Software Requirements Specification

Version 2.0

January 28, 2016

Institute Security Management

Submitted in partial fulfilment

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 were Tejas Reddy K, Raj Kumar Gutapu, Sravan Garimella and Karthik Katragadda. >>

Version Table

|  |  |  |
| --- | --- | --- |
| Version 1 | 16/1/2016 | Software Requirement Document |
| Version 2 | 28/1/2016 | UML Diagrams added |
| Version 3 | 8/3/2016 | Prototype Submission |

# Table of Contents

Table of Contents 1

List of Figures 2

1.0. Introduction 1

1.1. Purpose 1

1.2. Scope of Project 1

1.3 Constraints 1

1.4 Assumptions and Dependencies 2

1.3. Glossary 2

1.4. References 2

1.5. Overview of Document 2

Section 2 explains the overall description of the institute security system software. This includes the system environment, use case diagrams, user characteristics, and non-fictional requirements. 2

2.0. Overall Description 3

2.1 System Environment 3

2.2 Functional Requirements Specification 3

2.2.1 Use case 1 3

Use case: 3

2.3 User Characteristics 3

2.4 Non-Functional Requirements 3

3.0. Requirements Specification 4

3.1 Functional Requirements 4

3.1.1 Vehicle Entry 4

3.1.2 Vehicle exit 5

3.1.3 Solution of guard’s absence 5

3.1.4 Allocation of guards at buildings 6

3.1.5 Emergency 6

3.3 Detailed Non-Functional Requirements 7

3.4 Logical Structure of the Data 7

4.0 Supporting information 7

4.1 Table of contents and index 7

4.2 Appendixes 7

# List of Figures

Use Case Diagrams

Class Diagrams

Sequential Diagrams

Activity Diagrams

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to present a detailed description of a security management software. It explains the functionality of the system to ensure the safety and security of the institute, supervision of vehicle movements and to alert the authorities in case of a security breach. This document is intended for the Chief Security Officer and the security guards.

## 1.2. Scope of Project

This software system will be designed to help the Security Services of the institute by providing a database having the details of the personnel currently on duty, a record of the vehicle movement, and a trigger to the buzzer which is used in emergency conditions. Reallocation and backup from the security department is done in case of any shortage of the personnel for surveillance. In case of any mishap, a buzzer will be used to alert the authorities. The entire database will be accessible by the Chief of Logistics.

## 1.3 Constraints

1. All people come via vehicles only.

2. Each vehicle will have a unique identification number on its nameplate. Each driver has a valid ID card.

3. Security personnel must be familiar with operating the software.

4. All security guards must be loyal

## 1.4 Assumptions and Dependencies

1. Guards know a priori how to use the software irrespective of the guard.
2. All buzzers are working.
3. Backup security is available.
4. Information stored is backed up and safe from power cuts and hackers.

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Buzzer | An electronic sounding device |
| ID | Identity Details |
| Security | The set of guards that are responsible for safety |
| Personnel | Security guards (in this context) |
| Security Breach | Any happenings that are not accepted by the security for any reasons. |
| Backup | Extra guards intended for shortages or emergencies |
| miscreant | A terrorist, burglar or any person who will compromise security |
| Shift | The duration and period of the work |

## 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 is designed in the following way:

# Section 2 explains the overall description of the institute security system software. This includes the system environment, use case diagrams, user characteristics, and non-fictional requirements.

**Section 3** explains the functional and non-functional requirement specifications. Details the logical structure of the data used.

