****

**COMSATS University Islamabad (CUI)**

Software Requirement Specification  
(SRS DOCUMENT)

for

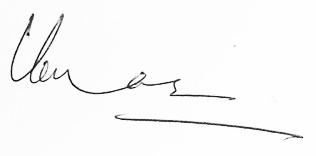
**EMPLOYEE IDENTIFICATION AND TRACKING**  
Version 1.0

***By***

**Muhammad Umer Naeem CIIT/FA17-BSE-140/ISB**

**Hamza Tahir CIIT/FA17-BSE-148/ISB**

***Supervisor*Mr. Umar Nauman**

**

*Bachelor of Science in Software Engineering (2017-2021)*

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Application Evaluation History

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Supervised by

Mr. Umar Nauman

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table of Contents

[1. Introduction 7](#_Toc55155641)

[1.1 Purpose 7](#_Toc55155642)

[1.2 Scope 7](#_Toc55155643)

[2. Overall Description 7](#_Toc55155644)

[2.1 Product Perspective 8](#_Toc55155645)

[2.2 Operating Environment 8](#_Toc55155646)

[2.3 Design and Implementation Constraints 8](#_Toc55155647)

[3. Requirement Identifying Technique 8](#_Toc55155648)

[3.1 Event Response Table 8](#_Toc55155649)

[3.2 Use Case Diagram 9](#_Toc55155650)

[3.3 Use Case Description 12](#_Toc55155651)

[4. Functional Requirements 46](#_Toc55155652)

[4.1 Functional Requirement 1 46](#_Toc55155653)

[4.2 Functional Requirement 2 46](#_Toc55155654)

[4.3 Functional Requirement 3 46](#_Toc55155655)

[4.4 Functional Requirement 4 47](#_Toc55155656)

[4.5 Functional Requirement 5 47](#_Toc55155657)

[4.6 Functional Requirement 6 48](#_Toc55155658)

[4.7 Functional Requirement 7 48](#_Toc55155659)

[4.8 Functional Requirement 8 49](#_Toc55155660)

[4.9 Functional Requirement 9 49](#_Toc55155661)

[4.10 Functional Requirement 10 50](#_Toc55155662)

[4.11 Functional Requirment 11 51](#_Toc55155663)

[4.12 Functional Requirement 12 51](#_Toc55155664)

[4.13 Functional Requirement 13 52](#_Toc55155665)

[4.14 Functional Requirement 14 52](#_Toc55155666)

[4.15 Functional Requirement 15 53](#_Toc55155667)

[4.16 Functional Requirement 16 53](#_Toc55155668)

[4.17 Functional Requirement 17 54](#_Toc55155669)

[4.18 Functional Requirement 18 54](#_Toc55155670)

[4.19 Functional Requirement 19 55](#_Toc55155671)

[4.20 Functional Requirement 20 55](#_Toc55155672)

[4.21 Functional Requirement 21 56](#_Toc55155673)

[4.22 Functional Requirement 22 56](#_Toc55155674)

[4.23 Functional Requirement 23 57](#_Toc55155675)

[4.24 Functional Requirement 24 57](#_Toc55155676)

[4.25 Functional Requirement 25 58](#_Toc55155677)

[4.26 Functional Requirement 26 58](#_Toc55155678)

[4.27 Functional Requirement 27 59](#_Toc55155679)

[4.28 Functional Requirement 28 59](#_Toc55155680)

[4.29 Functional Requirement 29 60](#_Toc55155681)

[4.30 Functional Requirement 30 60](#_Toc55155682)

[4.31 Functional Requirement 31 61](#_Toc55155683)

[4.32 Functional Requirement 32 61](#_Toc55155684)

[4.33 Functional Requirement 33 62](#_Toc55155685)

[4.34 Functional Requirement 34 62](#_Toc55155686)

[4.35 Functional Requirement 35 63](#_Toc55155687)

[5. Non-Functional Requirements 63](#_Toc55155688)

[5.1 Usability 63](#_Toc55155689)

[5.2 Performance 63](#_Toc55155690)

[6. Project Gantt Chart 63](#_Toc55155691)

[7. References 64](#_Toc55155692)

[8. Plagiarism Report 64](#_Toc55155693)

List of Figures

[Figure 1: Use Case Diagram of Manager. 10](#_Toc55304226)

[Figure 2: Usecase Diagram of Admin. 11](#_Toc55304227)

[Figure 3: Usecase diagram of Employee. 12](#_Toc55304228)

[Figure 4: Gantt chart. 64](#_Toc55304229)

List of Tables

[Table 1: Event Response Table. 9](#_Toc55300895)

[Table 2: UC 1 - Textual Description of Login. 13](#_Toc55300896)

[Table 3: UC 2 - Textual Description of Create User. 14](#_Toc55300897)

[Table 4: UC 3 - Textual Description of Edit User. 15](#_Toc55300898)

[Table 5: UC 4 - Textual Description of View User. 16](#_Toc55300899)

[Table 6: UC 5 - Textual Description of Edit Profile. 17](#_Toc55300900)

[Table 7: UC 6 - Textual Description of Issue RFID. 18](#_Toc55300901)

[Table 8: UC 7 - Textual Description of Revoke RFID. 19](#_Toc55300902)

[Table 9: UC 8 - Textual Description of View Video feed. 20](#_Toc55300903)

[Table 10: UC 9 - Textual Description of Delete User. 21](#_Toc55300904)

[Table 11: UC 10 - Textual Description of Setup Geofencing. 22](#_Toc55300905)

[Table 12: UC 11 - Textual Description of Edit Geofencing. 23](#_Toc55300906)

[Table 13: UC 12 - Textual Description of Delete User. 24](#_Toc55300907)

[Table 14: UC 13 - Textual Description of Forgot Password. 25](#_Toc55300908)

[Table 15: UC 14 - Textual Description of Search Employee. 26](#_Toc55300909)

[Table 16: UC 15 - Textual Description of Recent location. 27](#_Toc55300910)

[Table 17: UC 16 - Textual Description of Generate Report. 28](#_Toc55300911)

[Table 18: UC 17 - Textual Description of Create Managers. 29](#_Toc55300912)

[Table 19: UC 18 - Textual Description of Add Managers. 30](#_Toc55300913)

