

# Software Requirements Specification

Version 0.3

9<sup>th</sup> March 2016

## Objective Exam Management

Kushagra Surana

Shiv Mohan

Rohil Surana

Shubham Singh

Submitted in partial fulfilment  
Of the requirements of  
CS 223 Software Engineering

Mentor: Mr. Rimpesh Katiyar

This work is based upon the submissions of the course Software Engineering (CS223). The students who submitted this team projects were Kushagra Surana, Shiv Mohan, Rohil Surana and Shubham Singh.

## Revision History

Version 0.1	Dated: 17 Jan 2016	<ul style="list-style-type: none"><li>• Added purpose, scope, constraints, assumptions and dependencies</li><li>• Added features Notification to students, graphical representation, Login, progress report and test creation</li></ul>
Version 0.2	Dated: 27 Jan 2016	<ul style="list-style-type: none"><li>• Added Class Diagram, sequence diagrams, activity diagrams, use case diagrams</li></ul>
Version 0.3	Dated: 9 <sup>th</sup> March 2016	<ul style="list-style-type: none"><li>• Updated assumptions, use cases.</li></ul>

# Table of Contents

[Table of Contents](#)

[List of Figures](#)

[1.0. Introduction](#)

[1.1. Purpose](#)

[1.2. Scope of Project](#)

[1.3 Constraints](#)

[1.4 Assumptions and Dependencies](#)

[1.3. Glossary](#)

[1.4. References](#)

[1.5. Overview of Document](#)

[2.0. Overall Description](#)

[2.1 System Environment](#)

[2.2 Functional Requirements Specification](#)

[2.2.1 Use case 1](#)

[Use case:](#)

[2.3 User Characteristics](#)

[2.4 Non-Functional Requirements](#)

[3.0. Requirements Specification](#)

[3.1 Functional Requirements](#)

[3.1.1 Notification to students](#)

[3.1.2 Graphical progress report](#)

[3.3 Detailed Non-Functional Requirements](#)

[\*\*3.4 Logical Structure of the Data\*\*](#)

[4.0 Supporting information](#)

[4.1 Table of contents and index](#)

[4.2 Appendixes](#)

## **List of Figures**

**No table of figures entries found.**

## **1.0. Introduction**

### ***1.1. Purpose***

The purpose of this document is to explain the details of the requirements of the Objective Exam Management system to be created for the course CS223 Software Engineering. The document is meant to explain the functionalities and features of the system. It will also reveal the scope and the constraints of the system. The document will serve as an agreement between the stakeholders and the developers of the system.

### ***1.2. Scope of Project***

The Objective Exam Management system is a web based application to help the faculty and staff for proper organisation of an online objective exam. It help the faculty in creating a test paper, notifying the students about it, displaying the questions to the students at and for the time specified by the faculty, capturing the answers from students and generating reports for students as well as the faculty. At the end of each test a report will be generated and sent to the faculty assigned for the course that will display the number of students that took the test and all the statistics of the result.

### ***1.3 Constraints***

The developers have a time constraint of 1 month to complete the development of the software once this document is generated due to the length of one academic semester.

One invigilator will be required at the time of the examination at the computer centre.

As this will be a web app hence the students, faculties and the staff are required to have a browser installed on their terminals. It is recommended to have Chrome 32+ or Firefox 25+. The server side requirements for this system are all supposed to be cross platform.

#### **1.4 Assumptions and Dependencies**

- The staff (IT department) of an institute is equipped with a server either at the site or offsite. They will input all the information at the time of installation example name of institute,
- The institute has a Computer Centre where students will arrive to give the tests. The computers should be equipped with modern browsers.
- The paper is supposed to be only objective. The faculty will not have an option to add any subjective question to the system.
- The computer centre has the capacity to accommodate all the students of a course and there will be no connectivity issues and nor there will be any power cuts.

We plan to use the Material Design specifications from Google for the frontend of the system and Django framework for the backend.

#### **1.3. Glossary**

<b>Term</b>	<b>Definition</b>
system	The Objective exam management system that is to be developed
test	An online objective examination
faculty	The authority conducting the objective examination
staff	The IT support department of an institute equipped with the editing access to the database and the portal backend.
dashboard	The starting page after a person is logged in
Stakeholders	People that will be using the system and will be affected by it

notification	Email notification sent to student faculty etc.
Progress report	A report that will tell the observer a summary of the test of a student or a group of students for a particular test.
captcha	A program or system intended to distinguish human from machine input.
Computer center	A place where numerous computer systems are available to be used by students

#### **1.4. References**

IEEE. - *IEEE Std 830-1998 IEEE Recommended Practice for Software*

*Requirements Specifications*. IEEE Computer Society, 1998.

Web - [www.cse.msu.edu/~chengb/RE-491/Papers/SRSEExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSEExample-webapp.doc)

#### **1.5. Overview of Document**

The rest of the document is designed in the following way:

Chapter 2 Overall Description aims to describe informally all the requirements of this system. It will give a complete overview of the workflow of the system to the stakeholders and the developers and will help the developers to understand the contents of the Chapter 3.

Chapter 3 Requirements Specification will include complete and detailed technical details for the developers to understand the functionalities and the feature of the system.

