Software Requirements Specification

Version 3.0

8th March 2016

Infrastructure 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).

Version 2

Edit History:-

1. One new assumption added.
2. One new definition added in Glossary.
3. Exception paths defined in functional requirements.
4. Added UML diagrams.

The students who submitted this team projects were Pranav Arora, Pushpinder Garg, Manish Goyal, Sahil Dhiman and Saurav Suman.

Version 3

Edit History-:

1. Added remaining use case diagrams.
2. Change the class diagram according to problems found in implementation.

# Table of Contents

[Table of Contents i](#_Toc440664050)

[List of Figures ii](#_Toc440664051)

[1.0. Introduction 1](#_Toc440664052)

[1.1. Purpose 1](#_Toc440664053)

[1.2. Scope of Project 1](#_Toc440664054)

[1.3 Constraints 1](#_Toc440664055)

[1.4 Assumptions and Dependencies 1](#_Toc440664056)

[1.3. Glossary 1](#_Toc440664057)

[1.4. References 1](#_Toc440664058)

[1.5. Overview of Document 1](#_Toc440664059)

[2.0. Overall Description 2](#_Toc440664060)

[2.1 System Environment 2](#_Toc440664061)

[2.2 Functional Requirements Specification 2](#_Toc440664062)

[2.2.1 Use case 1 2](#_Toc440664063)

[Use case: 2](#_Toc440664064)

[2.3 User Characteristics 2](#_Toc440664065)

[2.4 Non-Functional Requirements 2](#_Toc440664066)

[3.0. Requirements Specification 3](#_Toc440664067)

[3.1 Functional Requirements 3](#_Toc440664068)

[3.1.1 << Name of the first feature>> 3](#_Toc440664069)

[3.1.2 << Name of the second feature>> 3](#_Toc440664070)

[3.3 Detailed Non-Functional Requirements 3](#_Toc440664071)

[***3.4*** ***Logical Structure of the Data*** 3](#_Toc440664072)

[4.0 Supporting information 3](#_Toc440664073)

[4.1 Table of contents and index 3](#_Toc440664074)

[4.2 Appendixes 3](#_Toc440664075)

# List of Figures

**No table of figures entries found.**

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of Infrastructure Management System. It explains the features, interface, working of system and constraints under which the system operates. This document is intended for both the stakeholders and developers of the system.

## 1.2. Scope of Project

This software system will be an Infrastructure Management System which mainly manages Hostel Infrastructure and issues related to it. It keeps the record of various necessary things in a hostel and their quantity, allocated room etc. It also keeps the track of complaints, ordering new items and can schedule various tasks such as pest control, inspection. The software has reduced the workload of managing all the events and has automated most of the work which was earlier supposed to be done manually.

It provides a very comfortable environment for the student to register a complaint about any problem and keeps all the record in a database and the issue is resolved through different processes. All the activities take place under the surveillance Manager i.e admin. The software can handle infrastructure management of maximum 1000 students at a time.

## 1.3 Constraints

The number of student register.

Admin have to handle all the management task.

Admin have to manually update order completion and complaint completion.

## 1.4 Assumptions and Dependencies

* The software is assumed to work on various ideal condition such as when a complaint is registered it is assumed to be resolved at a particular time.
* Admin is assumed to update all the information whenever the assigned worker reports.
* All the students have single room.
* Worker are assumed to be assign for a complaint and are highly skilled and honest.
* If an Order is placed it is assumed to be delivered or cancelled according to its status

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Admin | The person managing the infrastructure. |
| Student | The person using the infrastructure. |
| User | The person using the software. |
| Order | The infrastructure ordered to be used in hostel. |
| Items | Basic requirements like tables, chairs, fan, tube lights, bulbs |
| Infrastructure | Basic facilities needed for the operation of hostel. |
| Scheduled task | Tasks to be performed at regular interval. |

