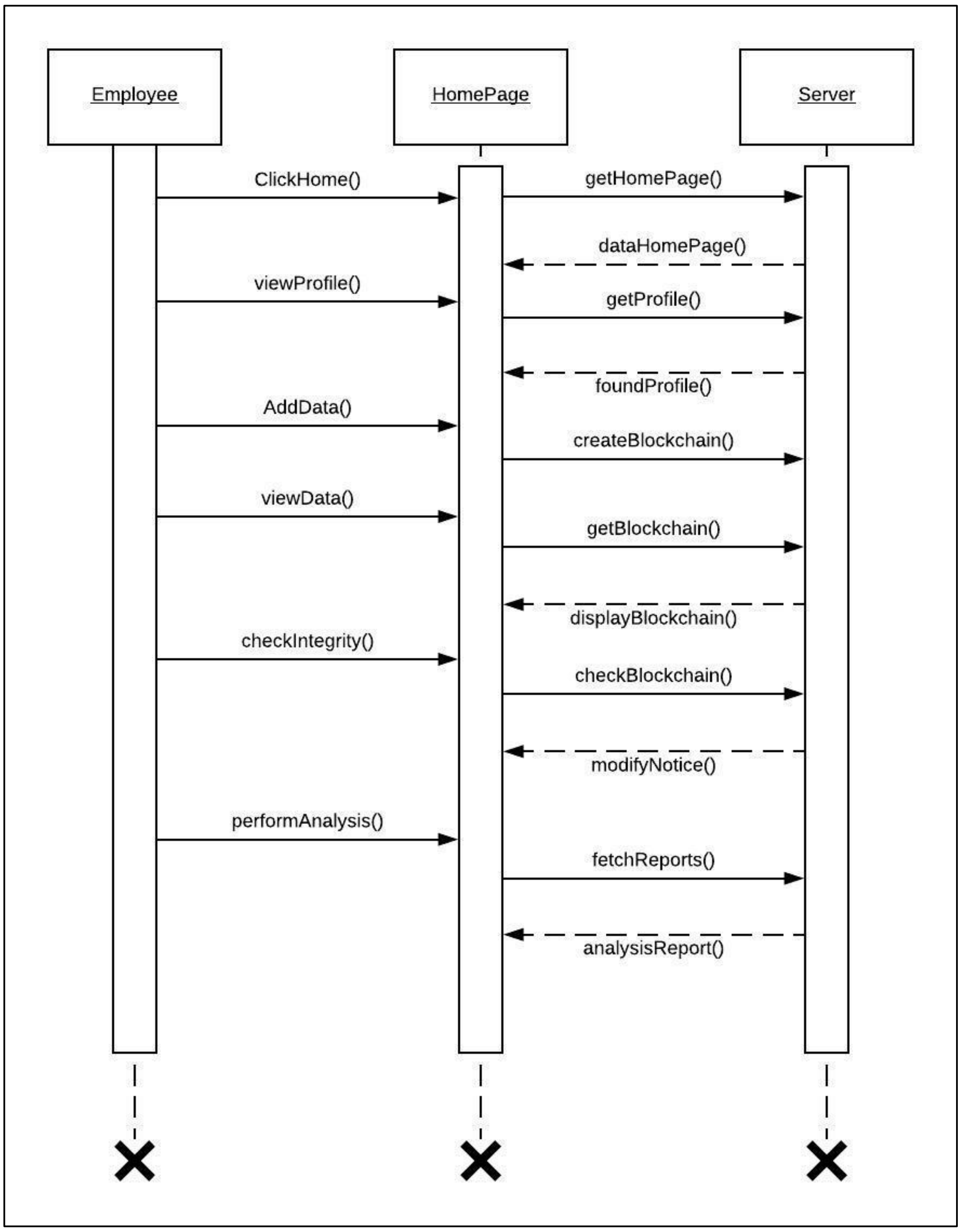
3.07 Sequence Diagram: Notification Page