[Table 20: UC 19 - Textual Description of Create Organization. 31](#_Toc55300914)

[Table 21: UC 20 - Textual Description of Edit Manager. 32](#_Toc55300915)

[Table 22: UC 21 - Textual Description of Edit Organization. 33](#_Toc55300916)

[Table 23: UC 22 - Textual Description of View Organization. 34](#_Toc55300917)

[Table 24: UC 23 - Textual Description of Delete Organization. 35](#_Toc55300918)

[Table 25: UC 24 - Textual Description of Delete Video Feed. 36](#_Toc55300919)

[Table 26: UC 25 - Textual Description of Delete Room. 37](#_Toc55300920)

[Table 27: UC 26 - Textual Description of Delete Manager. 38](#_Toc55300921)

[Table 28: UC 27 - Textual Description of Remove Camera. 39](#_Toc55300922)

[Table 29: UC 28 - Textual Description of Adding RFID Sensor. 40](#_Toc55300923)

[Table 30: UC 29 - Textual Description of Remove RFID. 41](#_Toc55300924)

[Table 31: UC 30 - Textual Description of Adding Camera. 42](#_Toc55300925)

[Table 32: UC 31 - Textual Description of Creating Room. 43](#_Toc55300926)

[Table 33: UC 32 - Textual Description of View Room. 44](#_Toc55300927)

[Table 34: UC 33 - Textual Description of Preview Attendance. 45](#_Toc55300928)

[Table 35: Description of FR-1: Check in. 46](#_Toc55300929)

[Table 36: Description of FR-2: Check out. 46](#_Toc55300930)

[Table 37: Description of FR-3: Modify Attendance. 47](#_Toc55300931)

[Table 38: Description of FR-4: Check Attendance. 47](#_Toc55300932)

[Table 39: Description of FR-5: Report Generation. 48](#_Toc55300933)

[Table 40: Description of FR-6: Monthly Report. 48](#_Toc55300934)

[Table 41: Description of FR-7: Create Organization. 49](#_Toc55300935)

[Table 42: Description of FR-8: Edit Organization. 49](#_Toc55300936)

[Table 43: Description of FR-9: View Organization. 50](#_Toc55300937)

[Table 44: Description of FR-10: Delete Organization. 50](#_Toc55300938)

[Table 45: Description of FR-11: Create Room. 51](#_Toc55300939)

[Table 46: Description of FR-12: Edit Room. 51](#_Toc55300940)

[Table 47: Description of FR-13: Create Room 52](#_Toc55300941)

[Table 48: Description of FR-14: Delete Room. 52](#_Toc55300942)

[Table 49: Description of FR-15: Add Camera. 53](#_Toc55300943)

[Table 50: Description of FR-16: Remove Camera. 53](#_Toc55300944)

[Table 51: Description of FR-17: Add RFID Sensor. 54](#_Toc55300945)

[Table 52: Description of FR-18: Remove RFID Sensor. 54](#_Toc55300946)

[Table 53: Description of FR-19: View Camera Feed. 55](#_Toc55300947)

[Table 54: Description of FR-20: Delete Camera Feed. 55](#_Toc55300948)

[Table 55: Description of FR-21: Login 56](#_Toc55300949)

[Table 56: Description of FR-22: Edit Profile. 56](#_Toc55300950)

[Table 57: Description of FR-23: Create Users. 57](#_Toc55300951)

[Table 58: Description of FR-24: Edit Users. 57](#_Toc55300952)

[Table 59: Description of FR-23: View Users. 58](#_Toc55300953)

[Table 60: Description of FR-26: Delete Users. 58](#_Toc55300954)

[Table 61: Description of FR-23: Issue RFID Cards. 59](#_Toc55300955)

[Table 62: Description of FR-28: Revoke RFID Cards. 59](#_Toc55300956)

[Table 63: Description of FR-29: Creating Geofencing. 60](#_Toc55300957)

[Table 64: Description of FR-30: Edit Geofencing. 60](#_Toc55300958)

[Table 65: Description of FR-31: Delete Geofencing. 61](#_Toc55300959)

[Table 66: Description of FR-32: Search Employee. 61](#_Toc55300960)

[Table 67: Description of FR-33: Search Employee using Images. 62](#_Toc55300961)

[Table 68: Description of FR-34: Alert Manager. 62](#_Toc55300962)

[Table 69: Description of FR-35: Check Attendance. 63](#_Toc55300963)

# Introduction

This document aims at providing the complete insight of the given application. The first section provides the purpose with a brief introduction of the scope and what this app aims to achieve.The Second portion will accommodate the overall description of our website and the products context and origin. It explains in great detail how and what our product is and its perspective and its operating environment. It also holds the key to our product’s implementation and design constraints.The Third portion entails the requirements identification techniques. The processes and resources that the product uses and the brief introduction to the data gathering strategy. This section also contains our use case diagram.

## Purpose

According to Forbes, during a survey where 89% of respondents said they wasted time every day, which was with an increase of 20% from the previous year. The manual process of employee tracking and identification is very time consuming and still not efficient.

Our efforts aim to produce a product that can help the managers automate the process of employee tracking and identification. This application enables the managers to keep search for a particular person in the real time security footage and generate a report.

## Scope

Our system is focused on offices that want to monitor their employee using an automated system. Once the user is registered on the system, he can create his organization and add users to the organization with different roles created by the user. Once it is done, the user can add cameras and RFID sensors, and label them with their positions. Now the system is ready for use, and the user with appropriate permissions defined in the roles can add an employee with their images to train the system. As soon as the employee is registered, the system will be trained with the employees face and issued with an RFID card. As the employees are registered, the employees will be clocked in based on the camera at the entrance and clocked out. An overall report of employees can also be reviewed at any time. Moreover, warning and challans can also be generated and sent to the employees. The system also allows the user with appropriate permissions to search a user with an image or name to search for the location. The same system will be used to monitor the visitors in the office.

# Overall Description

This section presents the overall description of the targeted project.

## Product Perspective

In this project we will be developing a web application and a mobile application. This system will allow the office management to automate the process of employee identification and tracking. This system offers a unique solution that used RFID sensors and security cameras to improve the efficiency and reduce the total cost to track the employees.