## 1.4. References

Hostel Management committee, GPRA, Jodhpur

## 1.5. Overview of Document

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

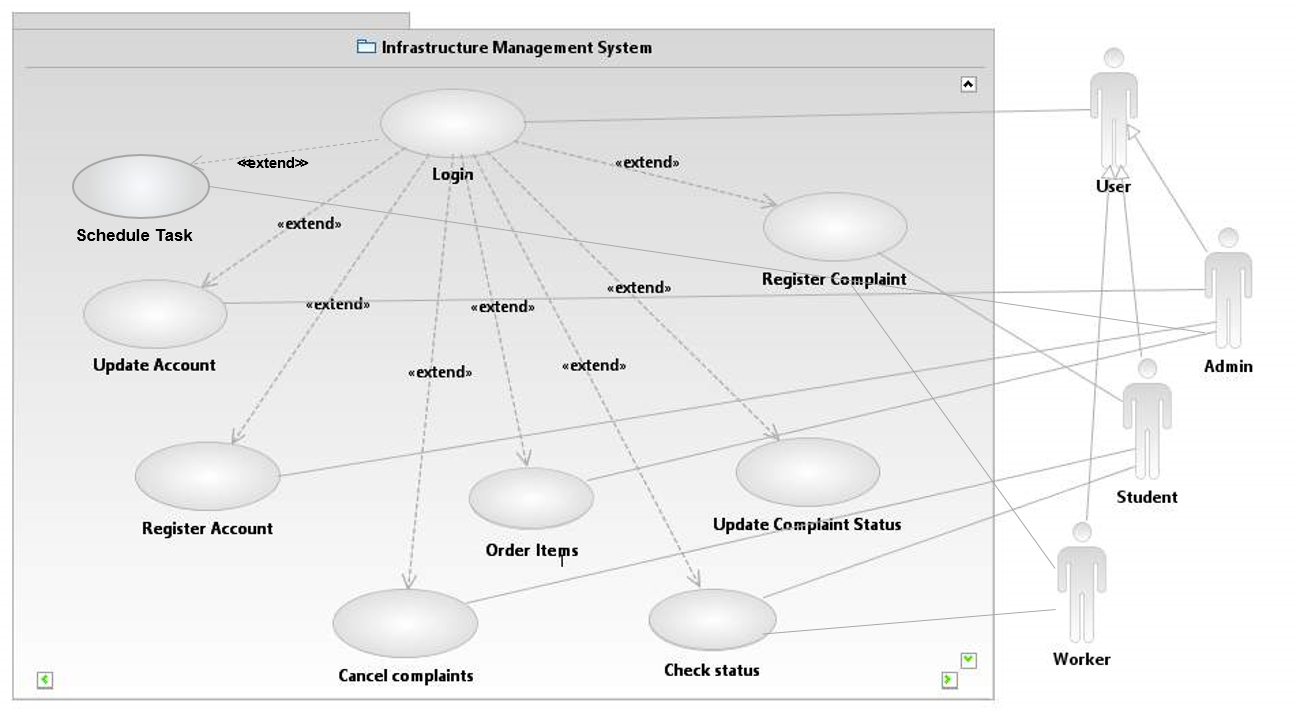
# 2.0. Overall Description

## 2.1 System Environment

<< Keep blank for the time being >>

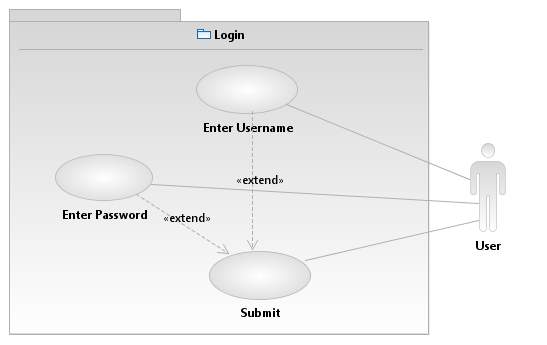
## 2.2 Functional Requirements Specification

This section describes the use cases for each of the actors separately. The student can use only one use case while the admin is the main actor in this system.



#### Use case: Login

**Diagram:**



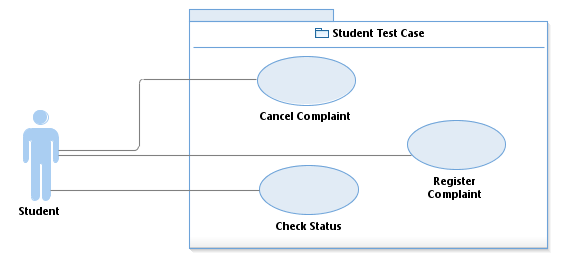
**Brief Description**

The user logs in the system through this use case.