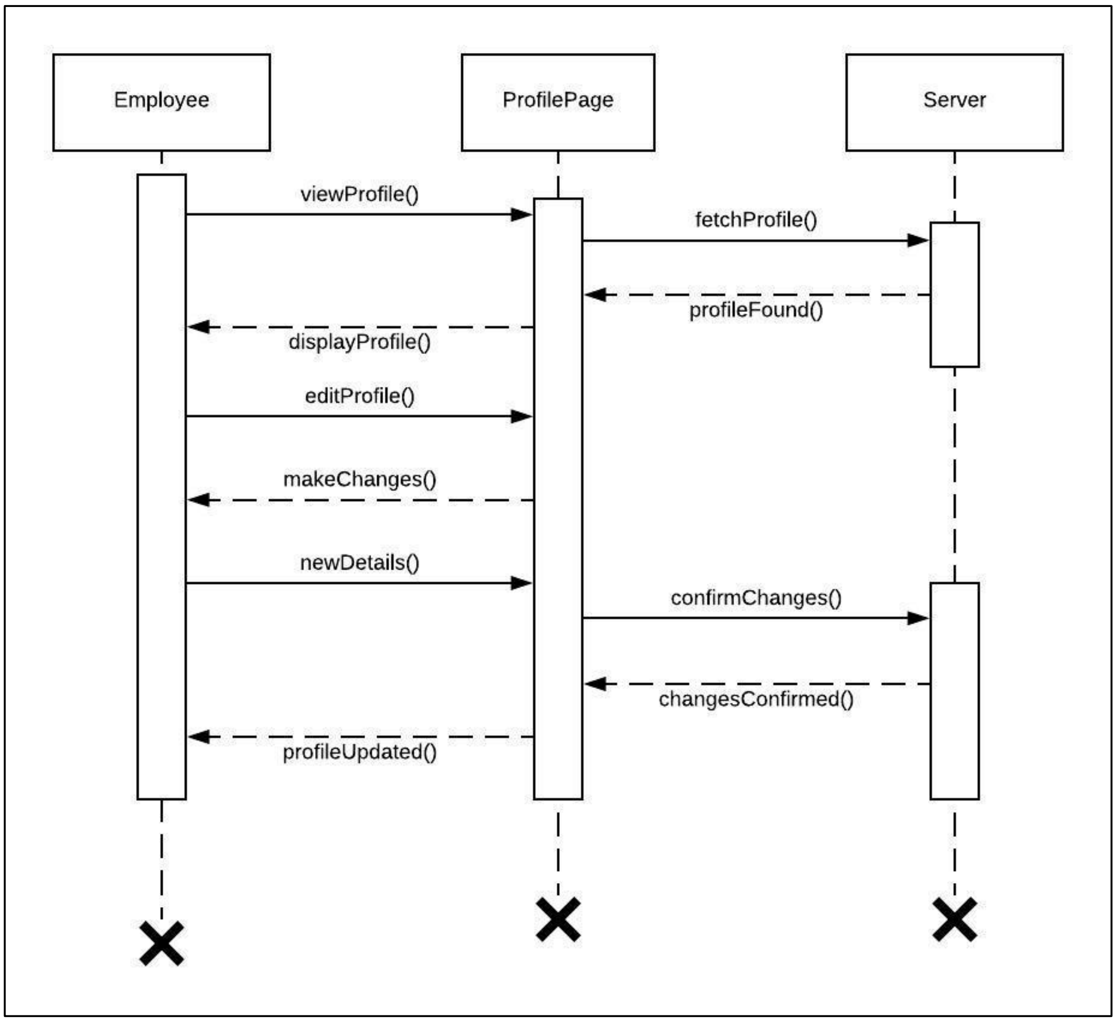
3.0.8 Sequence Diagram: Login Page



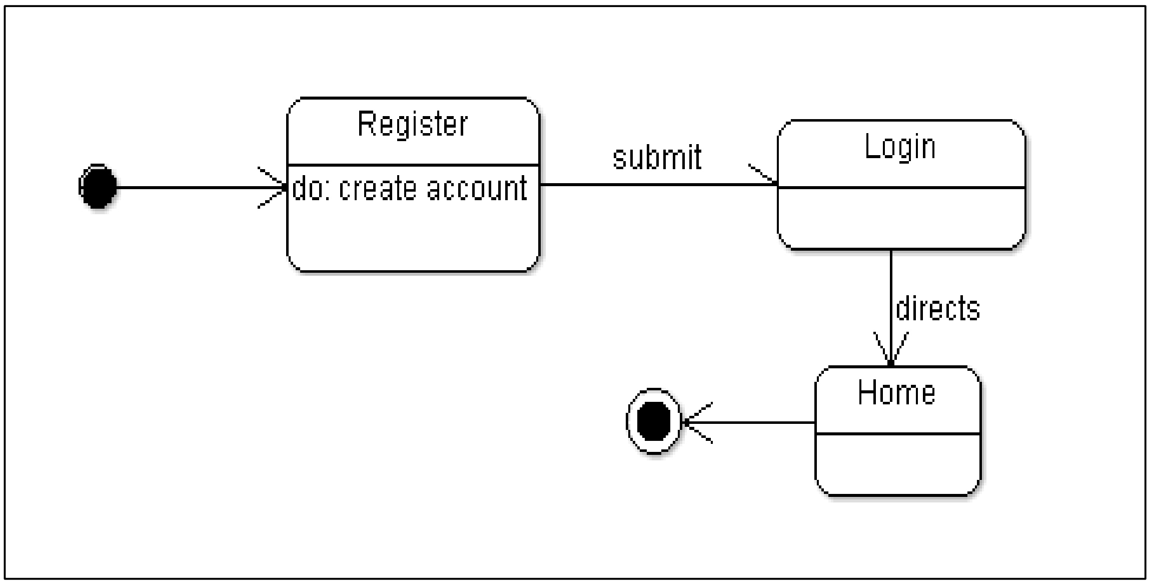
3.0.9 Sequence Diagram: Home Page



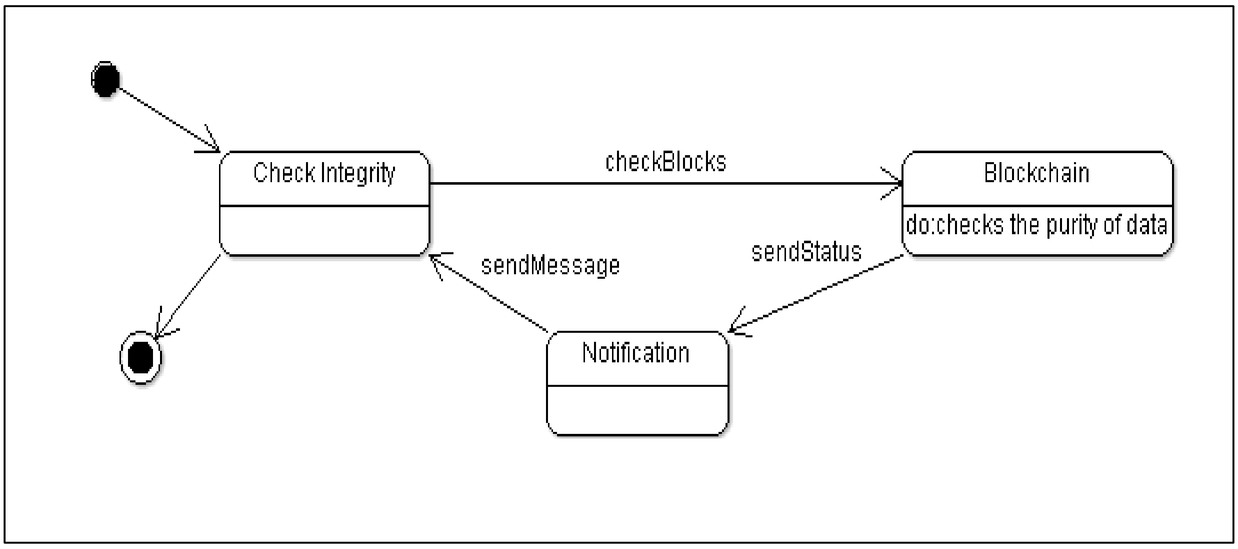
3.0.10 Sequence Diagram: Profile Page