**Section 4** gives the required supporting information.

# 2.0. Overall Description

## 2.1 System Environment

<< Keep blank for the time being >>

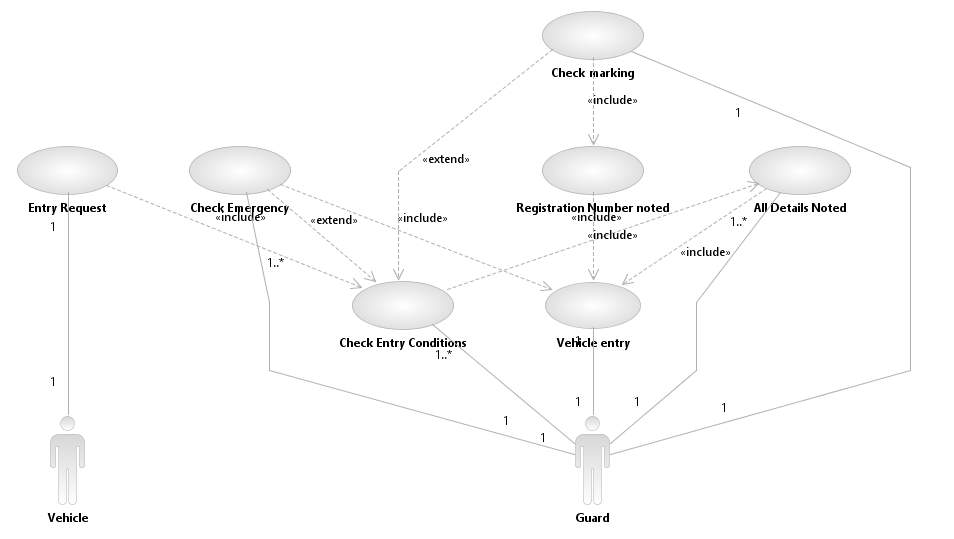
## 2.2 Functional Requirements Specification

. << Keep blank for the time being >>

### 2.2.1 Use case 1

#### Use case: Vehicle Entry

**Diagram:**



**Brief Description:** The above use case describes the process through which a vehicle is allowed or denied access into the campus.

**Initial Step-By-Step Description**

1. The vehicle appears at the gate.

2. The guard checks if the vehicle is marked or unmarked.

3. If marked, the vehicle is allowed inside and the registration number is noted down.

4. If unmarked the driver’s ID will be checked and if everything is fine his vehicle’s details are noted and he is granted permission.

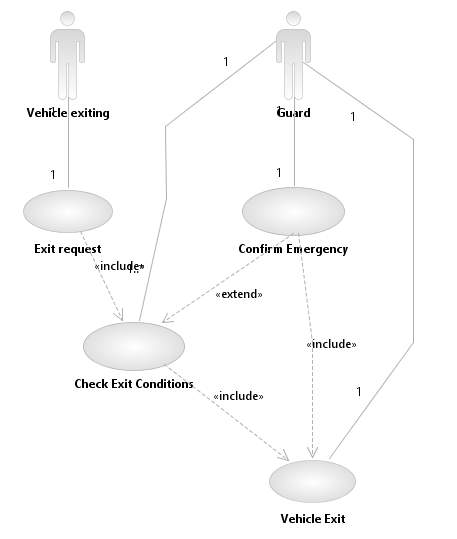
5. In both cases precondition is checked.

6. Vehicle is allowed in or sent out.

### 2.2.2 Use case 2

Use case: **Vehicle Exit**

**Diagram :**



**Brief Description:** This is the use case that allows the permission to leave the institute

**Step by step description:**

1. The vehicle appears at the gate.

2. The guard checks for precondition.

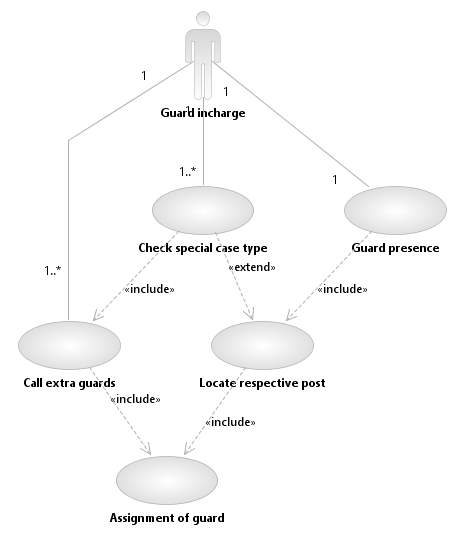
3. If the vehicle follows precondition, the vehicle is allowed out.