## 2.0. Overall Description

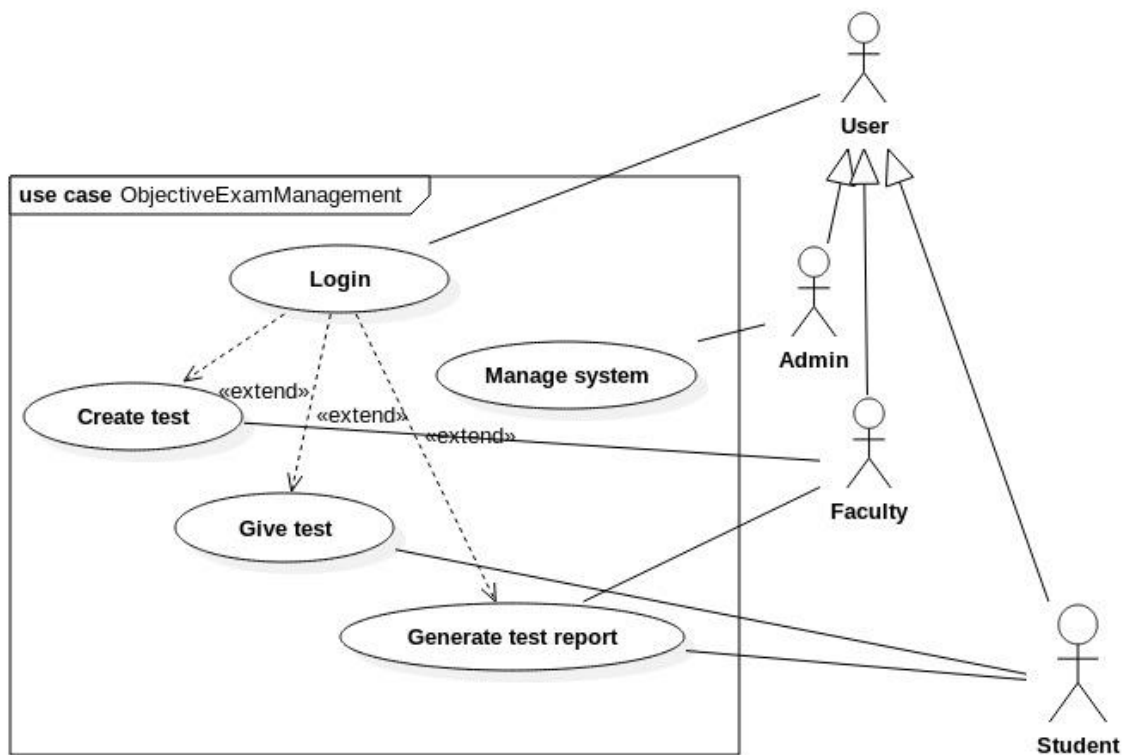
### 2.1 System Environment

### 2.2 Functional Requirements Specification

#### 2.2.1 Use case 1

Use case: **Objective Exam portal**

**Diagram:**



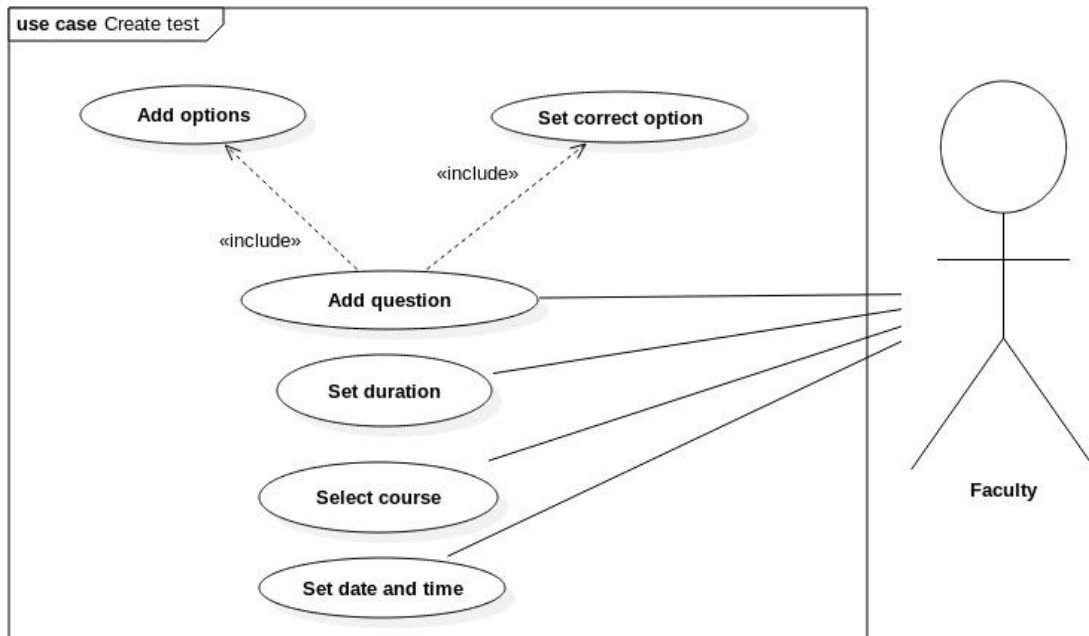
