Table of Contents

CHAPTER 1

Software Requirement Specification

Software Requirements Specification

for

Exam Preparation Portal

Version 1.0 approved

Prepared by

Arnav Ajay (014)

Ashwin Jawahar (017)

Gayathri Chandrasekar (039)

Neeraj Pandey (076)

Hrithik Raj (047)

NIIT University

05/09/18

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to build an online generic system that provides mock

examinations for students. As it is evaluated and generated by the server, it saves a lot of time of the students. The administrator has privileges to create, modify and delete the test papers and the questions, whereas users can register, login and give the test with his or her specific id, and can see the results as well.

1.2 Document Conventions

This document was prepared using the IEEE recommended practice for Software Requirements

Specification.

This document uses the following conventions.

DB	Database
SRS	Software Requirement Specification

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the developers, project managers, marketing staff, users, testers, and documentation writers. It is open to students, teachers as well as exam moderators to use and provide feedback on the project. The developers are advised to focus on modules 2.3, 2.4 and 2.5 to understand the implementation and integration of various classes in the project. The students and staff are suggested to read the module 2.6 to understand how to use the system. Whereas, the project managers are suggested to begin reading from module 3 to understand the system requirements. The rest of this SRS contains Overall Description, External Interface Requirements, System Features, Non-Functional Requirements and Other Requirements. The general suggested

sequence for reading the document - begin with the overview sections and proceed through the sections that are most pertinent to each reader type.

1.4 Product Scope

The goal of the online examination portal is to provide students with a free user-friendly mock examination that helps them prepare for their required examinations. The scope of this project is very broad as compared to manually taken examinations:

- This portal is not only limited to educational institutes but also to the corporate world
- There is no restriction as to the presence of an examiner during the period of the exam
- Less time consumption, thereby increasing the efficiency

1.5 References

This web application has been prepared on the basis of discussion with Team members and also

taken information from following- https://angular.io/ and the IEEE Website.

2. Overall Description

2.1 Product Perspective

This project is a more efficient and user-friendly replacement for the existing unpaid examination practice portals on the internet. Since, the websites that are free of cost usually lack in functionality, this project aims to provide better functionality at no cost for the numerous students who prepare for various examinations every year. It is accessible via the Internet, 24 hours a day, 7 days a week. This app will only allow the registered users to enter the test module. The various stages in the app are as follows:

- Login: This window offers the user two choices of logins- Candidate Login and Admin Login
- System Overview: It can only be accessed by an administrator. It allows the admin to add, modify and delete the test modules.
- Test: This window contains the exam that the candidate is supposed to attempt.
- Result: The result window displays the result of the exam that the candidate has just
- attempted.
- Rank: It displays the rank of the candidate among all the other users who have attempted the same mock examination.
- Review: This window provides feedback on the candidates' weak spots in the attempted examination.

2.2 Product Functions

The major functions that the product is required to perform are:

- Candidate

- Preparing for Examinations
- Tracking the performance of each student
- Realizing the weak subjects/topics of the student
- Competing with other candidates preparing for the same

- Administrator
  - Add test module/questions
  - Delete test module/questions
  - Modify test module/questions

2.3 User Classes and Characteristics

The main users of the system are the candidates and the admins/moderators. The candidates should be able to:

- Login to the website
- Take the examination
- View their score
- View their rank
- View their weak areas

The admins should be able to:

- Login to the website
- Set questions for the examination
- Help students with their problems

2.4 Operating Environment

Operating environment for the airline management system is as listed below.

- Any internet browser such as, Google Chrome, Mozilla Firefox, Internet Explorer, etc.
- Operating System does not matter

2.5 Design and Implementation Constraints

The candidate is allowed to give the exams any number of times, until specified otherwise by the administrator while building the test. While giving the exam the candidate is given only a set amount of time and the remaining time should be displayed, after which the exam should close and display the result. The only condition under which the project will face issues if there is no internet connectivity in the system or the internet connectivity is very slow resulting in inefficient use of the online system.

2.6 User Documentation

The basic functionality of the online examination portal will be stated in the website's 'Help' section. The user manual will include product overview, complete configuration of the required software and hardware, technical details and contact information which will include email address.

2.7 Assumptions and Dependencies

Proper working of this app is dependent on the internet connectivity of the users' computer.

Assumptions and dependencies:

- It is assumed that the user has basic knowledge of the system (i.e. he/she is not a first time
  user) as any action by the user is considered valid during an examination.
- It is assumed that the data entered by the user while registering is true.
- It is assumed that the candidate does not cheat during the exam as there are no supervisors around to monitor.

3. External Interface Requirements

3.1 User Interfaces

- Registration Screen:
  Various fields available on this screen will be:
  - Login Name
  - Email Id
  - Password

- Login Screen:
  Various fields available on this screen will be:
  - Login Name
  - Password

- Entering Questions:
  Various fields available on this screen will be:
  - Questions
  - Options-4
  - Correct Answer

- Examination Details Screen:
  Various fields available on this screen will be:
  - Examination Name
  - Number of Questions
  - Passing Marks
  - Time Limit

- Result Screen:
  Various fields available on this screen will be:
  - Number of Correct Questions
  - Number of Incorrect Questions
  - Total Marks
  - Result
- Review Screen:
  Various fields available on this screen will be:
  - Ranking in the Examination
  - Weak Topics

3.2 Hardware Interfaces

- Screen resolution of at least 800X600 is required for proper and complete viewing of screens. Higher resolution will be accepted.
- Any internet browser.
- Any operating system.

3.3 Software Interfaces

- Any Operating System
- MySql Server Database
- JavaScript
- HTML
- CSS
- Angular

3.4 Communications Interfaces

- HTTP
- TCP/IP: Database Connectivity

4. System Features

4.1 Candidate Module

4.1.1 Description and Priority

The candidate module includes all the actions that a candidate can perform which includes:

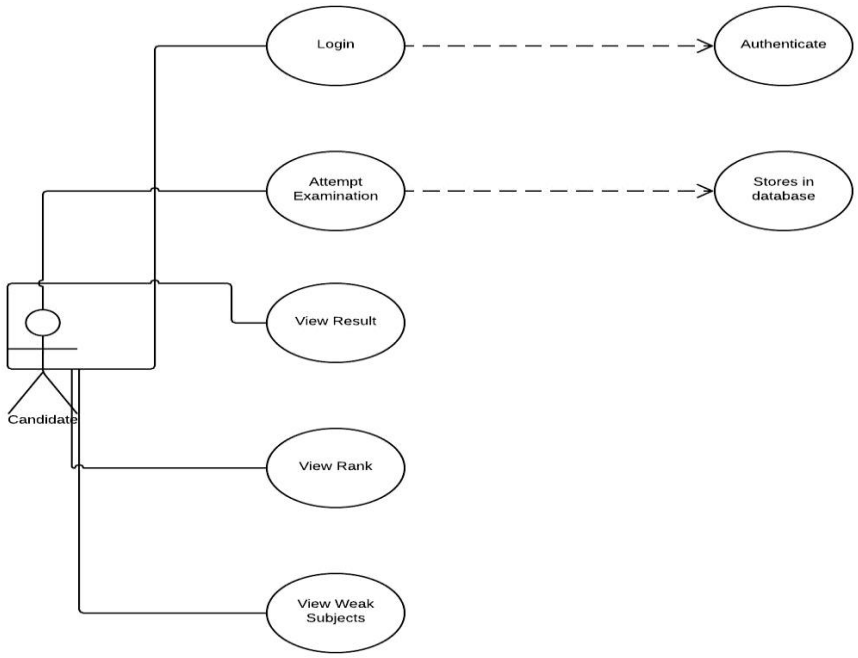
- Login to the website
- Take the examination
- View their score
- View their rank
- View their weak areas

4.1.2 Stimulus/Response Sequences

The response to the given functionalities will be:

- Login: The website will authenticate the username and password and redirect to user's dashboard in case of correct authentication, else displays an error message.
- Take the examination: The user chooses the rights answers among the multiple choices and it is stored in a database.
- View score, rank and review: Displays the score, rank and weak subjects after finishing the examination.

4.1.3 Functional Requirements



4.2 Administrator Module

4.2.1 Description and Priority

The administrator module includes all the actions that an admin can perform:

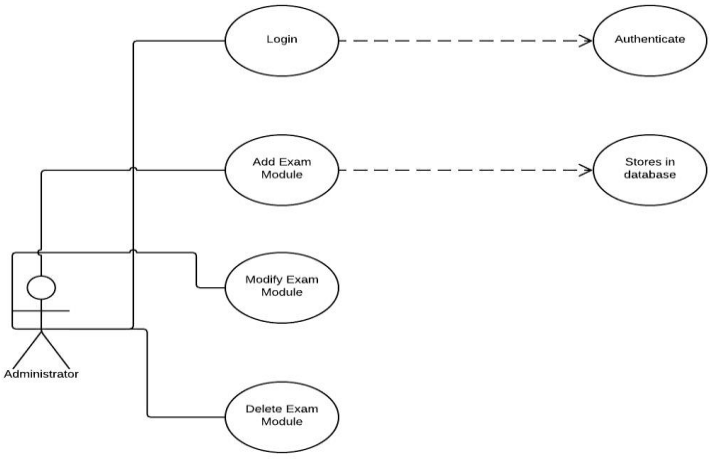
- Login to the website
- Add exam module
- Delete exam module
- Modify exam module

4.2.2 Stimulus/Response Sequences

The response to the given functionalities will be:

- Login: The website will authenticate the username and password and redirect to user's dashboard in case of correct authentication, else displays an error message.
- Modify Examination: The admin can modify, add or delete the questions or test modules.

4.2.3 Functional Requirements



5. Oth

5.1 Perf

Some Performance requirements identified is listed below:

- The database shall be able to accommodate a minimum of 1,000 records of Users.
- The software shall support use of multiple users at a time.
- There are no other specific performance requirements that will affect development.

5.2 Safety Requirements

The database might crash at any given time due to virus or operating system failure. It is advised to take a backup of the database.

5.3 Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use or modification are given below:

- Keep specific log or history data sets
- Assign certain functions to different modules

5.4 Software Quality Attributes

- AVAILABILITY: The mock exams should be available for all candidates, students as well as corporate employees.
- CORRECTNESS: The result and reviews displayed should be accurate.
- MAINTAINABILITY: The administrators should maintain the test modules and question banks.
- USABILITY: The exam portal should fulfil all the users' needs and requirements.

5.5 Business Rules

None

6. Other Requirements

None.

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List

Software Requirements Specification

for

Exam Preparation Portal

Version 2.0 approved

Prepared by

Arnav Ajay (014)

Ashwin Jawahar (017)

Gayathri Chandrasekar (039)

Neeraj Pandey (076)

Hrithik Raj (047)

NIIT University

10/09/18

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
SRS	05/09/18	N/A	1.0
SRS	10/09/18	Updated Functionalities	2.0

1. Introduction

1.1 Purpose

The purpose of this document is to build an online generic system that provides mock

examinations for students. As it is evaluated and generated by the server, it saves a lot of time of the students. The administrator has privileges to create, modify and delete the test papers and the questions, whereas users can register, login and give the test with his or her specific id, and can see the results as well.

1.2 Document Conventions

This document was prepared using the IEEE recommended practice for Software Requirements

Specification.

This document uses the following conventions.

DB	Database
SRS	Software Requirement Specification

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the developers, project managers, marketing staff, users, testers, and documentation writers. It is open to students, teachers as well as exam moderators to use and provide feedback on the project. The developers are advised to focus on modules 2.3, 2.4 and 2.5 to understand the implementation and integration of various classes in the project. The students and staff are suggested to read the module 2.6 to understand how to use the system. Whereas, the project managers are suggested to begin reading from module 3 to understand the system requirements. The rest of this SRS contains Overall Description, External Interface Requirements, System Features, Non-Functional Requirements and Other Requirements. The general suggested

sequence for reading the document - begin with the overview sections and proceed through the sections that are most pertinent to each reader type.

1.4 Product Scope

The goal of the online examination portal is to provide students with a free user-friendly mock examination that helps them prepare for their required examinations. The scope of this project is very broad as compared to manually taken examinations:

- This portal is not only limited to educational institutes but also to the corporate world
- There is no restriction as to the presence of an examiner during the period of the exam
- Less time consumption, thereby increasing the efficiency

1.5 References

This web application has been prepared on the basis of discussion with Team members and also taken information from following- https://angular.io/ and the IEEE Website.

2. Overall Description

2.1 Product Perspective

This project is a more efficient and user-friendly replacement for the existing unpaid examination practice portals on the internet. Since, the websites that are free of cost usually lack in functionality, this project aims to provide better functionality at no cost for the numerous students who prepare for various examinations every year. It is accessible via the Internet, 24 hours a day, 7 days a week. This app will only allow the registered users to enter the test module. The various stages in the app are as follows:

- Login: This window offers the user two choices of logins- Candidate Login and Admin Login
- System Overview: It can only be accessed by an administrator. It allows the admin to add, modify and delete the test modules.
- Test: This window contains the exam that the candidate is supposed to attempt.
- Result: The result window displays the result of the exam that the candidate has just
- attempted.
- Rank: It displays the rank of the candidate among all the other users who have attempted the same mock examination.
- Review: This window provides feedback on the candidates' weak spots in the attempted examination.

2.2 Product Functions

The major functions that the product is required to perform are:

- Candidate
  - Preparing for Examinations
  - Tracking the performance of each student
  - Realizing the weak subjects/topics of the student
  - Competing with other candidates preparing for the same

- Administrator
  - Add test module/questions
  - Delete test module/questions
  - Modify test module/questions

2.3 User Classes and Characteristics

The main users of the system are the candidates and the admins/moderators.
The candidates should be able to:

- Login to the website
- Take the examination
- View their score
- View their rank
- View their weak areas

The admins should be able to:

- Login to the website
- Set questions for the examination
- Help students with their problems

2.4 Operating Environment

Operating environment for the airline management system is as listed below.

- Any internet browser such as, Google Chrome, Mozilla Firefox, Internet Explorer, etc.
- Operating System does not matter

2.5 Design and Implementation Constraints

The candidate is allowed to give the exams any number of times, until specified otherwise by the administrator while building the test. While giving the exam the candidate is given only a set amount of time and the remaining time should be displayed, after which the exam should close and display the result. The only condition under which the project will face issues if there is no internet connectivity in the system or the internet connectivity is very slow resulting in inefficient use of the online system.

2.6 User Documentation

The basic functionality of the online examination portal will be stated in the website's 'Help' section. The user manual will include product overview, complete configuration of the required software and hardware, technical details and contact information which will include email address.

2.7 Assumptions and Dependencies

Proper working of this app is dependent on the internet connectivity of the users' computer.

Assumptions and dependencies:

- It is assumed that the user has basic knowledge of the system (i.e. he/she is not a first time
  user) as any action by the user is considered valid during an examination.
- It is assumed that the data entered by the user while registering is true.
- It is assumed that the candidate does not cheat during the exam as there are no supervisors around to monitor.

3. External Interface Requirements

3.1 User Interfaces

- Registration Screen:
  Various fields available on this screen will be:
  - Login Name
  - Email Id
  - Password

- Login Screen:
  Various fields available on this screen will be:
  - Login Name
  - Password

- Entering Questions:
  Various fields available on this screen will be:
  - Questions
  - Options-4
  - Correct Answer

- Examination Details Screen:
  Various fields available on this screen will be:
  - Examination Name
  - Number of Questions
  - Passing Marks
  - Time Limit

- Result Screen:
  Various fields available on this screen will be:
  - Number of Correct Questions
  - Number of Incorrect Questions
  - Total Marks
  - Result

- Review Screen:
  Various fields available on this screen will be:
  - Ranking in the Examination
  - Weak Topics

3.2 Hardware Interfaces

- Screen resolution of at least 800X600 is required for proper and complete viewing of screens. Higher resolution will be accepted.
- Any internet browser.
- Any operating system.

3.3 Software Interfaces

- Any Operating System
- MySql Server Database
- JavaScript
- HTML
- CSS
- Angular

3.4 Communications Interfaces

- HTTP
- TCP/IP: Database Connectivity

4. System Features

4.1 Candidate Module

4.1.1 Description and Priority

The candidate module includes all the actions that a candidate can perform which includes:

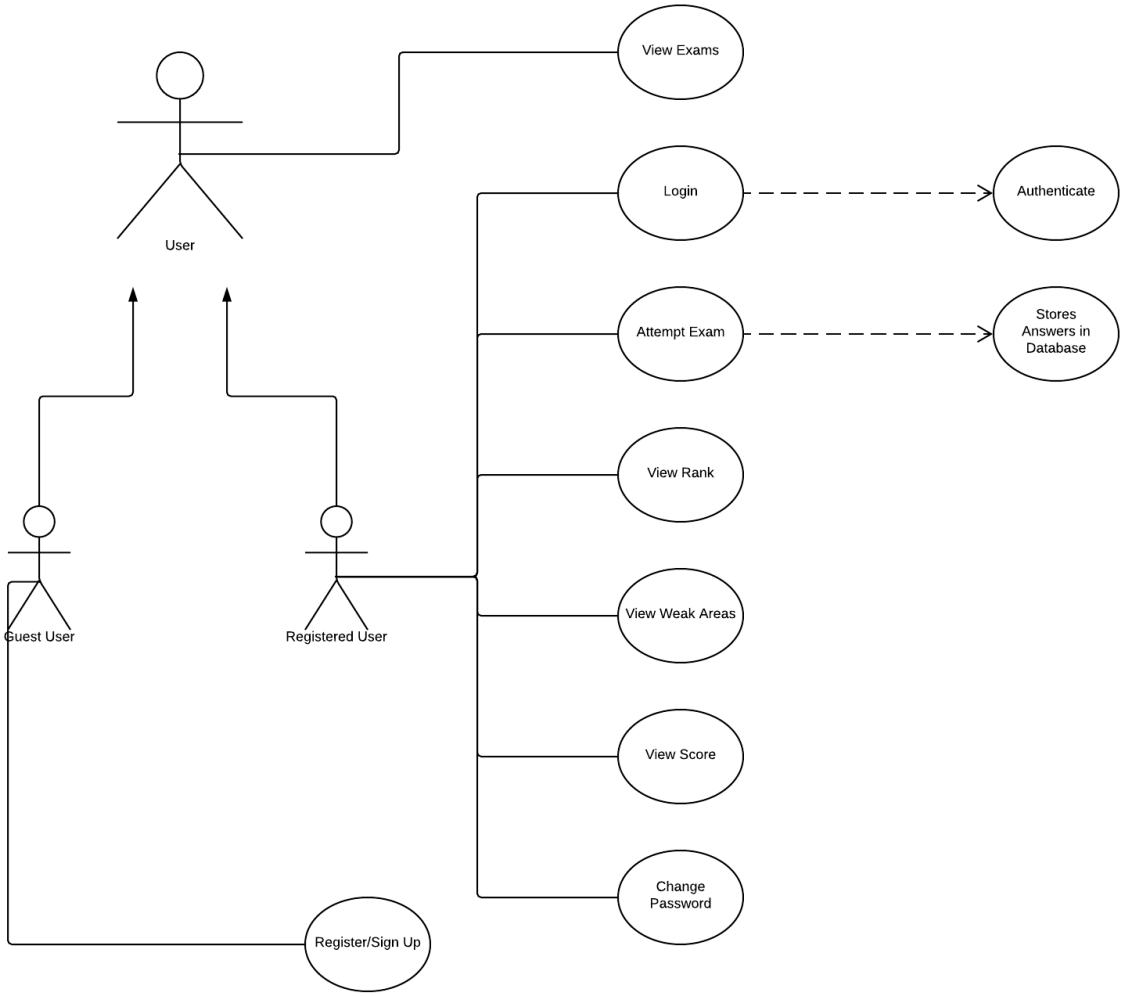
- Login to the website
- Take the examination
- View their score
- View their rank
- View their weak areas

4.1.2 Stimulus/Response Sequences

The response to the given functionalities will be:

- Login: The website will authenticate the username and password and redirect to user's dashboard in case of correct authentication, else displays an error message.
- Take the examination: The user chooses the rights answers among the multiple choices and it is stored in a database.
- View score, rank and review: Displays the score, rank and weak subjects after finishing the examination.

4.1.3 Functional Requirements



4.3 Administrator Module

4.2.1 Description and Priority

The administrator module includes all the actions that an admin can perform:

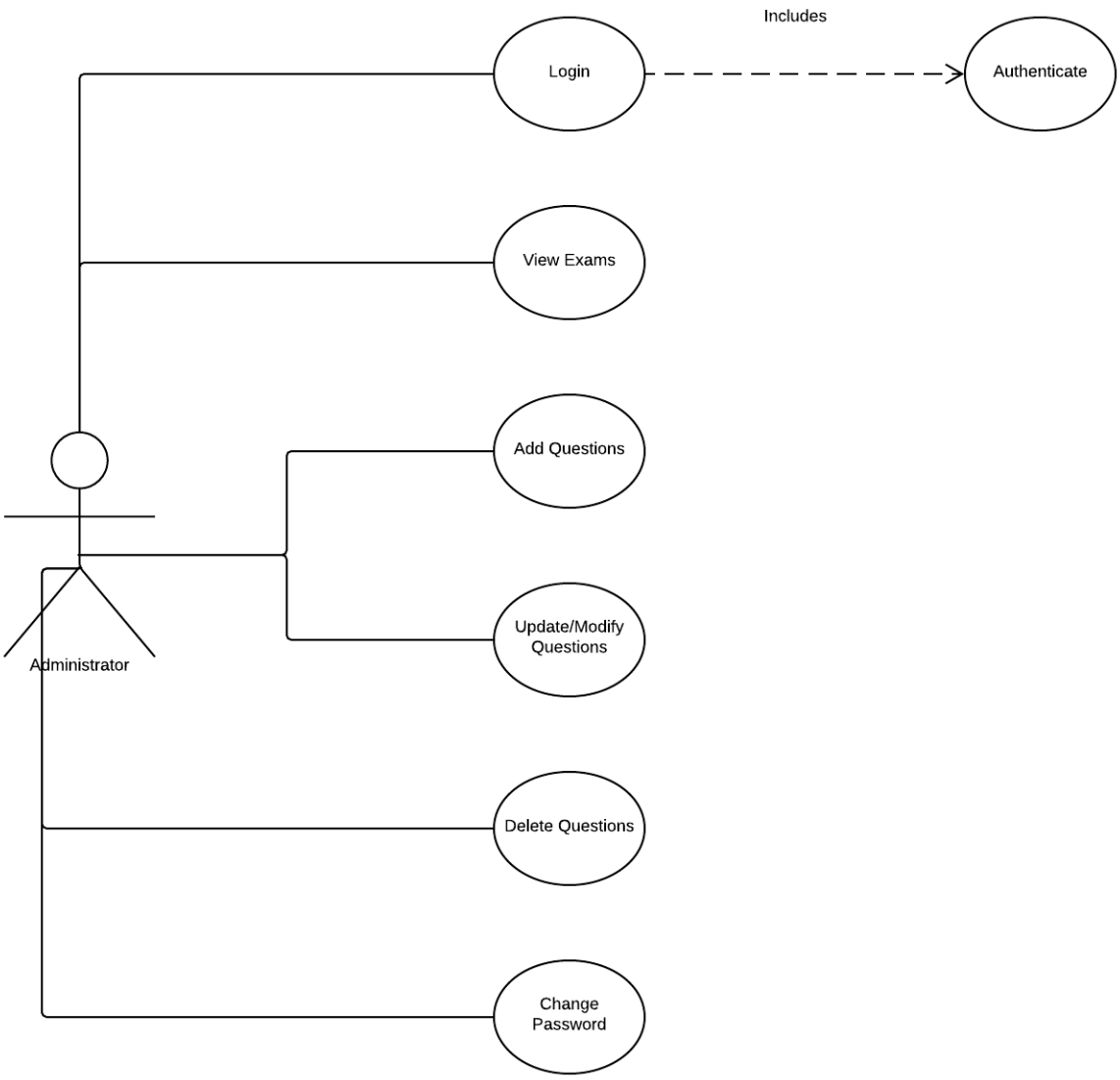
- Login to the website
- Add exam module
- Delete exam module
- Modify exam module

4.2.2 Stimulus/Response Sequences

The response to the given functionalities will be:

- Login: The website will authenticate the username and password and redirect to user's dashboard in case of correct authentication, else displays an error message.
- Modify Examination: The admin can modify, add or delete the questions or test modules.