4. If not, the vehicle is detained.

### 2.2.3 Use case 3

Use case: **Guard Allocation**

**Diagram :**



### **Brief Description:** This use case positions guards at various posts.

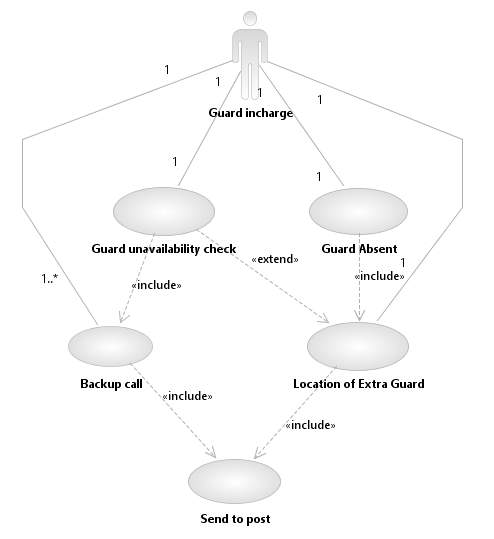
**Step by step description:**

1. This use case allocates guards at various positions
2. Check if any buildings are important.
3. Send extra guards to important buildings.

### 2.2.4 Use case 4

Use case: **Solution to guard absence**

**Diagram :**



**Brief Description:** This use case will allocate a guard whenever there is an absence.

**Step by step description:**

1. Identify when guards are absent
2. If guards are absent, use guards from backup to fill the positions.

### 2.2.5 Use case 5

Use case: **Emergency**

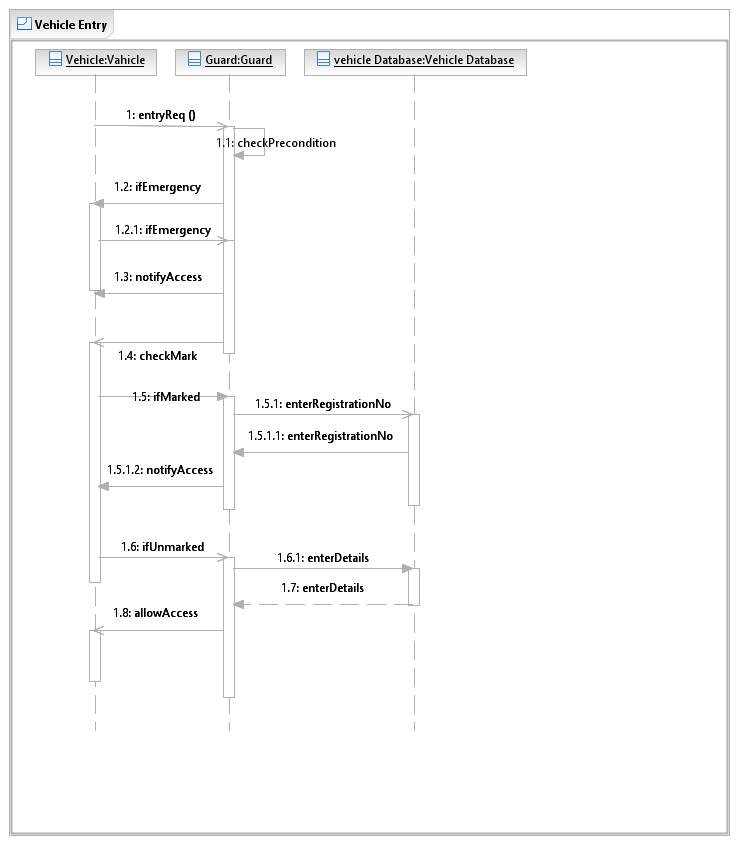
**Diagram:**

## H:\emergency.jpeg

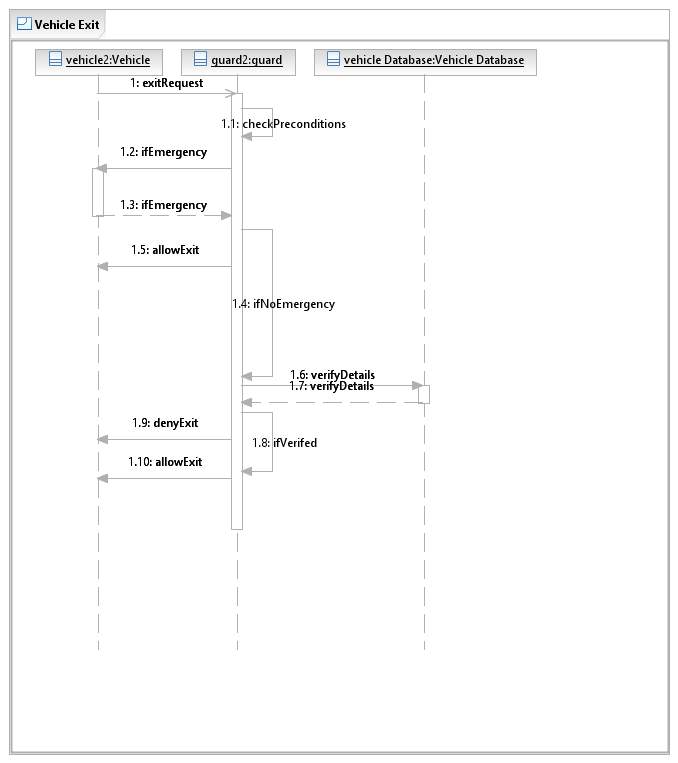
**Brief Description:** This use case alerts authorities in case of an emergency