**Brief Description**

**Initial Step-By-Step Description**

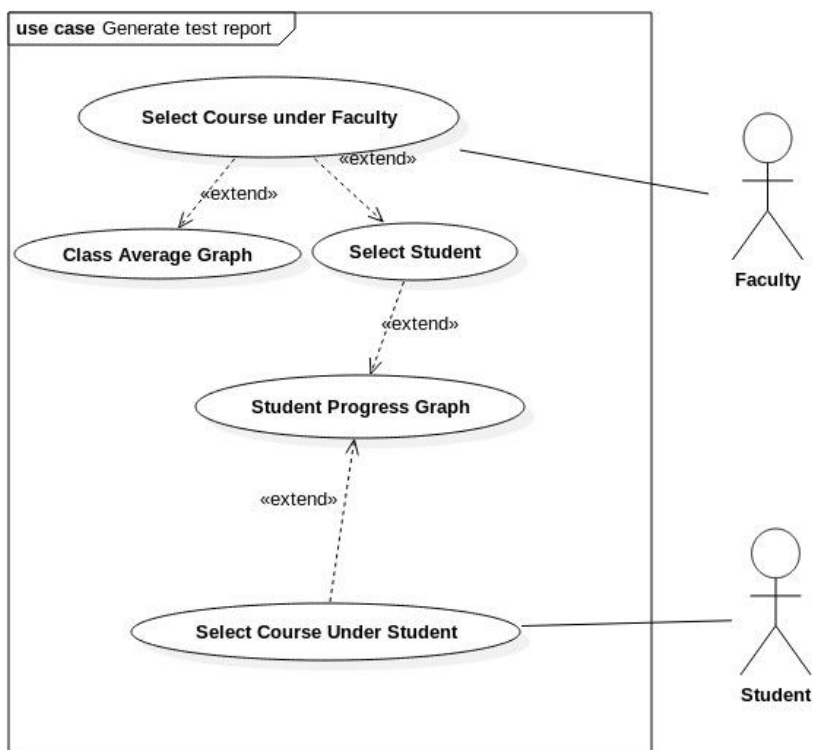
Use case: **Create Test**

**Diagram:**

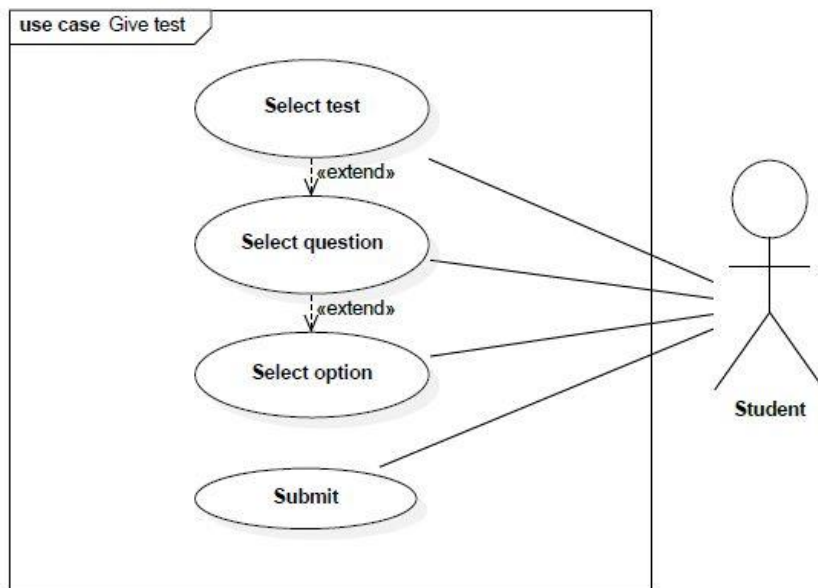




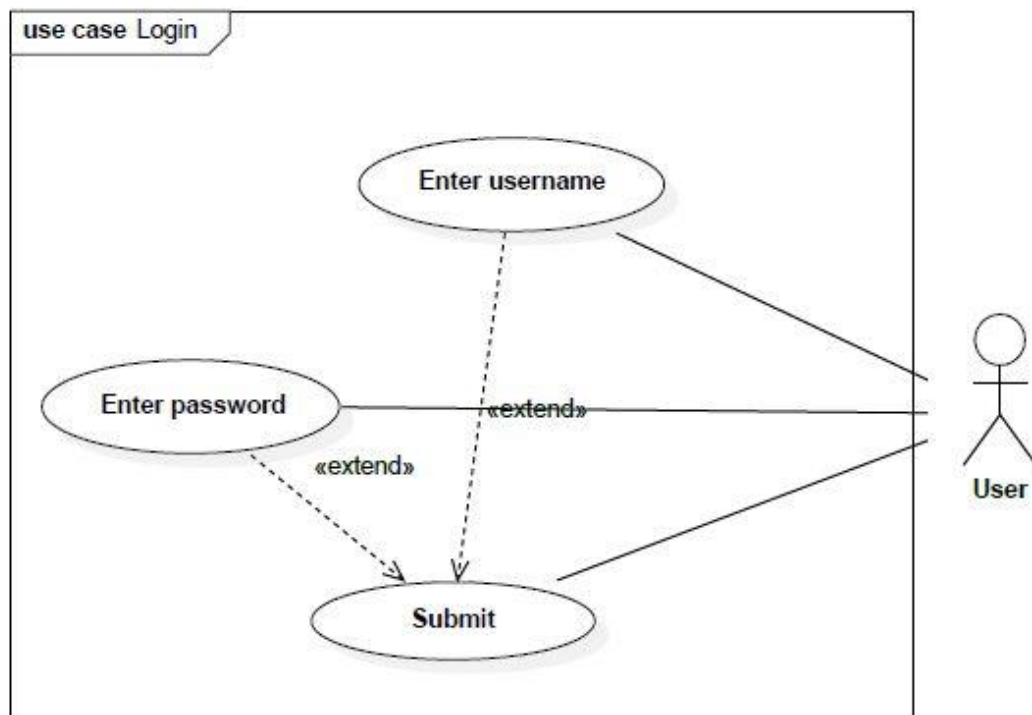
**Use case: Generate Test report**  
**Diagram:**



