Software Requirements Specification

Version 3.0

08.03.2016

Exam Management System

Submitted in partial fulfillment

Of the requirements of

CS 223 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223). The students who submitted this team projects are:

1. Dhruv Sharma (B14CS015)
2. Vijay Kumar Paliwal (B14CS040)
3. Bharat Singh (B14CS012)
4. Shubham Jain (B14CS035)

# 

# 

# 

**Document History :**

|  |  |  |
| --- | --- | --- |
| **VERSION** | **Date** | **Description of Changes** |
| **1.0** | 17-01-2016 | First draft of requirements specification including use cases descriptions and non-functional requirements. |
| **2.0** | 28-01-2016 | UML diagrams added |
| **3.0** | 08-03-2016 | Modified according to software implementation |

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **Table of Contents**

Table of Contents. i

List of Figures. ii

1.0. Introduction. 1

1.1. Purpose. 1

1.2. Scope of Project 1

1.3 Constraints. 1

1.4 Assumptions and Dependencies. 1

1.3. Glossary. 2

1.4. References. 2

1.5. Overview of Document 2

2.0. Overall Description. 3

2.1 System Environment 3

2.2 Functional Requirements Specification. 3

2.3 User Characteristics. 4

2.4 Non-Functional Requirements. 2

3.0. Requirements Specification. 3

3.1 Functional Requirements. 3

3.1.1 Verifying Time-table

3.1.2 Seating arrangement

3.1.3 Generating Attendance Sheet

3.1.4 Invigilation duty-chart generation

3.1.5 Exam guideline notification

3.1.6 Resource Allocation Management

3.1.7 Make-up exam notification

3.1.8 Make-up exam database updation

3.1.9 Mailing of Seating Plan

3.3 Detailed Non-Functional Requirements. 3

***3.4*** ***Logical Structure of the Data***. 3

4.0 Supporting information. 3

4.1 Table of contents and index. 3

4.2 Appendixes. 3

# 

# 

# **1.0. Introduction**

## **1.1. Purpose**

The purpose of this document is to define the specific requirements for the Exam Management System and to detail the specifications for the features, capabilities, critical attributes, and major characteristics of the proposed system. This document is intended to be studied by software design, development and testing team. This document does not describe about the inner soul of design of the software.

**1.2. Scope of Project**

This software will be used by an academic institute for exam management specifically mid semester exams and end semester exams. The software will take examination time-table, lists of registered students in each course and list of exam halls with their seating capacity as inputs and generate the seating arrangement for each room and exam and attendance sheets for every exam after verifying the input documents. Also, the software will generate instructors and TA’s duty chart if the list of instructors and TA’s per course is given.

## **1.3 Constraints**

The first version of this system must be available to be used by 10.04.2016. The customer can have any operating system as long as it supports Python2.7 or onwards.

## **1.4 Assumptions and Dependencies**

We are assuming that the exam time-table provided as an input to the system is in appropriate format and follow certain constraints:

1. The exam time-table will be given as an microsoft excel sheet with course no. course name and exam date for each course.
2. List of registered students for each course will be provided as separate excel file for each course.
3. List of available invigilators will be provided as a separate excel file.
4. List of available exam halls with capacity must be provided as text file.
5. The schedule must be such that total no. of students sitting for exam at a time does not exceed the total capacity of available exam halls.

## **1.3. Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| 1. Instructor     2. TA  3. Attendance Sheet  4. Seating Arrangement  5.Excel Sheet  6.Resources  7. User | Instructor is a person who instructs a course in the academic institution where this software is being used.  TA stands for teaching assistant of the respective course.  Attendance is a partially blank excel sheet which contains (in one column) the list of names and roll numbers of students to be seated in a room and blank space (in second column) for signatures of students .  Seating arrangement refers to a set of excel sheets, whose each element contains lists of students to be seated in one room in every exam time slot.  A Microsoft Excel Document.  The quantity of question papers, answer sheets and other resources (e.g. - graph papers etc.) required per hall.  Staff of office of academics will be the user who will be coordinating the exams. |

## **1.4. References**

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## **1.5. Overview of Document**

The rest of the document contains three sections.

Section 2 describes overall description of system requirement, functional requirements and non-functional requirements.

Section 3 describes detailed specification of functional and non-functional requirements.

Section 4 describes table of contents and index and appendixes.

