
Software Requirements Specification

for

surakshIIT

Version 0.1

Prepared by

Group Name: **GITHUB SPAMMERS**

Course: **CS253**

Mentor TA: **Mr. Aman Aryan**

Date: **30-01-2022**

Group Details

Name	Roll No.	Email ID
Aarchie	200004	aarchi20@iitk.ac.in
Arpit Kumar	200189	arpitk20@iitk.ac.in
Harsh Jain	200412	harshj20@iitk.ac.in
Kajal Deep	200483	kajald20@iitk.ac.in
Kuldeep Singh Chouhan	200530	kuldeepsc20@iitk.ac.in
Mannem Rishwan Reddy	200562	mannemr20@iitk.ac.in
Rohit	200813	rohitg20@iitk.ac.in
Sandeep Kumar Bijarnia	200856	sandeepb20@iitk.ac.in
Udit Prasad	201055	uditp20@iitk.ac.in
Yash Raj Mittal	201148	yashrajm20@iitk.ac.in

CONTENTS

REVISIONS

1	INTRODUCTION	1
1.1	PRODUCT SCOPE	1
1.2	INTENDED AUDIENCE AND DOCUMENT OVERVIEW	2
1.3	DEFINITIONS, ACRONYMS AND ABBREVIATIONS	2
1.4	DOCUMENT CONVENTIONS	2
1.5	REFERENCES AND ACKNOWLEDGMENTS	2
2	OVERALL DESCRIPTION	3
2.1	PRODUCT OVERVIEW	3
2.2	PRODUCT FUNCTIONALITY	4
2.3	DESIGN AND IMPLEMENTATION CONSTRAINTS	4
2.4	ASSUMPTIONS AND DEPENDENCIES	4
3	SPECIFIC REQUIREMENTS	6
3.1	EXTERNAL INTERFACE REQUIREMENTS	6
3.2	FUNCTIONAL REQUIREMENTS	9
3.3	USE CASE MODEL	10
4	OTHER NON-FUNCTIONAL REQUIREMENTS	22
4.1	PERFORMANCE REQUIREMENTS	22
4.2	SAFETY AND SECURITY REQUIREMENTS	22
4.3	SOFTWARE QUALITY ATTRIBUTES	23
5	OTHER REQUIREMENTS	24
	APPENDIX A – DATA DICTIONARY	25
	APPENDIX B - GROUP LOG	27

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Udit Prasad Aarchie Kajal Deep Mannem Rishwan Reddy Arpit Kumar Kuldeep Singh Chouhan	Initial Draft	30/01/22

1 Introduction

1.1 Product Scope

The purpose of this product is to provide the **IITK campus** community with a **digital portal** for security-related things. It will deal mainly with day to day functionalities like Campus and Hall level **entry** and **exit**, **Lost and Found** and related things.

As in the present, if someone arrives at the campus, they need to do paperwork, and then they are provided with a slip which they use while exiting the campus.

Something similar happens when we visit halls of the opposite gender.

This product will make these things smoother for both the security officials and the visitor/resident.

Also, we get many emails and messages when someone loses their items, which can also be resolved with the help of this product.

In short, the benefits of this system include digitisation of all these records instead of registers and having a paperless experience rather.

1.2 Intended Audience and Document Overview

The intended audience for this SRS are developers, students, professors, security officials, testers and documentation writers.

The rest of the document provides a general description of the system.

Section 2 gives a glimpse of the project's functions and a basic overview.

Section 3 includes the hardware, software requirements and functional requirements. This section also provides info about how users will interact with the system.

Section 4 contains the non-functional requirements and additional system requirements.

The sequence of reading the document:

- **Developers, Documentation writers and end-users (security officials and campus residents)**
They need to read the whole document. For better understanding, after reading 2.2, they should read 3.2 and 3.3.
- **Testers**
They should mainly read section 3 for getting a better idea of how the system works.

1.3 Definitions, Acronyms, and Abbreviation

CC	Computer Centre
DB	DataBase
IEEE	Institute of Electrical and Electronics Engineers
IITK	Indian Institute of Technology, Kanpur
ID	Identity
PF	Provident Fund
JS	JavaScript
MODEM	Modulator Demodulator
OTP	One Time Password
UI	User Interface
UIN	Unique Identification Number

1.4 Document Conventions

- Bolds are used for essential points/headings/subheadings.
- Italics are used for comments.
- IEEE standard formatting has been used while writing this document.