## Operating Environment

*OE-1: The web application for this system shall be deployed on Windows IIS Server 2014.*

*OE-2: The database for this server will be SQL Server 2014.*

*OS-3: The mobile application shall run on android 10.*

## Design and Implementation Constraints

*CO-1: The system shall use the SQL Server 2014.*

*CO-2: The system shall use ASP.NET Core For the development of backend API’s*

*CO-3: The system shall use Angular 7 for the development of Frontend.*

*CO-4: The system shall use Android SKD 29 for the Development of Mobile Application.*

*CO-5: The system shall use Python 3.8 to develop the machine learning code.*

*CO-6: The system shall use IronPython to integrate Python with C# code.*

*CO-7: The user needs to have a knowledge of basic computer skills to use the application.*

# Requirement Identifying Technique

This section describes the requirements identifying technique(s) which further help to derive

## Event Response Table

Table 1: Event Response Table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event | Trigger | Source | Use Case | Response | Destination |
| Employee is checked in to the system | Camera on the enterance detecting face | User | Check in the employee | Store the check in time in the database | System |
| Employee is checked out from the system | Camera on the exit detecting the face | User | Check out the employee | Store the check out time in the database | System |
| Update Location in the database | RFID sensor detecting RFID card and Camera detecting face | User | Update location | Store the latest location of the user in the database | System |
| Geofencing Alert | RFID sensor detecting RFID card and Camera detecting face | User | Check for Access Control | Check the person in the database and his access control else report to the Manager and store in the database | Manager,  System |
| Conflict Resolution requet | RFID sensor and Camera detecting in different places | User | RFID sensor and Camera conflict | Report to the manager and send the email as well as log in the system | Manager,  Employee,  System |
| Unknown Person identification | Camera detecting faces | User | Unknown person identification | Report to the Manager and send a notification | Manager,  System |
| Monthly Report Generation | End of the month |  | Monthly Report Generation | Annual Activity Report | Manager |

## Use Case Diagram

The use case diagrams of our web application is give below.

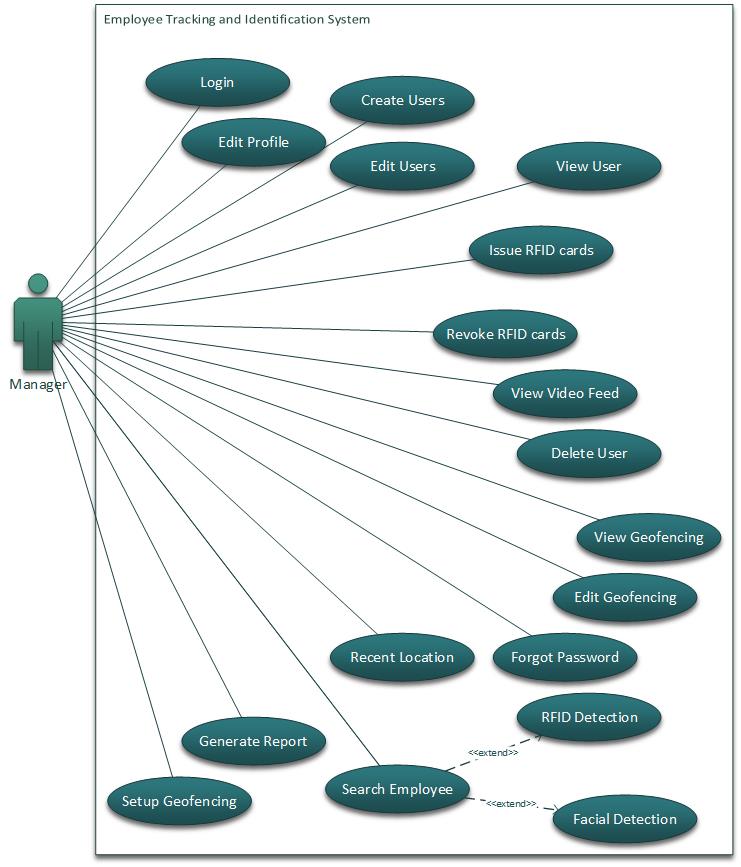


Figure : Use Case Diagram of Manager.

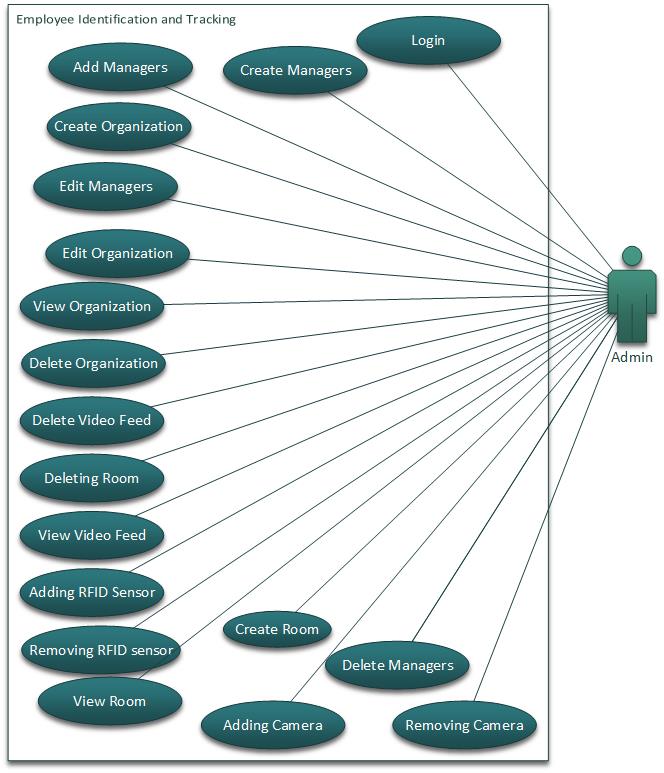


Figure : Usecase Diagram of Admin.