4.2.3 Functional Requirements



5.1 Performance Requirements

Some Performance requirements identified is listed below:

- The database shall be able to accommodate a minimum of 1,000 records of Users.

- The software shall support use of multiple users at a time.
- There are no other specific performance requirements that will affect development.

5.2 Safety Requirements

The database might crash at any given time due to virus or operating system failure. It is advised to take a backup of the database.

5.3 Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use or modification are given below:

- Keep specific log or history data sets
- Assign certain functions to different modules

5.4 Software Quality Attributes

- AVAILABILITY: The mock exams should be available for all candidates, students as well as corporate employees.
- CORRECTNESS: The result and reviews displayed should be accurate.
- MAINTAINABILITY: The administrators should maintain the test modules and question banks.
- USABILITY: The exam portal should fulfil all the users' needs and requirements.

5.5 Business Rules

None

6. Other Requirements

None.

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List

CHAPTER 2

Software Design Document

Software Design Specification

Version 1.0

Gayathri - U101116FCS039

Arnav Ajay – U101116FCS014

Neeraj Pandey – U101116FCS076

Ashwin Jawahar – U101116FCS017

Hrithik Raj – U101116FCS047

1. Introduction

1.1 Purpose of this document

This Software Design Specification Document is made with the purpose of outlining the software architecture and design of the Online Examination Portal in detail. This document will provide developers an insight in meeting the client's needs efficiently and effectively. Moreover, the document facilitates communication and understanding of the system by providing sequence diagrams, collaboration models, state diagrams and other required documents of the system design. This document will demonstrate how the design will accomplish the functional and non-functional requirements captured in the Software Requirement Specification Document. The document will provide a framework to the programmers through describing the architecture, sub-systems, interfaces, database design and functions.

1.2 Scope of the development project

The scope of this project is very broad as compared to manually taken examinations:

1. This portal is not only limited to educational institutes but also to the corporate world
2. There is no restriction as to the presence of an examiner during the period of the exam
3. Less time consumption, thereby increasing the efficiency

1.2 Definitions, acronyms, and abbreviations

IEEE: Institute of Electrical and Electronics Engineers
SDS: Software Design Specification

1.4 References

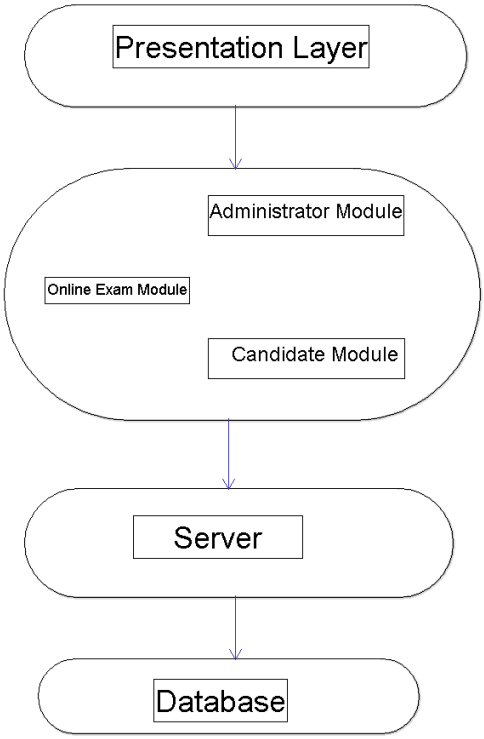
1. R. S. Pressman, Software Engineering: A Practioner's Approach, 5th Ed, McGraw-Hill, 2001.
2. IEEE SDS template

1.5 Overview of document

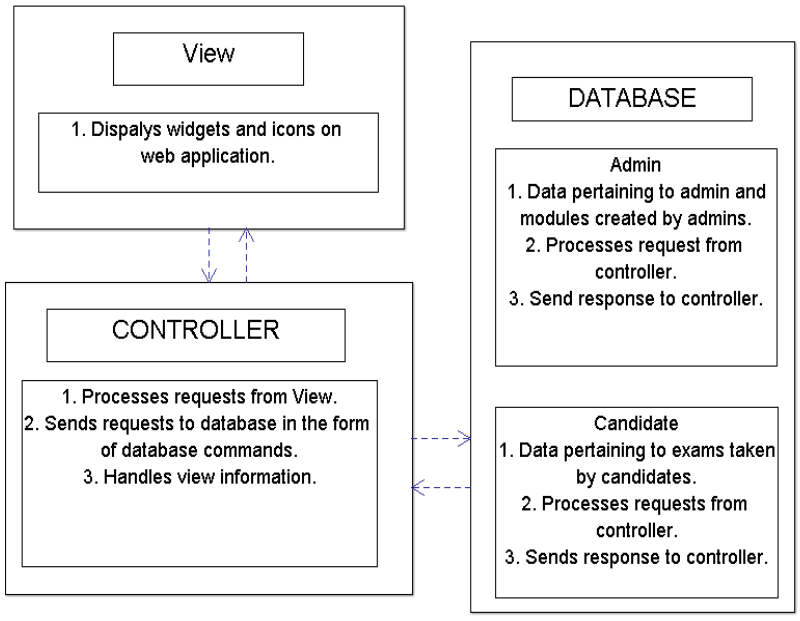
The SDS document is divided into the following sub-sections:

1. Introduction: It mainly describes the purpose of the document and scope of the development project.
2. Conceptual Architecture/Architecture Diagram: The intent of conceptual architecture is to direct attention at an appropriate decomposition of the system without delving into the details of interface specification.
3. Logical Architecture: Logical architecture is a structural design that gives as much detail as possible without constraining the architecture to a particular technology or environment.
4. Execution Architecture: The execution architecture determines largely the realtime and performance behavior of a system. Concepts such as latency, response time and throughput are illustrated.
5. Design Decisions and Trade-offs: It describes the decisions taken throughout the system design and why they were taken instead of their alternatives.
6. Pseudocode: Pseudocode/algorithms implemented in the system.
7. Appendices: Subsidiary matters, if any.

2. Conceptual Architecture/Architecture Diagram

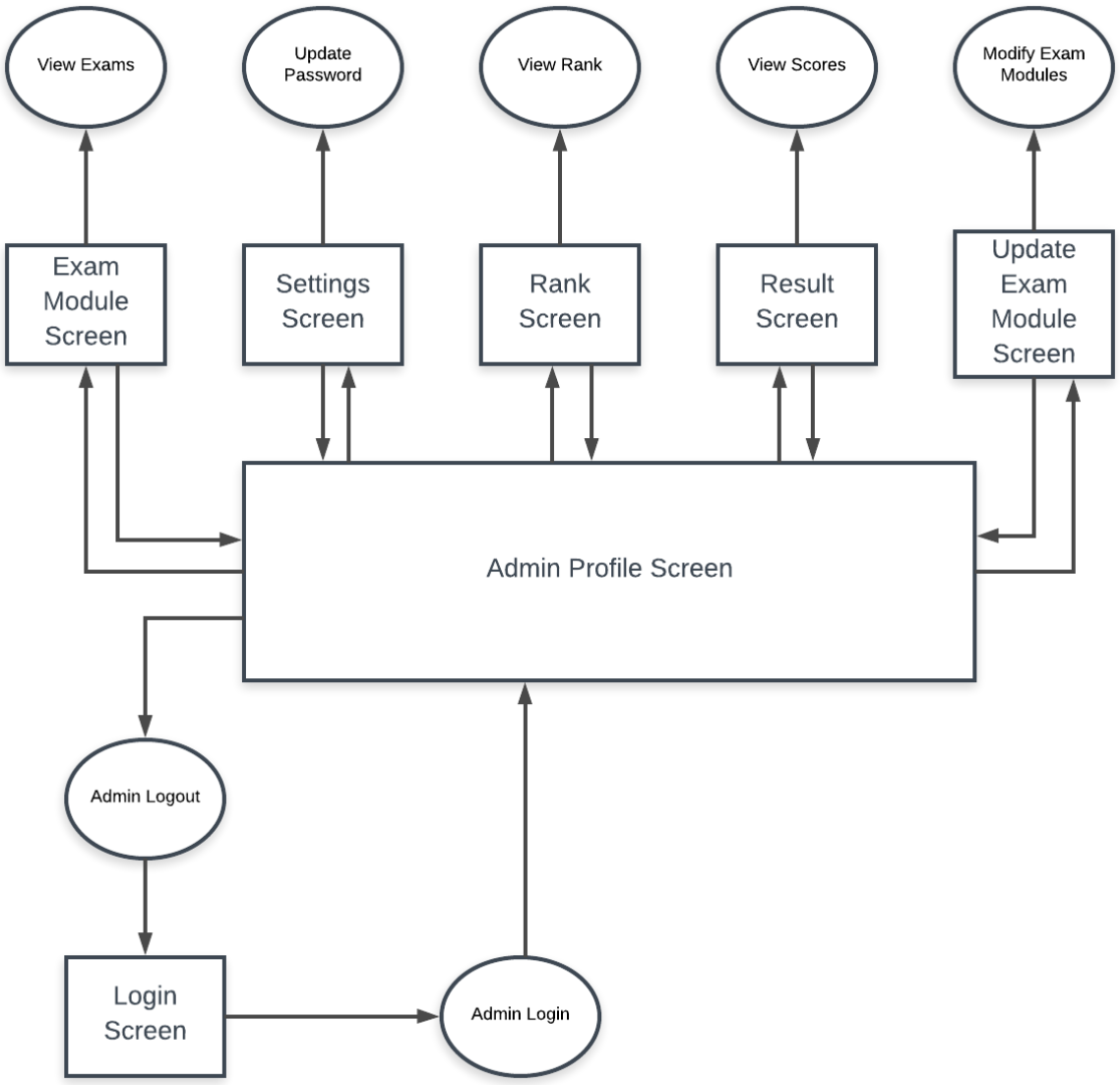


2.1 Overview of modules / components

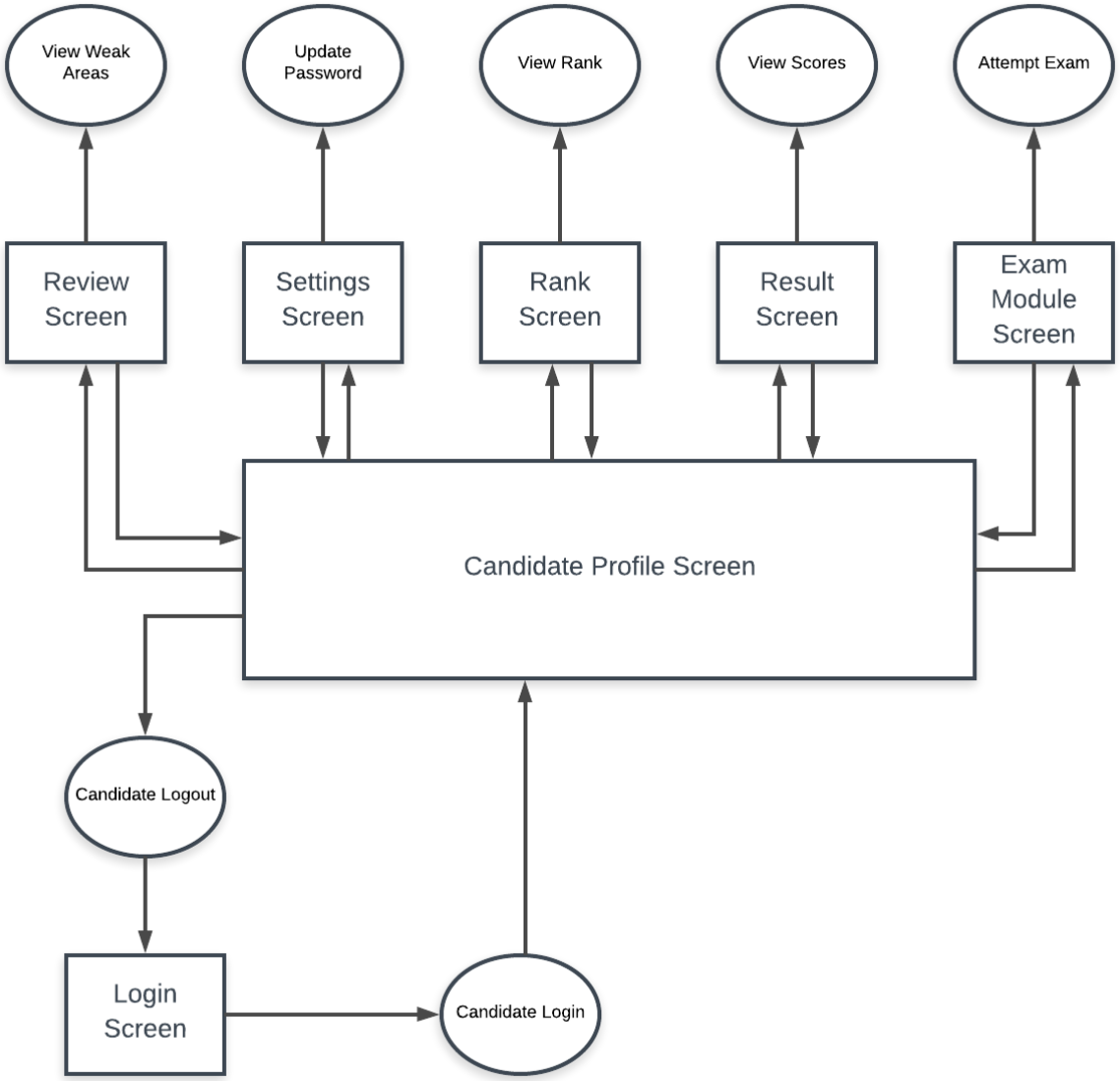


2.2 Structure and relationships

2.2.1 Administrator



2.2.2 Candidate

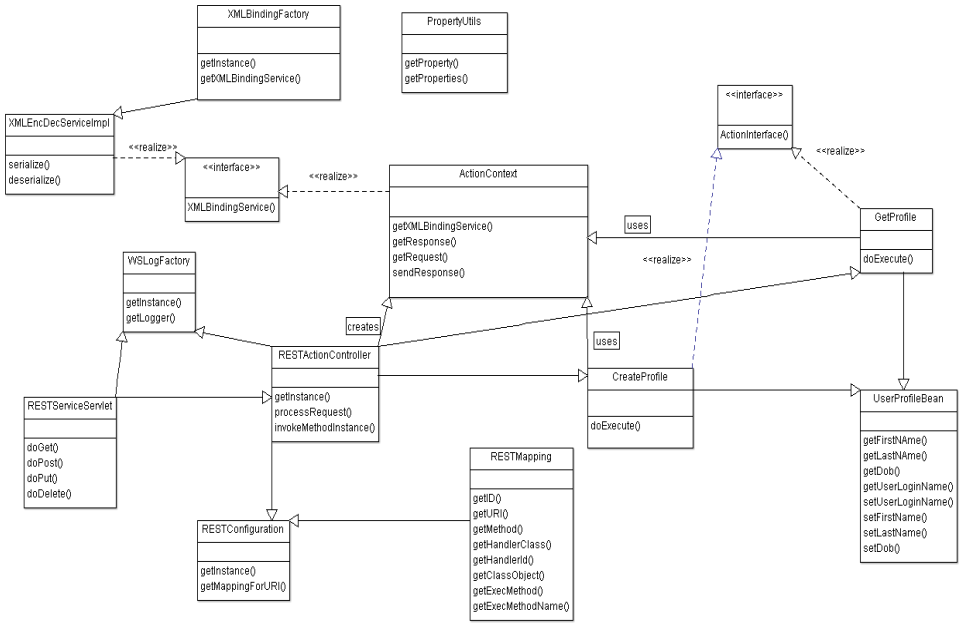


2.3 User interface issues

Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. No user would be able to access any part of the application without logging on to the system. User Interface Issues for the possible users of the system are:

1. User A is a 23-year-old student or corporate candidate who is looking to take a mock online examination using this system. As the user is young and technologically proficient in using computer applications and systems, the user interface will be of fairly common conventions on the candidates' screen.
2. User B is a 35-year-old person who is an administrator of the online system with responsibilities to update exam modules on the site by modifying the questions. Since, user B might not be completely well-versed in using online portals, directives will be given for ease of use.

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

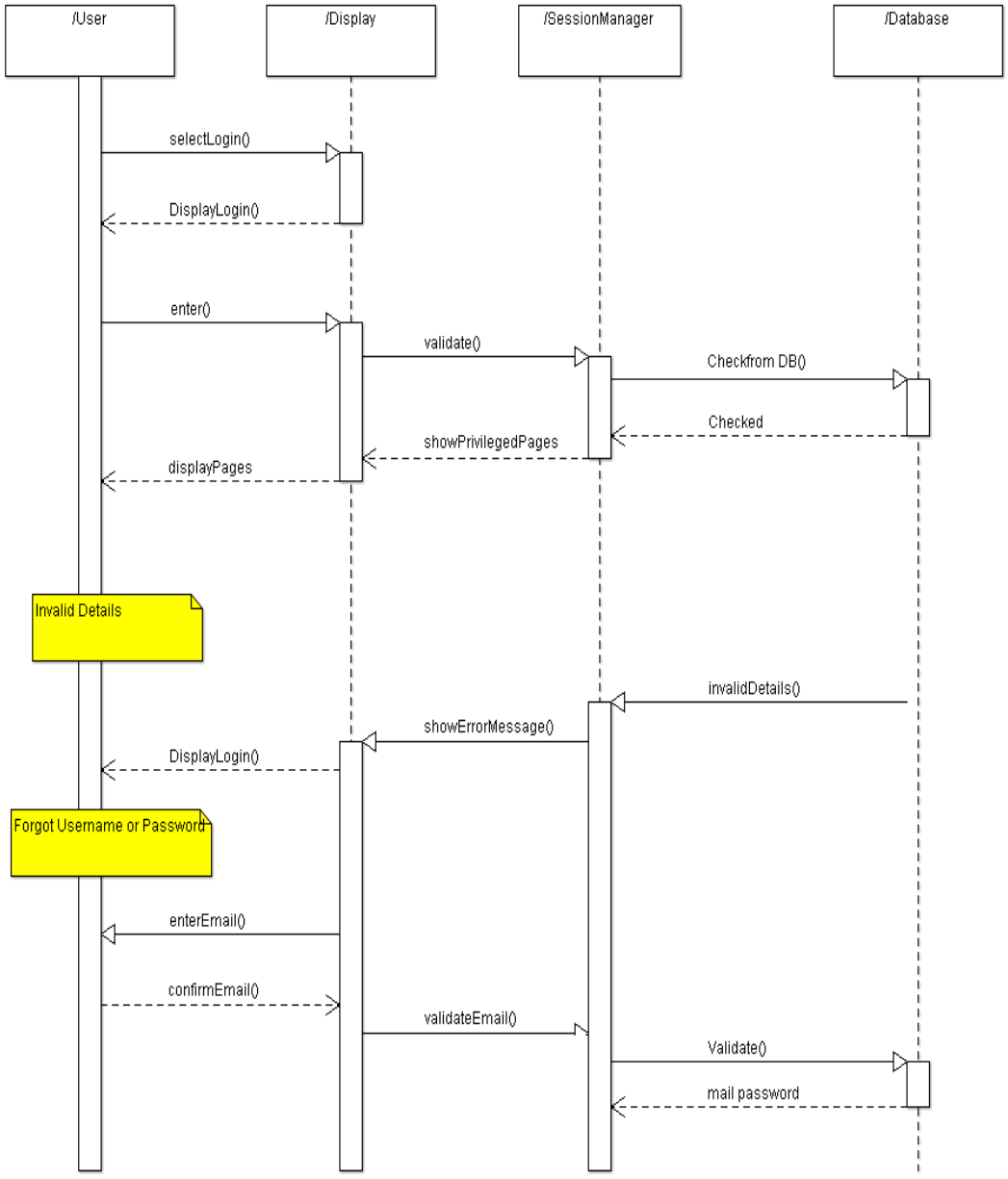


3.1 Logical Architecture Description

Representational State Transfer is a software architectural style that defines a set of constraints to be used for creating web services. Web services that conform to the REST architectural style, or RESTful web services, provide interoperability between computer systems on the Internet. One of the key advantages of REST APIs is that they provide a great deal of flexibility. Data is not tied to resources or methods, so REST can handle multiple types of calls, return different data formats and even change structurally with the correct implementation of hypermedia. This explains the class structure in Java. The rest of the functionalities use bootstrap and typescript, the architecture of which is given below.