**Initial Step-By-Step Description**

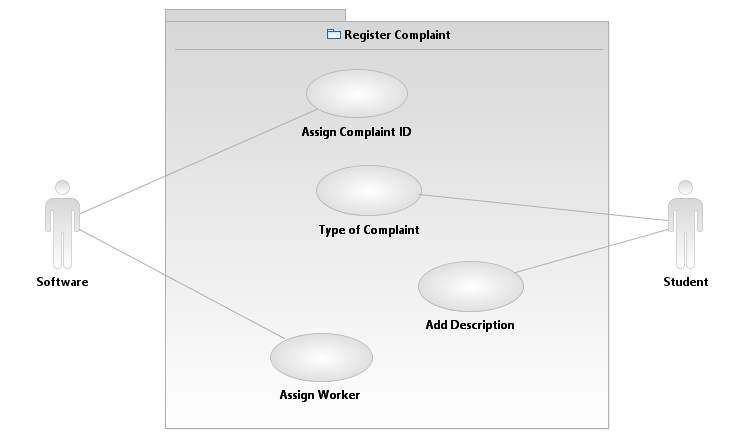
1. The system displays a login form.
2. The user enter a valid username and password.
3. The system logs in the users.

### 2.2.1 Student Use Case



#### Use case: Register Complaint

**Diagram:**



**Brief Description**

The user logs in the system and registers a complaint.

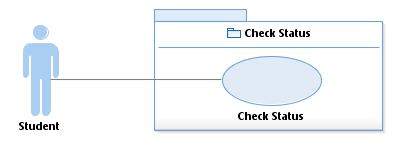
**Initial Step-By-Step Description**

Before the execution of the use case, the student logs in the system.

1. The student chooses to register a complaint.
2. The system displays a form and the desired options.
3. The student selects the type of complaint.
   1. Repair
   2. Service
   3. Other
4. The student adds a description of complaint.
5. The student clicks on submit and the system registers the complaint.
6. The system assigns the appropriate worker to the complaint.

#### Use case: Check Status

**Diagram:**



**Brief Description**

The user logs in the system and check the status of his/her registered complaints.

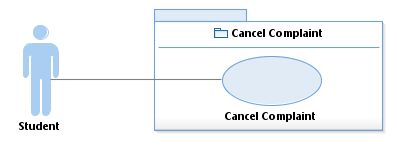
**Initial Step-By-Step Description**

Before the execution of the use case, the student logs in the system.

1. The student chooses to “check the status” of his/her complaints.
2. The system displays the current status of his/her registered complaints.

#### Use case: Cancel complaint

**Diagram:**

****

**Brief Description**

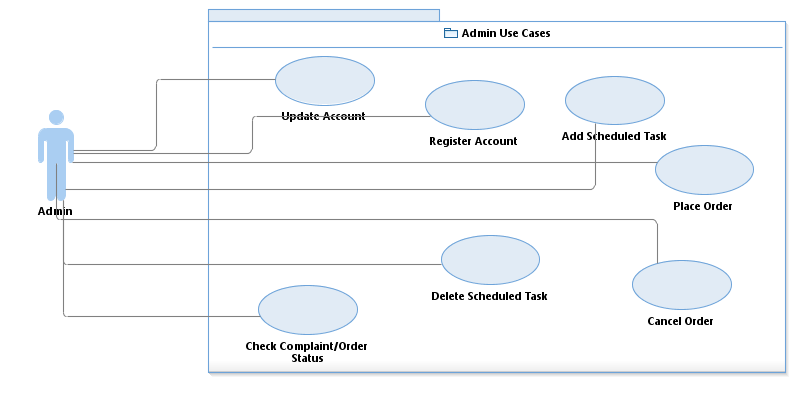
The user logs in the system and selects the option to cancel the complaint.

**Initial Step-By-Step Description**

Before the execution of the use case, the student logs in the system.

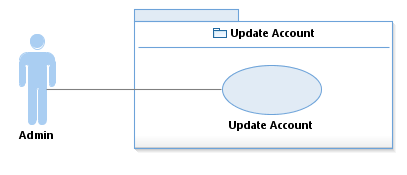
1. The student chooses to cancel a complaint.
2. The system displays a list of active registered complaints.
3. The student selects the complaint to be cancelled and click cancel.
4. The system cancels the complaint.

### 2.2.1 Admin Use Case



#### Use case: Update Account

**Diagram:**

****

**Brief Description**

The user logs in the system and selects the option to update the student account.

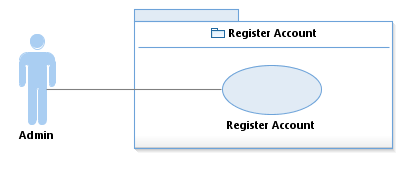
**Initial Step-By-Step Description**

Before the execution of the use case, the admin logs in the system.

