Online Examination Management System

Software Requirements Specification

19th February, 2019

Made by:-

Tarun Pratap Singh 16IT143

Akshay Pandita 16IT151

Nimish Mangal 16IT233

Lead Software Engineer

Prepared for: -

Software Engineering IT350

Under the guidance of Ms. Raksha Ma’am

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 21/02/19 | 1 | Tarun, Akshay, Nimish |  |
|  |  |  |  |
|  |  |  |  |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Tarun, Akshay, Nimish | Group members |  |
|  |  |  |  |

**Table of Contents**

**REVISION HISTORY II**

**DOCUMENT APPROVAL II**

**1. INTRODUCTION 1**

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

**2. GENERAL DESCRIPTION 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

**3. SPECIFIC REQUIREMENTS 2**

3.1 External Interface Requirements 3

*3.1.1 User Interfaces 3*

*3.1.2 Hardware Interfaces 3*

*3.1.3 Software Interfaces 3*

*3.1.4 Communications Interfaces 3*

3.2 Functional Requirements 3

*3.2.1 <Functional Requirement or Feature #1> 3*

*3.2.2 <Functional Requirement or Feature #2> 3*

3.3 Use Cases 3

*3.3.1 Use Case #1 3*

*3.3.2 Use Case #2 3*

3.4 Classes / Objects 3

*3.4.1 <Class / Object #1> 3*

*3.4.2 <Class / Object #2> 3*

3.5 Non-Functional Requirements 4

*3.5.1 Performance 4*

*3.5.2 Reliability 4*

*3.5.3 Availability 4*

*3.5.4 Security 4*

*3.5.5 Maintainability 4*

*3.5.6 Portability 4*

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

**4. ANALYSIS MODELS 4**

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

**5. CHANGE MANAGEMENT PROCESS 5**

**A. APPENDICES 5**

A.1 Appendix 1 5

A.2 Appendix 2 5

**1. INRODUCTION**

**1.1 Purpose**

*- This Web Application provides facility to conduct online*

*examination world wide*

*- It saves time as it allows number of students to give the exam at a time*

*and displays the results as the test gets over, so no need to wait for the*

*result. wait for the result. It is automatically generated by the server.*

*- Administrator has a privilege to create, modify and delete the test papers and its*

*particular questions.*

*- User can register, login and give the test with his specific id, and can see the*

*results as well.*

**1.2 Scope**

- *this system allows to student to give their exam at any palce.it s save paper and give result very fast.*

**1.3 Overview**

***-*** *this system provides an easy solution to the student to give exam online.*

## **1.4 References**

*This subsection should:*

*(1)* *The Complete Reference ----------------PatrikNaughton, Herbert Schildt*

*(2) Website: -* [*www.tutorialspoint.com/java/*](http://www.tutorialspoint.com/java/)

*(3)* *Java Server Pages---------------O’Reilly*

# **2. General Description**

*This Web Application provides facility to conduct online examination worldwide. It saves time as it allows number of students to give the exam at a time and displays the results as the test gets over, so no need to wait for the result. It is automatically generated by the server.*

*Administrator has a privilege to create, modify and delete the test papers and its particular questions. User can register, login and give the test with his specific id, and can see the results as well.*

## **2.1 Product Perspective**

*This project aims to eliminate the inconsistencies experienced when giving internship/placement tests in NITK. The platform will ensure that the UX is same for every single test, which will make it easier for users to attain their objectives.*

*This also ensures that poorly functioning platforms do not hamper the recruitment process.*

## **2.2 Product Functions**

*The software will enable users to write the tests as well as companies to monitor the test.*

*All the features and sub-features are a subset of aforementioned*

## **2.3 User Characteristics**

*This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).*

## **2.4 General Constraints**

*This subsection of the SRS should provide a general description of any other items that will*

*limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).*

## **2.5 Assumptions and Dependencies**

*This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.*

# **3. Specific Requirements**

*This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.*

*Each requirement in this section should be:*

* *Correct*
* *Traceable (both forward and backward to prior/future artifacts)*
* *Unambiguous*
* *Verifiable (i.e., testable)*
* *Prioritized (with respect to importance and/or stability)*
* *Complete*
* *Consistent*
* *Uniquely identifiable (usually via numbering like 3.4.5.6)*

*Attention should be paid to the carefuly organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.*

## **3.1 External Interface Requirements**

### **3.1.1 User Interfaces**

### **3.1.2 Hardware Interfaces**

1. The application demands that all the PCs must be present in the internet

2. Pc should be sufficiently fast with adequate memory at least 64 MB RAM

and 2 GB hard –disk space is required to run this application.