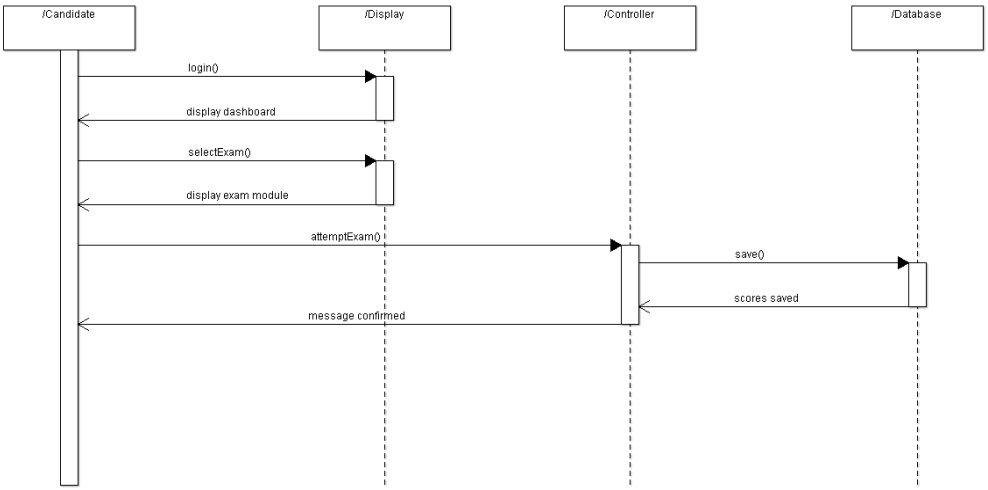
3.2 Login

3.2.1 Sequence Diagram

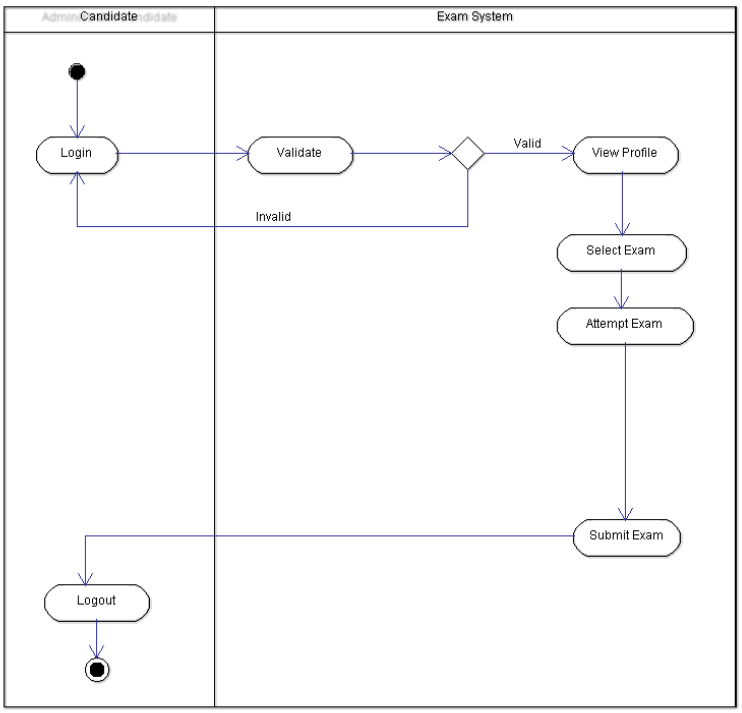


3.3 Attempt Exam

3.3.1 Sequence Diagram

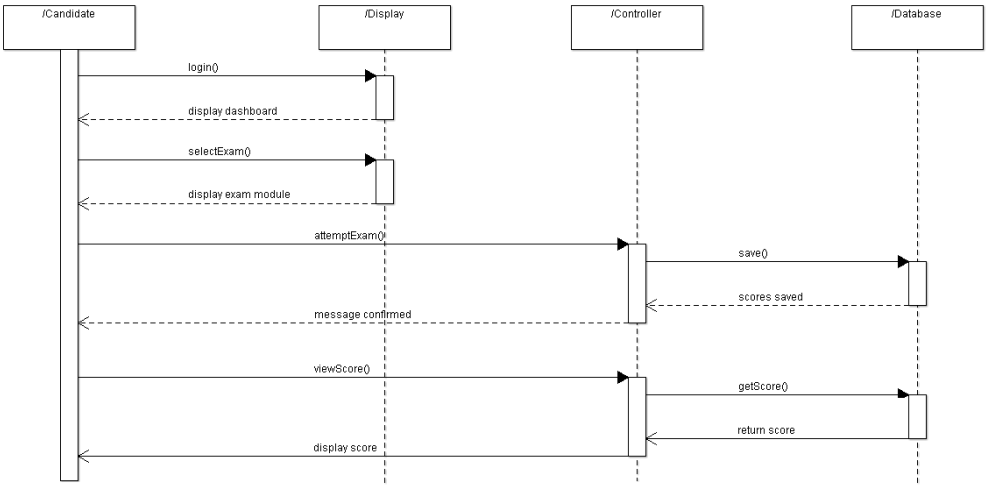


3.3.2 Activity Diagram

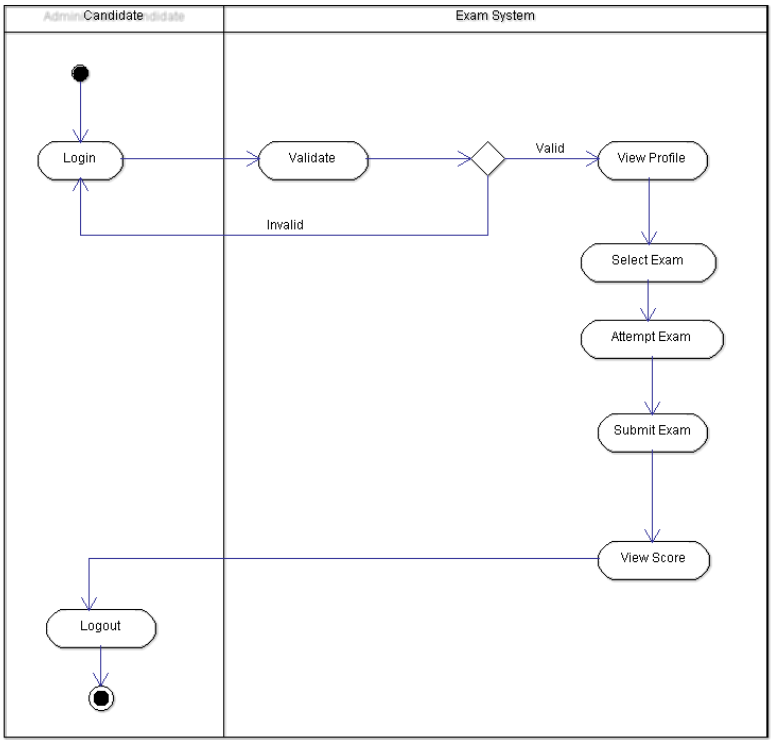


3.4 View Score

3.4.1 Sequence Diagram

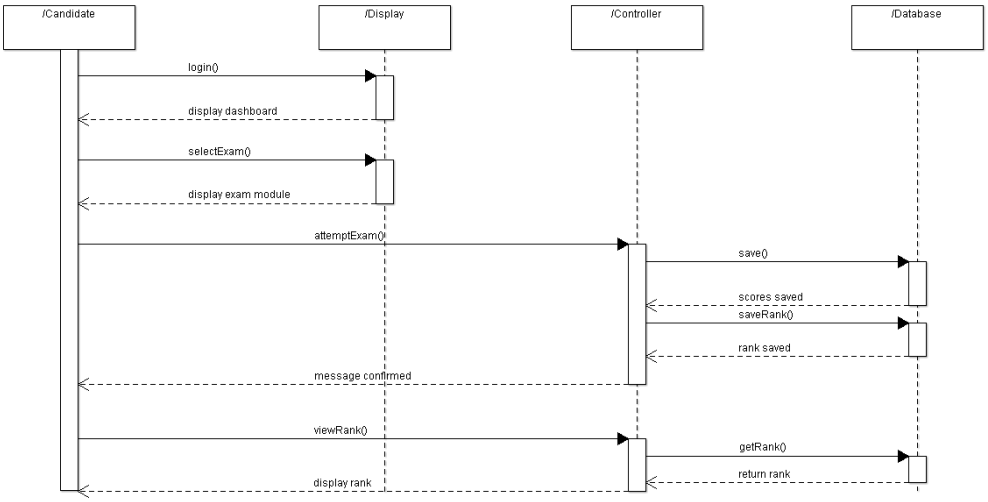


3.4.2 Activity Diagram

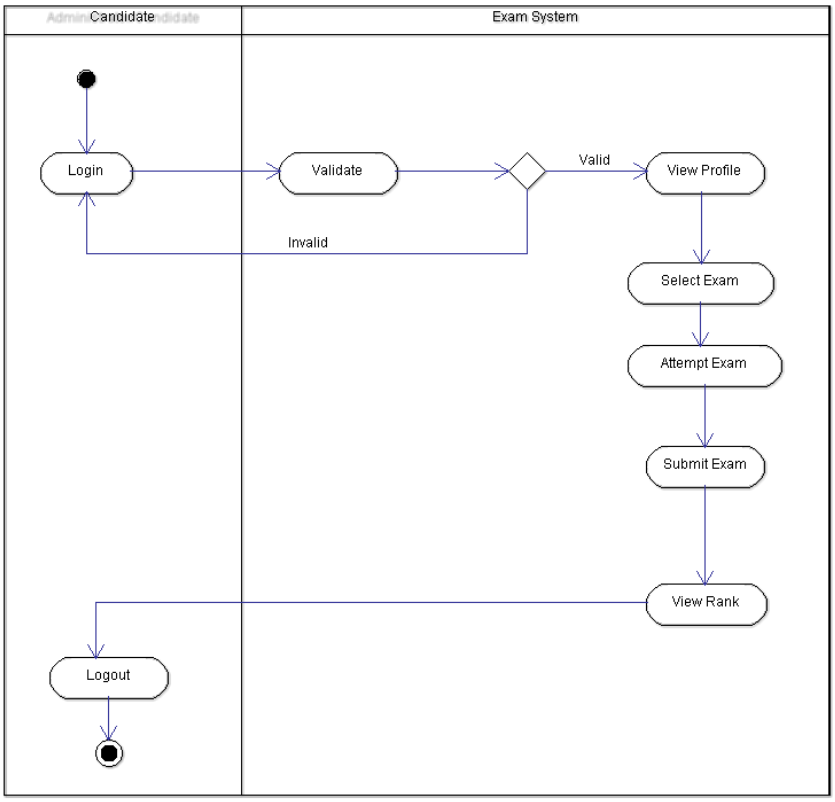


3.5 View Rank

3.5.1 Sequence Diagram

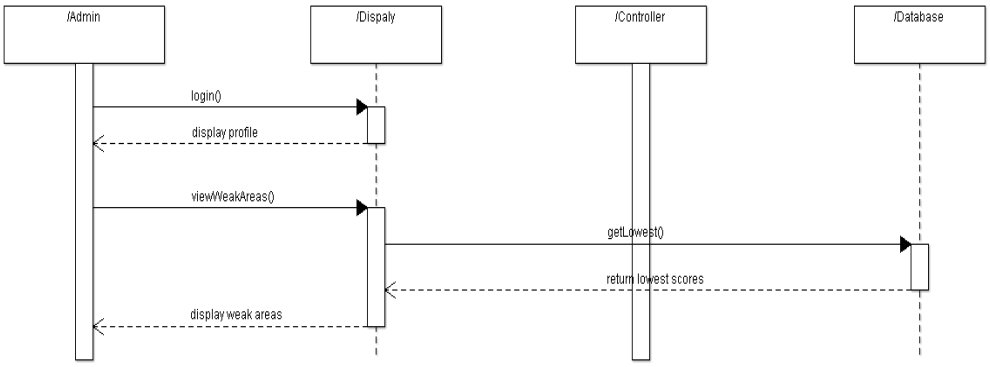


3.5.2 Activity Diagram

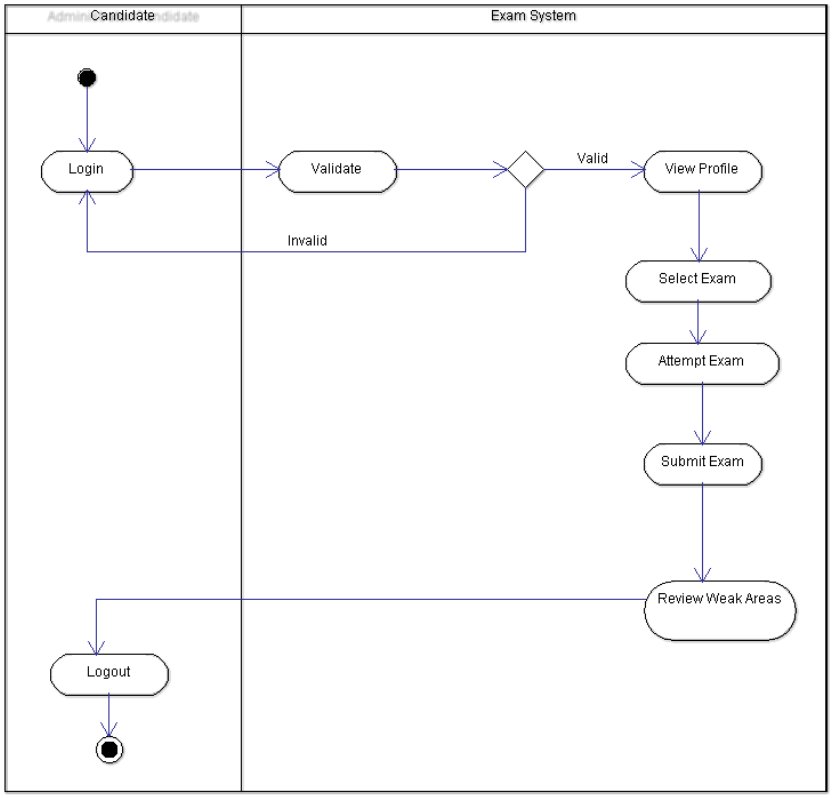


3.6 View Weak Areas

3.6.1 Sequence Diagram

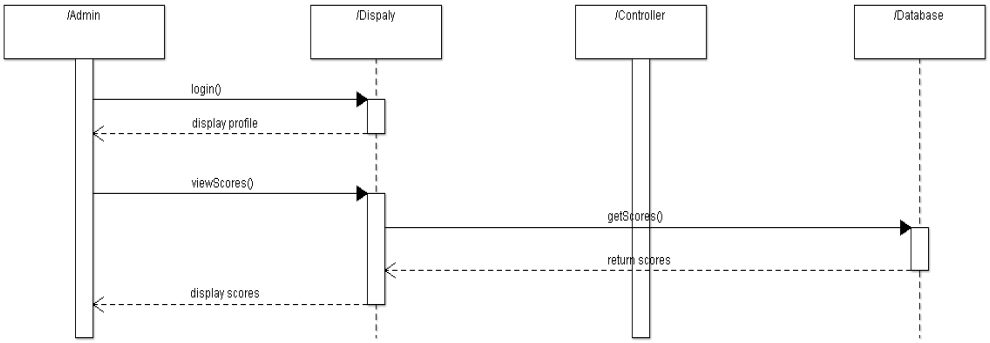


3.6.2 Activity Diagram

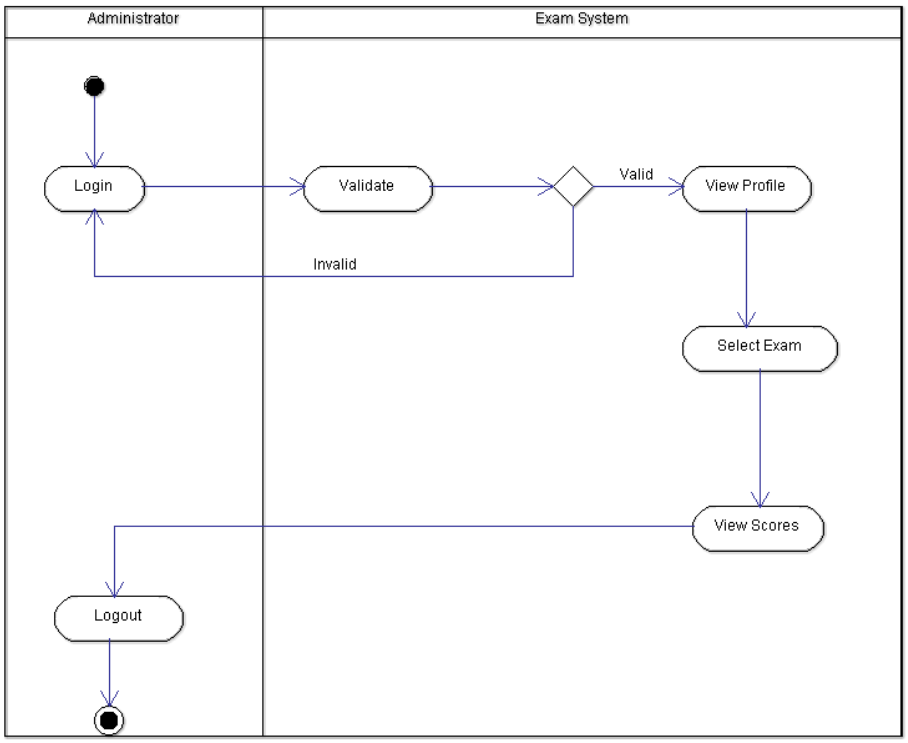


3.7 Admin- View Scores

3.7.1 Sequence Diagram

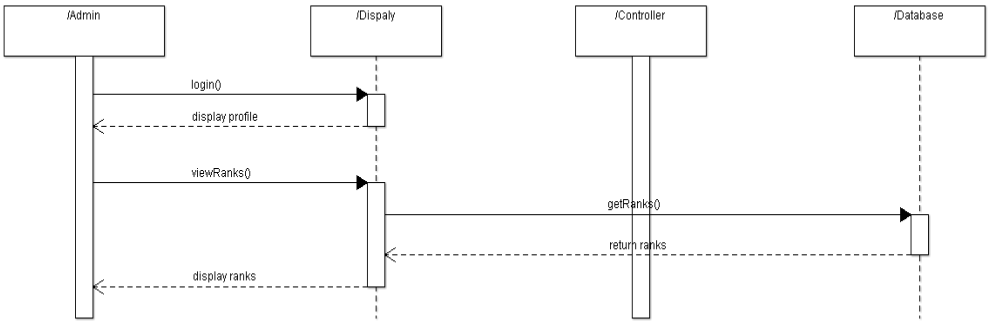


3.7.2 Activity Diagram

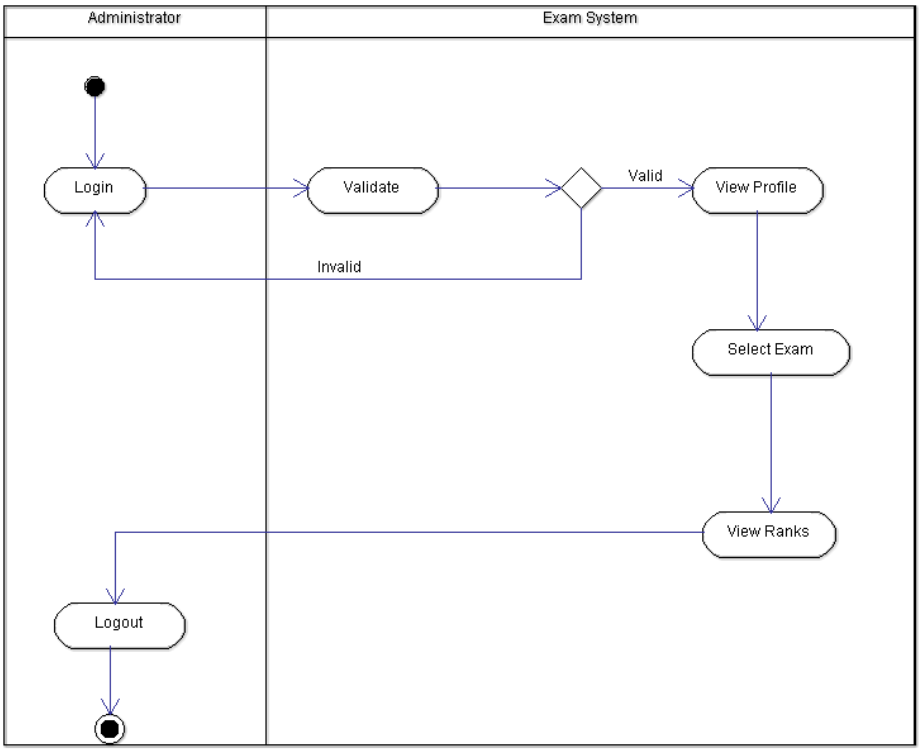


3.8 Admin- View Ranks

3.8.1 Sequence Diagram

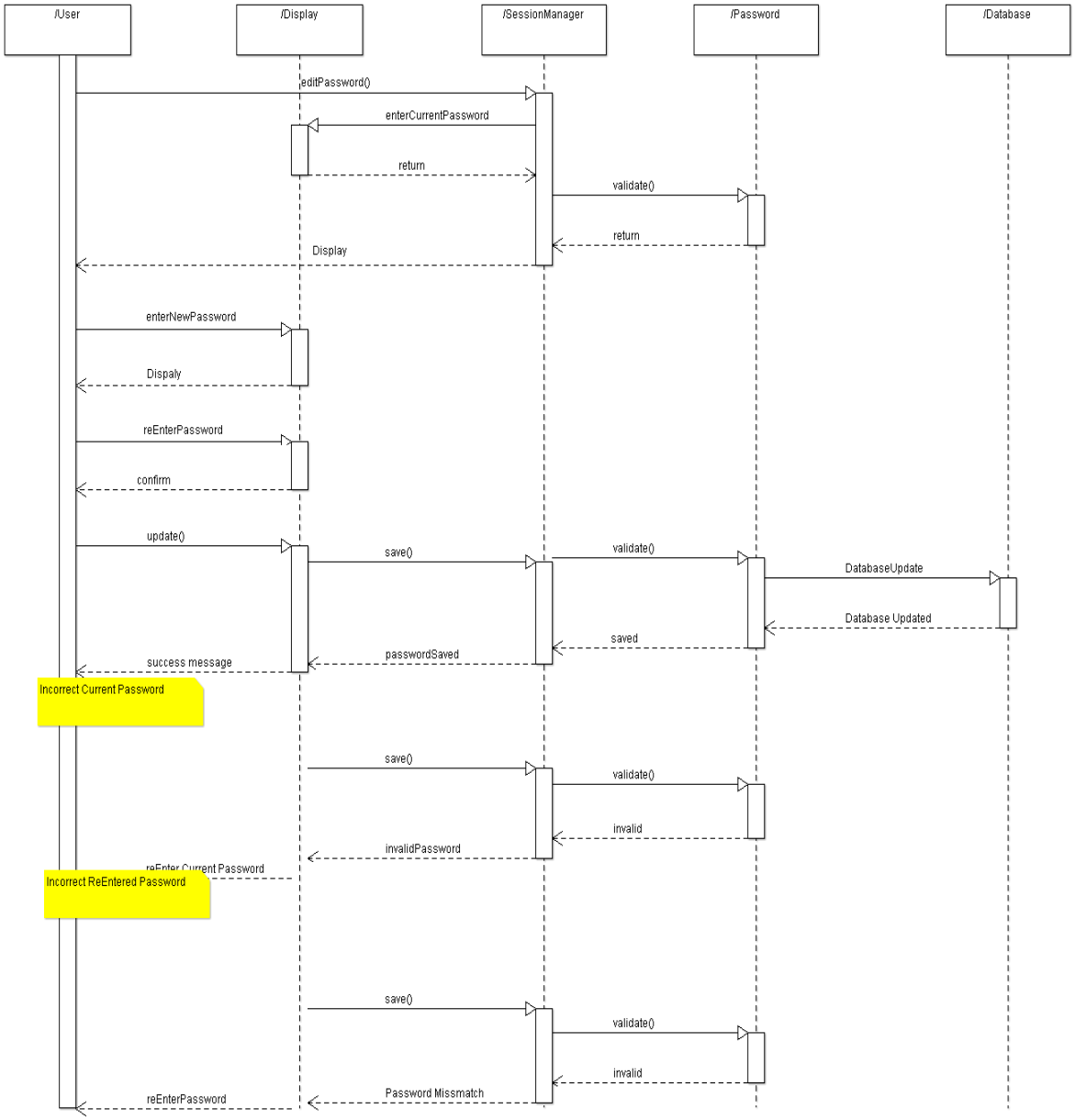


3.8.2 Activity Diagram

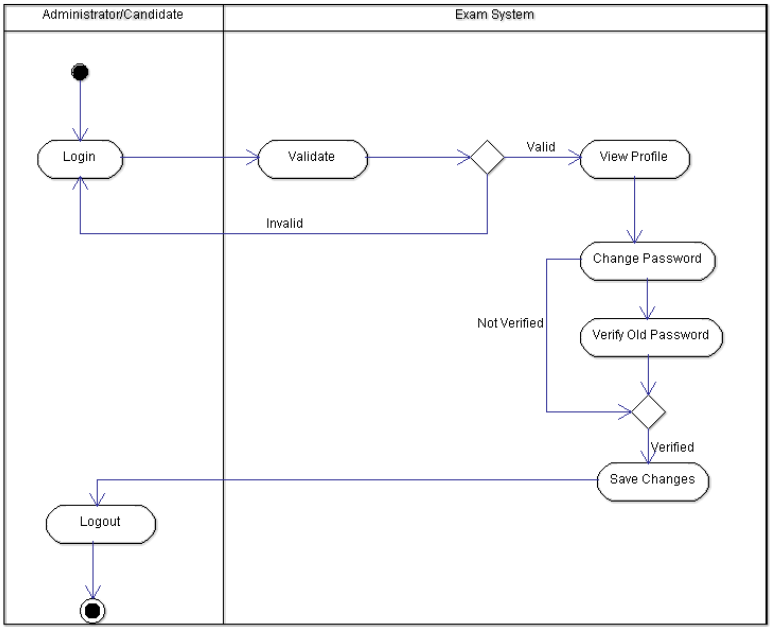


3.9 Admin/Candidate- Change Password

3.9.1 Sequence Diagram

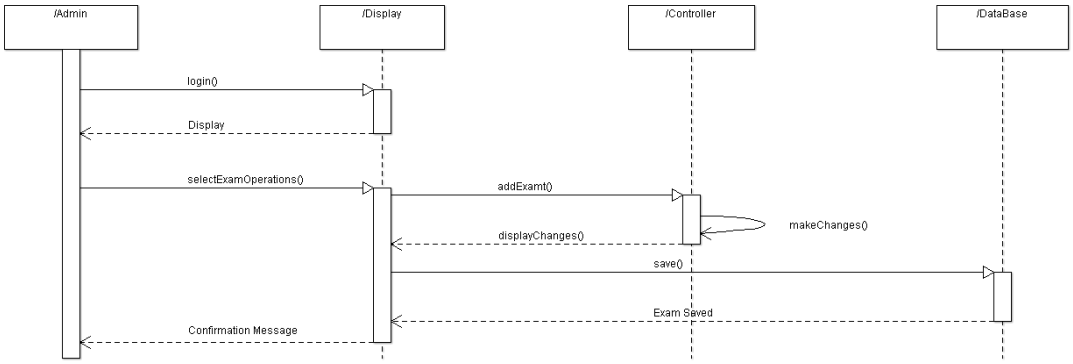


3.9.2 Activity Diagram

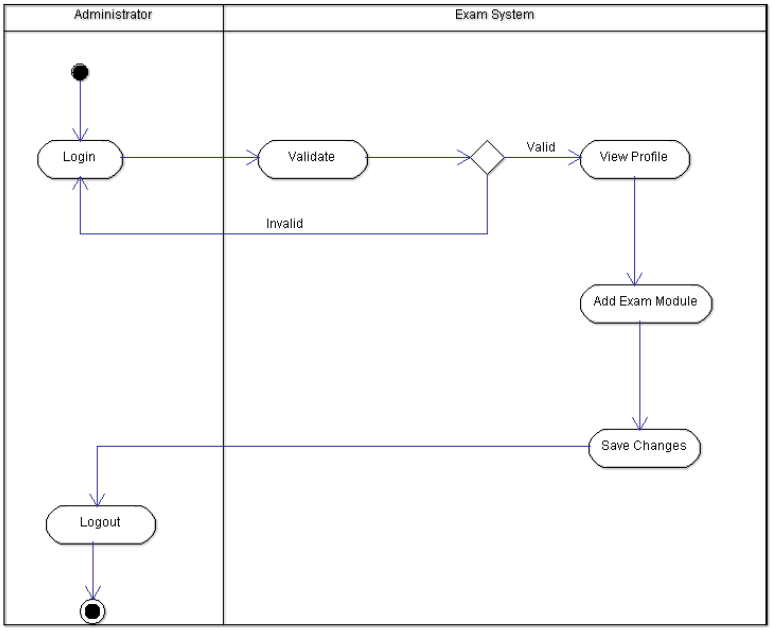


3.10 Add Exam Module

3.10.1 Sequence Diagram

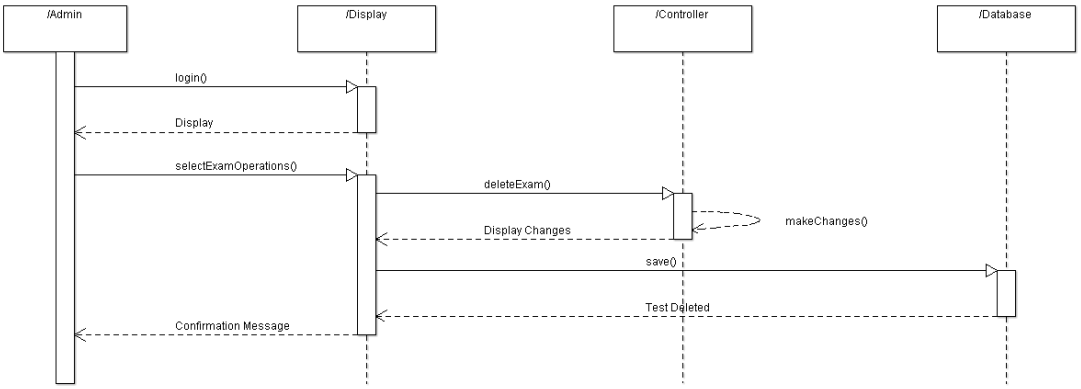


3.10.2 Activity Diagram

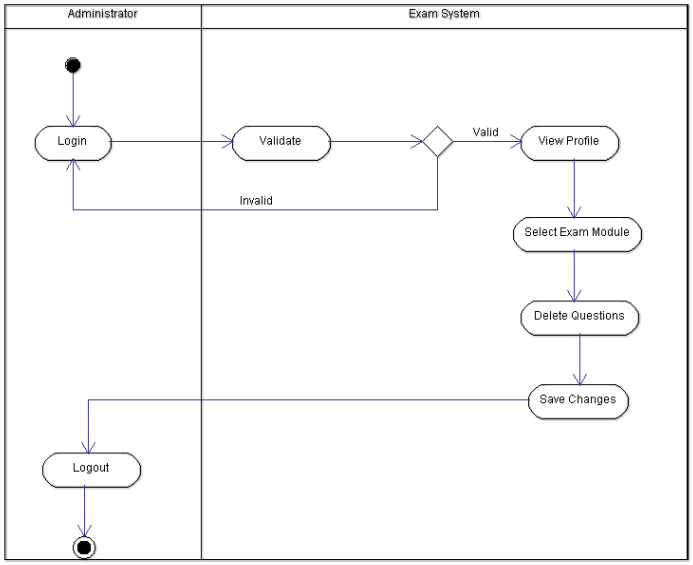


3.11 Delete Exam Module

3.11.1 Sequence Diagram

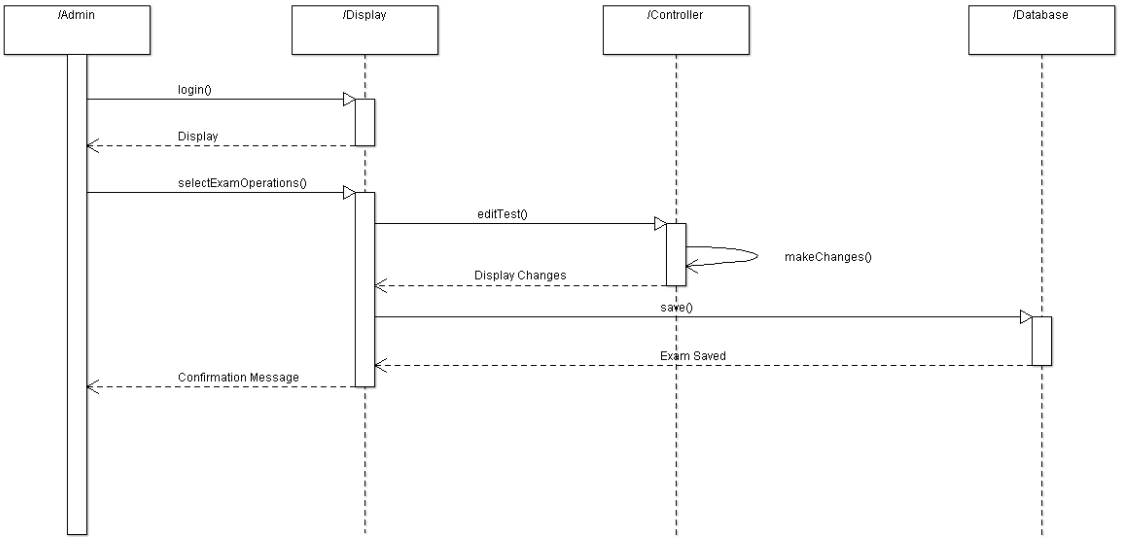


3.11.2 Activity Diagram

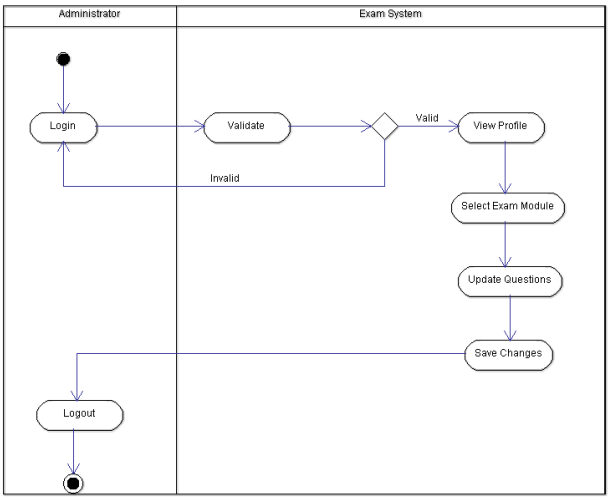


3.12 Update Exam Module

3.12.1 Sequence Diagram



3.12.2 Activity Diagram



4. Execution Architecture

Runtime environment required for the Online Exam Portal is any device with a web browser and an internet connection. The deployment platform used is Angular.

4.1 Reuse and relationships to other products

N/A

5. Design decisions and tradeoffs

By using trade-offs we want to give small software projects a simple way to have an efficient decision making process. This is an important criteria for us, since we want to use methods that are well known, and easy and intuitive to understand and use. We limited the number of functionalities as we made a trade-off between “sooner-but-worse” and “later-but-better”. AngularJS was used because it is a structural framework for dynamic web apps that makes much of the code you would otherwise have to write completely redundant. It is easy to integrate third party features with AngularJS as Angular integration comes pre-built into frameworks.

The decision to use SpringBoot was because it completely uses new development model to make Java Development very easy by avoiding some tedious development steps and boilerplate code and configuration and increases productivity.

6. Pseudocode for components

7. Appendices (if any)

N/A

Software Design Specification

Version 2.0

Gayathri - U101116FCS039

Arnav Ajay – U101116FCS014

Neeraj Pandey – U101116FCS076

Ashwin Jawahar – U101116FCS017

Hrithik Raj – U101116FCS047

1. Introduction

1.3 Purpose of this document