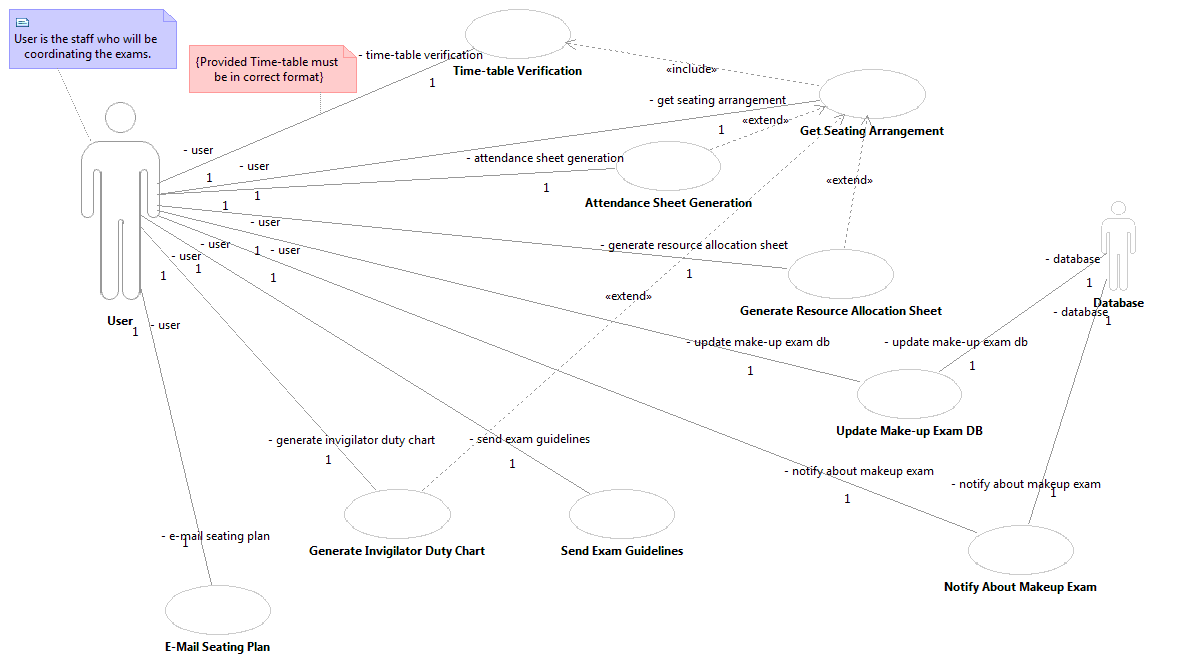
**2.0. Overall Description**

## **2.1 System Environment**

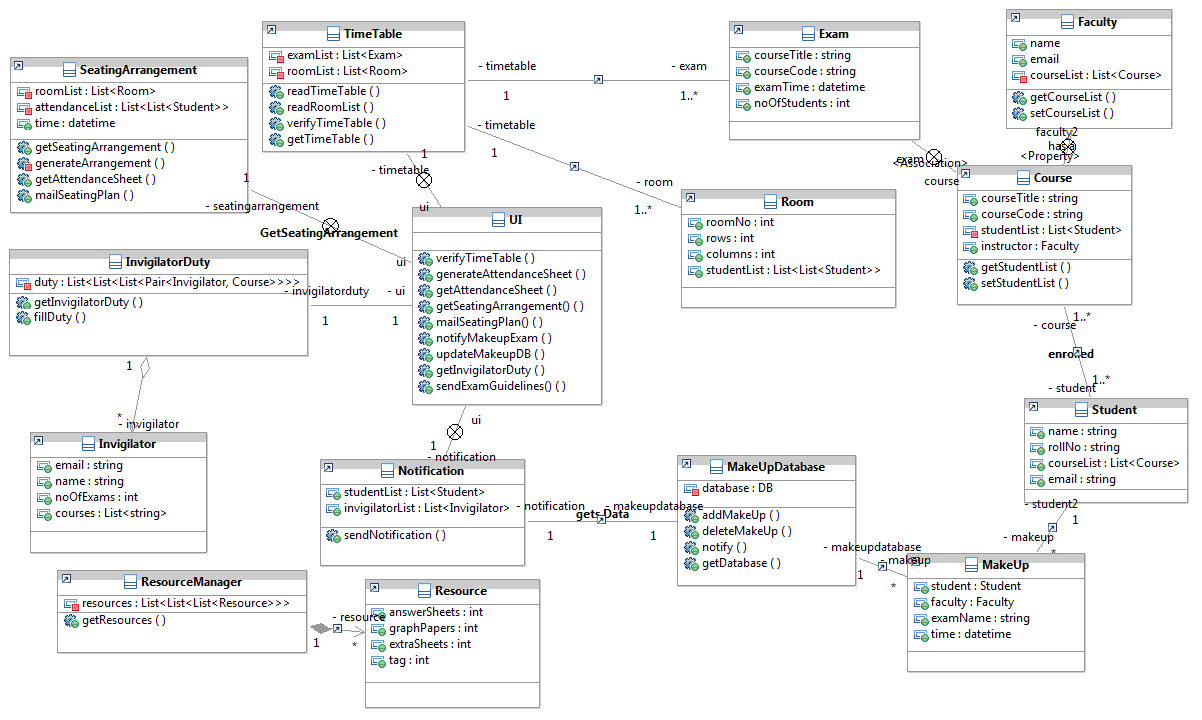
Currently system is developed for Ubuntu 14.04 os.