3. Screen resolution of at least 800\*600 required to properly view the screen.

4. It should be supporting the printers

### **3.1.3 Software Interfaces**

1. Any Window Operating System.

2. The PHP must be installed. For the database handling MYSQL must be installed.

3. The final application must be packaged in a set up program, so that the product can be easily installed on the clients-machine.

### **3.1.4 Communications Interfaces**

**-** The E-mail should be sent within one hour after the registration.

- The system should support registration for examination maximum of Students.

## **3.2 Functional Requirements**

**3.1 Description**

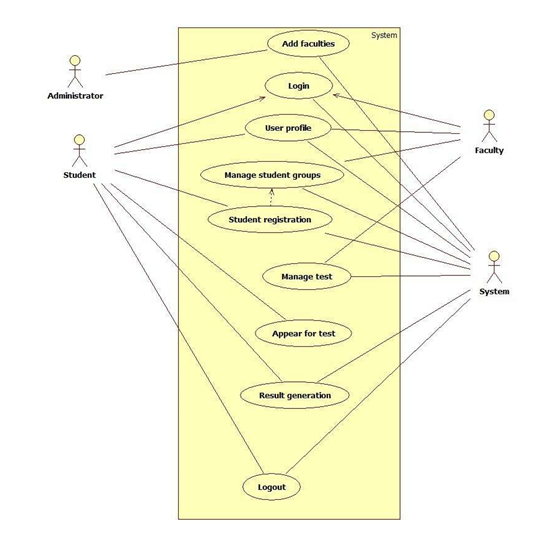
*The Online Examination System is developed for handling the activities for various users such as student, staff, exam staff. Every teacher should have laptop with wireless internet connection.*

**3.2 Technical Issues**

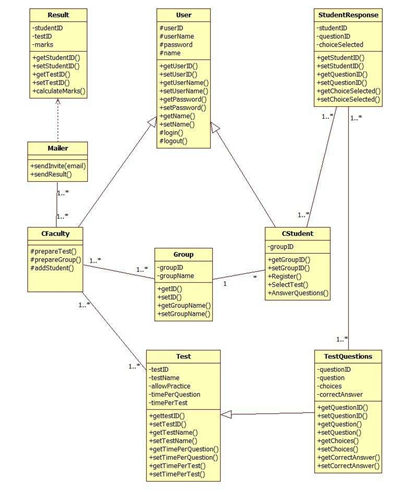
*This product will work on client-server architecture. It will be require an internet server and which will be able to run PHP application. The product should be support some commonly used browsers such as Internet Explorer, Mozilla Firefox. External interfaces include key board and mouse, enabling navigation s across the screens.*

## **3.3 Use Cases**

### **3.3.1 Use Case #1**



### **3.3.2 Use Case #2**



## **3.4 Classes / Objects**

### **3.4.1 User**

3.4.1.1 Attributes

-user ID

-username

-password

-name

3.4.1.2 Functions

-getUserID( )

-setUserID( )

-getUserName( )

-setUserName( )

-getPassword( )

-setPassword( )

-getName( )

-setName( )

-login( )

-logout( )

### **3.4.2 Student Response**

3.4.2.1 Attributes

-studentID

-questionID

-choiceSelected

3.4.2.2 Functions

-getStudentID( )

-setStudentID( )

-getQuestionID( )

-setQuestionID( )

-getChoiceSelected( )

-setChoiceSelected( )

### **3.4.3 Admin**

3.4.3.1 Attributes

-adminID( )

-adminPassword( )

3.4.3.2 Functions

-setQuestions( )

-removeQuestions( )

-setUsers( )

-getUsers( )

-getQueries( )

-deleteQueries( )

## **3.5 Non-Functional Requirements**

### **3.5.1 Performance**

The application should be efficient and should perform better in all conditions.

### **3.5.2 Reliability**

The application should be highly reliable and it should generate all the updated information in correct order.

### **3.5.3 Availability**

System will be available around the clock except for the time required for the back up of data.

### **3.5.4 Security**

Application will allow only valid users to access the system. Access to any will application resource depend upon user’s designation. There are two types of users namely Administrator and Student. Security is based upon the individual user ID and Password.