This Software Design Specification Document is made with the purpose of outlining the software architecture and design of the Online Examination Portal in detail. This document will provide developers an insight in meeting the client's needs efficiently and effectively. Moreover, the document facilitates communication and understanding of the system by providing sequence diagrams, collaboration models, state diagrams and other required documents of the system design. This document will demonstrate how the design will accomplish the functional and non-functional requirements captured in the Software Requirement Specification Document. The document will provide a framework to the programmers through describing the architecture, sub-systems, interfaces, database design and functions.

1.2 Scope of the development project

The scope of this project is very broad as compared to manually taken examinations:

4. This portal is not only limited to educational institutes but also to the corporate world
5. There is no restriction as to the presence of an examiner during the period of the exam
6. Less time consumption, thereby increasing the efficiency

1.4 Definitions, acronyms, and abbreviations

IEEE: Institute of Electrical and Electronics Engineers
SDS: Software Design Specification

1.4 References

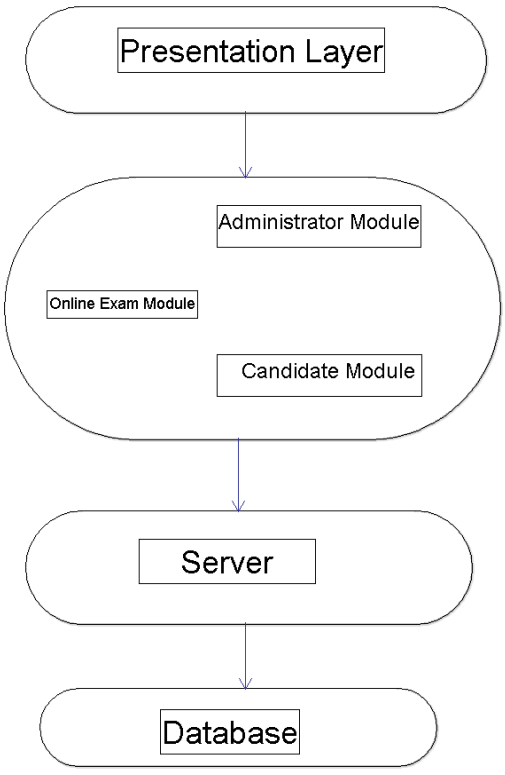
3. R. S. Pressman, Software Engineering: A Practioner's Approach, 5th Ed, McGraw-Hill, 2001.
4. IEEE SDS template

1.5 Overview of document

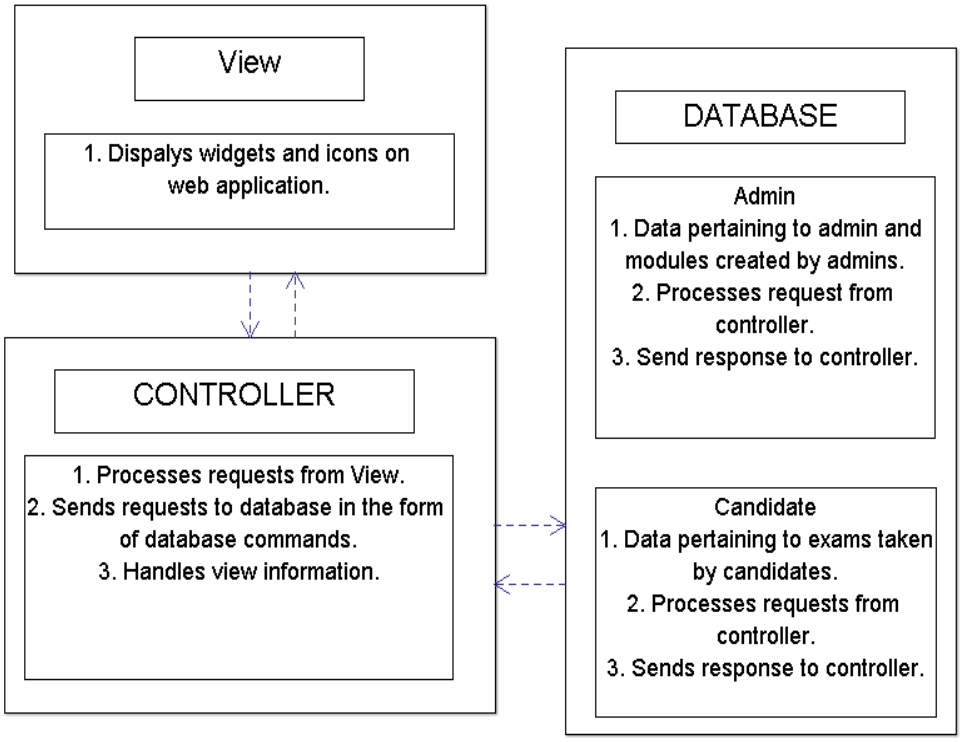
The SDS document is divided into the following sub-sections:

8. Introduction: It mainly describes the purpose of the document and scope of the development project.
9. Conceptual Architecture/Architecture Diagram: The intent of conceptual architecture is to direct attention at an appropriate decomposition of the system without delving into the details of interface specification.
10. Logical Architecture: Logical architecture is a structural design that gives as much detail as possible without constraining the architecture to a particular technology or environment.
11. Execution Architecture: The execution architecture determines largely the realtime and performance behavior of a system. Concepts such as latency, response time and throughput are illustrated.
12. Design Decisions and Trade-offs: It describes the decisions taken throughout the system design and why they were taken instead of their alternatives.
13. Pseudocode: Pseudocode/algorithms implemented in the system.
14. Appendices: Subsidiary matters, if any.

2. Conceptual Architecture/Architecture Diagram

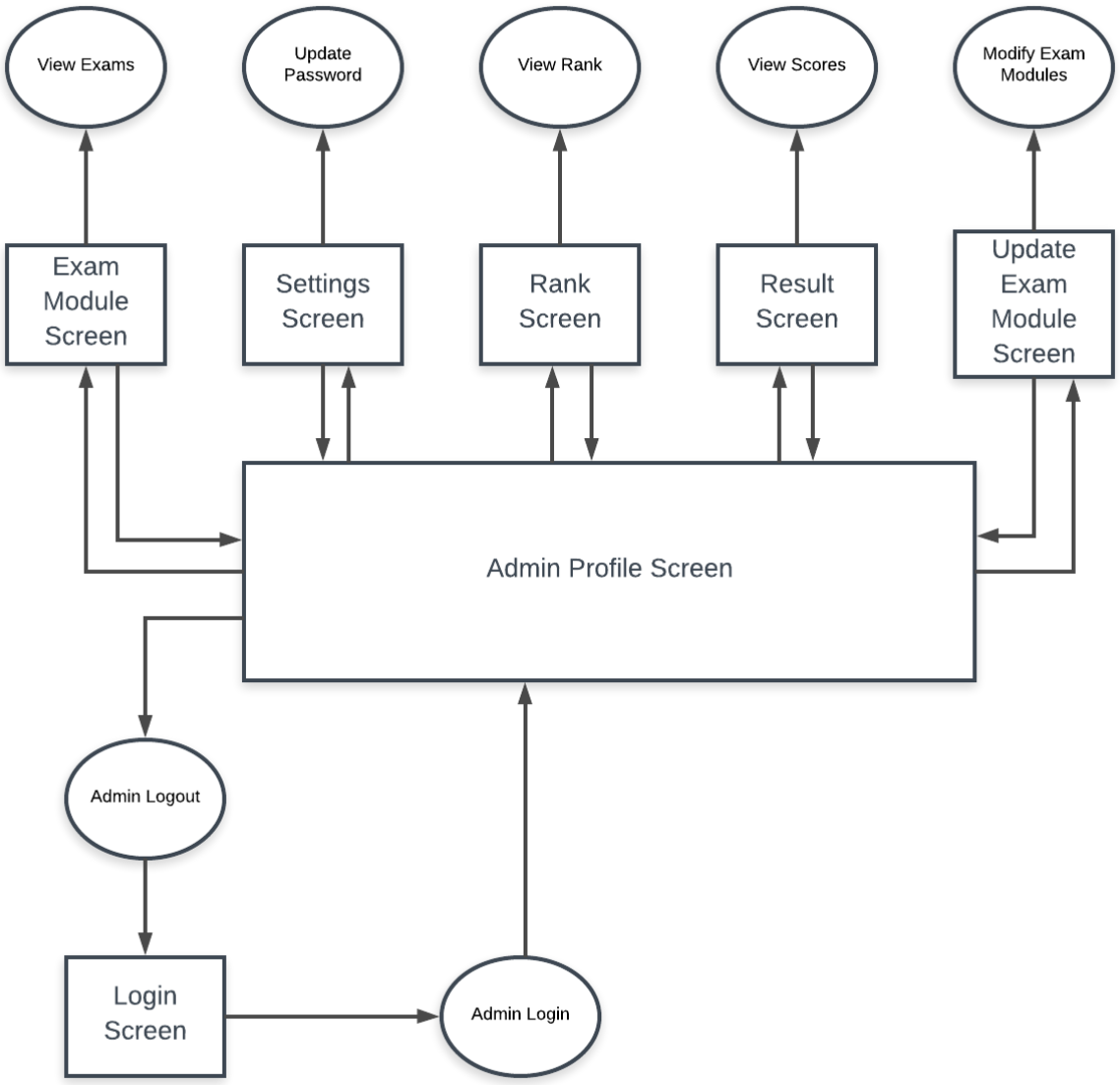


2.1 Overview of modules / components

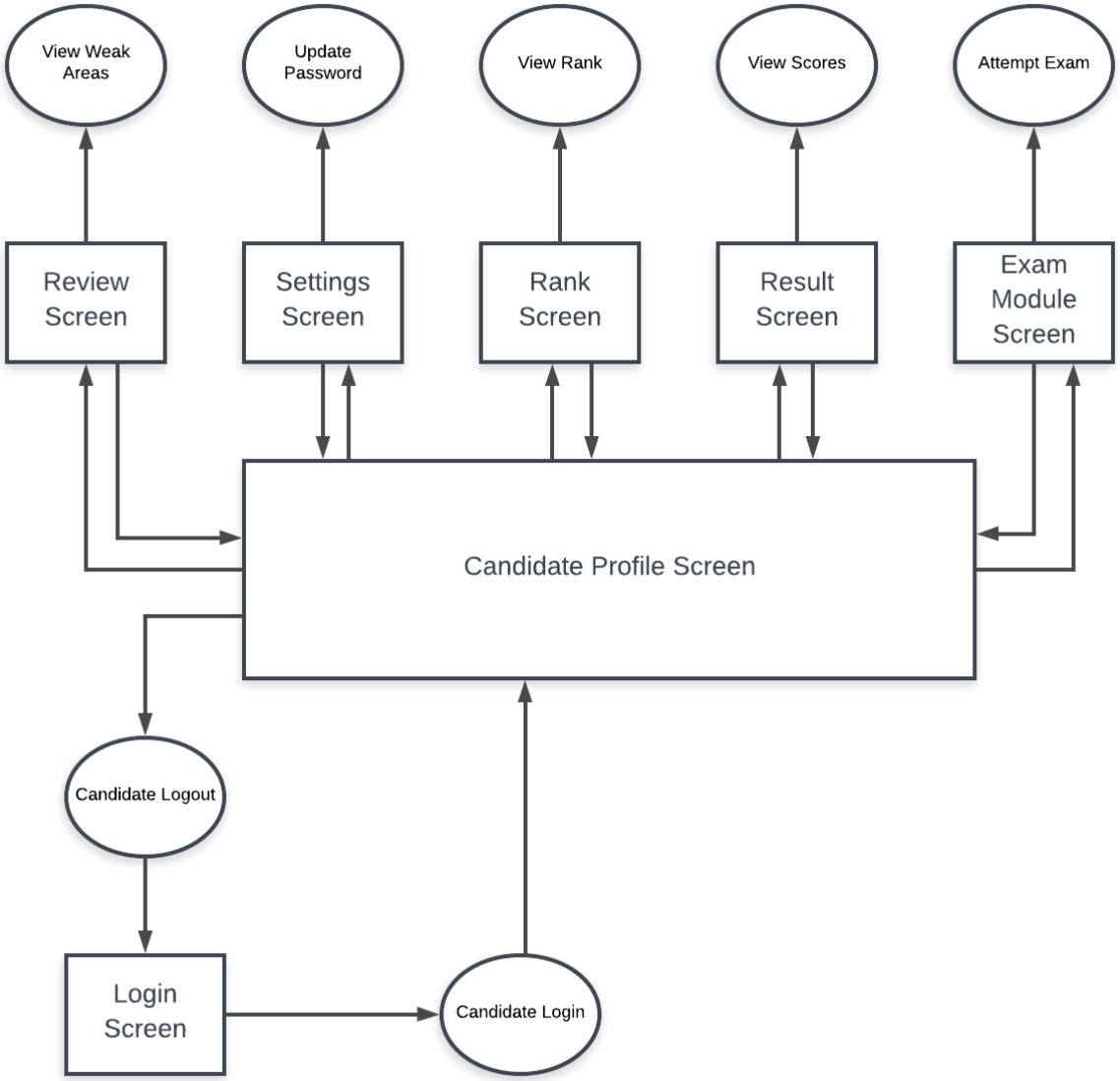


2.2 Structure and relationships

4.2.1 Administrator



4.2.2 Candidate

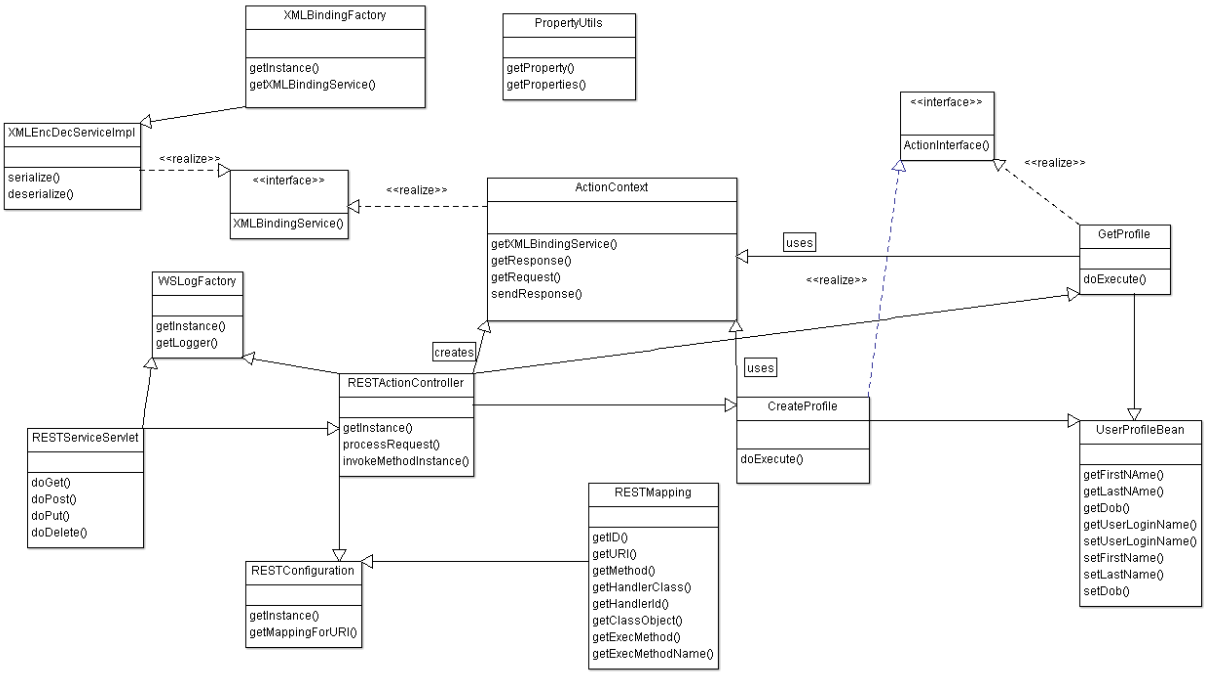


2.3 User interface issues

Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. No user would be able to access any part of the application without logging on to the system. User Interface Issues for the possible users of the system are:

3. User A is a 23-year-old student or corporate candidate who is looking to take a mock online examination using this system. As the user is young and technologically proficient in using computer applications and systems, the user interface will be of fairly common conventions on the candidates' screen.
4. User B is a 35-year-old person who is an administrator of the online system with responsibilities to update exam modules on the site by modifying the questions. Since, user B might not be completely well-versed in using online portals, directives will be given for ease of use.