**Step by step description:**

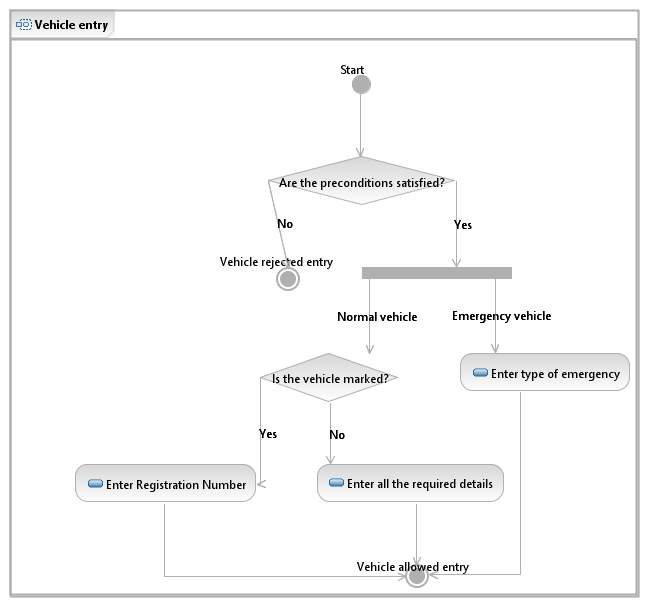
1. Guards press the panic button in case of an emergency.
2. The guards have to choose type of emergency.
3. Respective authorities are alerted.
4. Students are alerted by a buzzer.
   * 1. **Sequence Diagrams:** Vehicle Entry



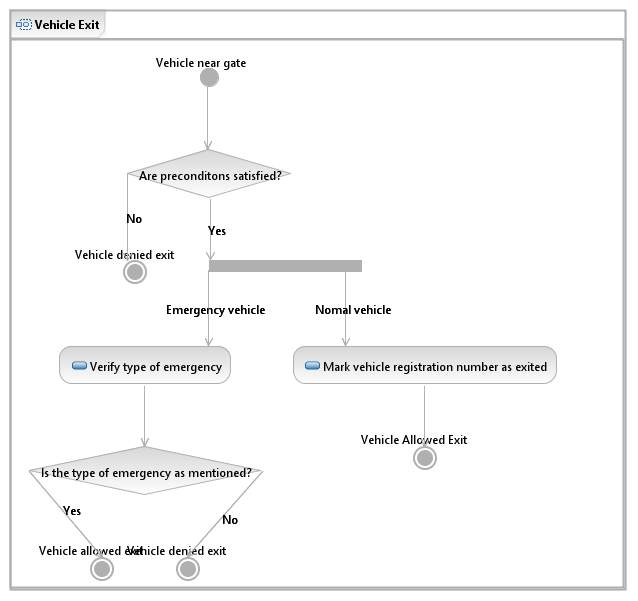
* + 1. **Sequence Diagrams:** Vehicle Exit