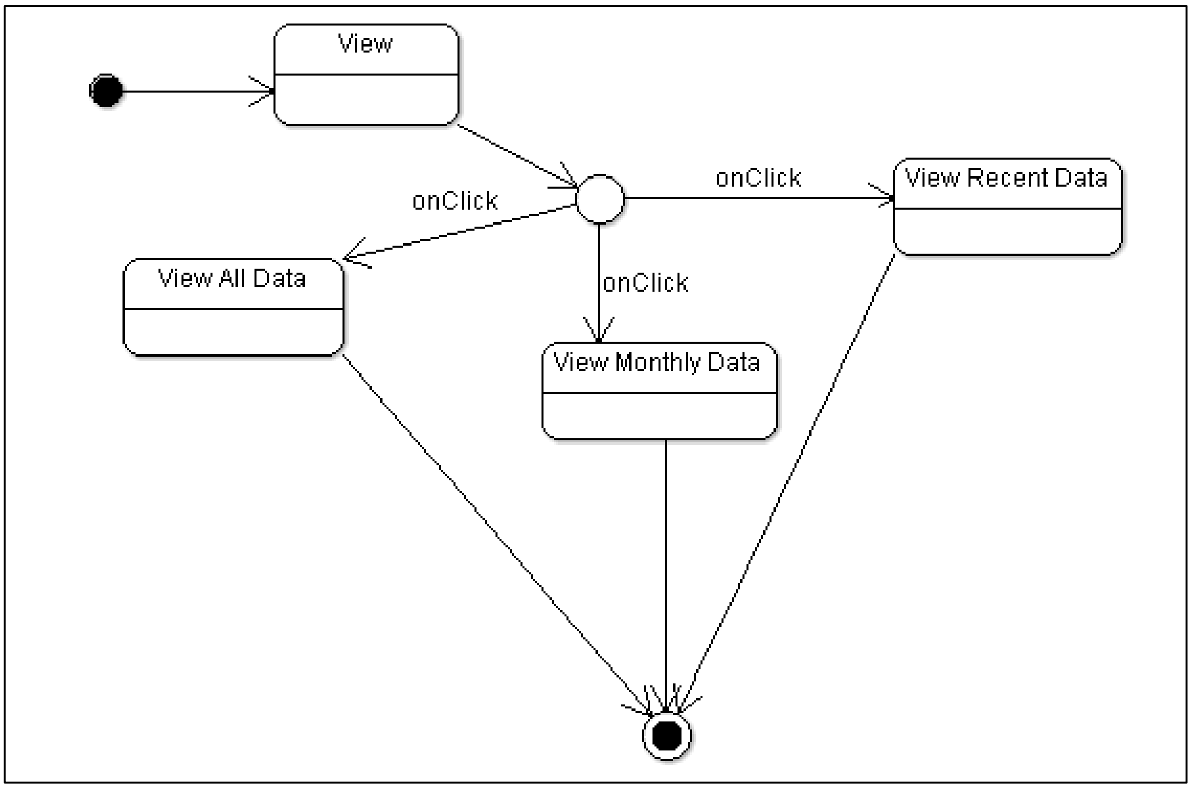
3.0.11 State Diagram: Register



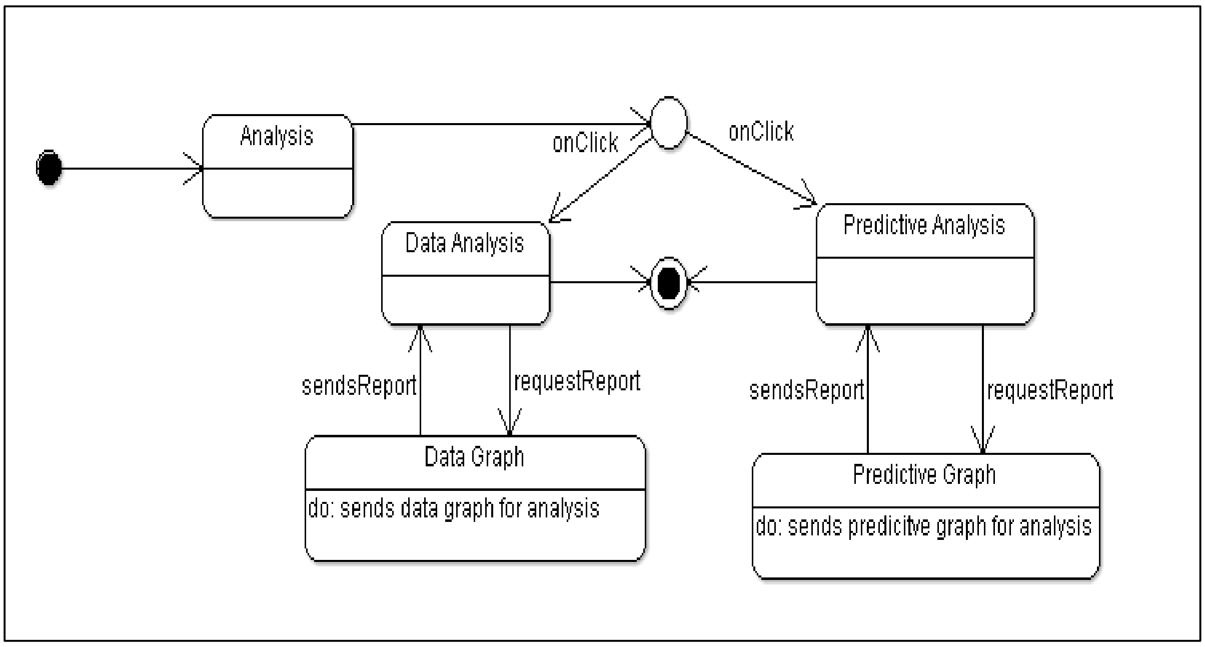
3.0.12 State Diagram: CheckIntegrity



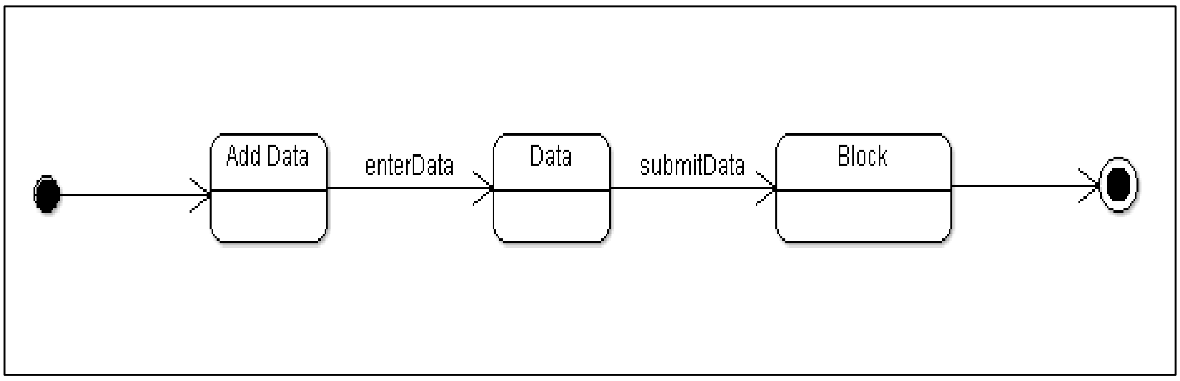
3.0.13 State Diagram: View