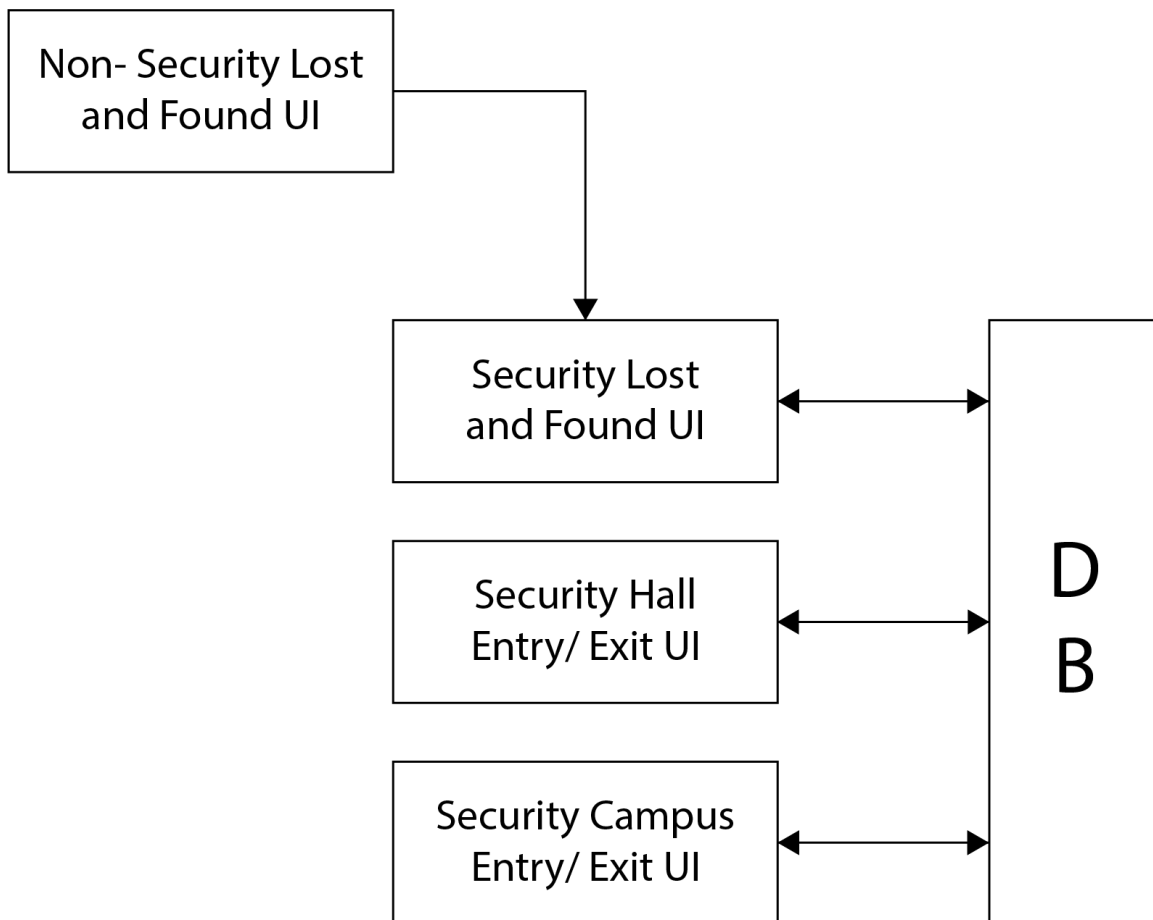
1.5 References and Acknowledgments

- IEEE 830-1998 standard for writing SRS Document.
- Security Section, IITK

2 Overall Description

2.1 Product Overview

- Our software will provide the stepping stones for digitising the interaction with the security team at IIT Kanpur. The software can be improved with regular updates.
- This software will smoothen the process of entry/exit for both residents and visitors of the campus, which is now based on registering on a register book, i.e. not a paperless transaction.
- These paperless entry/exit transactions will provide an easy and hassle-free movement to different halls and the main gate.
- Not having a dedicated Lost and Found Portal sometimes makes it challenging to report lost things and return found items to authentic owners. Our software will provide this platform with a simple to use UI.
- With our software, which will have a user-friendly interface, all the security measures can be handled in one place.
- Our software would be a **web-based application**, so any system which supports internet browsing will support **surakshIIT**.



2.2 Product Functionality

- **User Profile**

The profile will be created differently for different individuals. The amount of access provided will be in accordance with the individual's designation. For example, students can only use some features of the product, whereas a security employee can also access security-related features, the admin will have unrestricted access to the software.

- **Lost and Found portal**

Users can lodge a lost (found) complaint (details) which will be then approved by the security officers after this approval it would be displayed on the platform. Upon the resolution of the problem, the complaint would be removed from the system.

- **Hall entry/exit**

If a student wishes to enter the non-residence hostel, a security official will fill in their details in this portal and would be required to submit an id card to the security guard after its approval.

- **Campus entry/exit**

Any request regarding user entry/exit from campus can be approved using this portal.

- **Portal for paperless entry/exit for visitors**

Users can confirm with the security team if any of their visitors have reported to the main gate of the campus. Also, their details like Vehicle no/License and other basic information can be recorded here.

2.3 Design and Implementation Constraints

- Implement the database using a centralized database management system.
- Implement the backend of the system using C++.
- Implement the frontend of the system using React JS.
- The security office will be responsible for maintaining the delivered software.

2.4 Assumptions and Dependencies

- We have assumed all the security officials/ check posts will be provided with hardware/network connection and correct specifications to support our software.
- We have also assumed any of the users will not misuse the software.