## **2.2 Functional Requirements Specification**

#### **2.2.1 Use case diagram**

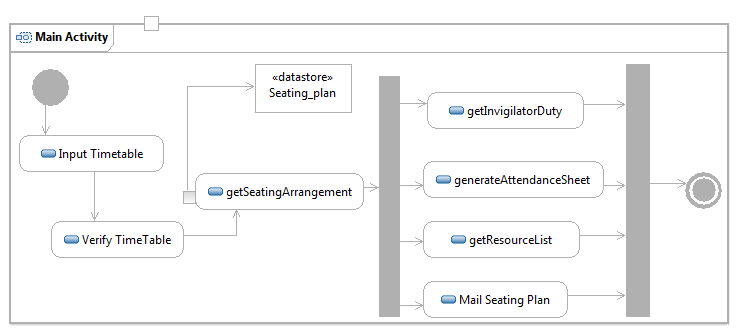


**2.2.2 Class Diagram:**

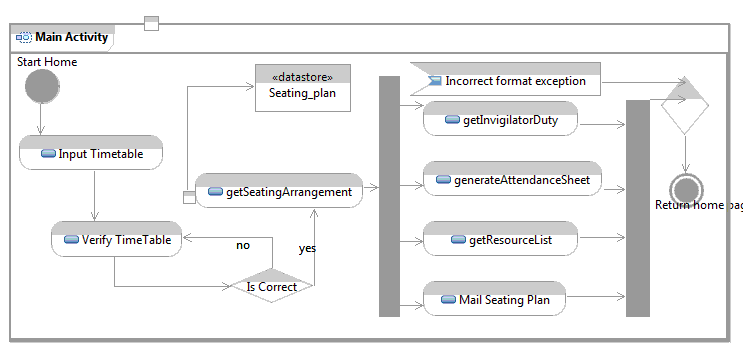


**2.2.3 Activity Diagram**

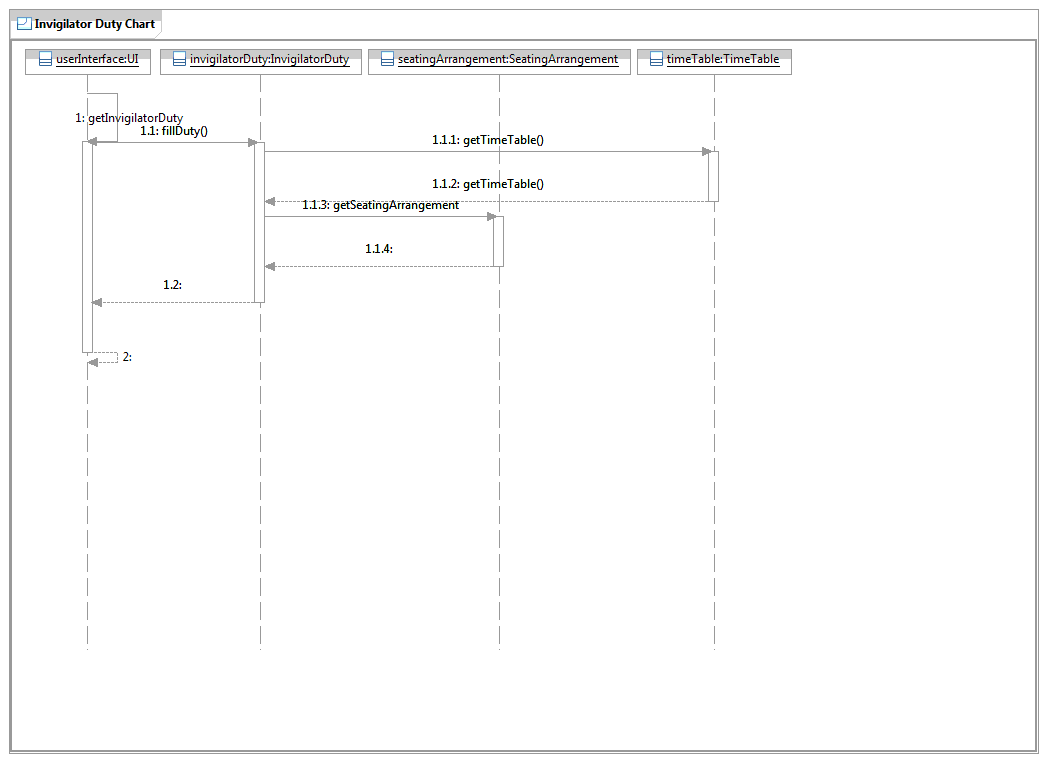
**2.2.3.1 Main Activity:**

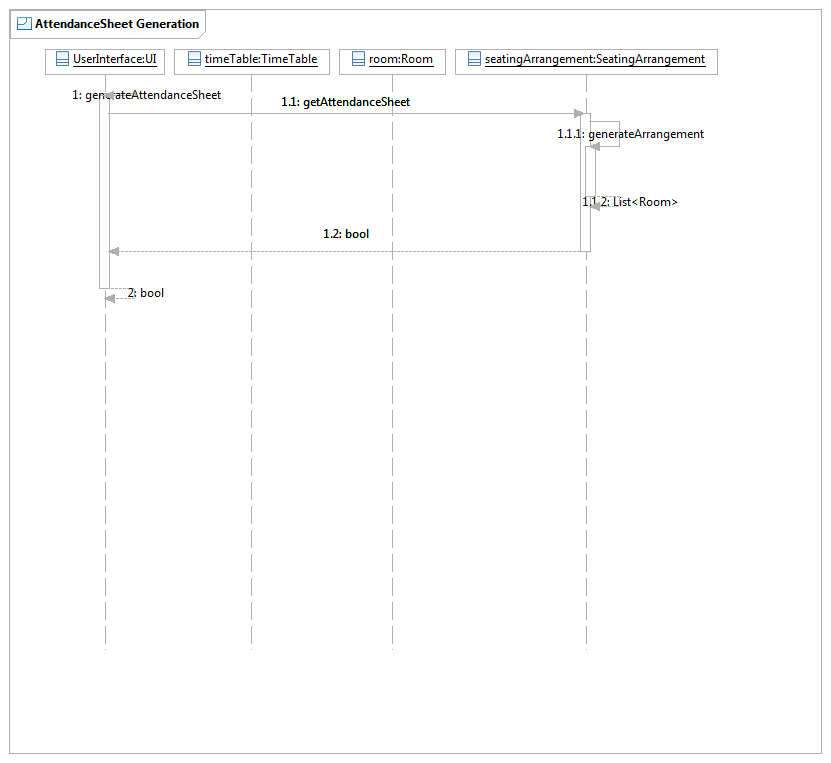


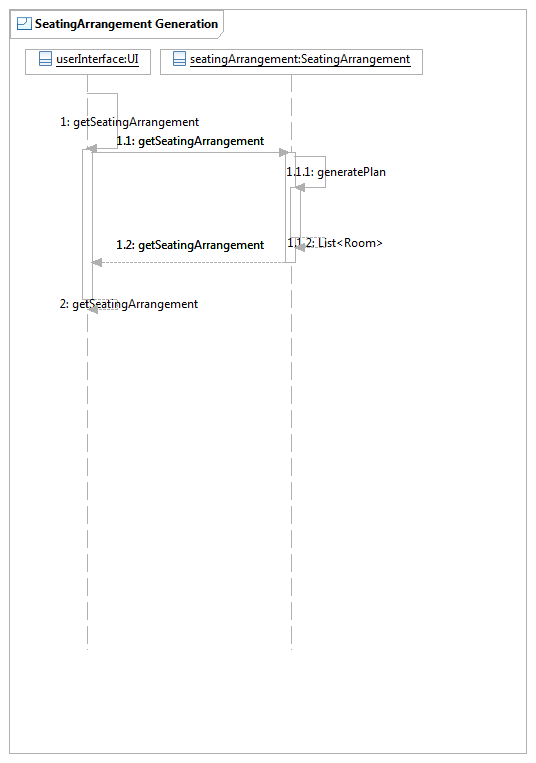
**2.2.3.2 MakeUp Exam Management**



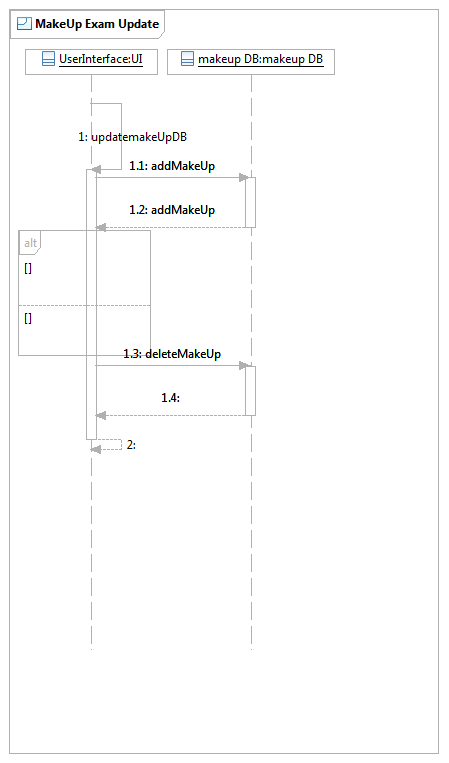
**2.2.4 Sequence Diagram:**

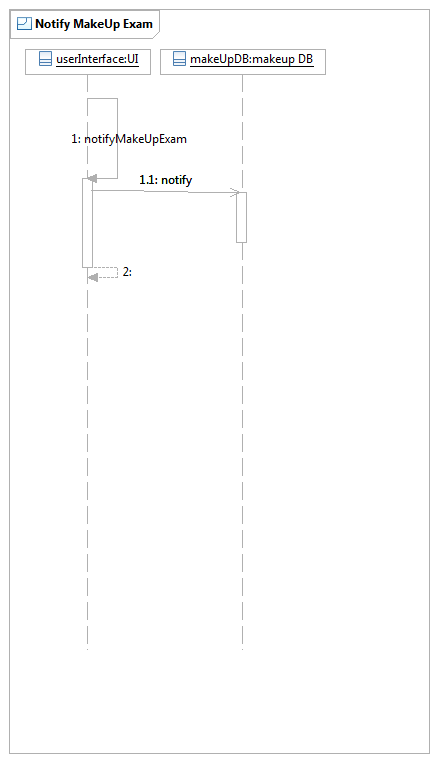
sequence1.gif





_O3ysA8UrEeWZT5QXZWhGxw.GIF





**Brief Description**

**Initial Step-By-Step Description**

## **2.3 User Characteristics**

The user of this software should be able to understand the basic terminologies of an academic institution i.e.- attendance sheet, seating arrangement, invigilator duty-chart etc. A user must have a good understanding of excel sheets.

## **2.4 Non-Functional Requirements**

The system must be able to identify failure cases and generate appropriate exceptions. The system must not damage existing data(e.g. - Input files) while operating on that.