### **3.5.5 Maintainability**

The installation and operation manual of examination management system will be provided to user.

### **3.5.6 Portability**

The application should be portable on any windows based system

## **3.6 Inverse Requirements**

*State any \*useful\* inverse requirements.*

## **3.7 Design Constraints**

*Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.*

## **3.8 Logical Database Requirements**

*Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.*

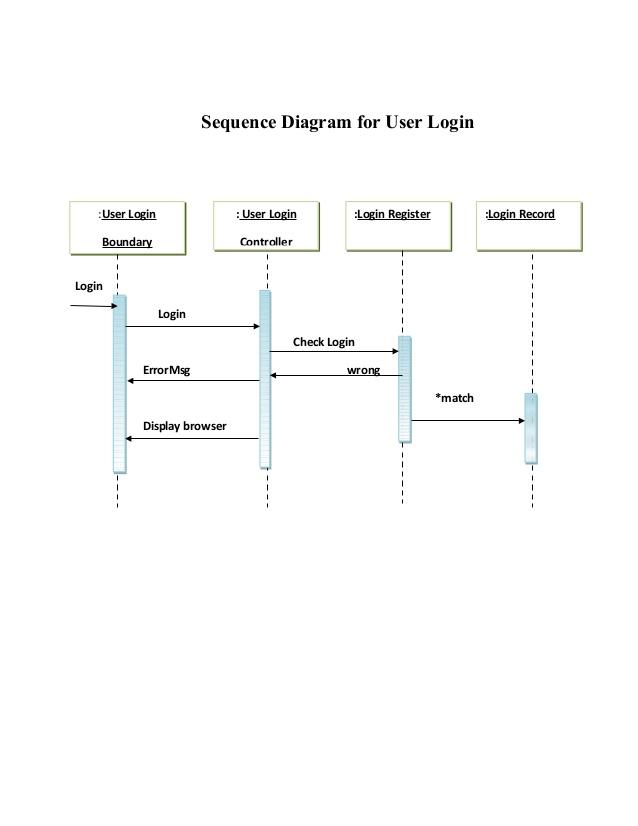
## **3.9 Operational Scenarios**

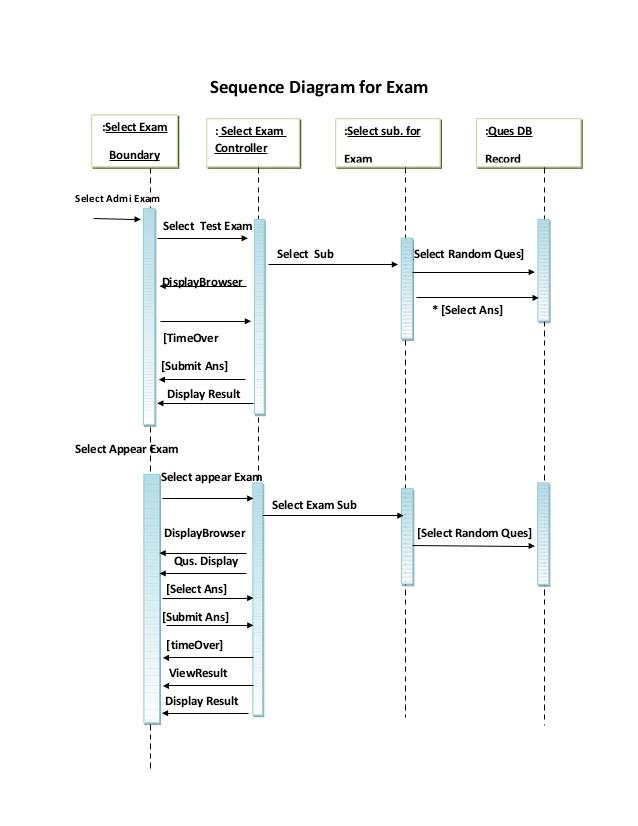
There will be student database, examination database. The student database will contain students name, e-mail address, address, phone number, qualification. The examination database contain exam date, time, exam hall ticket for student.