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

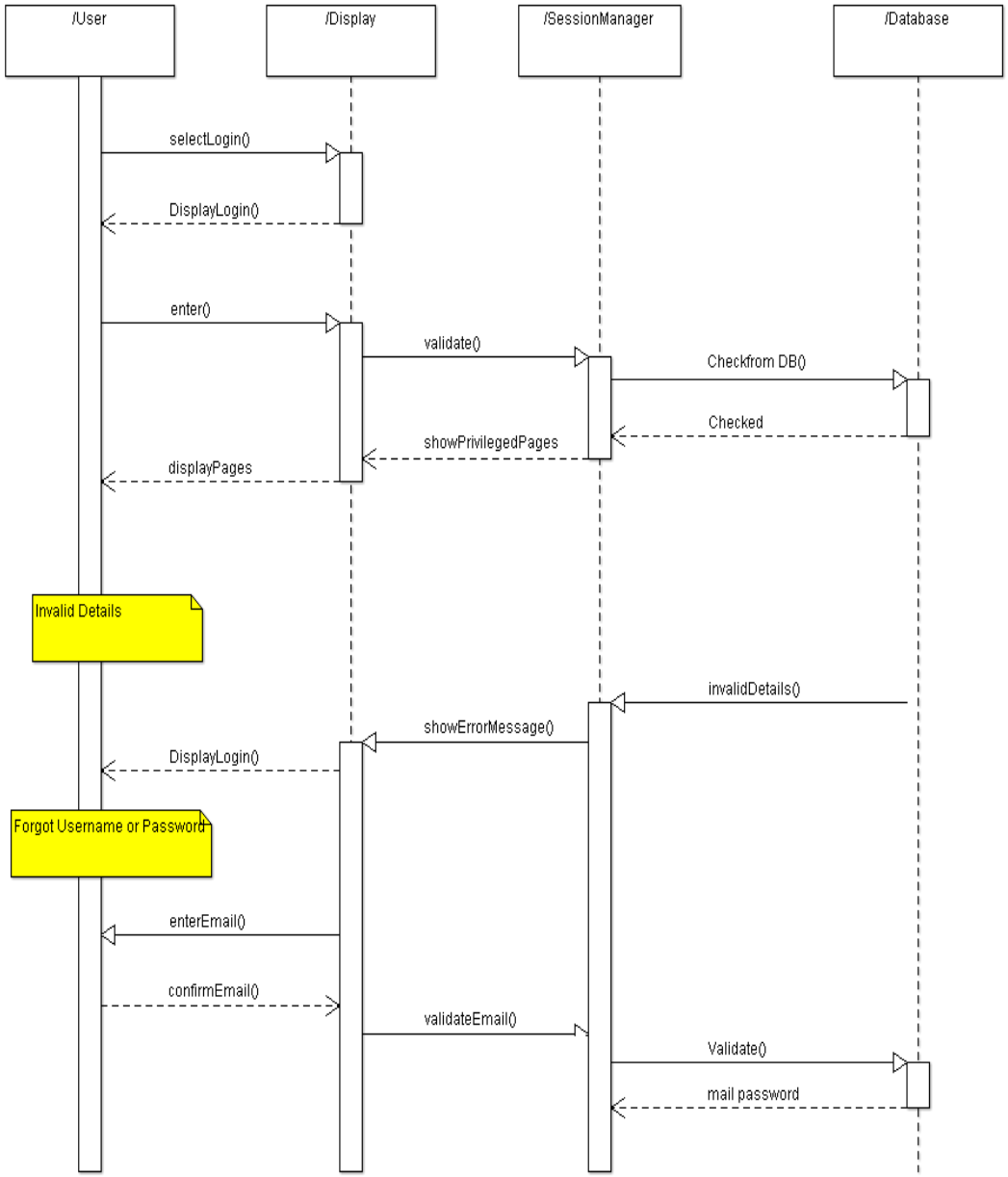


3.1 Logical Architecture Description

Representational State Transfer is a software architectural style that defines a set of constraints to be used for creating web services. Web services that conform to the REST architectural style, or RESTful web services, provide interoperability between computer systems on the Internet. One of the key advantages of REST APIs is that they provide a great deal of flexibility. Data is not tied to resources or methods, so REST can handle multiple types of calls, return different data formats and even change structurally with the correct implementation of hypermedia. This explains the class structure in Java. The rest of the functionalities use bootstrap and typescript, the architecture of which is given below.