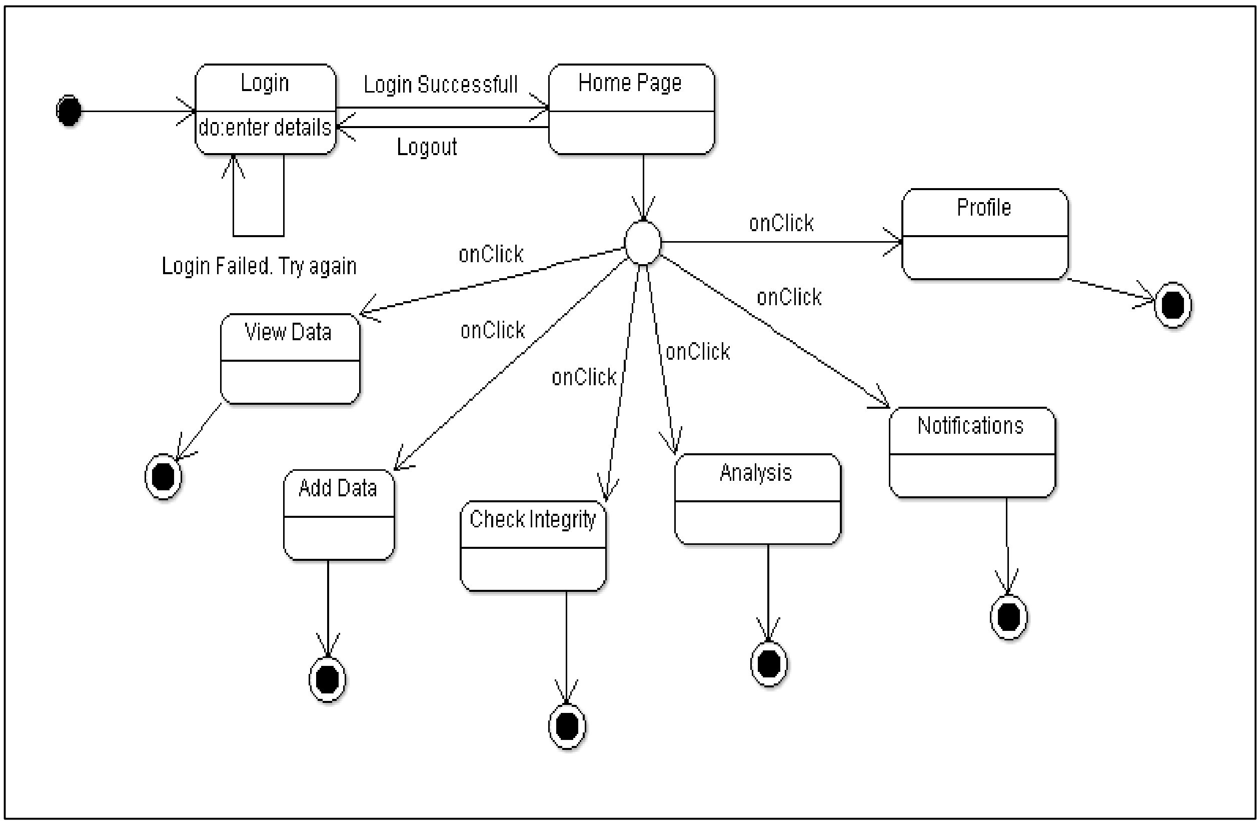
3.0.14 State Diagram: Analysis



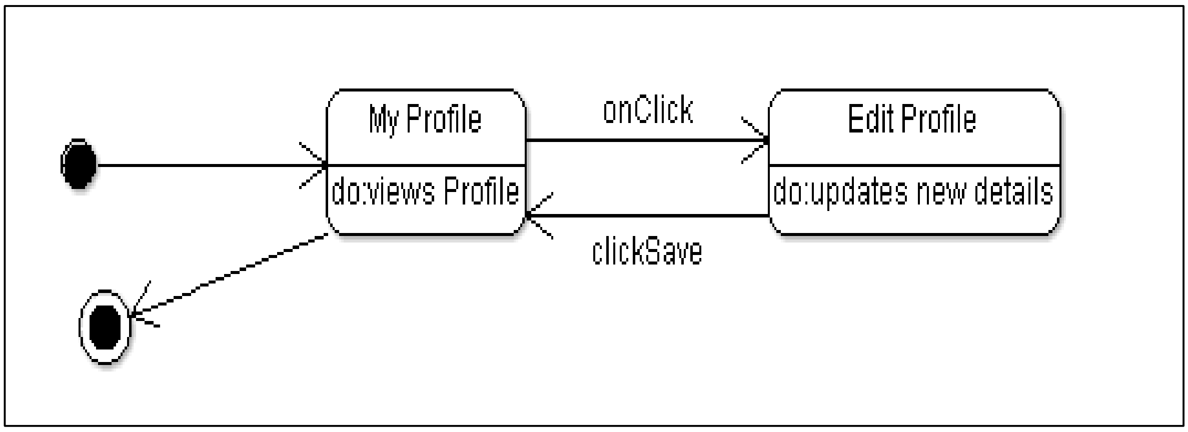
3.0.15 State Diagram: Add Data



3.0.16 State Diagram: Login



3.0.17 State Diagram: Profile



3.1 Logical Architecture Description

3.1.1 Class Diagram architecture

In the class diagram, the Block class is deriving Blockchain class which is further deriving View class and CheckIntegrity class. Program starts with the Login class which is deriving Password and it's authenticate class.
Analysis class is derived by the Graphs class and is deriving a Report class.
The project is stands on three bases: login, blockchain and analysis classes.