1. The admin selects the relevant student account.
2. The system displays the information about the student.
3. The admin updates the account as per the student requirement and clicks the update button.
4. The system updates the information.

#### Use case: Register Account

**Diagram:**



**Brief Description**

The user logs in the system and selects the option to create a student account.

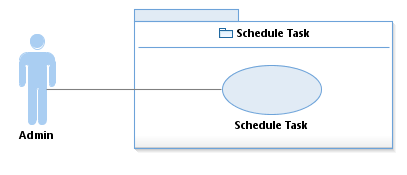
**Initial Step-By-Step Description**

Before the execution of the use case, the admin logs in the system.

1. The system displays the registration form.
2. The admin enters the required information as per the student and clicks register.
3. The system registers the account.

#### Use case: Schedule Task

**Diagram:**



**Brief Description**

The user logs in the system and selects the option to schedule a task and can remove an already scheduled task.

**Initial Step-By-Step Description**

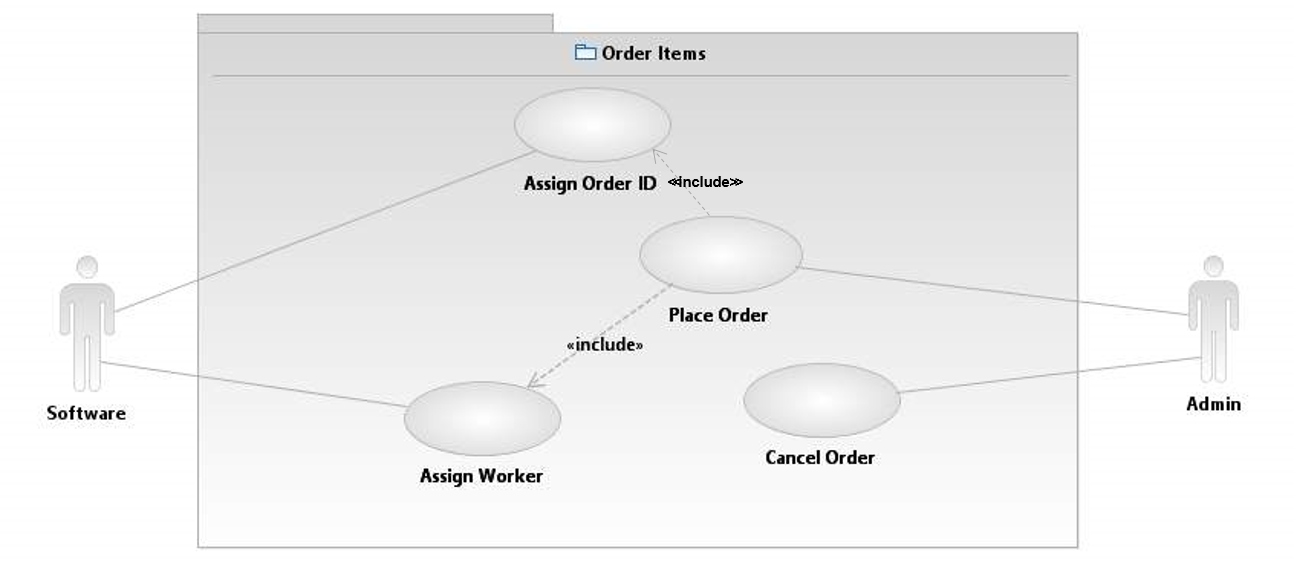
Before the execution of the use case, the admin logs in the system.

1. The admin selects the option to add/drop scheduled tasks.
2. The system shows a list of tasks scheduled.
3. The admin can either drop a task or add a new task.

#### Use case: Order

#### 

**Diagram:**

****

**Brief Description**

The user logs in the system and selects the order option.

**Initial Step-By-Step Description**

Before the execution of the use case, the admin logs in the system.