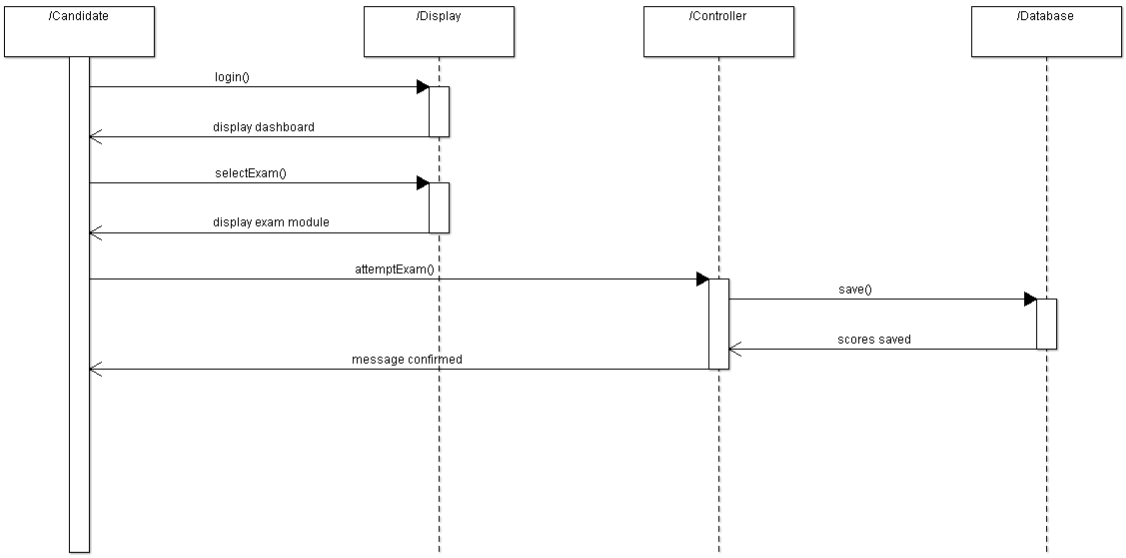
3.2 Login

3.2.1 Sequence Diagram

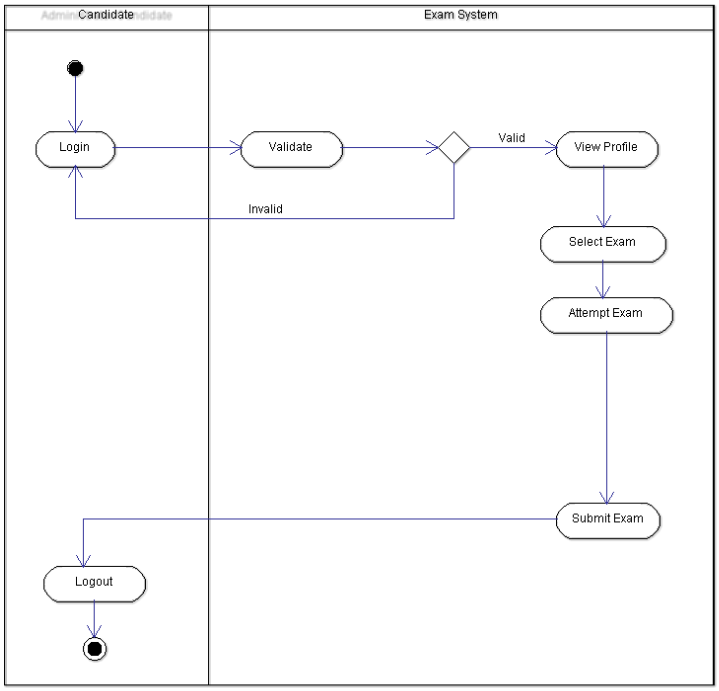


3.3 Attempt Exam

6.3.1 Sequence Diagram

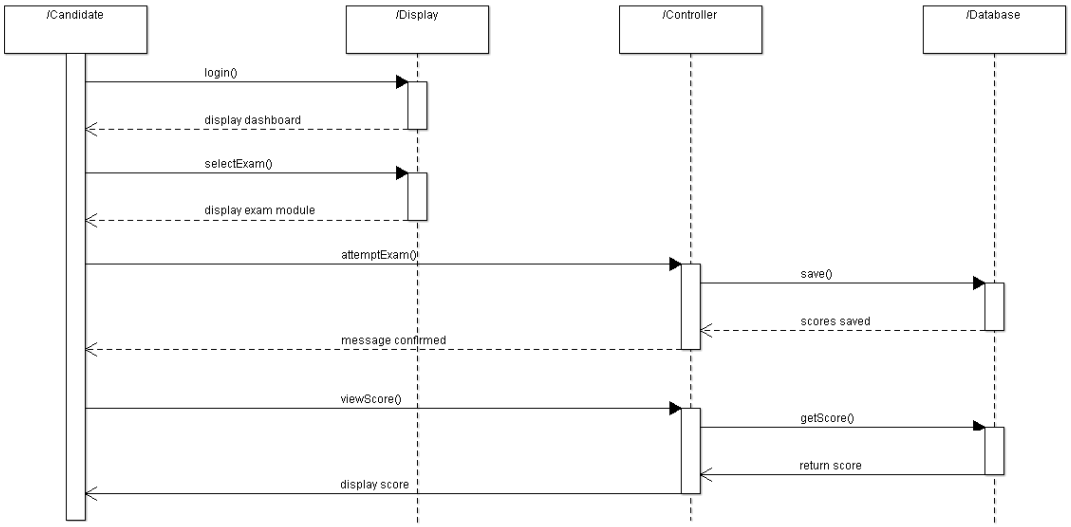


3.3.2 Activity Diagram

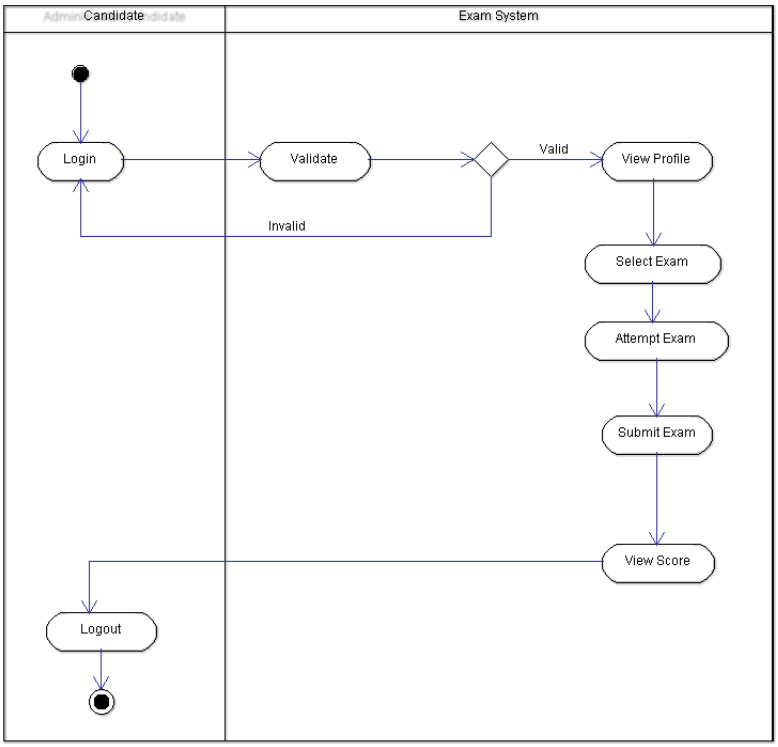


3.4 View Score

3.4.1 Sequence Diagram

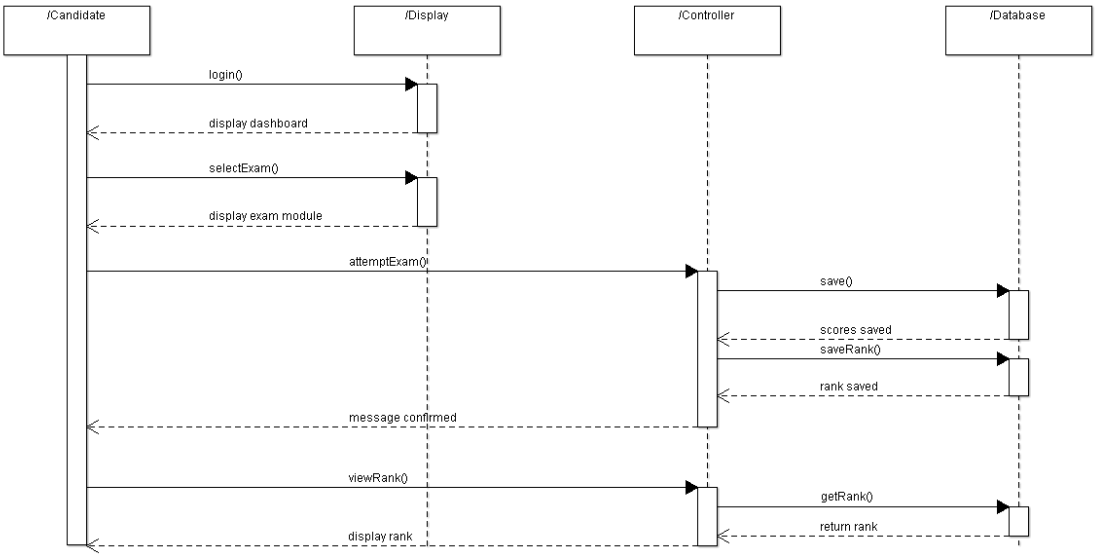


3.4.2 Activity Diagram

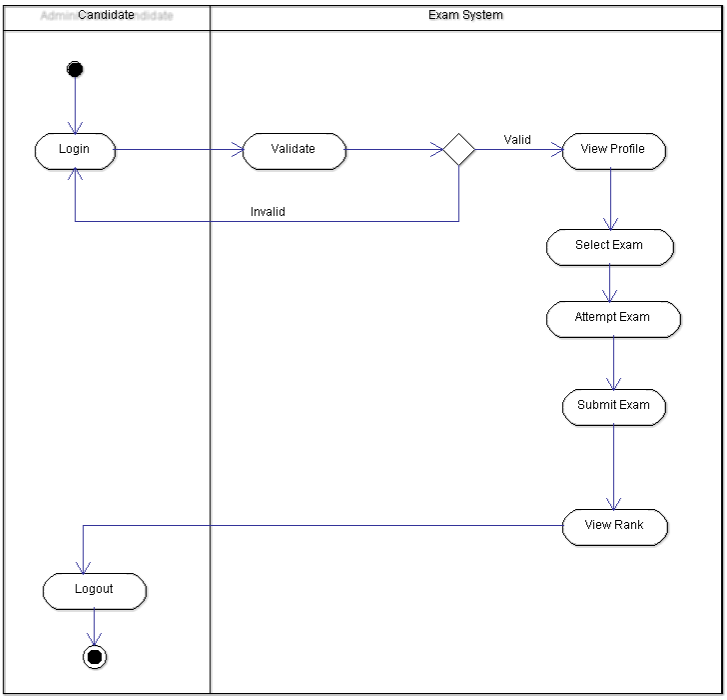


3.5 View Rank

3.5.1 Sequence Diagram

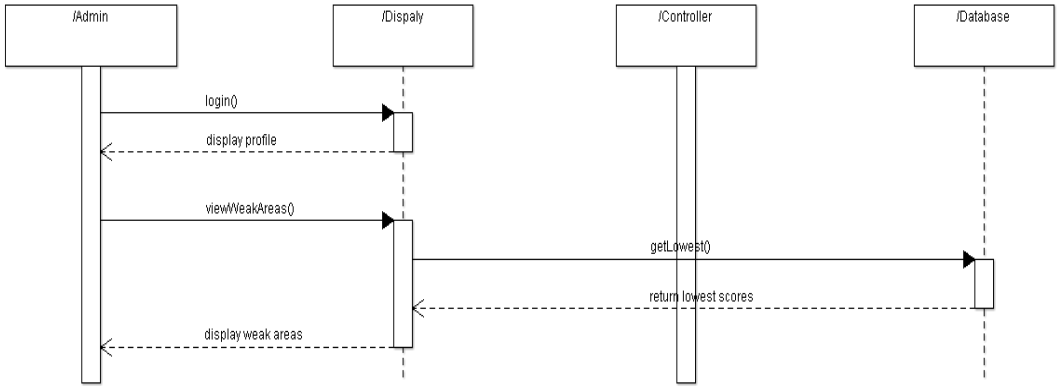


3.5.2 Activity Diagram

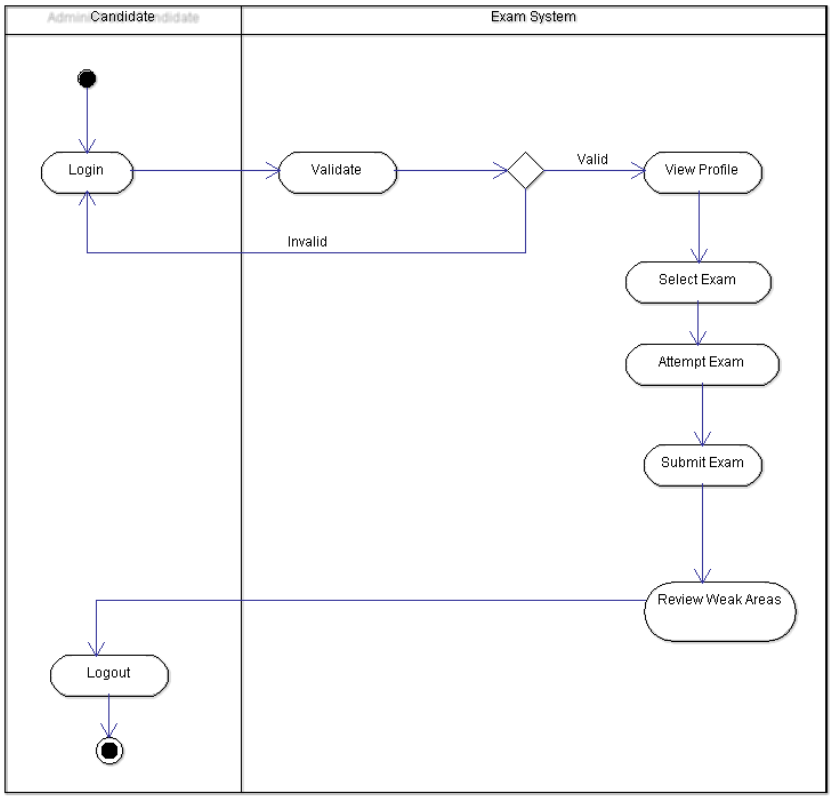


3.6 View Weak Areas

3.6.1 Sequence Diagram

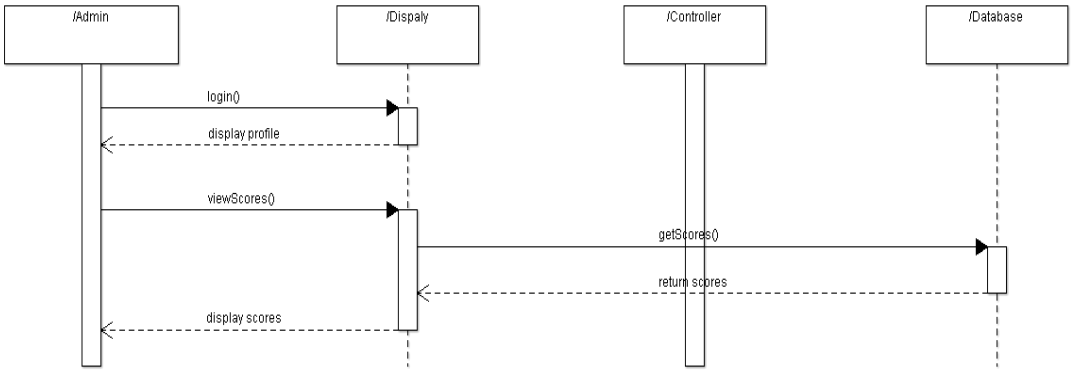


3.6.2 Activity Diagram

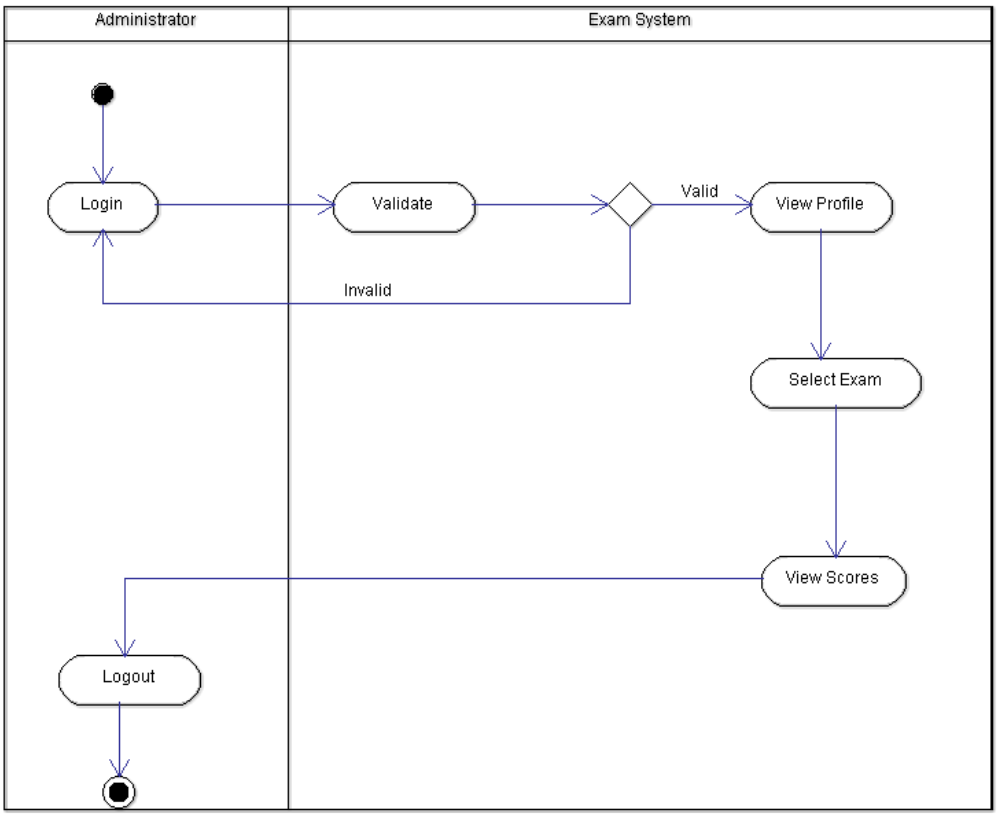


3.7 Admin- View Scores

3.7.1 Sequence Diagram

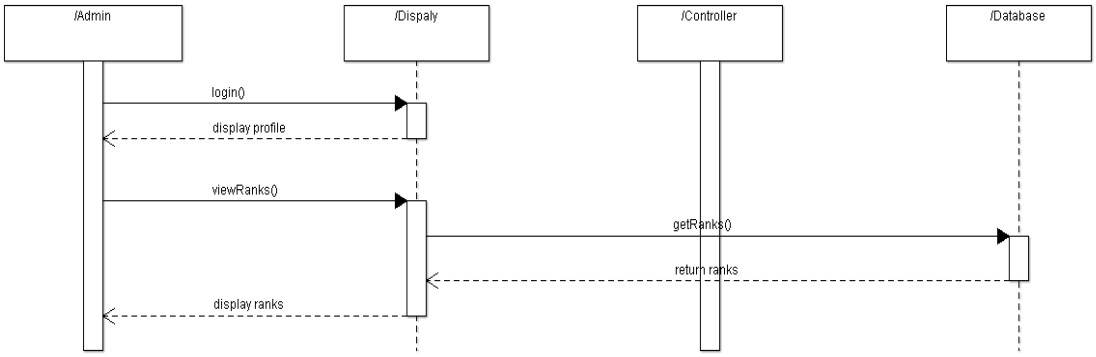


3.7.2 Activity Diagram

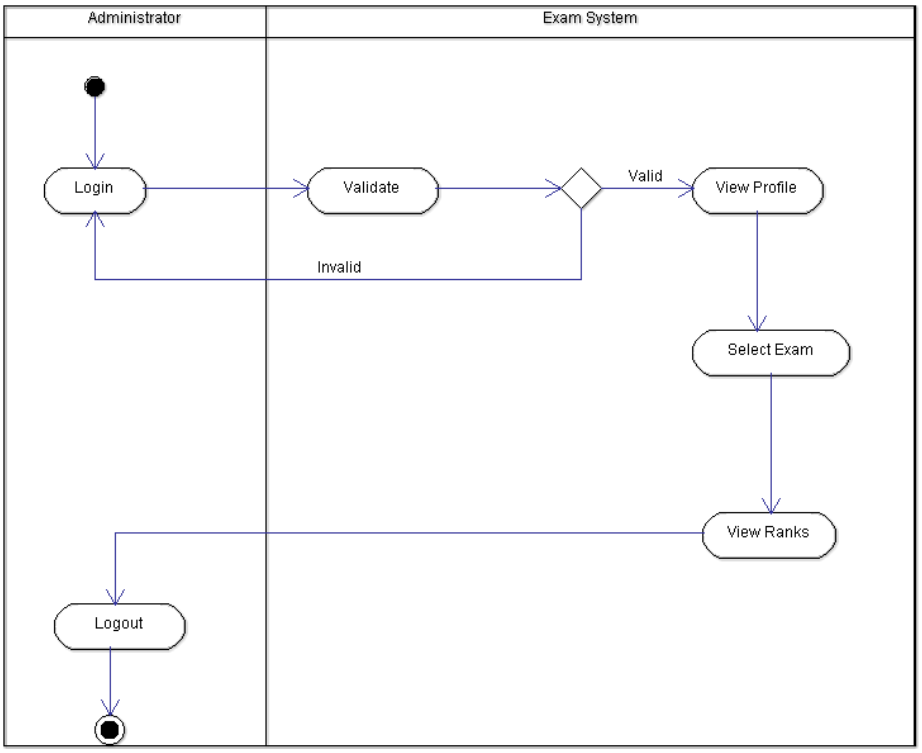


3.8 Admin- View Ranks

3.8.1 Sequence Diagram

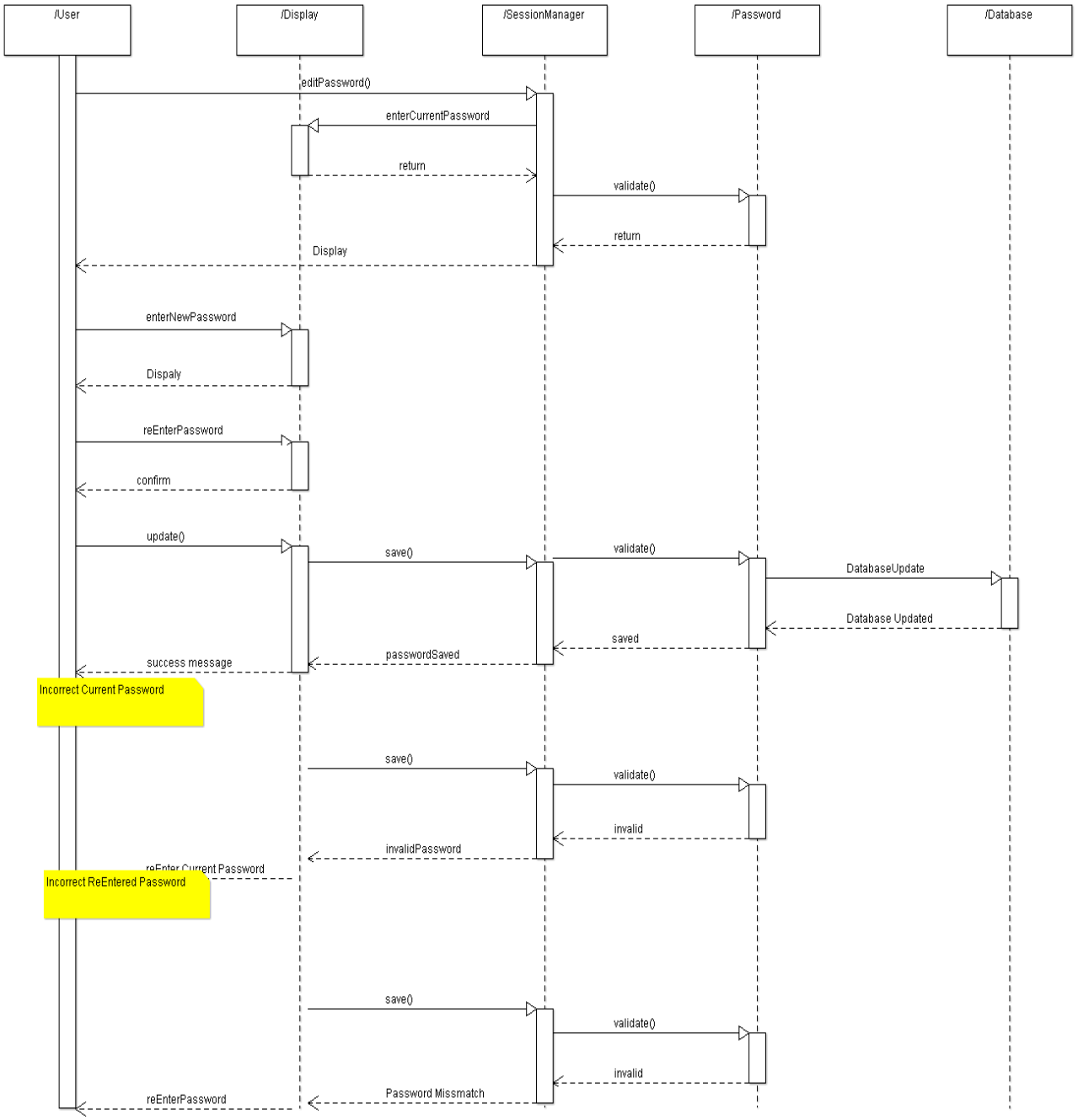


3.8.2 Activity Diagram

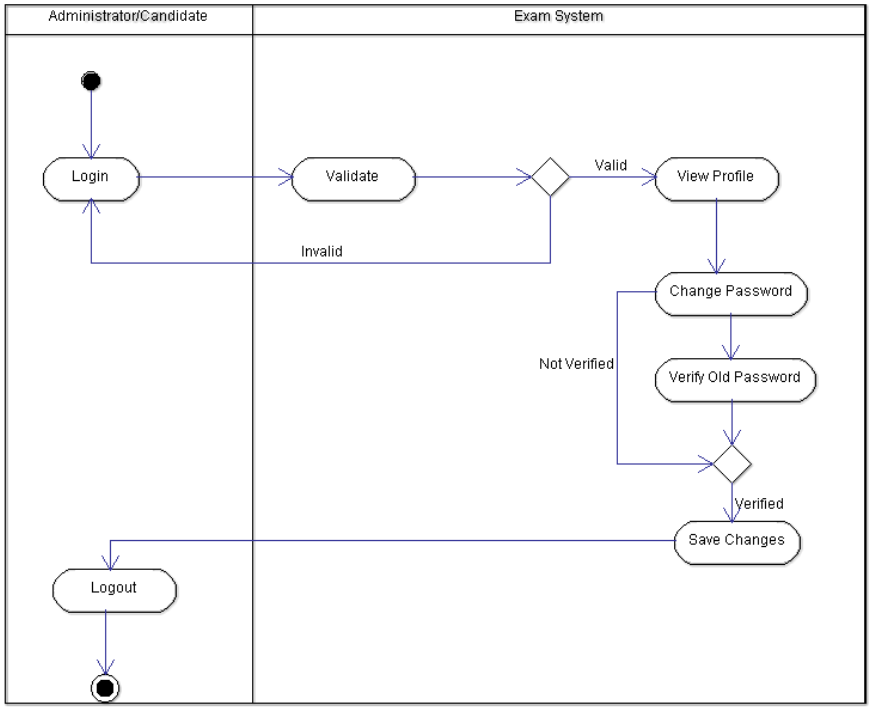


3.9 Admin/Candidate- Change Password

3.9.1 Sequence Diagram

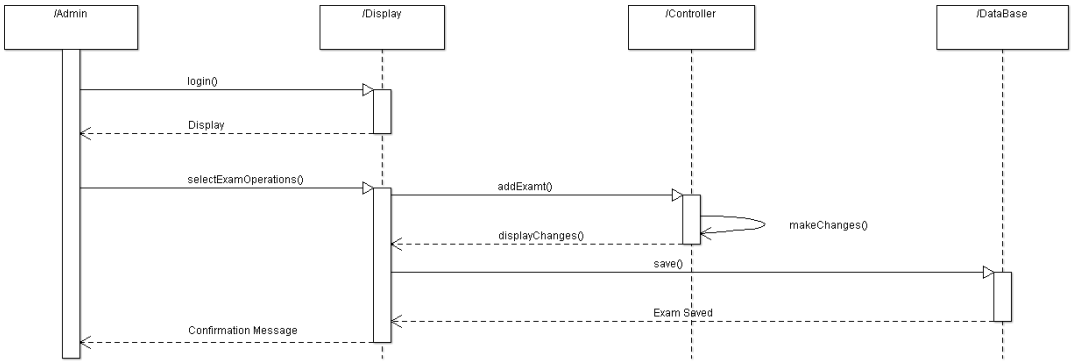


3.9.2 Activity Diagram

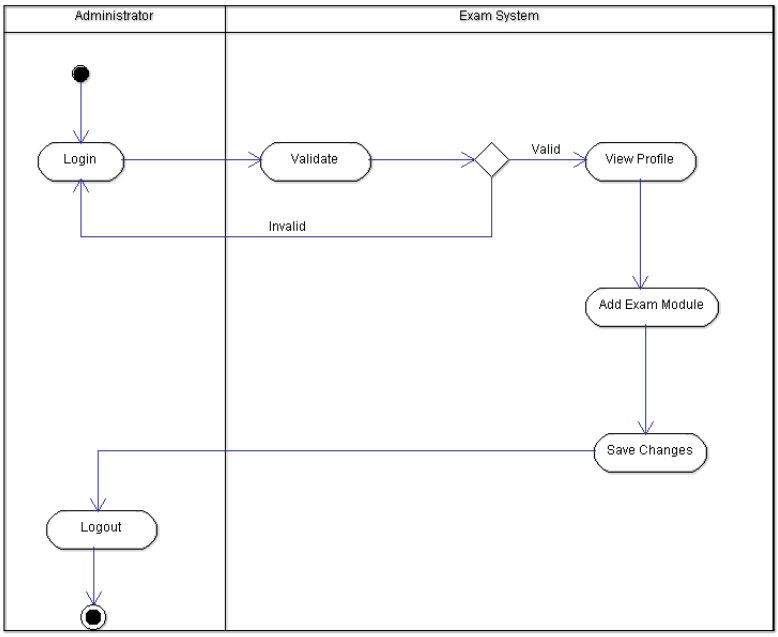


3.10 Add Exam Module

3.10.1 Sequence Diagram

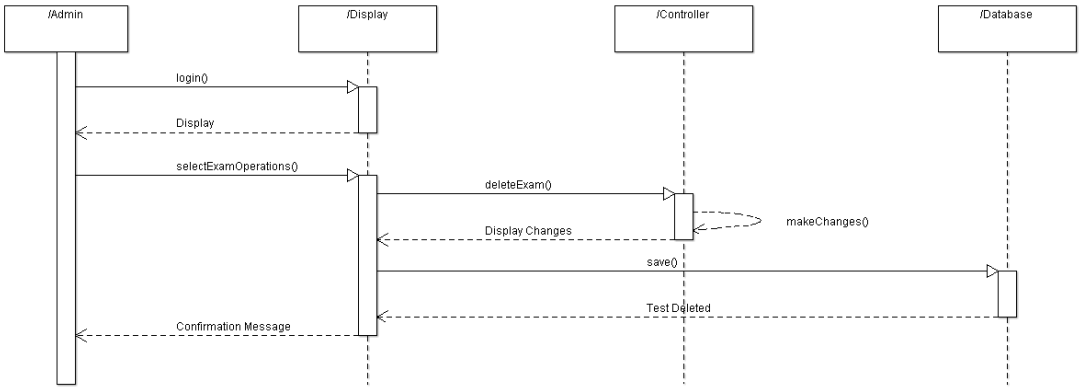


3.10.2 Activity Diagram

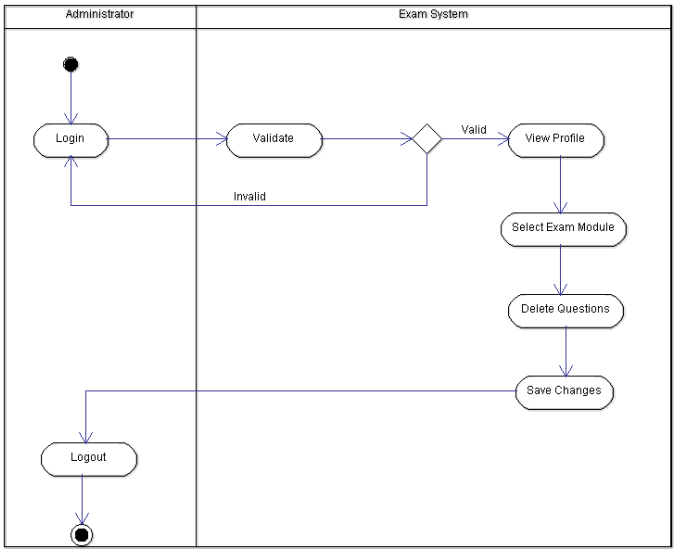


3.11 Delete Exam Module

3.11.1 Sequence Diagram

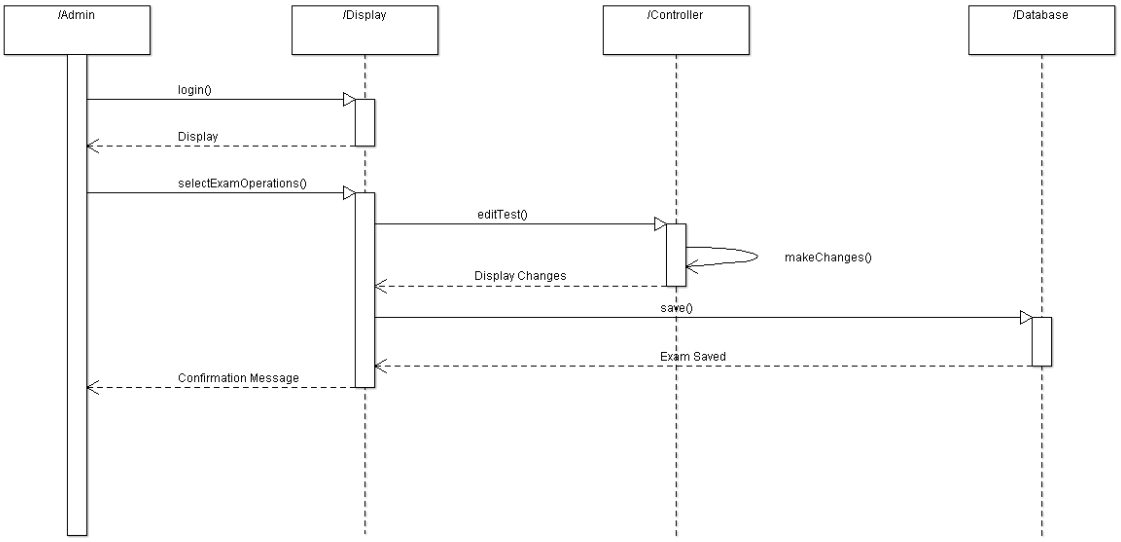


3.11.2 Activity Diagram

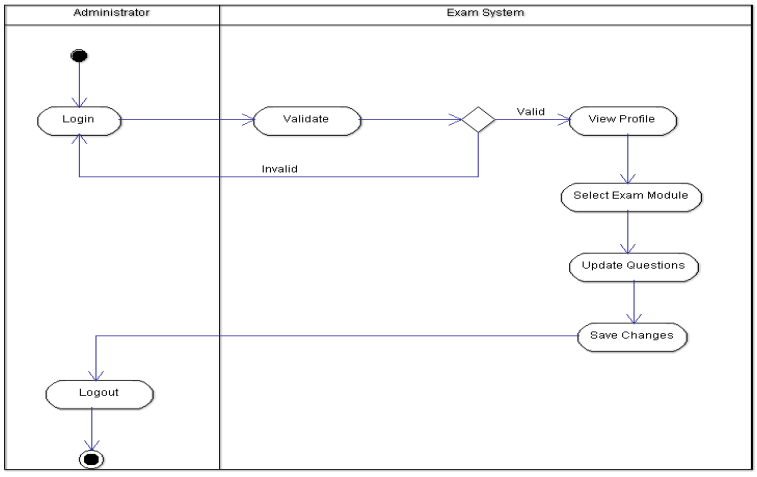


3.12 Update Exam Module

3.12.1 Sequence Diagram



3.12.2 Activity Diagram



4. Execution Architecture

Runtime environment required for the Online Exam Portal is any device with a web browser and an internet connection. The deployment platform used is Angular.

4.1 Reuse and relationships to other products

N/A

5. Design decisions and tradeoffs

By using trade-offs we want to give small software projects a simple way to have an efficient decision making process. This is an important criteria for us, since we want to use methods that are well known, and easy and intuitive to understand and use. We limited the number of functionalities as we made a trade-off between “sooner-but-worse” and “later-but-better”. AngularJS was used because it is a structural framework for dynamic web apps that makes much of the code you would otherwise have to write completely redundant. It is easy to integrate third party features with AngularJS as Angular integration comes pre-built into frameworks.

The decision to use SpringBoot was because it completely uses new development model to make Java Development very easy by avoiding some tedious development steps and boilerplate code and configuration and increases productivity.

6. Pseudocode for components

6.1 Sign-Up

1. Email field: input type = email, placeholder: “email address”
2. Password field: input type = password, placeholder: “Password”
3. Password confirmation field: input type = password, placeholder: “confirm password”
4. Username: input type = text, placeholder: “username”
5. Signup submit: value: “Sign Up”, default state, disabled
6. ##Signup Flow/Logic
7. ###On leaving email field
8. IF email is blank
9. Error message: “Please enter an email address.”
10. ELSE IF email field value is not a valid email address
11. Error message: “This doesn’t look like an email address. Please try
12. again.”
13. ###On leaving password field
14. IF password is not sufficiently strong
15. Error message: “Please replace with a stronger password.”
16. ###On leaving confirm password field
17. IF password confirmation does not match password
18. Error message: “Please replace with a stronger password.”
19. ###On leaving all fields
20. IF email AND password AND password confirmation all contain valid values
21. Enable Signup Submit

6.2 Sign-In

1. Username: input type = text, placeholder: “username”
2. Password field: input type = password, placeholder: “Password”
3. Signup submit: value: “Sign In”, default state, disabled
4. ##Sign In Flow/Logic
5. ###On leaving username field
6. IF username is blank
7. Error message: “Please enter a username”
8. ###On leaving password field
9. IF password is blank
10. Error message: “Please enter a password”
11. ###On leaving all fields
12. IF username AND password are registered

13. Enable Signup Submit
14. ELSEIF username and password combination wrong
15. Error message: “Please enter a valid username and password”