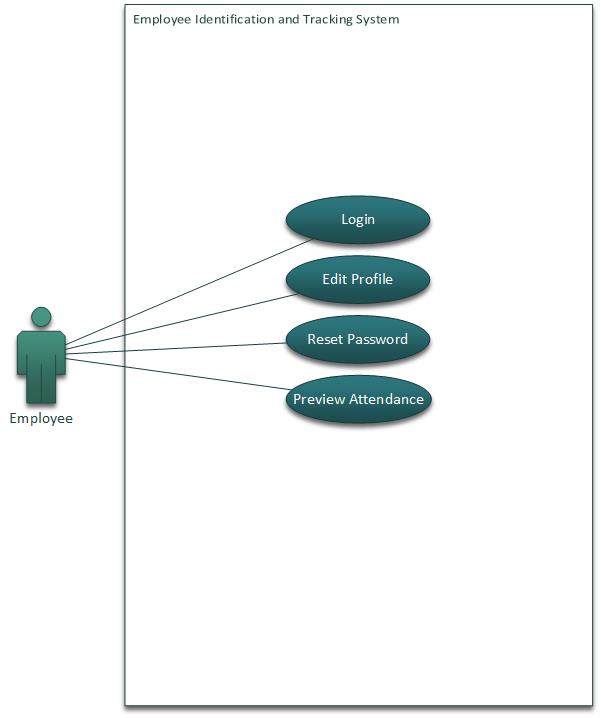


Figure : Usecase diagram of Employee.

## Use Case Description

Table 2: UC 1 - Textual Description of Login.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1 |
| **Use Case Name:** | Login |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | User | |
| **Description:** | This will allow the user to login to the system. |
| **Trigger:** | When the user presses “Login” button. |
| **Preconditions:** | PRE-1. The user open the application.  PRE-2. The user enters the email address.  PRE-3. The user enters the password. |
| **Postconditions:** | POST-1 The user is taken to the dashboard. |
| **Normal Flow:** | 1. User opens the application. 2. User enters the email address and the password. 3. User presses the login button. 4. If the credentials are correct the user is taken to the dashboard. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. Email is not correct.  E2. Password is not correct.  E3. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. User know how to user a browser. 2. User enter the correct password. 3. User has an active internet connection. |

Table : UC 2 - Textual Description of Create User.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-2 |
| **Use Case Name:** | Create User |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to add users to the organization. |
| **Trigger:** | When the user presses “Create” button. |
| **Preconditions:** | PRE-1. The manager fill in all the input fields.  PRE-2. The manager uploads the picture. |
| **Postconditions:** | POST-1 The user is registered in the system. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the create new user page. 3. Manager fill in the input field. 4. The manager presses the Save button. 5. The input fields are validated. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. Email already exists..  E2. The API backend is down. |
| **Business Rules** | An email address can only be associated with a single user. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 3 - Textual Description of Edit User.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3 |
| **Use Case Name:** | Edit User |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to edit user. |
| **Trigger:** | When the user presses “Update” button. |
| **Preconditions:** | PRE-1. The manager updates the input fields. |
| **Postconditions:** | POST-1 The user data is updated in the system. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of users. 3. Manager select the user to be updated. 4. Manager fills in the input field. 5. The manager presses the Save button. 6. The input fields are validated. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. Email already exists..  E2. The API backend is down. |
| **Business Rules** | An email address can only be associated with a single user. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 4 - Textual Description of View User.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4 |
| **Use Case Name:** | View User |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to view user’s details. |
| **Trigger:** | When the user presses “Details” button. |
| **Preconditions:** | PRE-1. The manager is logged in.  PRE-2. The manager select the user. |
| **Postconditions:** | POST-1 The user details are displayed |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of users page. 3. Manager selects a particular user. 4. The manager presses the Details button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | 1. Manager logs in to the application. 2. Manager click the details from the attendance table on the dashboard. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 5 - Textual Description of Edit Profile.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5 |
| **Use Case Name:** | Edit Profile |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | User | |
| **Description:** | This will allow the user to edit his profile. |
| **Trigger:** | When the user presses “Save” button. |
| **Preconditions:** | PRE-1. The user fill in all the input fields.  PRE-2. The user uploads the picture. |
| **Postconditions:** | POST-1 The users profile is updated in the system. |
| **Normal Flow:** | 1. User logs in to the application. 2. User open the edit profile. 3. User updates the input field. 4. The user presses the Save button. 5. The input fields are validated. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. User is logged in to the system 2. User has an active internet connection. |

Table : UC 6 - Textual Description of Issue RFID.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6 |
| **Use Case Name:** | Issue RFID |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to assign RFID cards to the users. |
| **Trigger:** | When the manager presses “Save” button. |
| **Preconditions:** | PRE-1. The manager fills in the RFID unique id.  PRE-2. The manager uploads the picture. |
| **Postconditions:** | POST-1 The user is registered in the system. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the create new user page. 3. Manager fill in the input field. 4. The manager presses the Save button. 5. The input fields are validated. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. RFID already exists..  E2. The API backend is down. |
| **Business Rules** | An email address can only be associated with a single user. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 7 - Textual Description of Revoke RFID.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7 |
| **Use Case Name:** | Revoke RFID |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to revoke RFID card for a particular users. |
| **Trigger:** | When the manager presses “Revoke” button. |
| **Preconditions:** | PRE-1. The manager selects the employee.  PRE-2. The manager presses the button “Revoke RFID” |
| **Postconditions:** | POST-1 The user’s RFID cards access will be revokes in the system. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the create new user page. 3. Manager fill in the input field. 4. The manager presses the Save button. 5. The input fields are validated. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | 1. Manager opens the List of RFID cards. 2. Manager Searches for the RFID unique ID. 3. Manager presses “Revoke access” button. |
| **Exceptions:** | E1. The user enters the wrong RFID unique id.  E2. The API backend is down. |
| **Business Rules** | A manager shall revoke the access of the RFID card when the employee is fired or resigns from the position. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 8 - Textual Description of View Video feed.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8 |
| **Use Case Name:** | View Video Feed |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to view the video footage of each camera. |
| **Trigger:** | When the manager presses “View Video” button. |
| **Preconditions:** | PRE-1. The manager is logged in.  PRE-2. The manager selects the room.  PRE-3. The manager selects the camera. |
| **Postconditions:** | POST-1 The manager can start viewing the video stream.  POST-2 The manager can rewind the video. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the camera feed page. 3. Manager select the room. 4. Manager selects the camera. 5. Manager presses the “View Video” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The cloud hosting storage is full.  E2. The API backend is down. |
| **Business Rules** | A manager shall be able to review the video feed. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 9 - Textual Description of Delete User.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-9 |
| **Use Case Name:** | Delete User |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to delete a user from the organization. |
| **Trigger:** | When the manager presses “Delete” button. |
| **Preconditions:** | PRE-1. The manager is logged in to the system.  PRE-2. The manager selects the user from the list. |
| **Postconditions:** | POST-1 The user is deleted from the system. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of users. 3. Manager selects the user. 4. Manager presses “delete” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | The user can be deleted by the manager anytime. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 10 - Textual Description of Setup Geofencing.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-10 |
| **Use Case Name:** | Setup geofencing |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to setup geofencing for the rooms. |
| **Trigger:** | When the manager presses “Save” button. |
| **Preconditions:** | PRE-1. The manager is logged in to the system.  PRE-2. The rooms are created by the admin.  PRE-3. The manger selects the roles allowed for a room. |
| **Postconditions:** | POST-1 The roles are allowed for a particular room. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of rooms. 3. Manager enables the geofencing. 4. Manager add the appropriate roles. 5. Manager presses “save” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down.  E2. No rooms have been created.  E3. No roles have been created.  E5. The camera in the room in not working.  E6. The RFID in the room in not working. |
| **Business Rules** | The manager shall be able to setup the geofencing for the room. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. 3. Camera is installed in the room. 4. RFID is installed in the room. |