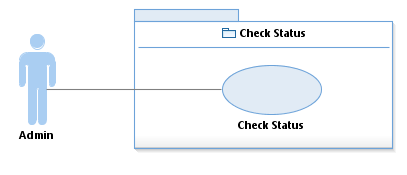
1. The admin selects the order option.
2. The system display the option :-
3. Place Order
4. Cancel Order
5. If admin selects Place Order
6. The system display the option :-

* Consumable
* Non - Consumable

1. The admin selects the desired option.
2. The system display a list of the items according to the selected option.
3. The admin selects the particular item and its quantity.
4. The system register the order

Use case: Check Complaint/Order Status

**Diagram:**



**Brief Description**

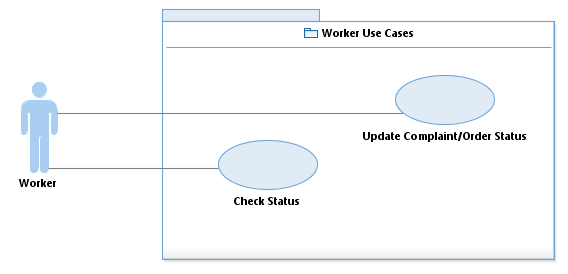
The admin can check the status of complaints/orders.

**Initial Step-By-Step Description**

Before the execution of the use case, the admin logs in the system.

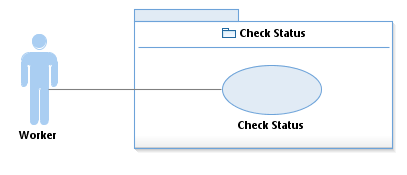
1. The admin click the Check Status option.
2. The system displays all the complaints/orders.

### 2.2.3 Worker Use Case



#### Use case: Check status

**Diagram:**



**Brief Description**

The worker logs in the system and checks the status of complaints/orders assigned to him.

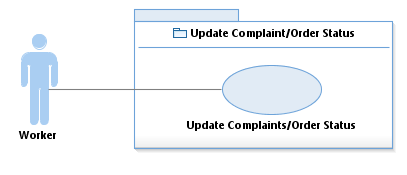
**Initial Step-By-Step Description**

Before the execution of the use case, the worker logs in the system.

1. The system displays a list of complaints/orders assigned to the worker.

#### Use case: Update complaint/Order status

**Diagram:**



**Brief Description**

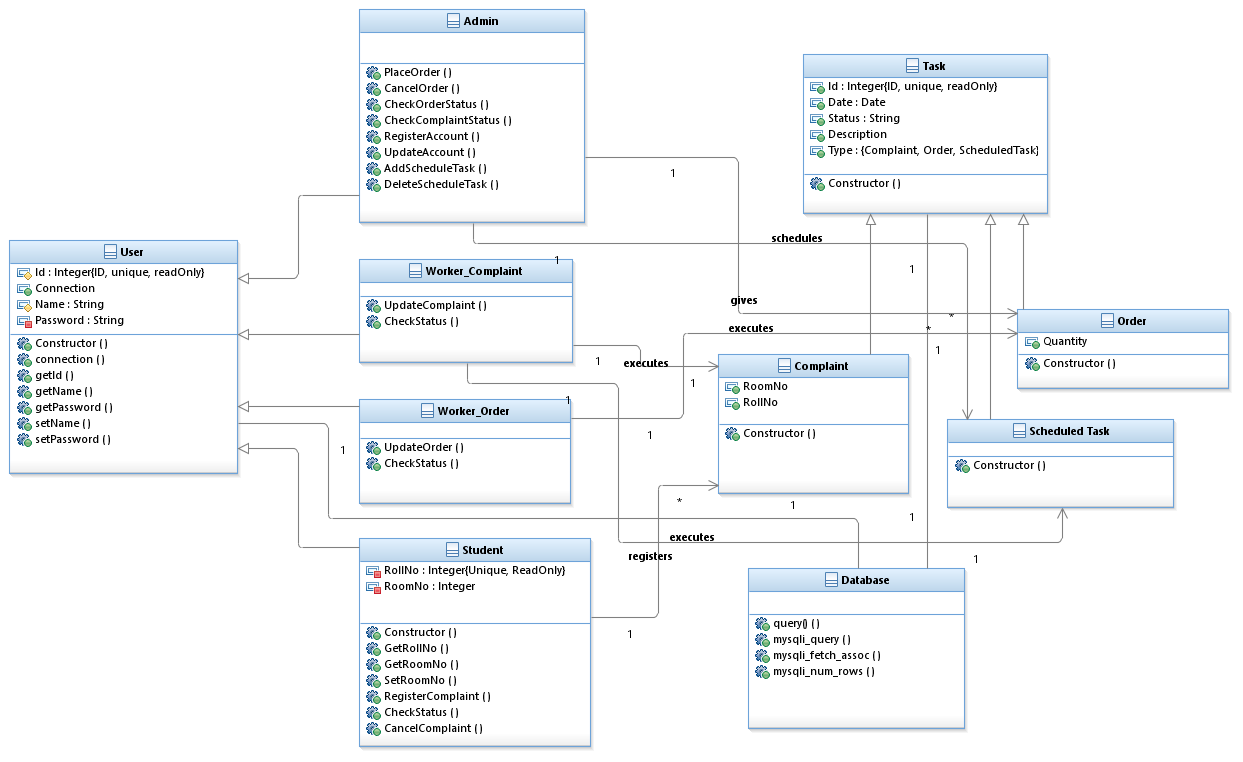
The worker logs in the system and checks the status of complaints/orders assigned to him.