- The third party we are planning to use is the database from the automation centre and the hall offices.
- All users must be able to operate the software according to the functionalities he/she can access.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- A new user will see an option to register herself using UIN, which would be Roll/ PF no/ any unique ID given by the campus administration.
- After successful registration, the user can log in using the credentials, i.e. UIN and password.
- After successful login, the dashboard will contain different options according to the user's role (security official or others)
- If a user is already registered and forgets the password, the portal shall show a chance to reset it after verifying through CC ID and password or OTP.

UIs of different options available are explained in the latter part of this subsection.

3.1.1.1 User profile

Here, the user can see their basic details like Name, Roll, email, contact, Address, notifications.

3.1.1.2 Lost and Found

- **Security officials:** Here, the user will see two columns, one for lost items and the other for the found ones. They can click to expand the thing to view all the details, and also, they will have the option to filter items.
- **Non-security officials:** Here, the user will see options to report any lost or found cases by simply clicking on the button and then a form will appear to fill in the information. Some fields will have a dropdown to select categories while some will be subjective fields

3.1.1.3 Hall entry and exit

- Only security officials will be able to use this option
- Here, a form will appear where the user can fill the UIN of the visiting person and after that, a picture of that person will appear which the user can match. After that, the user can allow her to enter/exit the hall.

3.1.1.4 Campus entry and exit

- This section will only be available for security personnel.

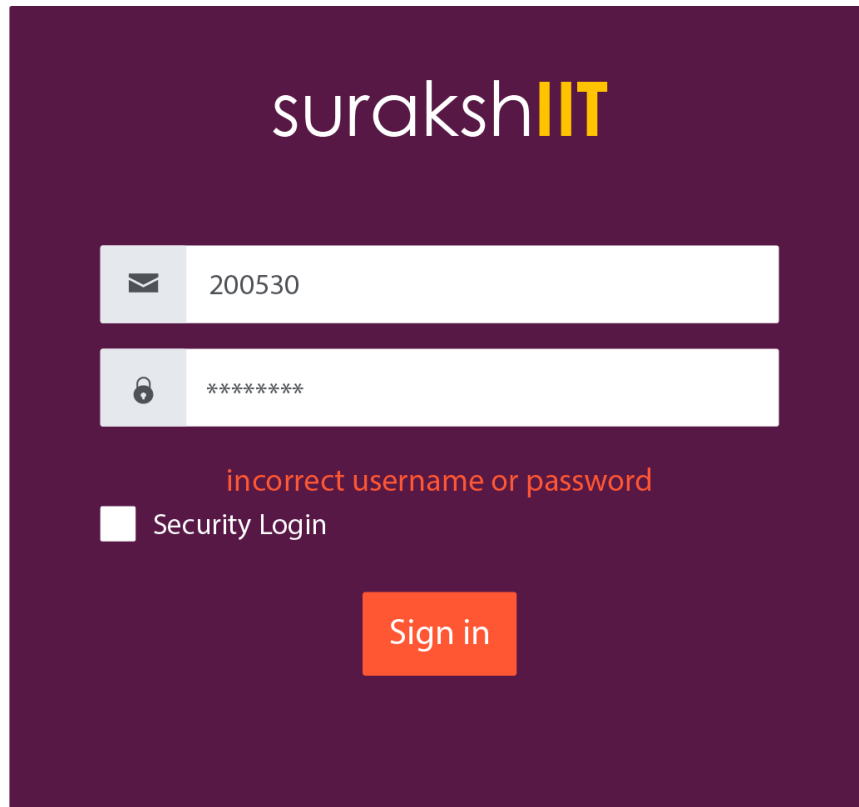
3.1.1.4.1 For Campus residents and daily basis travellers

Here also a form will appear where the user can fill the UIN of the visiting person if he/she is a resident/student of campus and then a picture of that person will appear with which the user can match and allow her to enter/exit the campus.

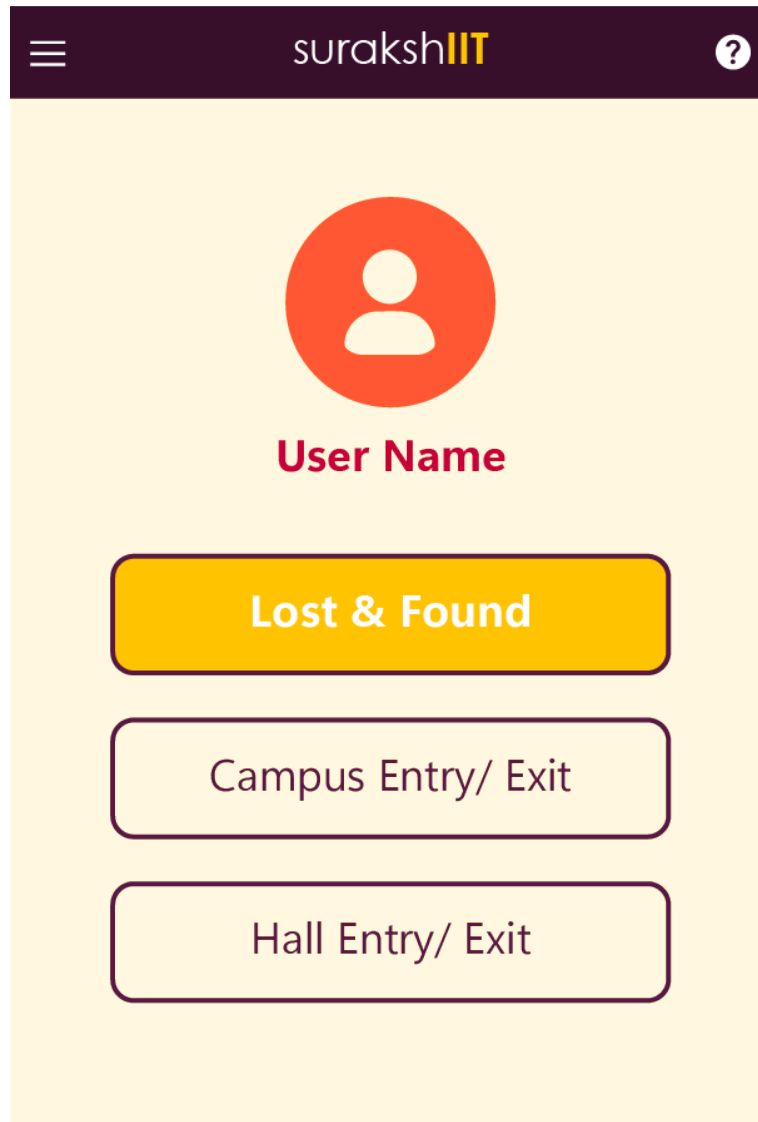
3.1.1.4.2 For visitors and other non-campus residents