# **4. Analysis Models**

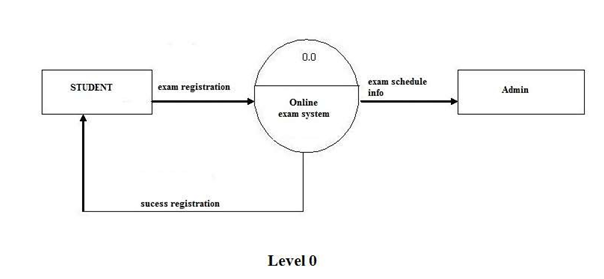
*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

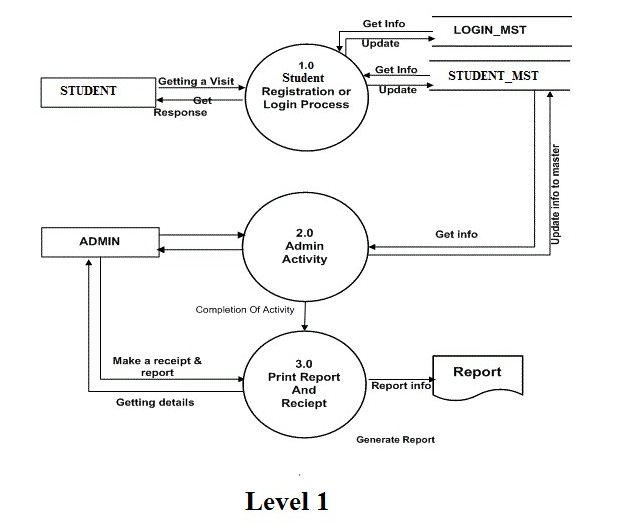
## **4.1 Sequence Diagrams**

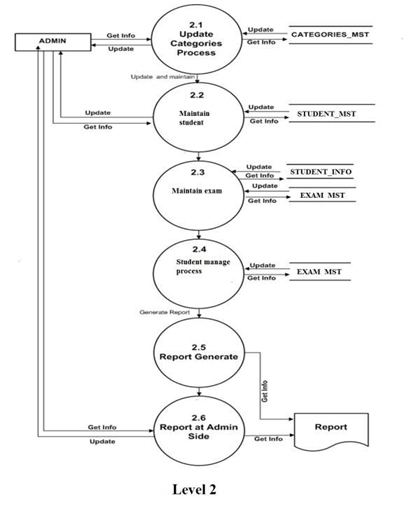




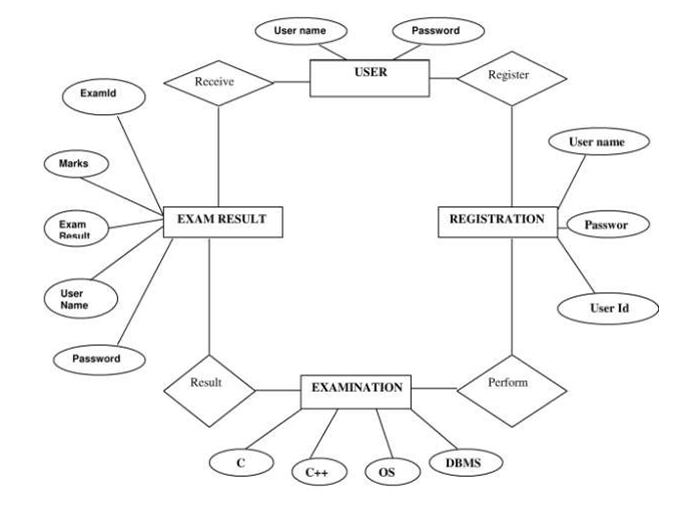
## **4.3 Data Flow Diagrams (DFD)**







## **4.2 State-Transition Diagrams (STD)**



# **5. Change Management Process**

# GitHub will be used to commit the changes whenever requirement changes and all the group members will be able to contribute in making the SRS using GitHub platform. Every contributor can work individually and change whenever requirement changes.

**6. Other Requirements**

Currently there are no other known requirements for the project. However this may change in the event of unforeseen circumstances encountered during the duration of the project.

# **A. Appendices**

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## A.1 Appendix 1

## Term Description

## 

## SRS Software Requirements Specification

## 

## IEEE Institute of Electrical and Electronics Engineers

## 

## User/Customer Person using the App.

## 

## API Application Program Interface

## 

## GUI Graphical User Interface

## 

## IDE Integrated Development Environment

## 

## Appendix 2: To Be Determined List

## ● Not yet done as the application is still in requirement phase.

## 

## Project Budget Estimation

## Number of Team Members 3

## Price (per hour) NA Number of working hours per day 2 hrs Total price (per day) NA Number of working days per week 5 days

## 

## 