# **3.0. Requirements Specification**

## **3.1 Functional Requirements**

## **3.1.1 Verifying Time-Table**

|  |  |
| --- | --- |
| **Use Case Name** | Time-Table Verification |
| **Trigger** | On the click of a button named “Verify time-table” |
| **Precondition** | The time-table provided should be in the correct format |
| **Basic Path** | 1. The exam schedule will be given as input. 2. Input the room availability list. 3. Click on the button “Verify Time-Table” 4. If not verified, notifies the user 5. Else go to next step |
| **Alternative Paths** |  |
| **Postcondition** | Will tell whether the exam schedule is feasible or not |
| **Exception Paths** | 1. Invalid format of exam schedule |
| **Other** |  |

### **3.1.2 Seating Arrangement**

|  |  |
| --- | --- |
| **Use Case Name** | Get Seating Arrangement |
| **Trigger** | User clicks the “Get Seating Arrangement” button. |
| **Precondition** | The time-table must be verified, the list of rooms with seating capacities and list of students with names and roll nos. registered per course must be available. |
| **Basic Path** | 1. The User clicks the “Get Seating Arrangement” button. 2. The user inputs the time-table file 3. The user inputs the available rooms list 4. The user inputs the list of enrolled students per course. 5. The system gives some files(No. of files = Rooms used for exam seating arrangement) |
| **Alternative Paths** | None |
| **Postcondition** | Some excel files giving seating arrangement for each room per exam. Each field of sheet will be having a roll no. |
| **Exception Paths** | 1. No of students for exam in any slot is greater than the total capacity of the available rooms for examination in that slot. 2. The input files are given in wrong format. 3. The memory available is not enough for storing output files. 4. The time-table is not verified. |
| **Other** | The output excel file can be converted into pdf. |