7. Appendices (if any)

N/A

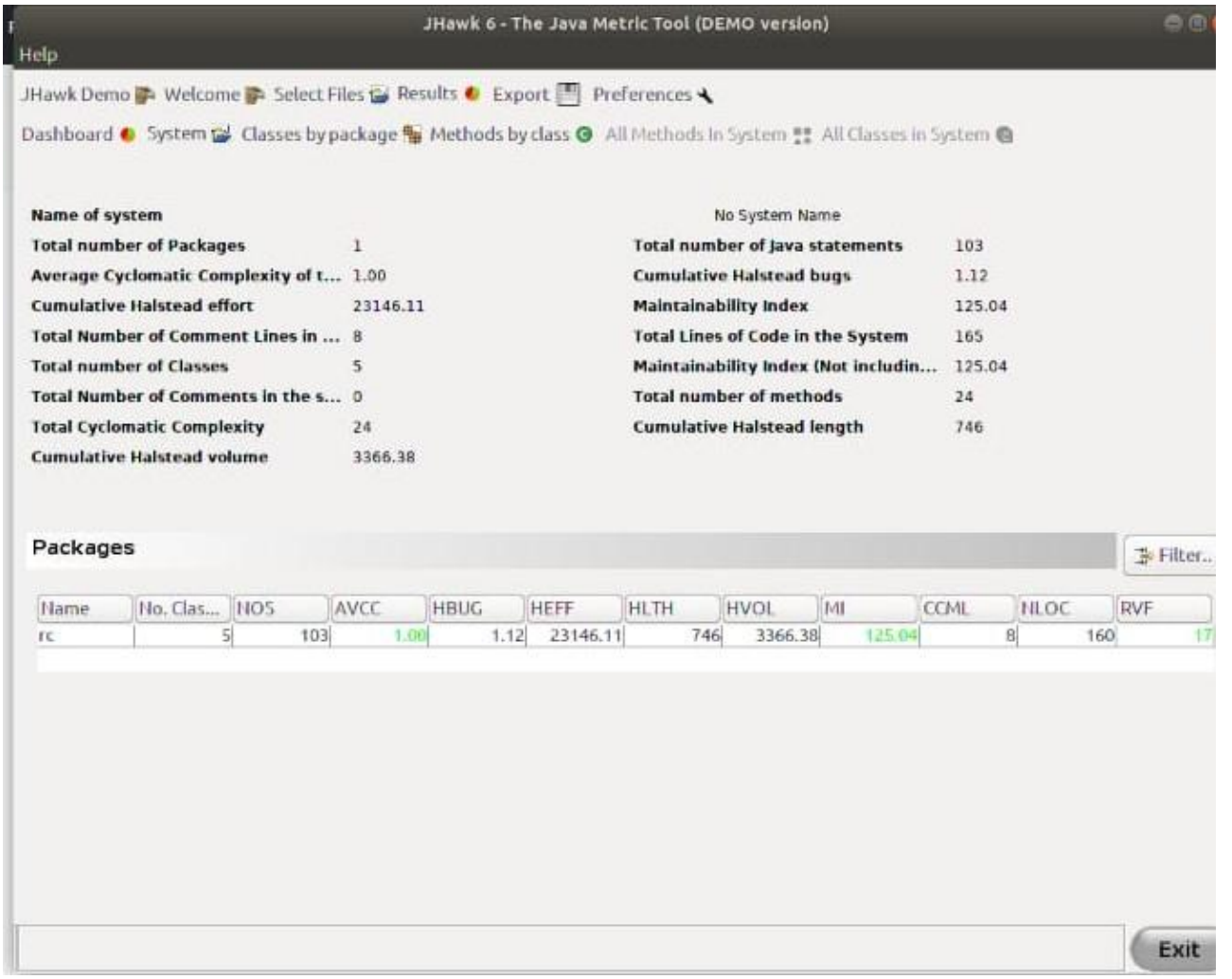
CHAPTER 3

Software Coding Metrics Document

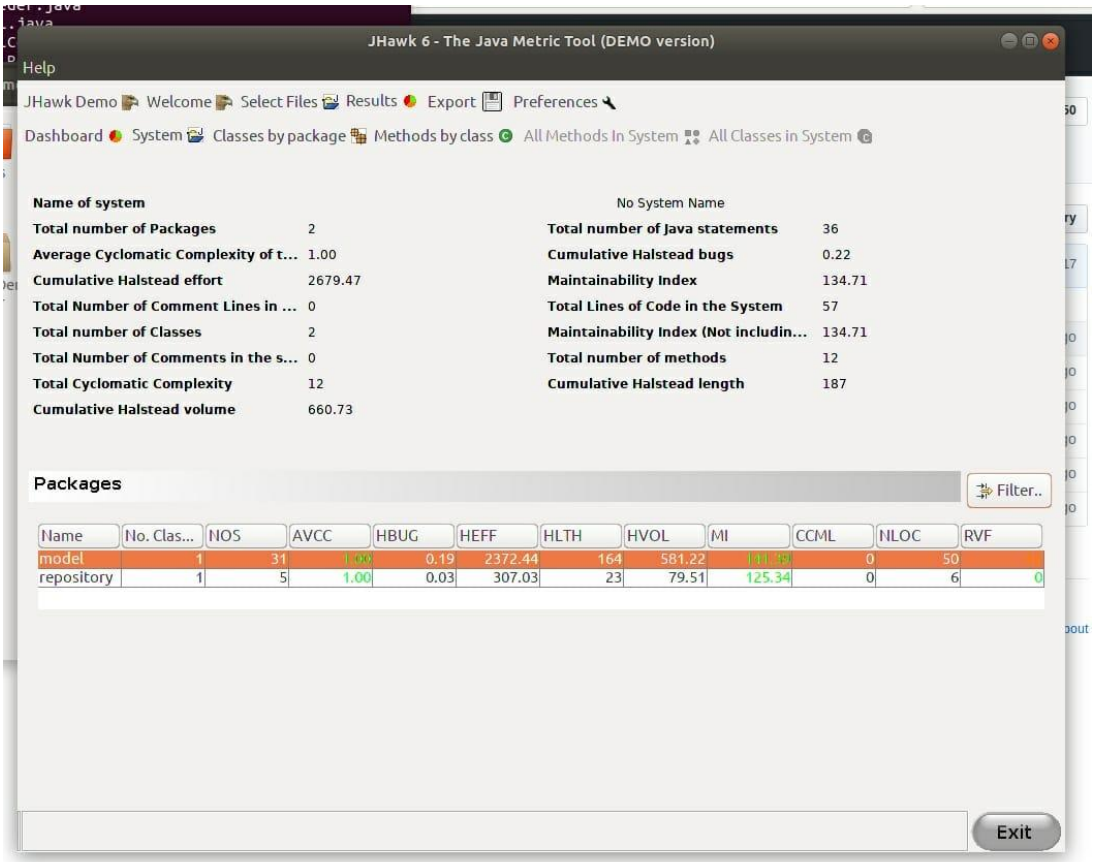
Coding Metrics

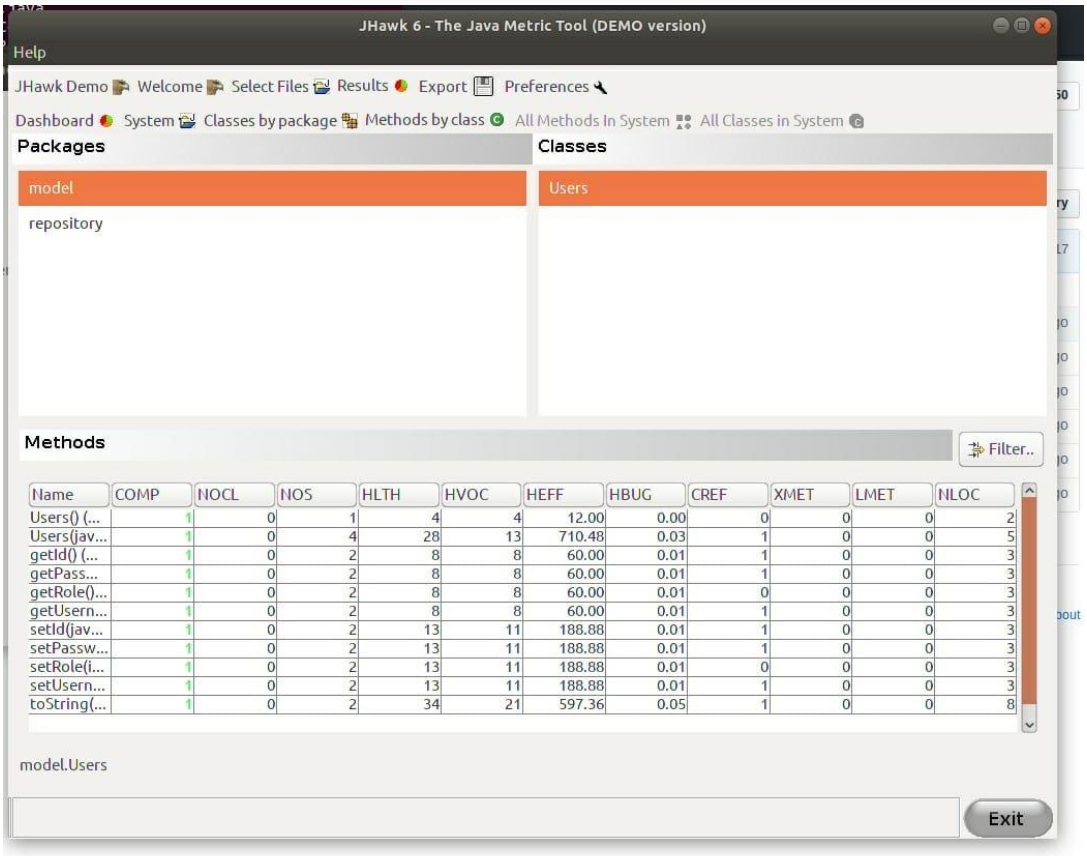
In this particular metric calculation, we are using the Jhawk Metric Tool. JHawk collects metrics at four different levels. The lowest level is the method level, then up to the class level, then the package level then, finally, the system level. The method, class and package levels correlate with the Java artefacts with these names. The 'system' level represents a group of these artefacts that you wish to examine collectively e.g. all the code in an application. Since only our backend, a small part of our project, is coded in java, mainly using Spring Boot and MongoDB , JHawk collects all metrics related to the classes, modules and sub-classes used.

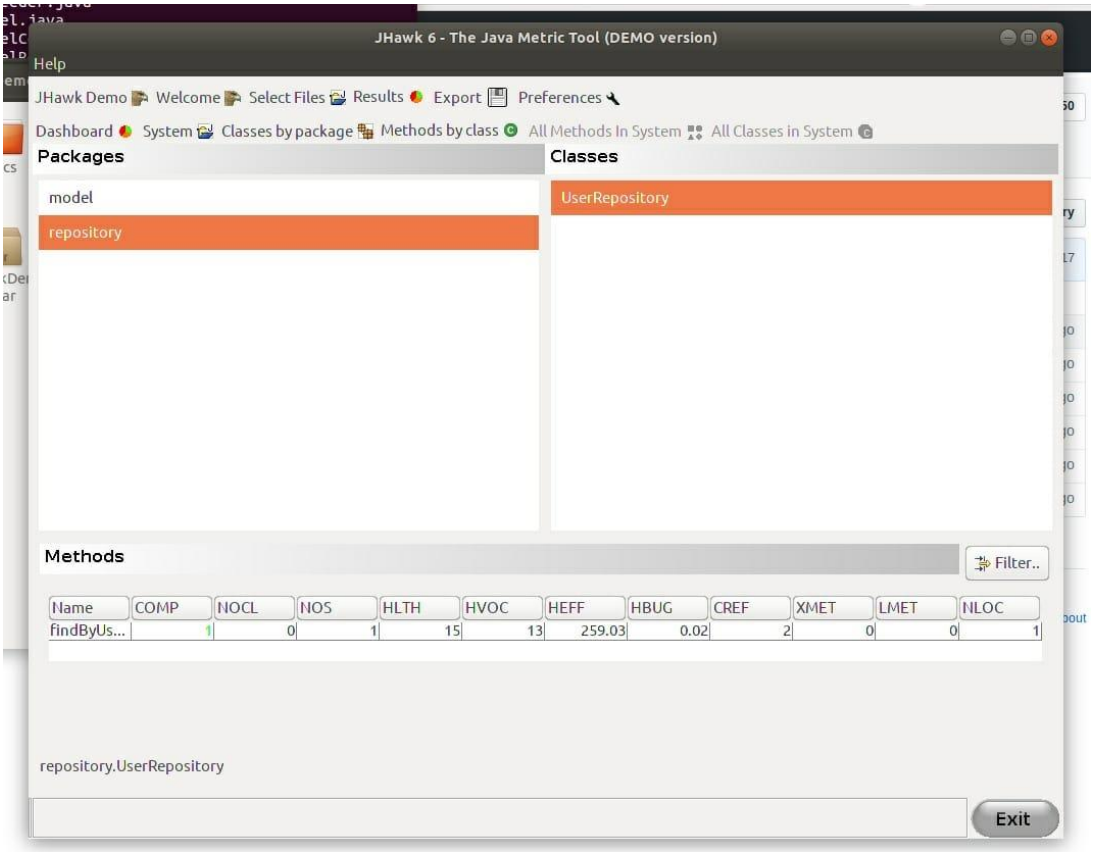
1. Database Connectivity using MongoDB

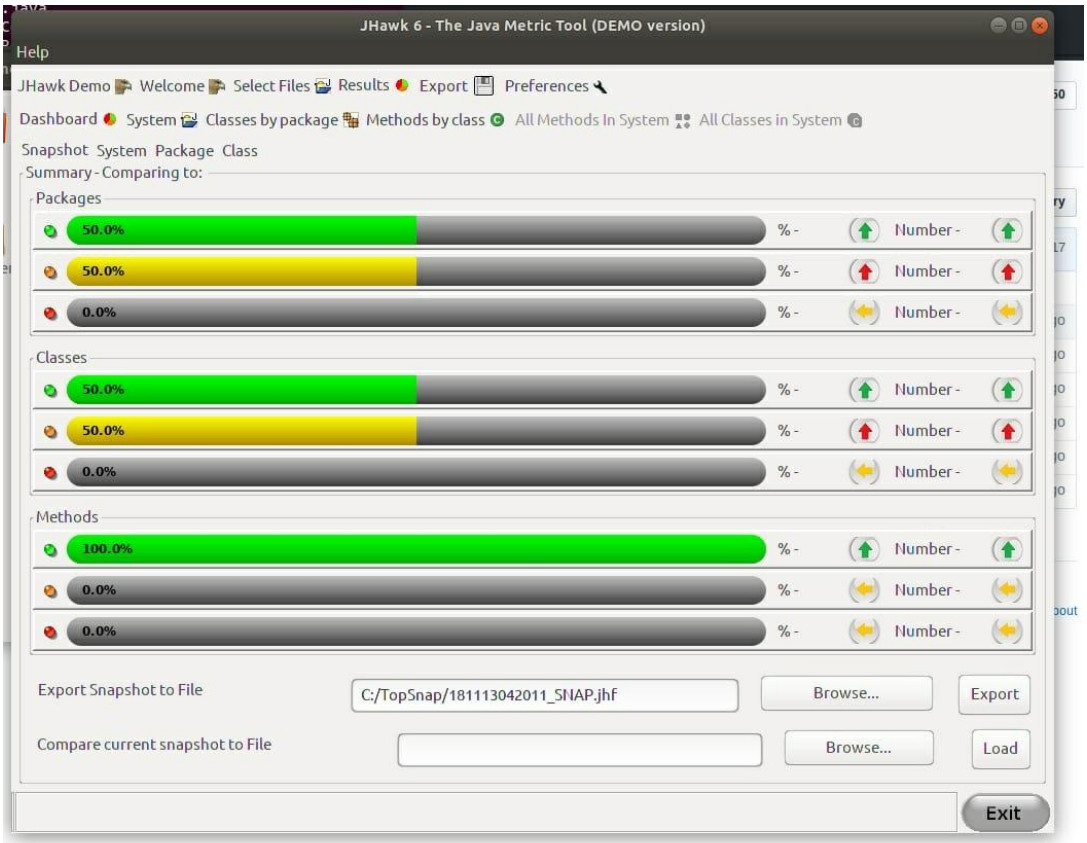


2. User Portal

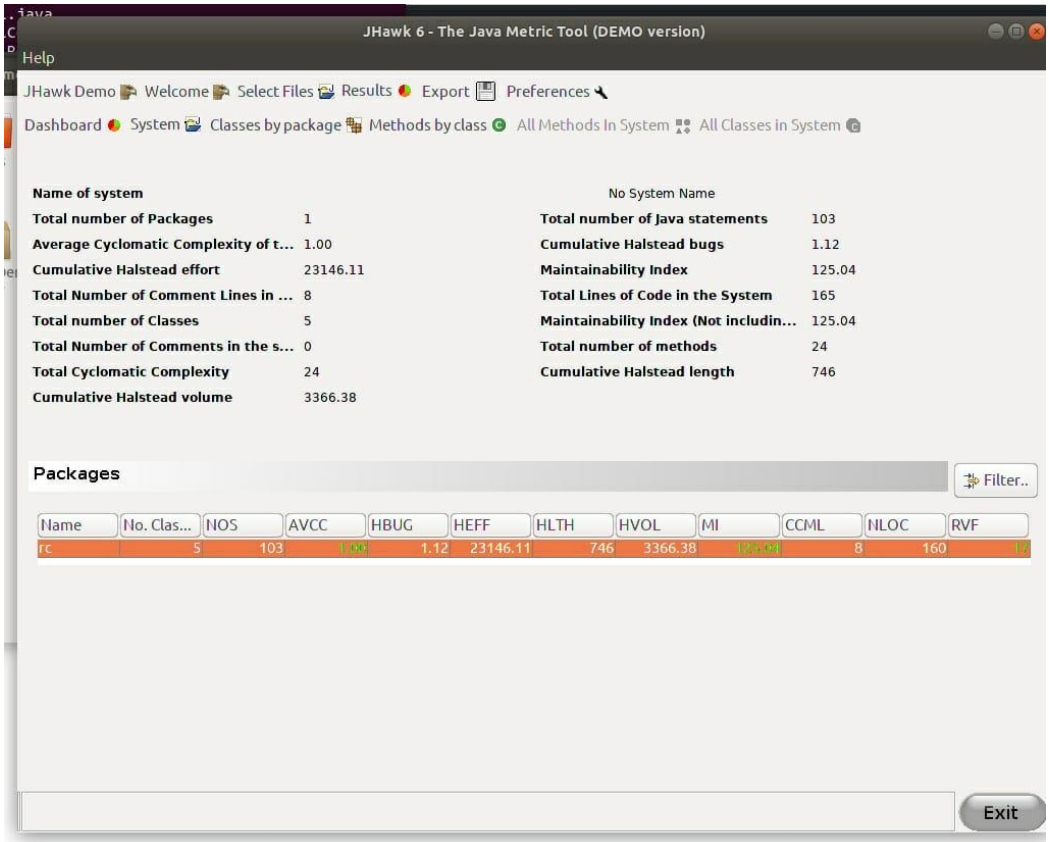


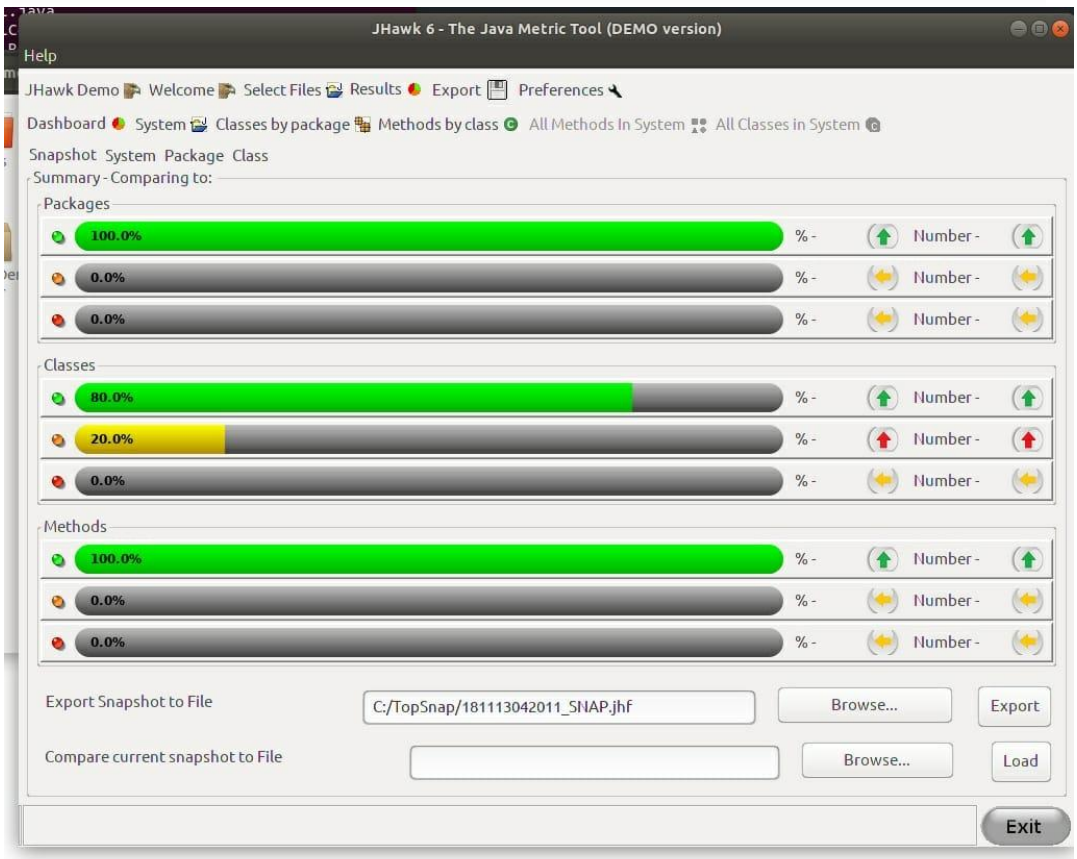



JHawk 6 - The Java Metric Tool (DEMO version)
Help
JHawk Demo Welcome Select Files Results Export Preferences
Dashboard System Classes by package Methods by class All Methods In System All Classes in System
Packages
Classes
model
repository
UserRepository
Methods
Filter..
Name COMP NOCL NOS HLTH HVOC HEFF HBUG CREF XMET LMET NLOC
findByUs... 1 0 1 15 13 259.03 0.02 2 0 0 1
repository.UserRepository
Exit

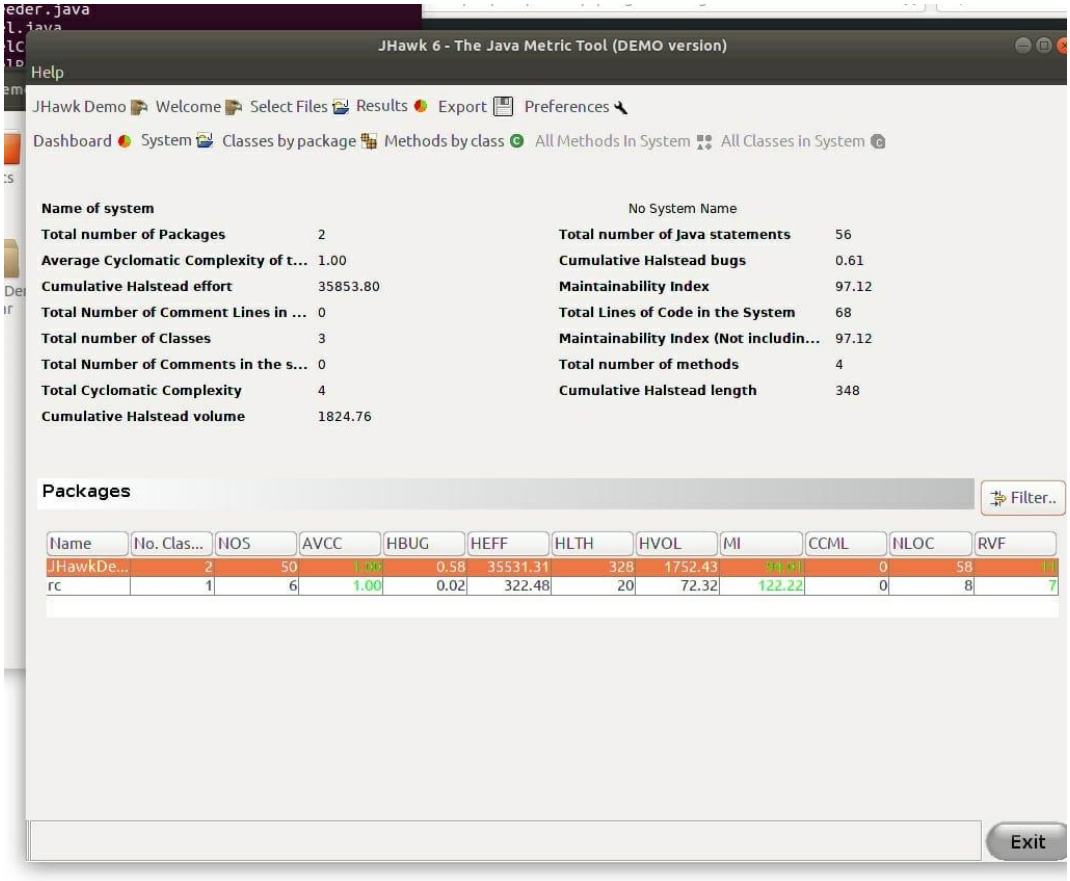

JHawk 6 - The Java Metric Tool (DEMO version)
Help
JHawk Demo Welcome Select Files Results Export Preferences
Dashboard System Classes by package Methods by class All Methods In System All Classes in System
Snapshot System Package Class
Summary - Comparing to:
Packages
50.0% % - Number -
50.0% % - Number -
0.0% % - Number -
Classes
50.0% % - Number -
50.0% % - Number -
0.0% % - Number -
Methods
100.0% % - Number -
0.0% % - Number -
0.0% % - Number -
Export Snapshot to File
C:/TopSnap/181113042011_SNAP.jhf
Browse...
Export
Compare current snapshot to File
Browse...
Load
Exit