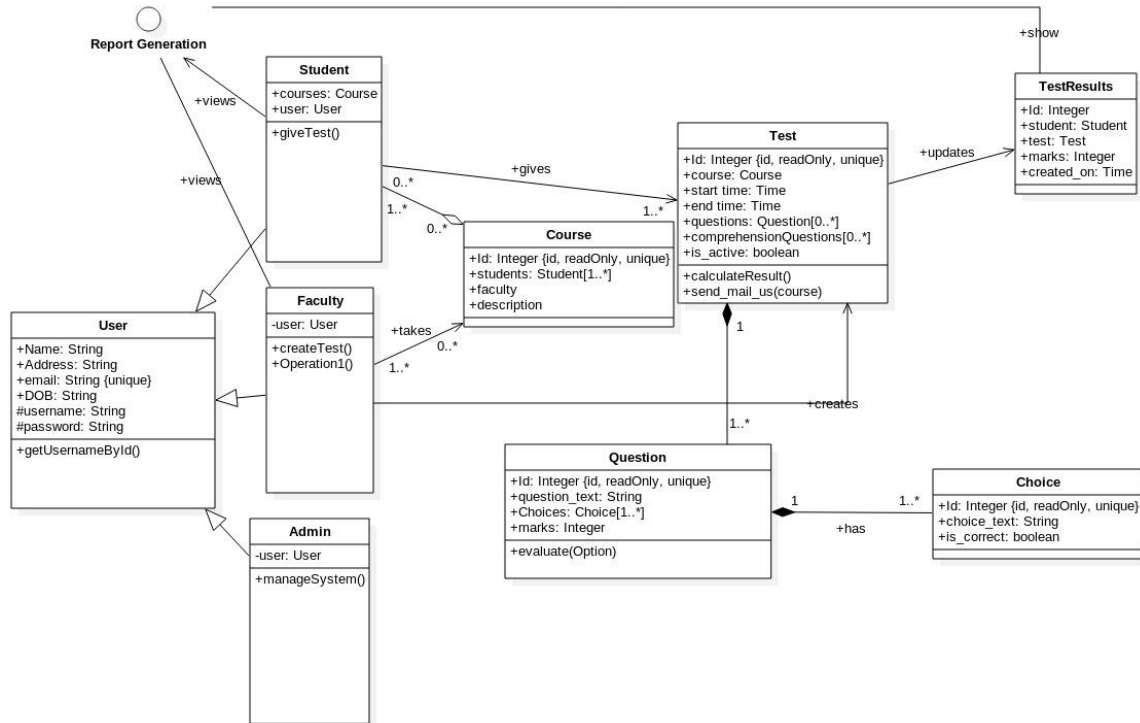
**Use case: Give Test**  
**Diagram:**



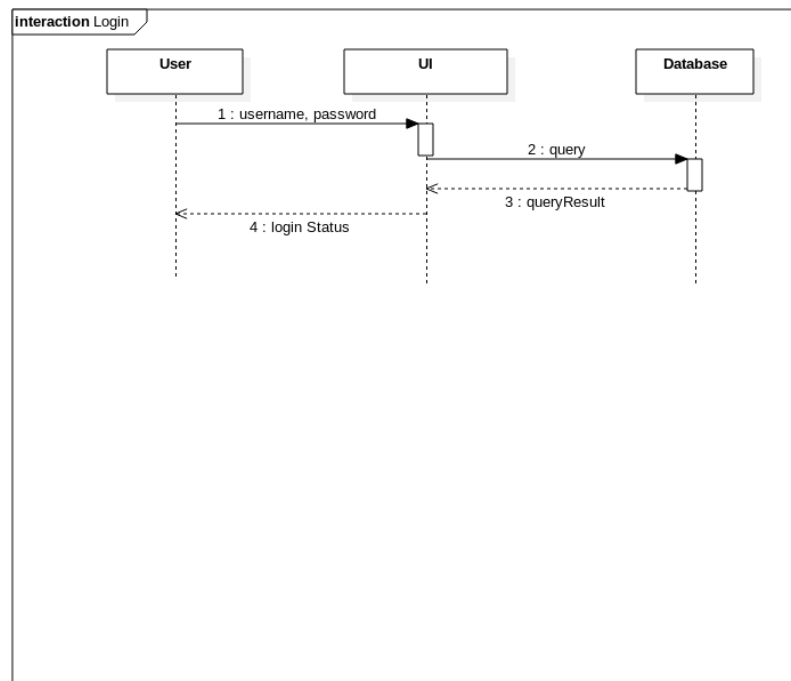
Use case: **Login**  
Diagram:



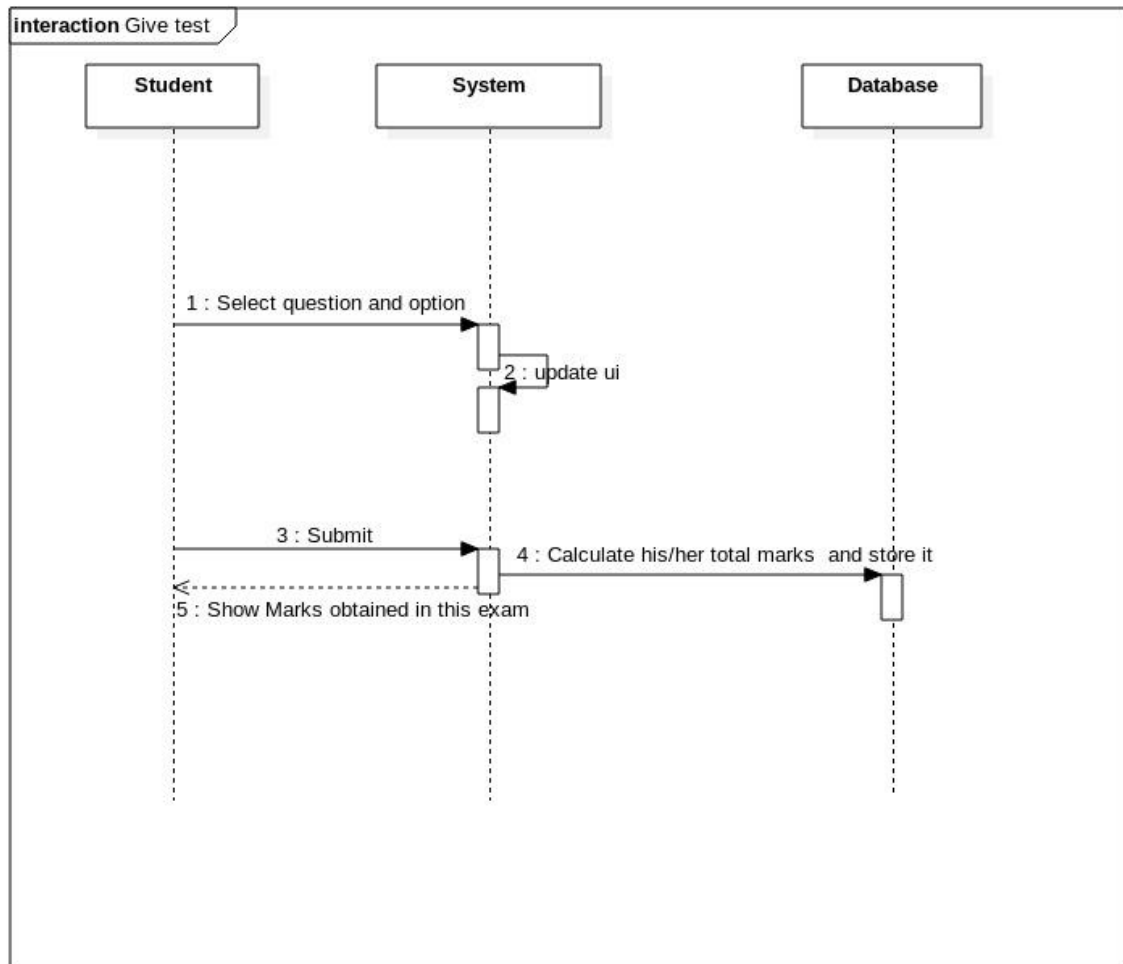
## Class Diagram:

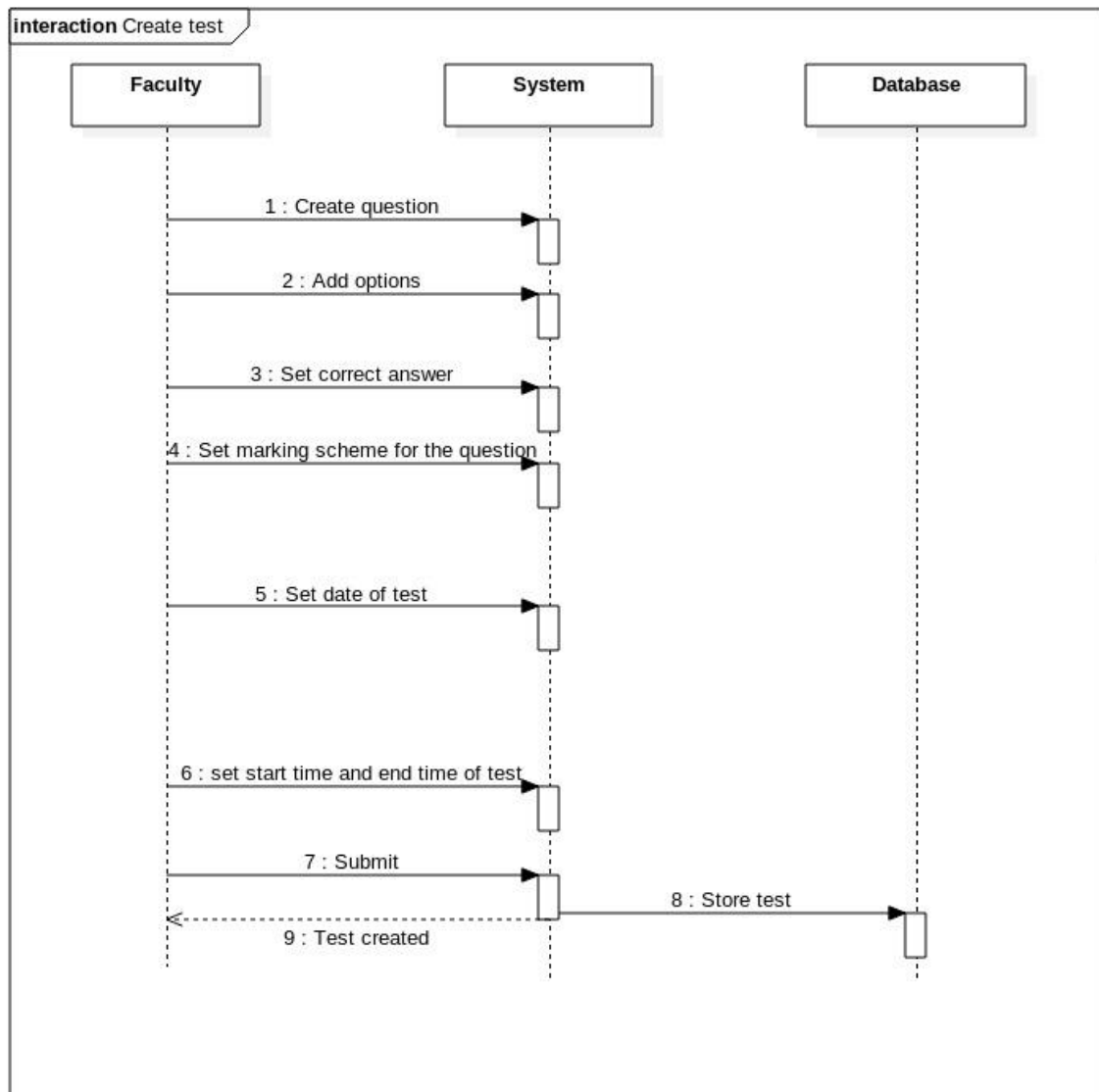


## Sequence Diagram:



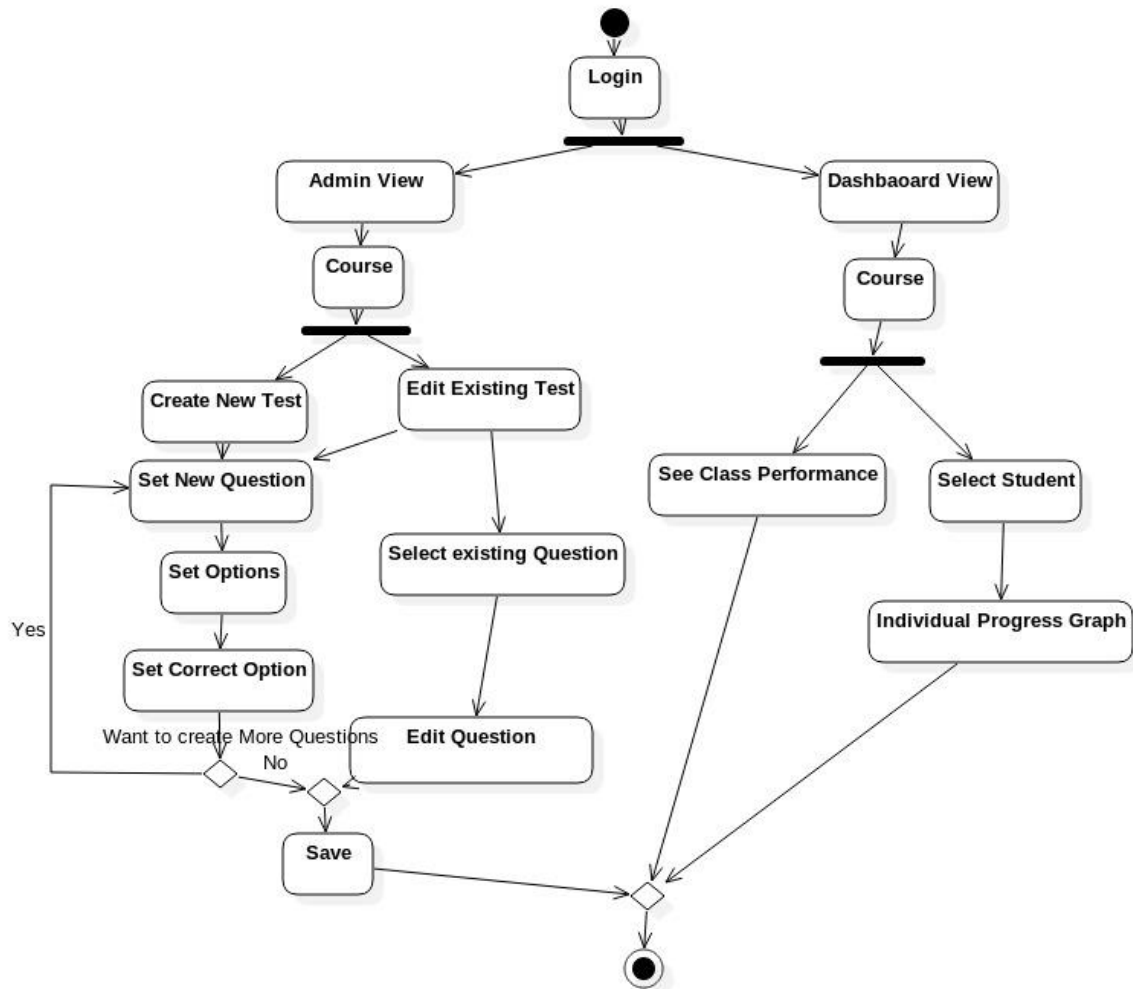
Login



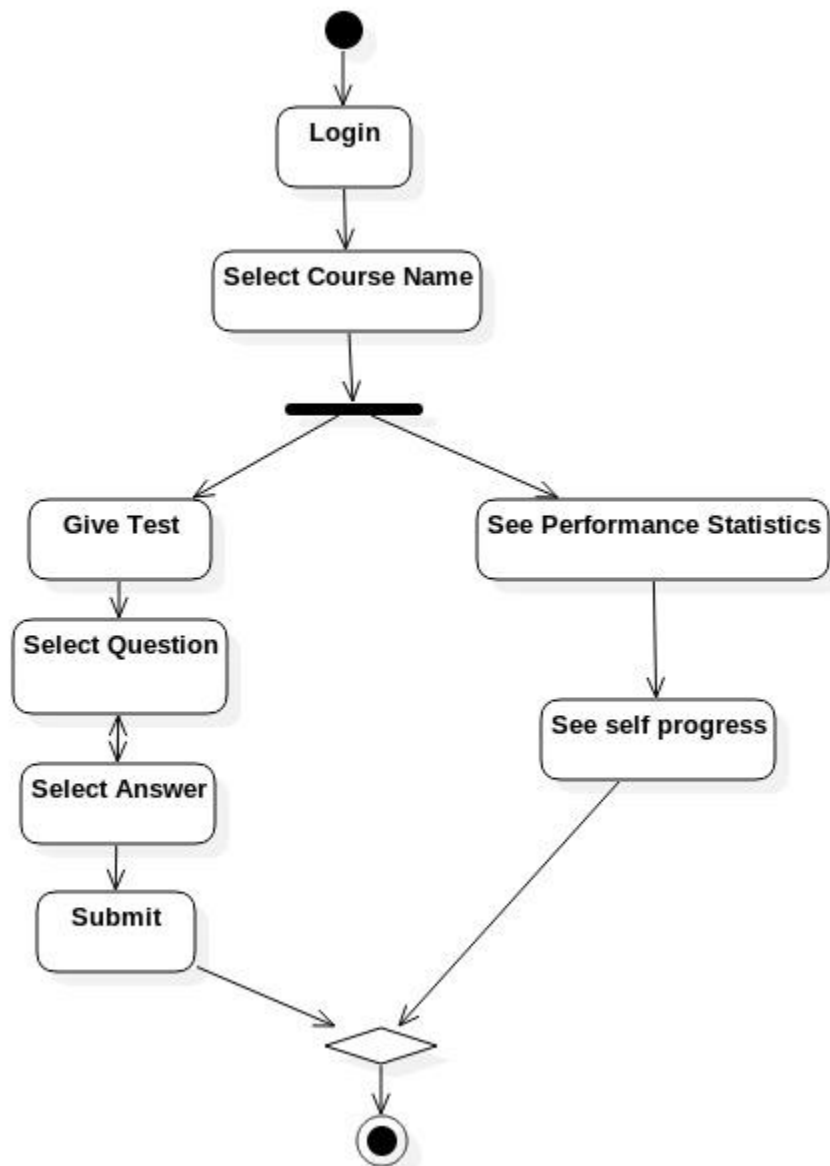


### Activity Diagram:

Faculty:



## Student



### 2.3 User Characteristics

The user of this software system requires the following skills to use this software