- If the visiting person is someone else, the user will fill in all the details, and then she can send a confirmation message to the person who has called that person.
- Also, users can navigate to any other option from any page of the system.

Later part of this subsection has some graphics which will help user to get an rough idea of how the UI looks.



UI of going to be login page of the system



UI of going to be home page of the system

3.1.2 Hardware Interfaces

- Our system will interact through different hardware like mobile, desktop, laptop, etc
- As our system will interact with the internet for various functions, all hardware interfaces should be connected through the internet via any hardware like MODEM, Router, Ethernet cable, etc

3.1.3 Software Interfaces

- The system must be able to connect to the DB of all the campus residents
- The system must be able to connect to the DB of different hall residents.
- The system must be able to interact with the DB to keep records of what the system does.

- The system must be able to connect to a message/ email-server to send a message/ mail to the person whom someone has come to meet.

3.2 Functional Requirements

3.2.1 Lost and Found:

The system shall help users (non-security officials) to report any lost belongings with complete details like its picture, location where it was last seen, when it got lost, etc., along with their contact details which would be auto-filled after login. Also, it will have the option to cancel the lost query if found by the same person. The system shall help users (security officials) to maintain the 'found' database and also have access to the 'lost' database with the ability to remove the lost query after sorting the issue. These databases shall also have the option to filter according to colour or item categories.

3.2.2 Hall entry/exit:

The system shall help users (security officials) to maintain the DB of all those who are non-resident of that hall i.e., students of different halls and other workers containing their enter and exit timings.

For students, the user after entering the student's unique roll number shall fill up the details like where the student wants to visit (canteen or someone's room) and also the roll number of other resident students if they come to meet. The roll number entered shall give all details like their room number, hall number, etc.

For others, the user shall enter the UIN allotted.

By submitting, The system shall have a record with a timestamp.

3.2.3 Campus entry/exit:

The system shall help users i.e., security officials to maintain a database of everyone entering or exiting campus with timestamp gate wise.