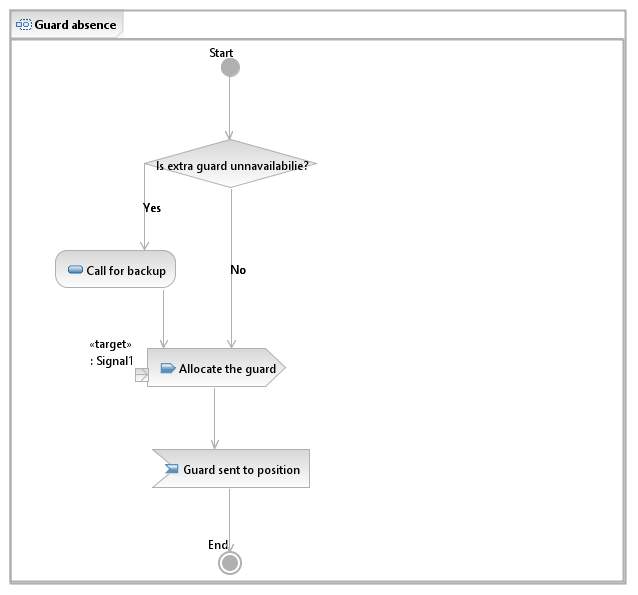
* + 1. **Activity Diagrams:** Vehicle Entry



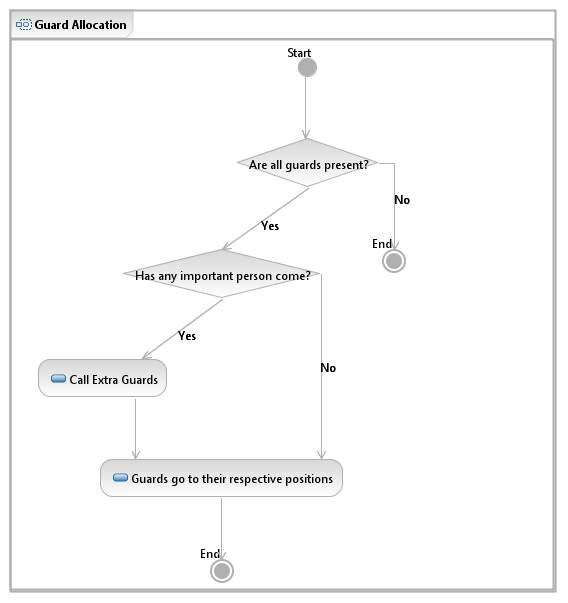
* + 1. **Activity Diagrams:** Vehicle Exit



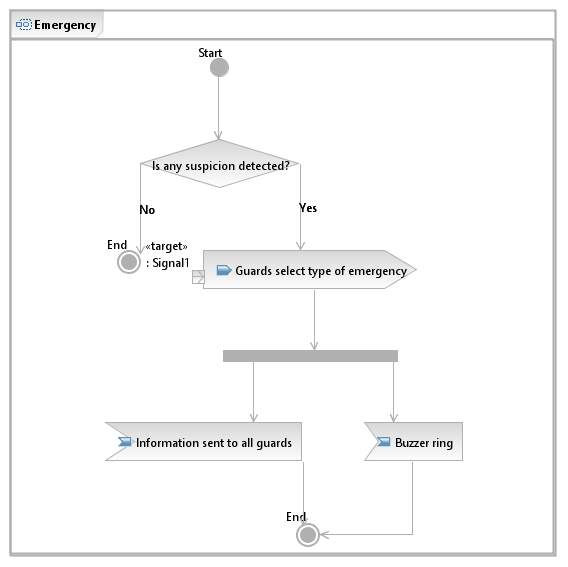
* + 1. **Activity Diagrams:** Guard Absence



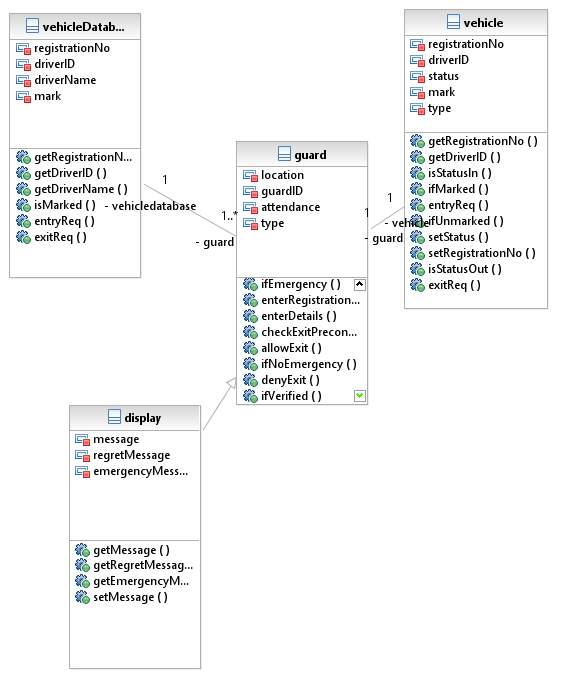
* + 1. **Activity Diagrams:** Guard Position