**Initial Step-By-Step Description**

Before the execution of the use case, the worker logs in the system.

1. The system displays a list of complaints/orders assigned to the worker.
2. The worker selects the complaint to be updated.
3. The system updates the status of the selected complaint.

**CLASS DIAGRAM:**



**SEQUENCE DIAGRAMS**

## 

**ACTIVITY DIAGRAM:**

## 

## 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 Register complaint

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **Trigger** | The user chooses to register the complaint. |
| **Precondition** | The user logs in the system. |
| **Basic Path** | 1. Log in to the system. 2. Select register the complaint option. 3. Fill the form. 4. Click submit. |
| **Alternative Paths** | None |
| **Postcondition** | The complaint is registered. |
| **Exception Paths** | In case of power loss ,data will be lost |
| **Other** |  |

### 3.1.2 Cancel Complaint

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **Trigger** | The user chooses to cancel the complaint. |
| **Precondition** | The user logs in the system. |
| **Basic Path** | 1. Log in to the system. 2. Select to cancel the complaint option. 3. Select the complaint. 4. Click Cancel. |
| **Alternative Paths** | None |
| **Postcondition** | The selected complaint is cancelled. |
| **Exception Paths** | In case of power loss ,data will be lost |
| **Other** |  |

### 3.1.3 Check status

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **Trigger** | The user wishes to check the status of his/her complaints. |
| **Precondition** | The user logs in the system. |
| **Basic Path** | 1. Log in to the system. 2. Select check status option. 3. Select the complaint. 4. If admin, user can update the status. |
| **Alternative Paths** | None |
| **Postcondition** | The selected complaint is displayed/updated. |
| **Exception Paths** | In case of power loss ,data will be lost |
| **Other** |  |

### 3.1.4 Register Account

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **Trigger** | The student wants to create his/her account. |
| **Precondition** | The admin logs in the system. |
| **Basic Path** | 1. Log in to the system. 2. Select “Register Account” option. 3. Fill the details. 4. Click Register. |
| **Alternative Paths** | None |
| **Postcondition** | The student is registered. |
| **Exception Paths** | In case of power loss ,data will be lost |
| **Other** |  |

### 3.1.5 Update Account

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **Trigger** | The student wants to update his/her own account. |
| **Precondition** | The admin logs in the system. |
| **Basic Path** | 1. Log in to the system. 2. Select “Update Account” option. 3. Select the student account to be updated. 4. Fill in the details. 5. Click Update. |
| **Alternative Paths** | None |
| **Postcondition** | The student account is updated. |
| **Exception Paths** | In case of power loss ,data will be lost |
| **Other** |  |

### 3.1.6 Schedule Tasks

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **Trigger** | The admin wants to regulate tasks after fixed intervals. |
| **Precondition** | The user logs in the system. |
| **Basic Path** | 1. Log in to the system. 2. Select “Schedule Tasks” option. 3. Select Add/DROP 4. Fill in the details and select the time interval. |
| **Alternative Paths** | None |
| **Postcondition** | The scheduled task is set. |
| **Exception Paths** | In case of power loss ,data will be lost |
| **Other** |  |

### 3.1.7 Check status

|  |  |
| --- | --- |
| **Use Case Name** |  |
| **Trigger** | The admin wants to order infrastructure. |
| **Precondition** | The admin logs in the system. |
| **Basic Path** | 1. Log in to the system. 2. Select Order option. 3. Select the option- 4. Place Order - for placing new order 5. Cancel Order – for cancelling/decreasing order 6. Order Status – for checking the status |
| **Alternative Paths** | None |
| **Postcondition** | The order will be placed/cancelled/checked. |
| **Exception Paths** | In case of power loss ,data will be lost |
| **Other** |  |

## 3.3 Detailed Non-Functional Requirements

### ***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