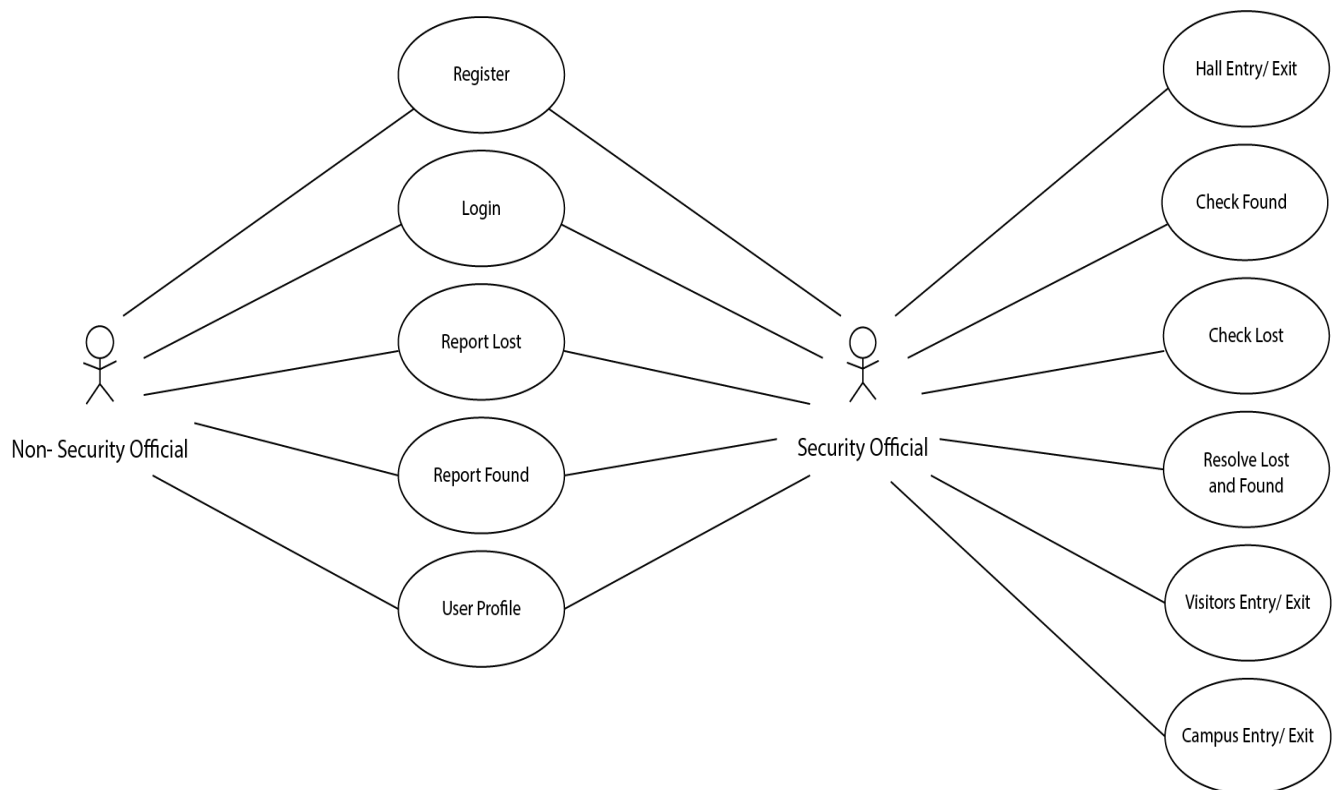
For residents (students or faculties or other staff): The user shall enter roll number, and auto-generated data containing phone number, campus address, etc., would be generated and stored in a database with a timestamp.

For daily workers or people with daily passes: The user shall enter the person's unique identification number and auto-generated data containing phone number, campus address, etc., would be generated and stored in a database with a timestamp.

3.2.4 Portal for paperless entry/exit for visitors

For other visitors without passes (like delivery persons, rental cabs or autos, visiting on purpose): The user shall fill in all the details like license, vehicle number, and model if came with the vehicle, the id card details, reason of the visit, visiting the location, details of the person (any resident) if came to meet or taking in charge. This will also generate the notification of confirmation to the person whomsoever it may concern.

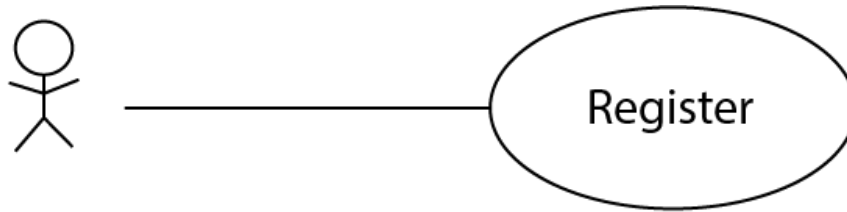
3.3 Use Case Model



Use-Case Diagram showcasing high-level working of surakshIIT

Later part of this subsection contains detailed information about different use cases

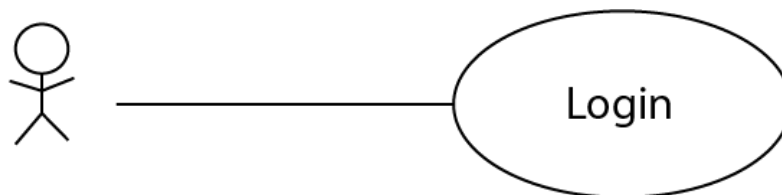
3.3.1 Use Case U1



New User

Use Case ID	U1
Use Case title	New User Registration
Author	Kajal
Purpose	Register the user for the first time
Requirements Traceability	UIN
Priority	High
Preconditions	UIN should be valid and available in database of IITK
Postconditions	User is able to Sign-Up
Actor	Guest users
Exceptions	<ul style="list-style-type: none"> • If the user is already registered, display the message - already registered • If the data entered is not available in the database, display error message
Includes	None
Note / Issues	None

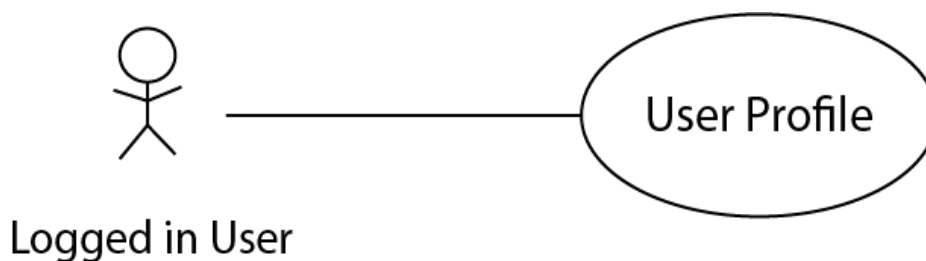
3.3.2 Use Case U2



Registered User

Use Case ID	U2
Use Case title	User Login
Author	Kajal
Purpose	Login to the portal
Requirements Traceability	<ul style="list-style-type: none"> • UIN • Password • User Type (Security / Non-Security)
Priority	High
Preconditions	User should be registered
Postconditions	User is able to login
Actors	Registered User
Exceptions	<ul style="list-style-type: none"> • If the user enters invalid password, he gets an option to change password by clicking on 'Forgot Password'
Includes	U1.2
Note / Issues	None