Table : UC 11 - Textual Description of Edit Geofencing.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-11 |
| **Use Case Name:** | Edit geofencing |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to edit the geofencing. |
| **Trigger:** | When the manager presses “Update” button. |
| **Preconditions:** | PRE-1. The manager is logged in to the system.  PRE-2. The manager selects the room from the list. |
| **Postconditions:** | POST-1 The geofencing settings for the room are updated. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of rooms. 3. Manager selects the room. 4. Manager updates the fields 5. Manager presses “update” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | The manager shall be able to edit the geofencing. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 12 - Textual Description of Delete User.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-12 |
| **Use Case Name:** | View Geofencing |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to view the geofencing. |
| **Trigger:** | When the manager presses “View” button. |
| **Preconditions:** | PRE-1. The manager is logged in to the system.  PRE-2. The manager selects the room from the list. |
| **Postconditions:** | POST-1 The manager can view the details of the room. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of room. 3. Manager selects the room. 4. Manager presses “view” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 13 - Textual Description of Forgot Password.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-13 |
| **Use Case Name:** | Forgot Password |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | User | |
| **Description:** | This will allow the user to reset his password. |
| **Trigger:** | When the user presses “Reset” button. |
| **Preconditions:** | PRE-1. The user enters the email adddress.  PRE-2. The user enter the valid OTP. |
| **Postconditions:** | POST-1 The user can enter new password. |
| **Normal Flow:** | 1. User opens the password reset page. 2. User enters the valid email address. 3. User enter the OTP sent to the email address. 4. System validates the OTP. 5. User enters the new password. 6. User presses “save” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | 1. User emails the admin with registered email. 2. Admin verifies the user. 3. Admin resets the password. 4. Admin send the new login link. |
| **Exceptions:** | E1. The API backend is down.  E2. The email is not confirmed. |
| **Business Rules** | None |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 14 - Textual Description of Search Employee.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-14 |
| **Use Case Name:** | Search Employee |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to search the employees real-time location. |
| **Trigger:** | When the manager presses “Search” button. |
| **Preconditions:** | PRE-1. The manager is logged in to the system.  PRE-2. The manager selects the user from the list. |
| **Postconditions:** | POST-1 The manager can view the locations of the employee |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the search form. 3. Manager enters the employee related information. 4. Manager presses “Search” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | 1. Manager logs in to the application. 2. Manager opens the search form. 3. Manager selects the image file. 4. Manager presses “Search” button. |
| **Exceptions:** | E1. The employee is not present today.  E2. The API backend is not running.  E3. Irrelevant information was used to search the employee. |
| **Business Rules** | The system shall enable the manager to search the user within the facility. |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 15 - Textual Description of Recent location.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-15 |
| **Use Case Name:** | Recent Location |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager to view the recent location of every user stored in the database. |
| **Trigger:** | When the manager presses “View” button. |
| **Preconditions:** | PRE-1. The manager is logged in to the system.  PRE-2. The manager selects the user from the list. |
| **Postconditions:** | POST-1 The manager can view the location of the user. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of users. 3. Manager selects the user. 4. Manager presses “view” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 16 - Textual Description of Generate Report.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-16 |
| **Use Case Name:** | Generate Report |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Manager | |
| **Description:** | This will allow the manager generate reports by applying various filters. |
| **Trigger:** | When the manager presses “Generate” button. |
| **Preconditions:** | PRE-1. The manager is logged in.  PRE-2. The manager applies the filters. |
| **Postconditions:** | POST-1 The manager can view the generated report. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the report generation. 3. Manager applies the filters.. 4. Manager presses “Generate” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Manager is logged in to the system 2. Manager has an active internet connection. |

Table : UC 17 - Textual Description of Create Managers.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-17 |
| **Use Case Name:** | Create Managers |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to create managers. |
| **Trigger:** | When the manager presses “Create” button. |
| **Preconditions:** | PRE-1. The admin is logged in to the system.  PRE-2. The admin fill in the valid input fields. |
| **Postconditions:** | POST-1 The admin has added a new managers. |
| **Normal Flow:** | 1. Admin open the form. 2. Admin fill the input fields with valid information. 3. Admin presses “Create” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down.  E2. The email already exists. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 18 - Textual Description of Add Managers.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-18 |
| **Use Case Name:** | Add Managers |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to assign managers to the organization. |
| **Trigger:** | When the admin presses “Assign” button. |
| **Preconditions:** | PRE-1. The manager selects the manager  PRE-2. The manager selects the organization |
| **Postconditions:** | POST-1 The manager can view the location of the user. |
| **Normal Flow:** | 1. Manager logs in to the application. 2. Manager open the list of managers. 3. Manager selects the manager. 4. Manager assigns the organization 5. Manager presses “Assign” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Manager is registered in the system 2. Admin has an active internet connection. |