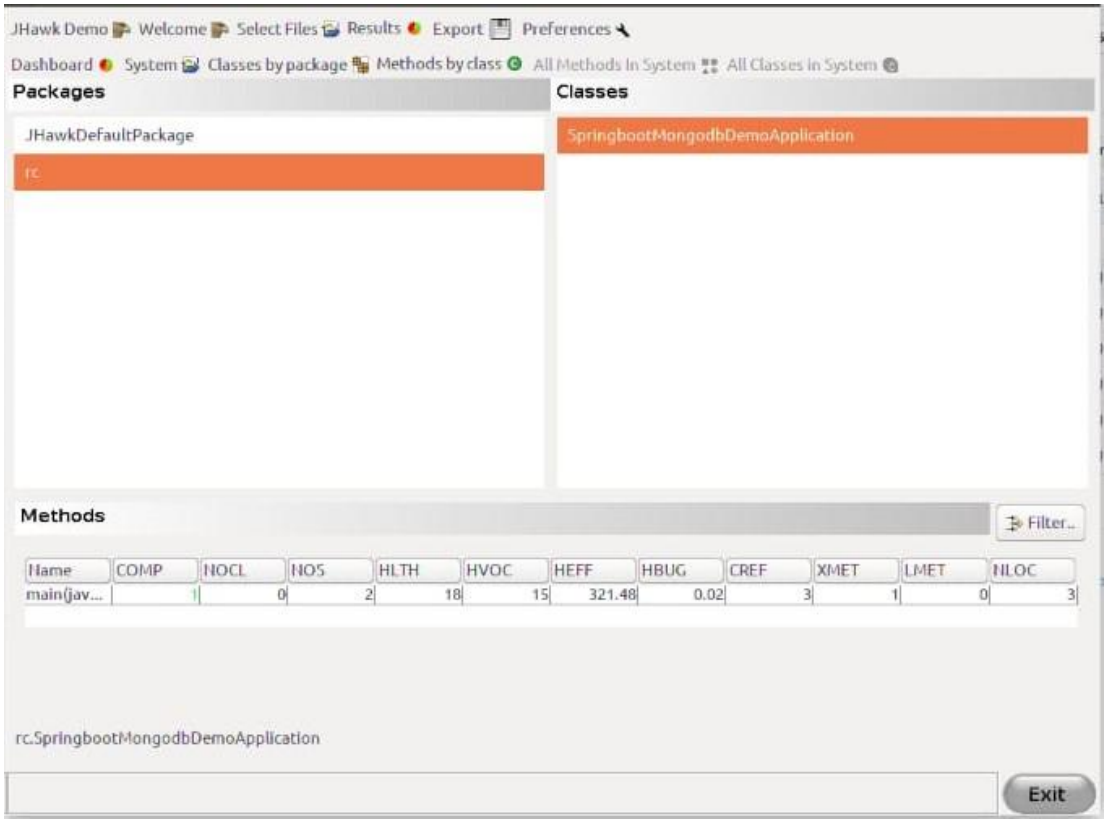
3. Admin Portal

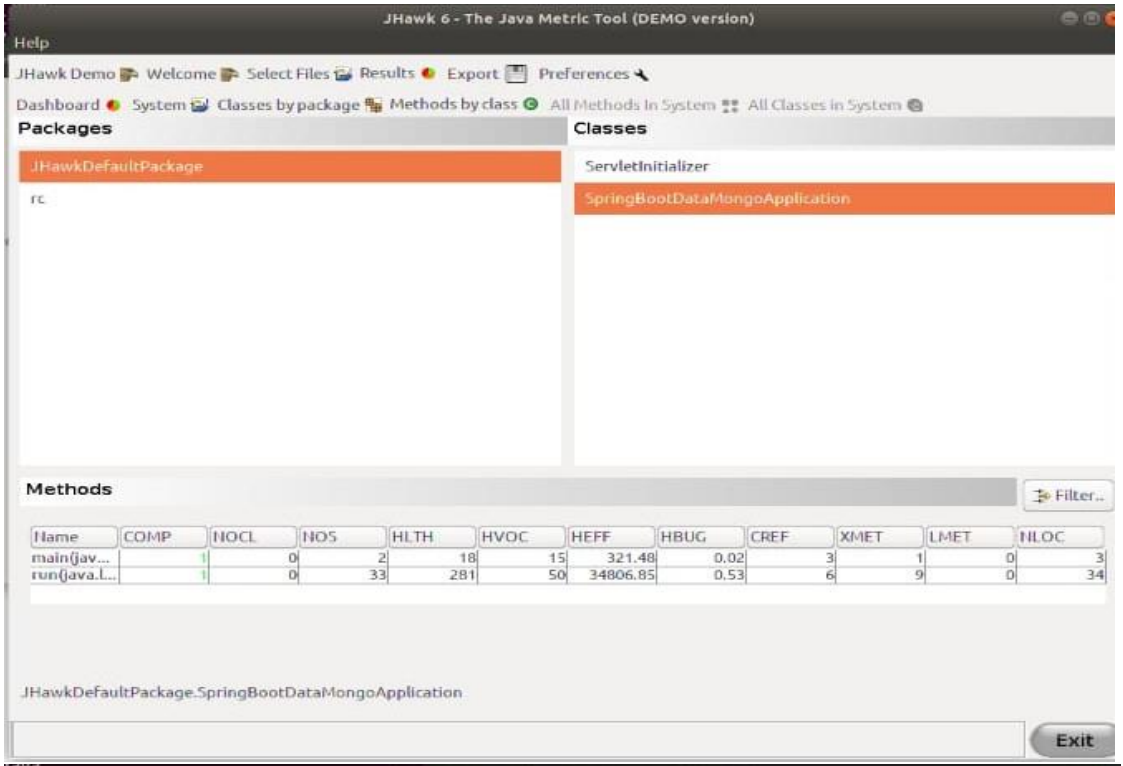


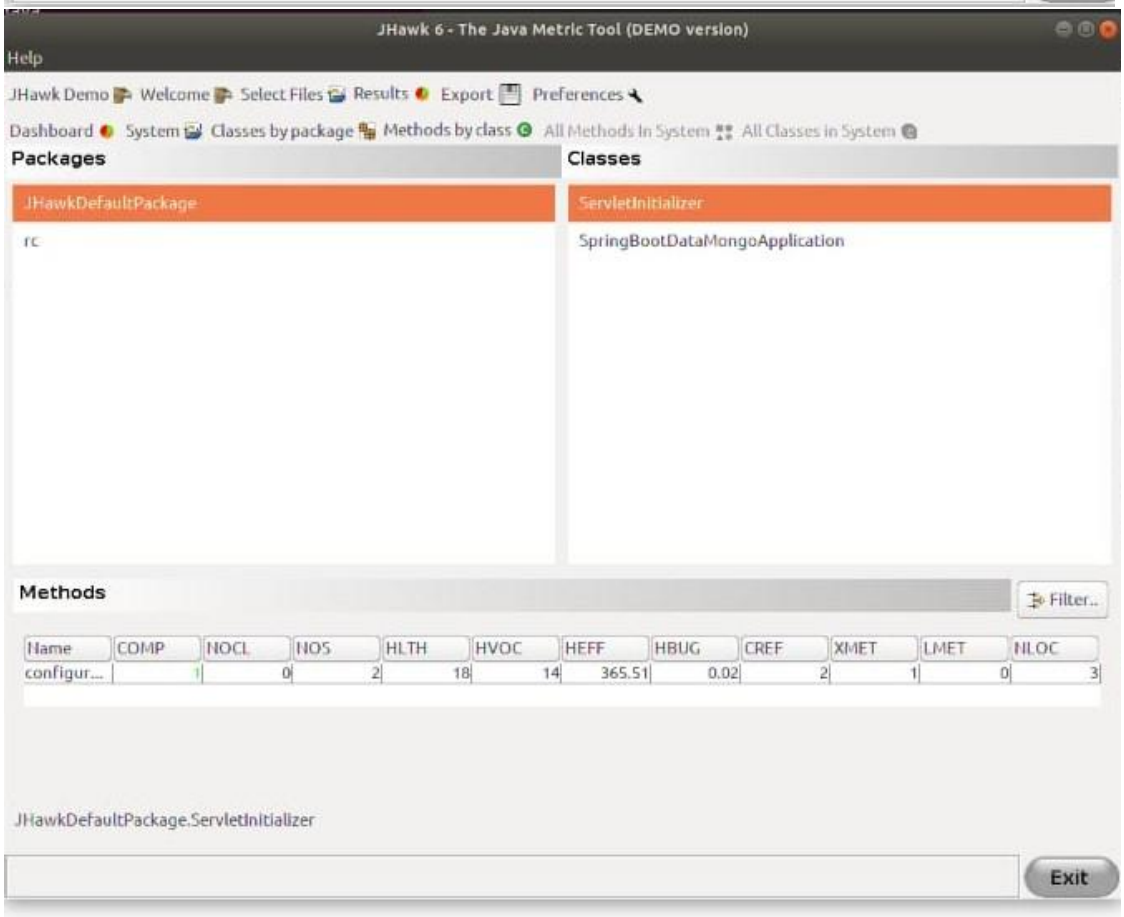


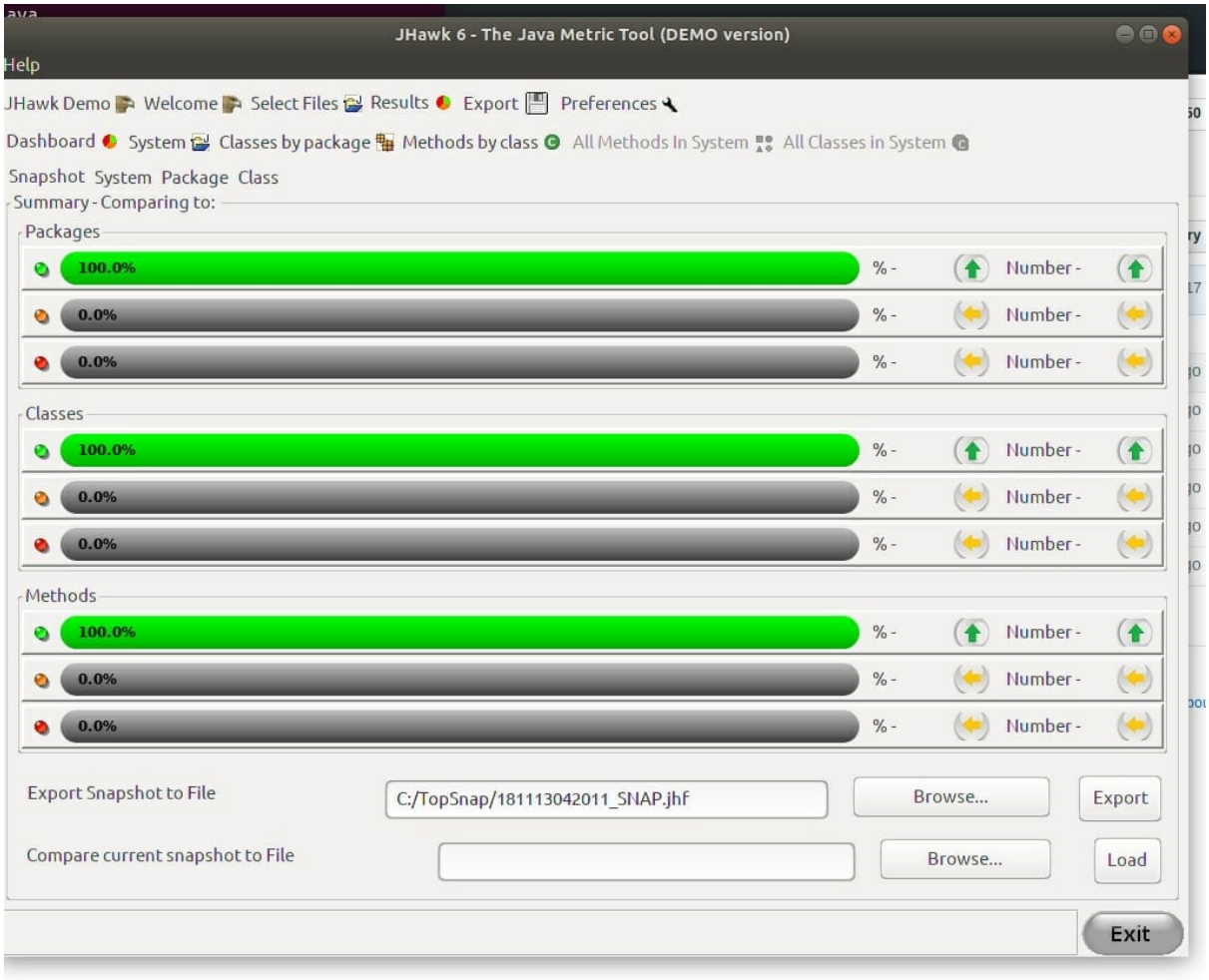
4. Spring Boot






JHawk 6 - The Java Metric Tool (DEMO version)
Help
JHawk Demo Welcome Select Files Results Export Preferences
Dashboard System Classes by package Methods by class All Methods In System All Classes in System
Packages
Classes
JHawkDefaultPackage
rc
ServletInitializer
SpringBootDataMongoApplication
Methods
Filter..
Name COMP NOCL NOS HLTH HVOC HEFF HBUG CREF XMET LMET NLOC
main(jav... 1 0 2 18 15 321.48 0.02 3 1 0 3
run(java.l... 1 0 33 281 50 34806.85 0.53 6 9 0 34
JHawkDefaultPackage.SpringBootDataMongoApplication
Exit


JHawk 6 - The Java Metric Tool (DEMO version)
Help
JHawk Demo Welcome Select Files Results Export Preferences
Dashboard System Classes by package Methods by class All Methods In System All Classes in System
Packages
Classes
JHawkDefaultPackage
rc
ServletInitializer
SpringBootDataMongoApplication
Methods
Filter..
Name COMP NOCL NOS HLTH HVOC HEFF HBUG CREF XMET LMET NLOC
configur... 1 0 2 18 14 365.51 0.02 2 1 0 3
JHawkDefaultPackage.ServletInitializer
Exit


JHawk 6 - The Java Metric Tool (DEMO version)
Help
JHawk Demo
Welcome
Select Files
Results
Export
Preferences
Dashboard
System
Classes by package
Methods by class
All Methods In System
All Classes in System
Snapshot System Package Class
Summary - Comparing to:
Packages
100.0%
0.0%
0.0%
% -
Number -
Classes
100.0%
0.0%
0.0%
% -
Number -
Methods
100.0%
0.0%
0.0%
% -
Number -
Export Snapshot to File
C:/TopSnap/181113042011_SNAP.jhf
Browse...
Export
Compare current snapshot to File
Browse...
Load
Exit

CHAPTER 4

Minutes of Meeting Document

Customer Meeting 1

17th August 2018
1:30 PM
Gurgaon

Meeting called by:	Amandeep Kamboj - Customer	**Type of meeting:**	Informal
Facilitator:	Amandeep Kamboj	**Note taker:**	Neeraj Pandey
Timekeeper:	Ashwin Jawahar		
Attendees:	Neeraj Pandey, Ashwin Jawahar, Hrithik Raj, Gayathri, Arnav Ajay		
Please read:	N/A		
Please bring:	N/A		

75 Minutes

Agenda item:	Understanding the aim of the project	**Presenter:**	Amandeep Kamboj

Discussion:

Mr. Amandeep Kamboj visualizes the project and the requirements for it. His project requirement is a Mock Examination Portal for the SSC examinations that could be incorporated in the Government Website in the future. His focus is on a website with great efficiency and functionality. The website is required to show the results, rank as well as the weak areas of the student taking the test so that he/she can improve on the same. He guides us to work on Angular CLI, Bootstrap and REST API as these are technologies and software that take website designing to a whole new level. He gives us a week or a week and a half to complete learning Angular and how it works.

Conclusions:

Action items	**Person responsible**	**Deadline**
• Learn Angular CLI	All Team Members	28th August 2018
• SRS Document	All Team Members	22nd August 2018
• Design Website	Arnav Ajay	24th August 2018

Customer Meeting 2

16th September 2018
10:30 AM
Gurgaon

Meeting called by: Amandeep Kamboj - Customer
Type of meeting: Informal

Facilitator: Amandeep Kamboj
Note taker: Gayathri

Timekeeper: Arnav Ajay

Attendees: Neeraj Pandey, Ashwin Jawahar, Hrithik Raj, Gayathri, Arnav Ajay

Please read: N/A

Please bring: N/A

60 Minutes

Agenda item: Discussing requirements in detail
Presenter: Amandeep Kamboj

Discussion:

This meeting is dedicated to understanding the requirements in detail. Firstly, Mr. Amandeep Kamboj has brought upon a slight in change in the project, wherein he requires us to make a Mock Examination Portal but needs it to be generic. It can be a mock exam site for any kind of examination taken by students and corporate employees as well. He follows up on our knowledge about Angular. He makes minor changes after checking up on our design of the website. He is open to all discussions regarding the problems faced in working with Angular and helps us accordingly. Shows us the efficiency that can be attained whilst using bootstrap to design the website.

Conclusions:

Action items	Person responsible	Deadline
• Implement Angular in Project	All Team Members	N/A
• Control Flow	All Team Members	25th Sept 2018
• Make a list of all difficulties found in Angular	All Team Members	N/A

Customer Meeting 3

7th October 2018
11:45 AM
Gurgaon

Meeting called by:	Amandeep Kamboj - Customer	**Type of meeting:**	Informal
Facilitator:	Amandeep Kamboj	**Note taker:**	Ashwin Jawahar
Timekeeper:	Hrithik Raj		
Attendees:	Neeraj Pandey, Ashwin Jawahar, Hrithik Raj, Gayathri, Arnav Ajay		
Please read:	N/A		
Please bring:	N/A		

30 Minutes

Agenda item:	Backend Working of Project	**Presenter:**	Amandeep Kamboj

Discussion:

This meeting is focused towards working of the backend which is a relatively small part of the project. Mr. Amandeep wants us to work with MongoDB for database connectivity. MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata. He explains to us the implementation of REST API in detail. He later on goes on to solve problems faced while making the project.

Conclusions:

Action items	**Person responsible**	**Deadline**
• Design Document	All Team Members	20th October 2018
• Get familiarized with REST API	All Team Members	15th October 2018
• Learn and implement MongoDB	All Team Members	2nd Nov 2018

CHAPTER 5

Nature of the Customer

NATURE OF THE CUSTOMER

The initial customer, Mr. Amandeep Kamboj, is an existing employee at Airtel. He is based in Gurgaon. He had been wanting to work on a mock examination portal for a particular exam but offered us an opportunity to work on the same but as a generic product for any exam that might require practice sets online.

It is a free mock examination management system which is not only intended for students but is also open to the corporate world. The administrators for the given examination can upload question sets which would be available for the candidates preparing for the same. The system can be modified as per the users' needs and requirements which will be based off the intended examination.

CHAPTER 6

Tools and Technologies

TOOLS AND TECHNOLOGIES

1. Software Requirement Specification
   - Argo UML
   - Lucid Chart

2. Software Design Document
   - Argo UML
   - Lucid Chart

3. Software Product Development
   - Angular CLI
   - Bootstrap
   - Java
   - Spring Boot
   - Visual Studio IDE
   - MySQL Workbench

4. Software Coding Metrics
   - JHawk Metrics Tool
   - Dependency Tester

CHAPTER 7

Novelty of Project Idea

Novelty of the Project

The project, Online Mock Examination Portal is designed to provide a website with better and easier functionality that is efficient to use for users as well as administrators included. It is aimed to be more effective as compared to already existing free online portals for certain exams. The product is generic and can be used for various exam preparations for students as well as corporate employees.

This project is a more efficient and user-friendly replacement for the existing unpaid examination practice portals on the internet. Since, the websites that are free of cost usually lack in functionality, this project aims to provide better functionality at no cost for the numerous students who prepare for various examinations every year. It is accessible via the Internet, 24 hours a day, 7 days a week.

The functionalities also include realization of weak areas in the given examination and displays the rank as opposed where other exam-takers stand which might help abundantly in the improvement of the student preparing to appear for the examination.

Along with the mentioned functionalities, the scope of this project is very broad as compared to manually taken examinations:

- This portal is not only limited to educational institutes but also to the corporate world
- There is no restriction as to the presence of an examiner during the period of the exam
- Less time consumption, thereby increasing the efficiency

CHAPTER 8

Sophistication Value of Project

Sophistication Value of the Project

This project is designed to be a more efficient and user-friendly replacement for the existing unpaid examination practice portals on the internet. Since, the websites that are free of cost usually lack in functionality, this project aims to provide better functionality at no cost for the numerous candidates who prepare for various examinations every year.

The portal has added functionalities like displaying the weak areas of each candidate appearing for the test and calculating the rank of the candidate as compared to the other candidates appearing for the same. This helps the user improve his/her abilities and helps focus on areas that might require more effort from the candidate's side.

CHAPTER 9

Applicability of Project

Applicability of the Project

The project has a very vast scope in future. The project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. Efficiency can be further enhanced and boosted up to a great extent by normalizing and de-normalizing the database tables used in the project as well as taking the kind of the alternative set of data structures and advanced calculation algorithms available. We can in future generalize the application from its current customized status wherein other vendors developing and working on similar applications can utilize this software and make changes to it according to their business needs.

The purpose of an online examination portal is to practice for tests in an efficient manner with no time wastage in checking the solutions. The main usability is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves a lot of time but also gives us an accurate result.

CHAPTER 10

Feedback of Customer

- How easy was it to install our software?

  ◉ Extremely easy ○ Very easy ○ Moderately easy ○ Slightly easy ○ Not at all easy
- How quick was the installation process for our software?

  ◉ Extremely quick ○ Very quick ○ Moderately quick ○ Slightly quick ○ Not at all quick
- How user-friendly is our software's interface?

  ◉ Extremely user-friendly ○ Very user-friendly ○ Moderately user-friendly ○ Slightly user-friendly ○ Not at all user-friendly
- How successful is our software in performing its intended task?

  ◉ Extremely successful ○ Very successful ○ Moderately successful ○ Slightly successful ○ Not at all successful

- Overall, are you satisfied with the performance of our software and our team , neither satisfied nor dissatisfied with it, or dissatisfied with it?

  ○ Extremely satisfied ◉ Moderately satisfied ○ Slightly satisfied ○ Neither satisfied nor dissatisfied ○ Slightly dissatisfied ○ Moderately dissatisfied ○ Extremely dissatisfied
- How likely are you to recommend our software and the team to others?

  ○ Extremely likely ◉ Very likely ○ Moderately likely ○ Slightly likely ○ Not at all likely

  - How approachable our team was

  ○ Extremely approachable ◉ Very approachable ○ moderately approachable ○ very difficult to meet ○ don't respond on time

  How do you rate the technical competence of the team

  ◉ Extremely Talented ○ Moderately Talented ○ Not talented at all

  Would you like to continue with our team in future

  ◉ Would love to work ○ Will at least give a thought ○ Will not consider at all

The project was done as a course project of CS 301. Would you like to participate in selecting the team from pool of students next time for another product?

◉ Would love to work ○ Will at least give a thought ○ Will not consider at all

How much business time is saved by the tool/product developed by our team compared to your traditional way of doing the business(e.g. without this product/tool/software)

○ 0-10 % ○ 10-30% ◉ 30-60% ○ 60-80% ○ more than 80%

Do you think the product is helpful to increase the business turnover

○ Quite sure ◉ It is likely to do so ○ Don't know/can't say ○ It was waste of time. No improvement at all.

How much percentage of your expectations /requirements are satisfied

○ 100% ◉ 70-90 % ○ 40-60% ○ below 40%

If you were the course supervisor how would you evaluate the project?

○ A+ ◉ A ○ B ○ B+ ○ C+ ○ C ○ D+ ○ D ○ Below D ○

How would you rate the sincerity of the team

◉ extremely sincere ○ Good level of sincerity is there ○ Moderate level of sincerity ○ Not sincere at all

Any other suggestion/feedback/comment