3.3.3



3.3.3.1 Use Case U3.1

Use Case ID	U3.1
Use Case title	User Profile (Security Users)

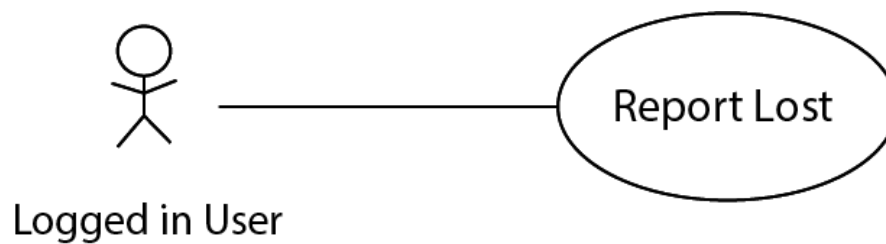
Author	Kajal
Purpose	To contain details of user (like name,phone number, address, email, programme enrolled) for portal and shows the options available for the security staff i.e, which section he wants to visit <ul style="list-style-type: none"> • Lost and Found • Hall Entry/Exit • Campus Entry/Exit
Requirements Traceability	None
Priority	High
Preconditions	User must be a Security staff
Postconditions	User can visit the desired section
Actors	Registered Security Staffs
Exceptions	None
Includes	U2
Note / Issues	None

3.3.3.2 Use Case U3.2

Use Case ID	U3.2
Use Case title	User Profile (Non-Security Users)
Author	Aarchie
Purpose	To contain details of user (like name,phone number, address, email, programme enrolled) for portal and has an option to report lost
Requirements Traceability	None
Priority	High
Preconditions	User should be a registered user
Postconditions	User is able to report about items lost

Actors	Registered Users
Exceptions	None
Includes	U2
Note / Issues	None

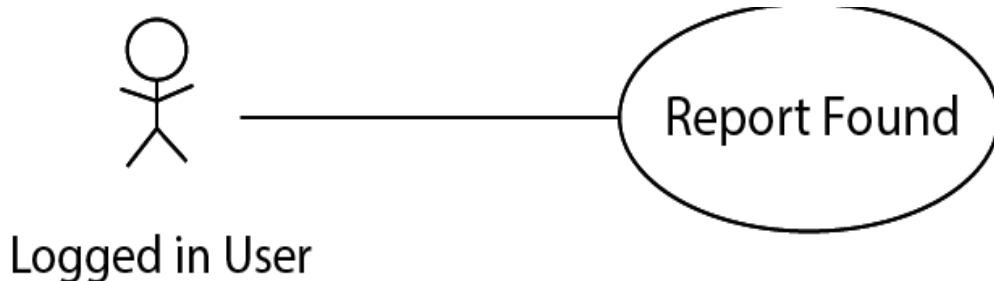
3.3.4.1 Use Case U4.1



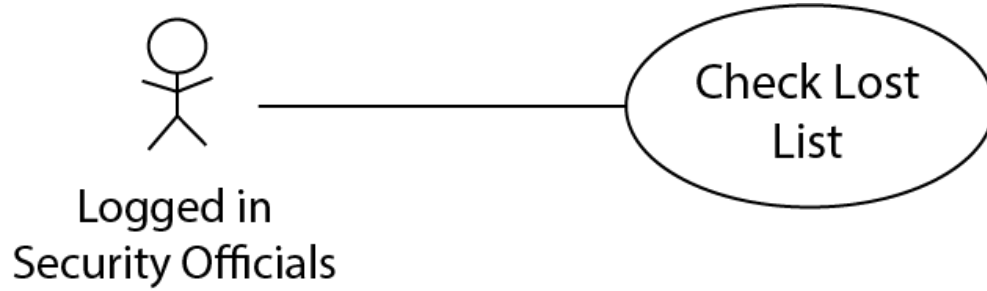
Use Case ID	U4.1
Use Case title	Report Lost
Author	Aarchie
Purpose	To file up query of lost belongings and also cancelling (if in case found it already)
Requirements Traceability	None
Priority	High
Preconditions	User has chosen the option to report about lost item
Postconditions	Data of lost item gets recorded in the lost list
Actors	Registered User
Exceptions	None
Includes	U3.1/ U3.2

Note / Issues	None
---------------	------

3.3.4.2 Use Case U4.2



Use Case ID	U4.2
Use Case title	Report Found
Author	Aarchie
Purpose	To file up query of found belongings
Requirements Traceability	None
Priority	Low
Preconditions	User has chosen the option to report about found item
Postconditions	Data of found item gets recorded in the lost list
Actors	Registered User
Exceptions	None
Includes	U3.1/ U3.2
Note / Issues	None