## **2.4    *Non-Functional Requirements***



### 3.0. Requirements Specification

#### 3.1 Functional Requirements

##### 3.1.1 Notification to students

<b>Use Case Name</b>	Notification to students about test
<b>Trigger</b>	Faculty approval for test notification
<b>Precondition</b>	1. Test questions already prepared 2. The date, time and duration of test are set
<b>Basic Path</b>	1. Test prepared by faculty 2. Test details are filled 3. Approval from faculty for notification 4. E-mail to students
<b>Alternative Paths</b>	Faculty has the option to send a notification to students registered to a course even if there is no test created.
<b>Post condition</b>	Students are provided test details prior to
<b>Exception Paths</b>	Incomplete and insufficient detailed test are not allowed to be sent as notification.
<b>Other</b>	

##### 3.1.2 Graphical progress report (Class average graph)

<b>Use Case Name</b>	Graphical chart generation
<b>Trigger</b>	Faculty selects the course
<b>Precondition</b>	1. Course exists.
<b>Basic Path</b>	1. Admin creates course 2. Faculty creates 0 or more tests. 3. Faculty selects the course.
<b>Alternative Paths</b>	
<b>Post condition</b>	Faculty have a report of the class's performance
<b>Exception Paths</b>	Absentee, incomplete test or failure to submit will result in 0 score of student for the test
<b>Other</b>	It will show the average marks of class for all the tests in a course.

##### 3.1.3 Graphical progress report (individual student)

<b>Use Case Name</b>	Graphical generation of students report
<b>Trigger</b>	Request from faculty or student
<b>Precondition</b>	2. Course exists and student exists.
<b>Basic Path</b>	1. Student selects the course. 2. Faculty selects the course and then selects the

	. student of that course.
<b>Alternative Paths</b>	
<b>Post condition</b>	Faculty or student have an updated report of the course
<b>Exception Paths</b>	Absentee, incomplete test or failure to submit will result in 0 score of student for the test
<b>Other</b>	It will show the score of student in all the tests of a course.

### 3.1.3 Test Creation

<b>Use Case Name</b>	Creation of test by the faculty
<b>Trigger</b>	Faculty request for test creation
<b>Precondition</b>	<ol style="list-style-type: none"> <li>1. Faculty must be already logged in</li> <li>2. Faculty must have at least one course registered under him</li> </ol>
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Faculty logs in and opens its admin view</li> <li>2. Requests for creating test</li> <li>3. Adds questions selects date, time and duration.</li> <li>4. Completion of test creation</li> </ol>
<b>Alternative Paths</b>	Drafts can also be saved that can be later edited and completed by the faculty.
<b>Post condition</b>	Test Created and request sent for notification to students. A preview for the test can also be generated for the faculty to view.
<b>Exception Paths</b>	Incomplete tests are not allowed to be created
<b>Other</b>	

### 3.1.4 User creation

<b>Use Case Name</b>	Creation of a user by admin
<b>Trigger</b>	Admin request for admin/student/faculty creation
<b>Precondition</b>	1. Admin must be logged in.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. Admin requests for admin/student/faculty creation.</li> <li>2. Admin fill the details and press 'save' button.</li> </ol>
<b>Alternative Paths</b>	
<b>Post condition</b>	A new admin/student/faculty is created.
<b>Exception Paths</b>	Incomplete user are not allowed to create.
<b>Other</b>	Admin can't create a test. To create a test faculty must be created.

### 3.1.4 Login

<b>Use Case Name</b>	Login
<b>Trigger</b>	If a person is not logged in and opens the application
<b>Precondition</b>	1.User is not logged in
<b>Basic Path</b>	1.User enters his user id and password 2.User presses log in button 3.User is redirected to his dashboard
<b>Alternative Paths</b>	
<b>Post condition</b>	User can perform all his available functions.
<b>Exception Paths</b>	In case the login fails at first the user will be prompted and will be given another chance to login
<b>Other</b>	

### 3.1.6 Test

<b>Use Case Name</b>	Test
<b>Trigger</b>	Student selects test.
<b>Precondition</b>	1.Student is logged in at the dashboard. 2.Student presses the appropriate test link 3.Test has been started or just started.
<b>Basic Path</b>	1.Test page will be opened. 2.Student will go through the complete test process 3. Student will submit the test in time.
<b>Alternative Paths</b>	
<b>Post condition</b>	Answers are stored at the backend and result is computed for each student.
<b>Exception Paths</b>	In case the student fails to submit the paper in time all his answers will be recorded and sent to the backend.
<b>Other</b>	A record will also be stored if the student failed to submit questions on time.

### 3.3 Detailed Non-Functional Requirements

### 3.4 Logical Structure of the Data

Database Structures:

#### 1. Course

Name	Faculty	Description

#### 2. Test

Name	Course	Start_time	End_time	Is_active

#### 3. TestResult

Test	Student	Marks	Created_on

#### 4. Question

Question_text	Marks	Test

#### 5. Choice

Question	Choice_text	Is_correct

#### 6. User

Name	Address	Date_of_birth

#### 7. Student(is\_staff = True)

User	Courses

#### 8. Faculty(is\_staff = True)

User

#### 9. Admin(is\_superuser = True)

User

## **4.0 Supporting information**

### ***4.1 Table of contents and index***

### ***4.2 Appendixes***