### **3.1.3 Generating attendance sheet**

|  |  |
| --- | --- |
| **Use Case Name** | Attendance Sheet Generation |
| **Trigger** | Generated with the generation of Seating Arrangement on the click of “Generate Seating Plan” |
| **Precondition** | The seating arrangement must be prepared. The list of students enrolled in all courses is available. |
| **Basic Path** | 1. The user clicks the “Generate Seating Plan” button. 2. The user inputs the seating arrangement file. 3. System generates the Attendance Sheet as referred in Section 1.3. |
| **Alternative Paths** | The user, after generating the seating arrangement, may choose to generate attendance sheets. |
| **Postcondition** | The attendance sheets in excel format per room per slot. |
| **Exception Paths** | 1. The memory available is not enough for storing output files. 2. The seating arrangement module output is in wrong format. |
| **Other** | Output can be converted into pdf files. |

### 

### **3.1.4 Invigilation duty-chart generation**

|  |  |
| --- | --- |
| **Use Case Name** | Generate invigilator duty chart |
| **Trigger** | The user clicks the button “Submit” |
| **Precondition** | The seating arrangement is given and the list of instructor and TA’s is available coursewise. |
| **Basic Path** | 1. The user inputs the Instructor’s list. 2. The user selects the output destination where to store the files generated. 3. The system generates the invigilator duty-chart per slot or appropriate exception. |
| **Alternative Paths** | None |
| **Postcondition** | The invigilator duty-chart per slot per room. It will be in excel file format. |
| **Exception Paths** | 1. Invalid file formats. 2. Available invigilators are less then as many as are required. |
| **Other** | The invigilator duty-chart must be mailed to all invigilators. |

### 

### **3.1.5 Exam Guidelines Notification**

|  |  |
| --- | --- |
| **Use Case Name** | Send exam guidelines |
| **Trigger** | A button “Mail Guidelines” to send mail to students, faculty and TA’s regarding the exam guidelines. |
| **Precondition** | The documents of guidelines should be ready before-hand. |
| **Basic Path** | 1. Browse the Guidelines to be mailed 2. Click on button “Mail Guidelines” to send them the notification. |
| **Alternative Paths** |  |
| **Postcondition** | Mail regarding the exam guidelines will be sent to all |
| **Exception Paths** | 1. Invalid entries occurring in the recipient list |
| **Other** | None |

### 

### 

### 

### **3.1.6 Make-Up Exam Notification**

|  |  |
| --- | --- |
| **Use Case Name** | Notifying about make-up exam |
| **Trigger** | On the click of a button-”Notify about Makeup Exam” |
| **Precondition** | Updating of the database of the make-up exam |
| **Basic Path** | 1. Update the make-up exam database(if needed) 2. Click on the button ”Notify about Makeup Exam” to send notification to the student and the concerned faculty |
| **Alternative Paths** | 1. Manually, enter the name of the recipients 2. Send them the notification |
| **Postcondition** | Notification send regarding the make-up exam to the concerned student and faculty. |
| **Exception Paths** | 1. Invalid entry 2. Disconnected from internet |
| **Other** | None |

### 

### **3.1.7 Make-Up exam database Updation**

|  |  |
| --- | --- |
| **Use Case Name** | Update Make-Up Exam Database |
| **Trigger** | The user clicks on the button “Make-up Manager” |
| **Precondition** | The list of the absentees should be provided. |
| **Basic Path** | 1. Click on the button “Make-up Manager” 2. Consequently, add or delete a field according to the requirement |
| **Alternative Paths** | None |
| **Postcondition** | Generates a database of students who have a pending make-up exam or update existing database. |
| **Exception Paths** | 1. Entry already exists 2. Invalid entry |
| **Other** | None |

### 

### 

### 

## **3.3 Detailed Non-Functional Requirements**

1. User’s system must support Python2.7 or later version.
2. Software output must be in excel or pdf format.
3. Software must not damage existing data or input files.
4. The first version software must be available to be used by 10.04.2016
5. User of this system shall authenticate themselves as office of academics staff members.
6. Software must be able to handle around 6000 students database and 300 courses list.

### ***3.4 Logical Structure of the Data***

<< Keep this blank for the time being>>

# **4.0 Supporting information**

# 

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

## **4.2 Appendixes**