3.3.4.3 Use Case U4.3

Use Case ID	U4.3
Use Case title	Check Lost List
Author	Udit
Purpose	Navigate through the list of lost items
Requirements Traceability	None
Priority	High
Preconditions	None
Postconditions	None
Actors	Registered Security officials
Exceptions	None
Includes	U2
Note / Issues	None

3.3.4.4 Use Case U4.4

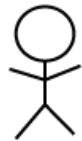


Check Found

Logged in
Security Officials

Use Case ID	U4.4
Use Case title	Check Found List
Author	Udit
Purpose	To fill up details of 'found' queries also sorting and removing the lost case if item present in 'lost' database
Requirements Traceability	None
Priority	High
Preconditions	None
Postconditions	Data of the item found is removed from the lost list
Actors	Registered Security officials
Exceptions	None
Includes	U2
Note / Issues	None

3.3.5.1 Use Case U5.1



Hall Entry

Logged in
Security Officials

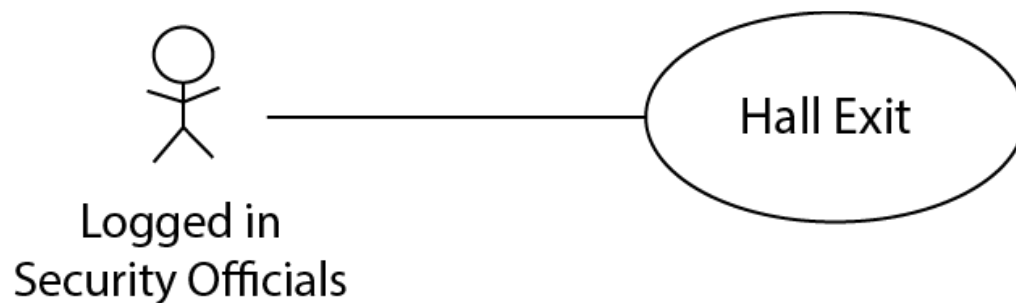
Use Case ID	U5.1
Use Case title	Hall Entry
Author	Kajal
Purpose	To fill the details of the person entering the hall (student from other halls / workers)
Requirements Traceability	<ul style="list-style-type: none"> • UIN
Priority	High
Preconditions	User should be registered
Postconditions	User can look for the details of the person entering the hall and validate it
Actors	Security officers
Exceptions	None
Includes	U2
Note / Issues	Entry time should be valid

3.3.5.2 Use Case U5.2

Use Case ID	U5.2
Use Case title	Hall entry user Profile
Author	Kajal
Purpose	Contains the details of the person entering the hall (student from other halls / workers)
Requirements Traceability	<ul style="list-style-type: none"> • UIN
Priority	High
Preconditions	User has entered the UIN of the person entering
Postconditions	Person's entry will be recorded

Actors	Security officers
Exceptions	None
Includes	U5.1
Note / Issues	None

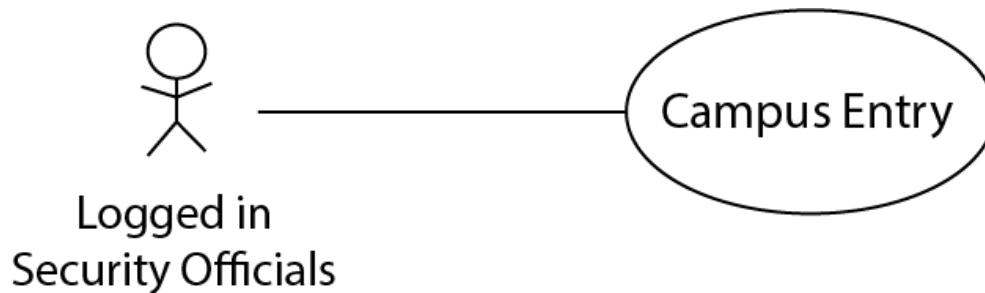
3.3.5.3 Use Case U5.3



Use Case ID	U5.3
Use Case title	Hall Exit
Author	Aarchie
Purpose	Entering the exit time and removing the data of the user exiting the hall
Requirements Traceability	<ul style="list-style-type: none"> • UIN
Priority	High
Preconditions	Record of the person exiting has been recorded in the entry list
Postconditions	Data of the person is updated with exit records
Actors	Security Officers
Exceptions	None
Includes	U5.2

Note / Issues	None
---------------	------

3.3.6.1 Use Case U6.1



Use Case ID	U6.1
Use Case title	Campus Entry
Author	Udit
Purpose	To enter the record of the person entering with timestamp associated with unique identification number
Requirements Traceability	<ul style="list-style-type: none"> • UIN in case of residents/daily pass holders • Name, Vehicle No. / Model, Driving License
Priority	High
Preconditions	User has logged into his account
Postconditions	The record of the person entering is recoded
Actors	Security Officers
Exceptions	None
Includes	U2
Note / Issues	Entry time should be valid

3.3.6.2 Use Case U6.2



Campus Exit

Logged in
Security Officials

Use Case ID	U6.2
Use Case title	Campus Exit
Author	Udit
Purpose	To enter the record of the person exiting with timestamp associated
Requirements Traceability	<ul style="list-style-type: none"> • UIN in case of residents/daily pass holders • Name in other cases
Priority	High
Preconditions	Record of the person exiting has been recorded
Postconditions	Data of the person is removed from the entry record
Actors	Security officers
Exceptions	None
Includes	U6.1
Note / Issues	None

4 Other Non-functional Requirements

4.1 Performance Requirements

Speed of Response: Being an interactive web-based application at worst there could be a delay time because of popping forms and sometimes due to error messages because of the wrong login. For saving details to the DB or for fetching them from DB, there could be a delay of much below 3 seconds. For saving a user's settings or sessions software should not take more keeping seconds. In worst cases, the software shall respond at a maximum of 4 seconds.