Table : UC 19 - Textual Description of Create Organization.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-19 |
| **Use Case Name:** | Create Organization |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to create organization. |
| **Trigger:** | When the admin presses “Create” button. |
| **Preconditions:** | PRE-1. The admin fill in the valid input fields. |
| **Postconditions:** | POST-1 The admin has added a new organization. |
| **Normal Flow:** | 1. Admin open the form. 2. Admin fill the input fields with valid information. 3. Admin presses “Create” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down.  E2. The organization already exists. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 20 - Textual Description of Edit Manager.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-20 |
| **Use Case Name:** | Edit Manager |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to edit manager. |
| **Trigger:** | When the admin presses “Update” button. |
| **Preconditions:** | PRE-1. The admin updates the input fields. |
| **Postconditions:** | POST-1 The manager data is updated in the system. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of manager. 3. Admin select the manager to be updated. 4. Admin fills in the input field. 5. The admin presses the Save button. 6. The input fields are validated. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. Email already exists..  E2. The API backend is down. |
| **Business Rules** | An email address can only be associated with a single user. |
| **Assumptions:** | 1, Admin is logged in to the system  2, Admin has an active internet connection. |

Table : UC 21 - Textual Description of Edit Organization.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-21 |
| **Use Case Name:** | Edit Organization |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the manager to edit organization. |
| **Trigger:** | When the user presses “Update” button. |
| **Preconditions:** | PRE-1. The admin updates the input fields. |
| **Postconditions:** | POST-1 The organization data is updated in the system. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of orgnaization. 3. Admin select the orgnaization to be updated. 4. Amdin fills in the input field. 5. The admin presses the Save button. 6. The input fields are validated. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. Organization already exists..  E2. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 22 - Textual Description of View Organization.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-22 |
| **Use Case Name:** | View Organization |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to view organization’s details. |
| **Trigger:** | When the admin presses “Details” button. |
| **Preconditions:** | PRE-1. The admin select the organization. |
| **Postconditions:** | POST-1 The organization details are displayed |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of organization page. 3. Admin selects a particular organization. 4. The admin presses the Details button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 23 - Textual Description of Delete Organization.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-23 |
| **Use Case Name:** | Delete Organization |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to delete an organization. |
| **Trigger:** | When the admin presses “Delete” button. |
| **Preconditions:** | PRE-1. The admin selects the organization from the list. |
| **Postconditions:** | POST-1 The organization is deleted from the system. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of organization. 3. Admin selects the oranization. 4. Admin presses “delete” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | The organization can be deleted by the manager anytime. |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 24 - Textual Description of Delete Video Feed.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-24 |
| **Use Case Name:** | Delete Video Feed |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to delete a video feed from the cloud. |
| **Trigger:** | When the admin presses “Delete” button. |
| **Preconditions:** | PRE-1. The admin selects the video from the list. |
| **Postconditions:** | POST-1 The video is deleted from the cloud. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of videos. 3. Admin selects the video. 4. Admin presses “delete” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 25 - Textual Description of Delete Room.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-25 |
| **Use Case Name:** | Delete Room |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to delete a room from the organization. |
| **Trigger:** | When the admin presses “Delete” button. |
| **Preconditions:** | PRE-1. The admin selects the room from the list. |
| **Postconditions:** | POST-1 The room is deleted from the system. |
| **Normal Flow:** | 1. admin logs in to the application. 2. admin open the list of rooms. 3. admin selects the room. 4. admin presses “delete” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 26 - Textual Description of Delete Manager.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-26 |
| **Use Case Name:** | Delete Manager |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to delete a manager from the organization. |
| **Trigger:** | When the admin presses “Delete” button. |
| **Preconditions:** | PRE-1. The admin selects the manager from the list. |
| **Postconditions:** | POST-1 The manager is deleted from the system. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of manager. 3. Admin selects the manager. 4. Admin presses “delete” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 27 - Textual Description of Remove Camera.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-27 |
| **Use Case Name:** | Remove Camera |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to delete a camera from the room. |
| **Trigger:** | When the admin presses “Delete” button. |
| **Preconditions:** | PRE-1. The admin selects the camera from the list in the room. |
| **Postconditions:** | POST-1 The camera is deleted from the system. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of camera. 3. Admin selects the camera. 4. Admin presses “delete” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 28 - Textual Description of Adding RFID Sensor.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-28 |
| **Use Case Name:** | Adding RFID Sensor |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to add a RFID sensor to the room. |
| **Trigger:** | When the admin presses “Add” button. |
| **Preconditions:** | PRE-1. The admin configures the RFID |
| **Postconditions:** | POST-1 The RFID sensor is added to the room. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin opens the RFID add form. 3. Admin configures the RFID sensor. 4. Admin presses “Create” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 29 - Textual Description of Remove RFID.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-29 |
| **Use Case Name:** | Remove RFID |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to delete a RFID from the room. |
| **Trigger:** | When the admin presses “Delete” button. |
| **Preconditions:** | PRE-1. The admin selects the RFID from the list in the room. |
| **Postconditions:** | POST-1 The RFID sensor is deleted from the system. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin open the list of RFID sensors. 3. Admin selects the RFID sensor. 4. Admin presses “delete” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 30 - Textual Description of Adding Camera.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-30 |
| **Use Case Name:** | Adding Camera |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to add a Camera to the room. |
| **Trigger:** | When the admin presses “Add” button. |
| **Preconditions:** | PRE-1. The admin configures the Camera |
| **Postconditions:** | POST-1 The Camera is added to the room. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin opens the Camera add form. 3. Admin configures the Camera feed. 4. Admin presses “Create” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 31 - Textual Description of Creating Room.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-31 |
| **Use Case Name:** | Creating Room |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to add a room to the organization |
| **Trigger:** | When the admin presses “Add” button. |
| **Preconditions:** | PRE-1. The admin configures the Room |
| **Postconditions:** | POST-1 The room is added to the organization. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin opens the add room form. 3. Admin fills the input fields. 4. Admin presses “Create” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 32 - Textual Description of View Room.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-32 |
| **Use Case Name:** | View Room |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Admin | |
| **Description:** | This will allow the admin to view the room and its components |
| **Trigger:** | When the admin presses “View” button. |
| **Preconditions:** | PRE-1. The admin opens the room page |
| **Postconditions:** | POST-1 The admin can view the room details. |
| **Normal Flow:** | 1. Admin logs in to the application. 2. Admin opens the room list page. 3. Admin select the room. 4. Admin presses “View” button. |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Admin is logged in to the system 2. Admin has an active internet connection. |

Table : UC 33 - Textual Description of Preview Attendance.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-33 |
| **Use Case Name:** | Preview Attendance |
| **Actors:** | |  |  | | --- | --- | | Primary Actor: | Employee | |
| **Description:** | This will allow the employee to preview his attendance. |
| **Trigger:** | When the employee enter the attendance tab. |
| **Preconditions:** | PRE-1. The employee is allowed to access the portal. |
| **Postconditions:** | POST-1 The employee can see the attendance report. |
| **Normal Flow:** | 1. Employee logs in to the application. 2. Employee opens the dashboard. 3. Employee opens the report |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | None. |
| **Exceptions:** | E1. The API backend is down. |
| **Business Rules** | None |
| **Assumptions:** | 1. Employee is logged in to the system 2. Employee has an active internet connection. |

# Functional Requirements

Following are the functional requirements for our proposed solution.