****

* + 1. **Activity Diagrams:** Emergency



* + 1. **Class Diagrams :**



## 2.3 User Characteristics

Knowledge of use of computer  
 Knowledge of basic english

## 2.4 Non-Functional Requirements

# 3.0. Requirements Specification

## 3.1 Functional Requirements

### 3.1.1 Vehicle Entry

|  |  |
| --- | --- |
| **Use Case Name** | Vehicle entry |
| **Trigger** | When a vehicle appears at the entry point. |
| **Precondition** | It should either be a new vehicle or a vehicle that previously exited the institute |
| **Basic Path** | 1. Vehicle appears at the entry point. 2. Security notes down vehicle details (Registration no., model of car,Driver ID) 3. Check for precondition 4. If it follows precondition allow the vehicle in or do not allow. 5. Faculty or Staff will have their vehicle marked with a sticker. |
| **Alternative Paths** | 1. If the vehicle is marked, only the registration number will be noted. 2. The vehicle is then allowed in without further checks. |
| **Postcondition** | A vehicle is either allowed or rejected entry into the institute . |
| **Exception Paths** | 1. Access should be granted immediately to vehicles like ambulances, fire brigades and police vehicles. |
| **Others** | This use case will ensure that miscreants do not enter the  Campus. |

### 3.1.2 Vehicle exit

|  |  |
| --- | --- |
| **Use Case Name** | Vehicle Exit |
| **Trigger** | When a vehicle appears at the edit point. |
| **Precondition** | It should a vehicle that previously entered the institute |
| **Basic Path** | 1. Vehicle appears at the exit point. 2. Security verifies the registration no. of the vehicle 3. Check for precondition 4. If it follows precondition allow the vehicle out or stop the vehicle. |
| **Alternative Paths** | None. |
| **Postcondition** | Vehicle exits or is stopped. |
| **Exception Paths** | 1. Access should be granted immediately to vehicles like ambulances, fire brigades and police vehicles. |
| **Other** | Guards are to be present and also alert during the process. |

### 3.1.3 Solution to guard’s absence

|  |  |
| --- | --- |
| **Use Case Name** | Solution of guard’s absence |
| **Trigger** | In case of guard absence due to some valid reason |
| **Precondition** | There is a guard present to replace the absent guard. |
| **Basic Path** | 1. Locate an extra guard 2. Send him to the empty post. |
| **Alternative Paths** | None |
| **Postcondition** | All posts are occupied by guards |
| **Exception Paths** | 1. In case of no extra guards, use backup. |
| **Other** | None |

## 3.1.4 Allocation of guards at buildings

|  |  |
| --- | --- |
| **Use Case Name** | Guard Positioning |
| **Trigger** | At the beginning of their duty shift |
| **Precondition** | The respective guard is present. |
| **Basic Path** | At the beginning of each shift, guards are allocated in their respective positions. |
| **Alternative Paths** | None |
| **Postcondition** | Guards are present at their respective positions |
| **Exception Paths** | If any important person enters a particular area in the institute, extra guards are assigned. |
| **Other** | Guards are to be present and also alert during the process. |

### 3.1.5 Emergency

|  |  |
| --- | --- |
| **Use Case Name** | Emergency |
| **Trigger** | If a guard detects some suspicious conditions |
| **Precondition** | None |
| **Basic Path** | 1. Guard hits an emergency button 2. Information regarding the situation is notified to all the guards. 3. A buzzer is rung. |
| **Alternative Paths** | None |
| **Postcondition** | All guards and local authorities are on high alert. |
| **Exception Paths** | None |
| **Other** | None |

## 3.3 Detailed Non-Functional Requirements

**Reliability :** The software is reliable and cannot be hacked as it is present only inside the institute. The software will never fail the user and work at all times.

**Precision** : The vehicle entry, exit details and their times of arrival are noted and presented with no error. The records of past data and other details are maintained carefully.

**Availability :** The application will be available to the guards at checkpoints and chief of logistics at all times

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

# 4.0 Supporting information

## 4.1 Table of contents and index

## 4.2 Appendices