Execution Time: As a person enters/exits from a gate, the timestamp shall be recorded, and for it to be precise, the execution should be done in real-time. Lost and Found also need to be recorded along with the timestamp. So, it is necessary for our software to execute the task in real-time.

Storage:

For storage allocation, we would assume-

- it must be capable of storing 10,000 profiles, including students and other residents
- for lost and found, we expect 1000 entries at a time.
- for entry into opposite gender halls, we hope at max 400 students. Also, we expect to add 1500 entries for workers operating in the hall
- for temporary exit from campus, we expect around 2000 students at max.

We would allocate storage space for our software according to these predictions.

4.2 Safety and Security Requirements

- There would be three-tier of roles
 - **Admin** - will have all read, write, and executable power.
 - **Security officials**- will have read, write, and executable power over his hall (or area).
 - **User** - will have read (only for lost found list) and can upload a query for lost found and hall entry.
- Login will require a Unique Identification Number and a password. UIN will be the Roll number for students and the PF number for other residents. Security officials would be able to access all the DB.
- Confidentiality must be maintained, and under no circumstances should the user be allowed to look into the database. Admin and Security officials would be responsible for maintaining the confidentiality of their DB.

- Proper authentication from CC and UIN will make this system non-repudiable. OTPs would be sent to the registered email-IDs on CC to implement two-factor authentication.
- System shall not store any of the user's confidential information in any way.

4.3 Software Quality Attributes

- **Functional Suitability** - The should store queries aptly according to the function called, and there should always be output for every valid input and an error prompt for invalid input. Functional correctness, completeness, and appropriateness are must, for our software performs three independent tasks.
- **Compatibility** - The web-based interaction enhances compatibility over different devices. This also makes the software portable and modular. Different users have different operating systems on their devices, and they shall experience the same environment over different systems. It would be adaptable to other internet browsers.
- **Resource Utilization** - Major resources required by our software are just a good internet connection and a stable and reliable server, which smoothens and fastens the tasks.

5 Other Requirements

5.1 Compatibility Requirements

This system shall support different hardware, Operating systems and browsers which are in current use and are based on industry standards.

5.2 Training related Requirements

Security officials will be able to use this software with fundamental training and will not take more than some hours to get familiarised with the system.

5.3 Legal Requirements and Copyright

No special Legal Requirements exist.

Copyright of the source code and the documents must be retained by the developers of the system.

Appendix A – Data Dictionary

Variable name	Variable Definition	Data type	Other information
user_ID	Roll no in case of students and PF no or any other UIN given by IITK for others	String	-
role	Security officials and non-security officials will have different roles	Boolean	0 for non-security 1 for security
if_Registered	If a user is registered or not	Boolean	0 for not registered 1 for registered
lost_ID	unique number given to every lost entry	Number	-
found_ID	unique number assigned to every found entry	Number	-
item_category	Items will be divided into different categories to ease the process of resolving lost and found queries like Cards, Keys, etc	String	-
item_Brand	If an item has some brand associated, it can be stored here. It will also help in lost and found queries.	String	If item is not associated with any brand it will be empty string
item_Color	It will store the color of the lost/found item.	String	-
if_Solved	If a lost or found query is resolved or not	Boolean	0 for Unsolved 1 for Solved
hall_No	Every hall of Residence has a unique number	Numeric	-
gate_No	From which gate entry/exit has been done in case of campus entry/exit	Numeric	3 options for 3 different gates

address	It will store the address of all the registered users	String	-
entry_time	Entry time in case of hall or campus entry	Time and Date	-
exit_time	Exit time in case of hall/campus entry	Time and Date	-
to_meet	It will store the record of the person to whom someone has come to meet in hall/campus entry	Object	-

Appendix B - Group Log

Date	Activity
9-01-2022	Discussed various project ideas among the team.
11-01-2022	Discussed with the professor and finalised the idea.
12-01-2022	Pondered over the pre-existing system and methods, and how they can be improved.
20-01-2022	Decided to make the software in the form of a website.
24-01-2022	Met Mr. A K Sharma Ji, Security Officer to understand their requirements. Shared the idea with the TA.
25-01-2022	Conducted a survey among the students to realise the problems they face due to the existing system.
26-01-2022	Distributed work among ourselves and started documentation.
29-01-2022	Created the first draft of the document.
30-01-2022	Got the document reviewed by the TA.
31-01-2022	Updated the document.