## Functional Requirement 1

Table : Description of FR-1: Check in.

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | Check in |
| **Requirement** | The system shall allow the user to automatically check in to the system when they enter the facility. |
| **Source** | Supervisor |
| **Rationale** | To maintain the record of employee attendance. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 2

Table : Description of FR-2: Check out.

|  |  |
| --- | --- |
| **Identifier** | FR-2 |
| **Title** | Check out |
| **Requirement** | The system shall allow the user to automatically check out from the system when they enter the facility. |
| **Source** | Supervisor |
| **Rationale** | To maintain the record of employee attendance. |
| **Business Rule (if required)** | None |
| **Dependencies** | FR-1 |
| **Priority** | High |

## Functional Requirement 3

Table : Description of FR-3: Modify Attendance.

|  |  |
| --- | --- |
| **Identifier** | FR-3 |
| **Title** | Modify Attendance |
| **Requirement** | The system shall not allow anyone other than admin to modify the attendance. |
| **Source** | Supervisor |
| **Rationale** | To maintain the record of employee attendance. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 4

Table : Description of FR-4: Check Attendance.

|  |  |
| --- | --- |
| **Identifier** | FR-4 |
| **Title** | Check attendance |
| **Requirement** | The system shall allow the employee to check his attendance report. |
| **Source** | Supervisor |
| **Rationale** | To maintain the record of employee attendance. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | Medium |

## Functional Requirement 5

Table : Description of FR-5: Report Generation.

|  |  |
| --- | --- |
| **Identifier** | FR-5 |
| **Title** | Report Generation |
| **Requirement** | The system shall allow the manager to generate report of the attendance. |
| **Source** | Supervisor |
| **Rationale** | To maintain the record of employee attendance. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | Medium |

## Functional Requirement 6

Table : Description of FR-6: Monthly Report.

|  |  |
| --- | --- |
| **Identifier** | FR-6 |
| **Title** | Monthly Report |
| **Requirement** | The system shall automatically generate an attendance report at the end of month and send it to the manager |
| **Source** | Supervisor |
| **Rationale** | To maintain the record of employee attendance. |
| **Business Rule (if required)** | None |
| **Dependencies** | FR-1 |
| **Priority** | High |

## Functional Requirement 7

Table : Description of FR-7: Create Organization.

|  |  |
| --- | --- |
| **Identifier** | FR-7 |
| **Title** | Create Organization |
| **Requirement** | The system shall allow the admin to create organization |
| **Source** | Supervisor |
| **Rationale** | To have separate tenant. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 8

Table : Description of FR-8: Edit Organization.

|  |  |
| --- | --- |
| **Identifier** | FR-8 |
| **Title** | Edit Organization |
| **Requirement** | The system shall allow the admin to edit organization |
| **Source** | Supervisor |
| **Rationale** | To have separate tenant. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 9

Table : Description of FR-9: View Organization.

|  |  |
| --- | --- |
| **Identifier** | FR-9 |
| **Title** | View Organization |
| **Requirement** | The system shall allow the admin to view organization |
| **Source** | Supervisor |
| **Rationale** | To have separate tenant. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 10

Table : Description of FR-10: Delete Organization.

|  |  |
| --- | --- |
| **Identifier** | FR-10 |
| **Title** | Delete Organization |
| **Requirement** | The system shall allow the admin to delete organization |
| **Source** | Supervisor |
| **Rationale** | To have separate tenant. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirment 11

Table : Description of FR-11: Create Room.

|  |  |
| --- | --- |
| **Identifier** | FR-11 |
| **Title** | Create Room |
| **Requirement** | The system shall allow the admin to create room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 12

Table : Description of FR-12: Edit Room.

|  |  |
| --- | --- |
| **Identifier** | FR-11 |
| **Title** | Edit Room |
| **Requirement** | The system shall allow the admin to edit room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 13

Table : Description of FR-13: Create Room

|  |  |
| --- | --- |
| **Identifier** | FR-13 |
| **Title** | Create Room |
| **Requirement** | The system shall allow the admin to view room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 14

Table : Description of FR-14: Delete Room.

|  |  |
| --- | --- |
| **Identifier** | FR-14 |
| **Title** | Delete Room |
| **Requirement** | The system shall allow the admin to delete room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 15

Table : Description of FR-15: Add Camera.

|  |  |
| --- | --- |
| **Identifier** | FR-15 |
| **Title** | Add Camera |
| **Requirement** | The system shall allow the admin add camera to room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 16

Table : Description of FR-16: Remove Camera.

|  |  |
| --- | --- |
| **Identifier** | FR-16 |
| **Title** | Remove Camera |
| **Requirement** | The system shall allow the admin remove camera from room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 17

Table : Description of FR-17: Add RFID Sensor.

|  |  |
| --- | --- |
| **Identifier** | FR-17 |
| **Title** | Add RFID sensor |
| **Requirement** | The system shall allow the admin add RFID sensor from room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 18

Table : Description of FR-18: Remove RFID Sensor.