3.1.2 Sequence Diagram:

Arrow line signifies there is a send message taken place. Response is being shown by dotted arrows.

3.1.2.1 RegisterPage

When the user will click on the register button, RegisterPage will communicate with the server and accordingly send the response back to the Employee class. This is used to enter the employee details via using employeeDetails() function.

3.1.2.2 Server

Server will get a request and it will send a response via a serverStatus() function back to the RegisterPage. Also, it will also give validation on creating an account by sending a accountStatus function as a response.

3.1.2.3 Notification

It will act as an intermediate between CheckIntegrity and the BlockChain class as this will notify the admin if any data has been meddled or not as it gets a sendNoChangeMesg() from the CheckIntegrity class. Similarly, if data has been changed by any external authority, then it will send a message of Alert!! Data modified back to the Employee class so that all the employees get to know that data has been misused.

3.1.2.4 CheckIntegrity

Employee will be directly communicating with the CheckIntegrity class.
This will send a checkBlockChain() function to the Blockchain which will act as our security function of the blockchain as it will be sending back changeAlert() message if anyone is misusing data .

3.1.2.5 ViewPage

In the Employee class the user will call viewData() method and ViewPage's work will come into play. ViewPage will fetch data from the Block class to get data of that particular employee .The Block class will send the data as a data() method back to the ViewPage ,which will further display the data to the Employee.

3.1.2.6 AnalysisPage

The user will click on the "Do data analysis button" and then dataAnalysis() function will call the AnalysisPage which will further fetch the data report from the Graphs class.
The data will be analyzed and a Report will be sent back to the Employee class.

Similarly predictive analysis will be done whenever user click on "do predictive analysis" button .AnalysisPage will fetch Predictive data report from the Graphs and send it back to Employee class .

3.1.2.7 AddDataPage

There will be a data submit request by the Employee class as per the need whenever it will addData(). AddDataPage will send proceed request back to the Employee which will be followed by a confirmation.
When all the details of the employee will be added then AddDataPage will send a createBlock() function to Block class's constructor which will respond as its status or it can simply send a Failure response directly to Employee class .

3.1.2.8 Report

The user will click on the Send Report button. Report class will send a function reportStatus() to the NotificationPage and report will be displayed to the Employee .

3.1.2.9 LoginPage

The user will request a login , LoginPage will send a request to the server and if the server is ready , LoginPage will ask the user to enter the Login parameters .After entering the details , LoginPage will check for authenticity in the server and will log the user into the software.
If authentication fails login will fail.

3.1.2.10 HomePage

The user can ask for multiple requests from the HomePage like viewing profile, adding data, checking integrity, performance analysis . In every such request HomePage will further send a request to the server and asks for its response. For every request there is a response in the form of a unique function sent back to HomePage by the Server.

3.1.2.11 ProfilePage

ProfilePage will work on employee's personal data, it will send a request to server in the form of a function fetchProfile() which will be displayed to the user .
Personal data of the Employee can be changed so no CheckIntegrity class is required here.
User can simply send a request to ProfilePage to change his personal (unofficial) data. The Server class will confirm the changes and the profile will be updated.

3.1.3 State Diagram

Initial state is being shown by starting with a black dot. Final State is being shown by the black dot surrounded by an empty circle.

3.1.3.1 Register

The user when clicks on the create account button It will land on Register class then when user press submit button it will redirect him to Login page and then to Home page.

3.1.3.2 CheckIntegrity

From checkIntegrity class the user will redirected to a notification page all which is shown in frontend but in the back-end purity of data will be checked(misuse or not) . Then it will go back to checkIntegrity.

3.1.3.3 View

From the View, the user can be redirected to either View Recent Data or Monthly data or All data depending upon which button he clicks.

3.1.3.4 Analysis

The user may click on Data Analysis or Predictive Analysis button depending upon his choice .In data analysis he will be directed to a Data Graph which will send back a report to Data Analysis method and the report / graph will be displayed there.
Similarly, in predictive analysis he will be directed to a Predictive Graph which will send back a report to Predictive Analysis method and the report / graph will be displayed there.