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | Remove RFID sensor |
| **Requirement** | The system shall allow the admin remove rfid sensor from room |
| **Source** | Supervisor |
| **Rationale** | To have separate rooms in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 19

Table : Description of FR-19: View Camera Feed.

|  |  |
| --- | --- |
| **Identifier** | FR-19 |
| **Title** | View Camera Feed |
| **Requirement** | The system shall allow the admin and manager to view the camera feed. |
| **Source** | Supervisor |
| **Rationale** | To have a record. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 20

Table : Description of FR-20: Delete Camera Feed.

|  |  |
| --- | --- |
| **Identifier** | FR-20 |
| **Title** | Delete Camera Feed |
| **Requirement** | The system shall allow the admin delete the camera feed. |
| **Source** | Supervisor |
| **Rationale** | To have a space on the storage. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 21

Table : Description of FR-21: Login

|  |  |
| --- | --- |
| **Identifier** | FR-21 |
| **Title** | Login |
| **Requirement** | The user shall be able to login to the system. |
| **Source** | Supervisor |
| **Rationale** | Authentication |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 22

Table : Description of FR-22: Edit Profile.

|  |  |
| --- | --- |
| **Identifier** | FR-22 |
| **Title** | Edit Profile |
| **Requirement** | The system shall allow the user to edit their profile |
| **Source** | Supervisor |
| **Rationale** | Profile management |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 23

Table : Description of FR-23: Create Users.

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | Create Users |
| **Requirement** | The system shall allow the manager to create accounts for employees |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 24

Table : Description of FR-24: Edit Users.

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | Edit Users |
| **Requirement** | The system shall allow the manager to edit accounts for employees |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 25

Table : Description of FR-23: View Users.

|  |  |
| --- | --- |
| **Identifier** | FR-25 |
| **Title** | View Users |
| **Requirement** | The system shall allow the manager to view accounts for employees |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 26

Table : Description of FR-26: Delete Users.

|  |  |
| --- | --- |
| **Identifier** | FR-26 |
| **Title** | Delete Users |
| **Requirement** | The system shall allow the manager to delete accounts for employees |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 27

Table : Description of FR-23: Issue RFID Cards.

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | Issue RFID cards |
| **Requirement** | The system shall allow the manager to issue a single card to each user. |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 28

Table : Description of FR-28: Revoke RFID Cards.

|  |  |
| --- | --- |
| **Identifier** | FR-28 |
| **Title** | Revoke RFID cards |
| **Requirement** | The system shall allow the manager to revoke the RFID cards for employees |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 29

Table : Description of FR-29: Creating Geofencing.

|  |  |
| --- | --- |
| **Identifier** | FR-29 |
| **Title** | Creating Geofencing |
| **Requirement** | The system shall allow the manager to create geofencing for a single room by assigning a list of roles permissible to enter. |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 30

Table : Description of FR-30: Edit Geofencing.

|  |  |
| --- | --- |
| **Identifier** | FR-30 |
| **Title** | Edit Geofencing |
| **Requirement** | The system shall allow the manager to edit the geofencing |
| **Source** | Supervisor |
| **Rationale** | To have separate areas in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 31

Table : Description of FR-31: Delete Geofencing.

|  |  |
| --- | --- |
| **Identifier** | FR-31 |
| **Title** | Delete geofencing |
| **Requirement** | The system shall allow only the admin to remove the geofencing. |
| **Source** | Supervisor |
| **Rationale** | To have separate areas in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 32

Table : Description of FR-32: Search Employee.

|  |  |
| --- | --- |
| **Identifier** | FR-32 |
| **Title** | Search Employee |
| **Requirement** | The system shall allow the manager to search for employee real-time location with their name. |
| **Source** | Supervisor |
| **Rationale** | To have search the employee. |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 33

Table : Description of FR-33: Search Employee using Images.

|  |  |
| --- | --- |
| **Identifier** | FR-33 |
| **Title** | Search Employee using images |
| **Requirement** | The system shall allow the manager to search for employee using their images |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 34

Table : Description of FR-34: Alert Manager.

|  |  |
| --- | --- |
| **Identifier** | FR-34 |
| **Title** | Alert Manager |
| **Requirement** | The system shall notify the user using push notification about any trespassing of geofencing. |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

## Functional Requirement 35

Table : Description of FR-35: Check Attendance.

|  |  |
| --- | --- |
| **Identifier** | FR-35 |
| **Title** | Check Attendance |
| **Requirement** | The system shall allow the employees to check their attendance. |
| **Source** | Supervisor |
| **Rationale** | To have separate profiles in an organization |
| **Business Rule (if required)** | None |
| **Dependencies** |  |
| **Priority** | High |

# Non-Functional Requirements

## Usability

*USE-1: All the deletions made in the database should be soft deletion.*

*USE-2: The application should be able to assist atleast 500 people at a time.*

*USE-3: All the icons used should easily depict what they do.*

*USE-4: Buttons in the application should be big enough so that user can easily press them.* (Erikson, 2012)

## Performance

*PER-1: The system shall not take more than 10 seconds to detect a face.*

*PER-2: The system shall not take more than 10 seconds to detect RFID signal.*

*PER-3: The system shall not take more than 10 seconds assosicate found person with the database.*

# Project Gantt Chart

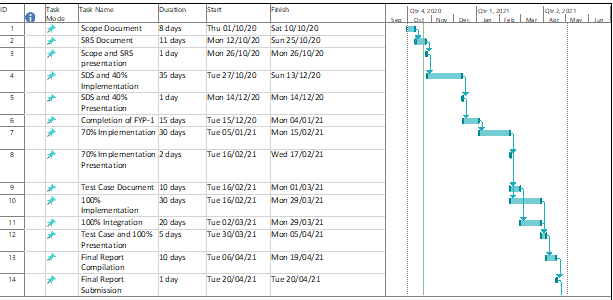


Figure : Gantt chart.

# References

1. Chu Luo, "Video Summarization for Object Tracking in the Internet of Things", *Next Generation Mobile Apps Services and Technologies (NGMAST) 2014 Eighth International Conference on*, pp. 288-293, 2014.
2. Ching-Sheng Wang, Chien-Liang Chen, "RFID-based and Kinect-based indoor positioning system", *Wireless Communications Vehicular Technology Information Theory and Aerospace & Electronic Systems (VITAE) 2014 4th International Conference on*, pp. 1-4, 2014.
3. Sommerville, I. (2011). *Software engineering* (9th ed.). Boston: Pearson.
4. Pressman, R. (2020). *Software engineering* (7th ed.). Boston, Mass.: McGraw Hill.
5. Meyer, B. *Object-oriented software construction*. London: Prentice-Hall International.
6. Mitchell, T. (2017). *Machine learning*. New York: McGraw Hill.

# Plagiarism Report