3.1.3.5 Add Data

The user will traverse from enterData with Data class as the second state and when data is submitted it will go to the Block class to the final state.

3.1.3.6 Login

At Login class there can be two next states one can be Login class itself (if the details are wrong or server fails to authenticate) and the other when successful login. If successful it will go to home page. From Home Page depending upon the click , the user will be directed to Profile or Notifications or Analysis or CheckIntegrity or AddData or ViewData state .This depend upon which button the user clicks upon.

3.1.3.7 Profile

In the My Profile after click we can go to Edit Profile to update new details and then save them too. Then, it will go back to My Profile then to final state.

3.2 Class name: Block

3.2.1 Method 1: genesis

Method Description: This method creates the first block of the BlockChain. The first block is called genesis block and is of high importance.

3.2.2 Method 2: hash

Method Description: This method creates a hash which is a one-way function. Every hash is unique to maintain the exclusivity.

3.2.3 Method 3: blockHash

Method Description: This method gives every block of data a unique hash. It also provides that particular block the hash of the previous block.

3.3 Class Name: BlockChain

3.3.1 Method 1: addBlock

Method description: It is used to add block to the blockchain. It basically converts user data into a block.

3.3.2 Method 2: isValid

Method Description: Checks whether the block is valid or not.

3.3.3 Method 3: newChain

Method Description: It will create a new chain when requested.

3.4 Class Name: Date

3.4.1 Method 1: dateToday

Method Description: It provides with the current date.

3.5 Class Name: View

3.5.1 Method 1: viewAttendance**3.5 Class Name:**

Method Description: It will view the attendance entered into the block.

3.5.2 Method 2: viewTasks

Method Description: It will view the tasks performed by the Employee.

3.5.3 Method 3: viewDetail

Method Description: It will view the detailed information of an employee

3.6 Class Name: Check_Integrity

3.6.1 Method 1: modifynotice

Method Description: checks whether the integrity of the BlockChain is maintained or not.

3.7 Class Name: Employee

3.7.1 Method 1: addEmp

Method Description: It adds a new employee into the system i.e. it registers a new employee into the system.

3.7.2 Method 2: deleteEmp

Method Description: It deletes the account of an employee from the system.

3.7.3 Method 3: searchEmp

Method Description: It can search an employee and his details from the system.

3.8 Class Name: Department

3.8.1 Method 1: departmentDetails

Method Description: Contains the details of the department the employees belong to as well as their job description.

3.9 Class Name: Login

3.9.1 Method 1: checkLogin

Method Description: Checks whether the user has previously registered or not. If user has an account it allows him to perform actions.

3.10 Class Name: Password

3.10.1 Method 1: addPassword

Method Description: It can be used to provide further security to the system

3.10.2 Method 2: deletePassword

Method Description: It is used to delete the password that ensures the second level of security to the system.

3.10.3 Method 3: changePassword

Method Description: It is used to change the password.

3..11 Class Name: Graphs

3.11.1 Method 1: analyseGraph

Method Description: Produces the graph analysing the performance of the employees.

3.11.2 Method 2: predictGraph

Method Description: Produces the Predictive Graph by analysing past behaviour.

3.12 Class Name: Analysis

3.12.1 Method 1: dataAnalysis

Method Description: Analyses the performance of the employee by studying his pattern.

3.12.2 Method 2: predictiveAnalysis

Method Description: Analyses and predicts the future performance of the employee by studying his past behaviour.

3.13 Class Name: Report

3.13.1 Method 1: dataReport

Method Description: Produces full report of the performance analysis of the employee.

3.13.2 Method 2: predictiveReport

Method Description: Produces full report of the predictive analysis of the employee’s performance.

3.14 Class Name: Feedback

3.14.1 Method 1: sendAlert

Method Description: Sends alert to the employee according to his performance

4.0 Execution Architecture

Runtime environment required is a device having a Linux machine (64 bit) running on a localhost.

4.1 Reuse and relationships to other products

NIL

5.0 Design decisions and tradeoffs

The software will be as user friendly as possible with a simple user interface.
The possible tradeoff is the use of buttons to traverse to other pages. Hence, it becomes easier for the user to do the particular task he is willing to do.

6.0 Pseudocode for components

TBD

7.0 Appendices (if any)

NIL