



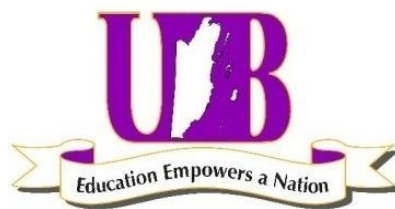
Software Engineering

CMPS4131

Tamika Chen, Raynisha Cornelio, Javier Castellanos, Abner Mencia, Alex Peraza

Report 1



Feb 28, 2024



## Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>Contribution Breakdown</b>	<b>3</b>
<b>Responsibility Matrix</b>	<b>4</b>
<b>Responsibility Allocation Chart</b>	<b>5</b>
<b>Customer Statement Requirements</b>	<b>6</b>
Problem Statement	6
Glossary of Terms	9
<b>System Requirements</b>	<b>10</b>
Enumerated Functional Requirements	10
Enumerated Nonfunctional Requirements	11
On Screen Appearance Requirements	12
<b>Functional Requirements Specification</b>	<b>14</b>
Stakeholders	14
Actors and Goals	14
Casual Description	17
Use Cases Diagram	20
Traceability Matrix	22
Fully Dressed Descriptions	23
System Sequence Diagram	1
<b>User Interface Specification</b>	<b>1</b>
Preliminary Designs (MOCKUPS)	1
User Effort Estimation	1
<b>System Architecture</b>	<b>1</b>
Architecture Style	1
Mapping Subsystem to Hardware	1
Connectors and Network Protocols	1
Global Control Flow	1
Hardware Requirements	1
<b>Plan of Work</b>	<b>1</b>
Gantt Chart	1

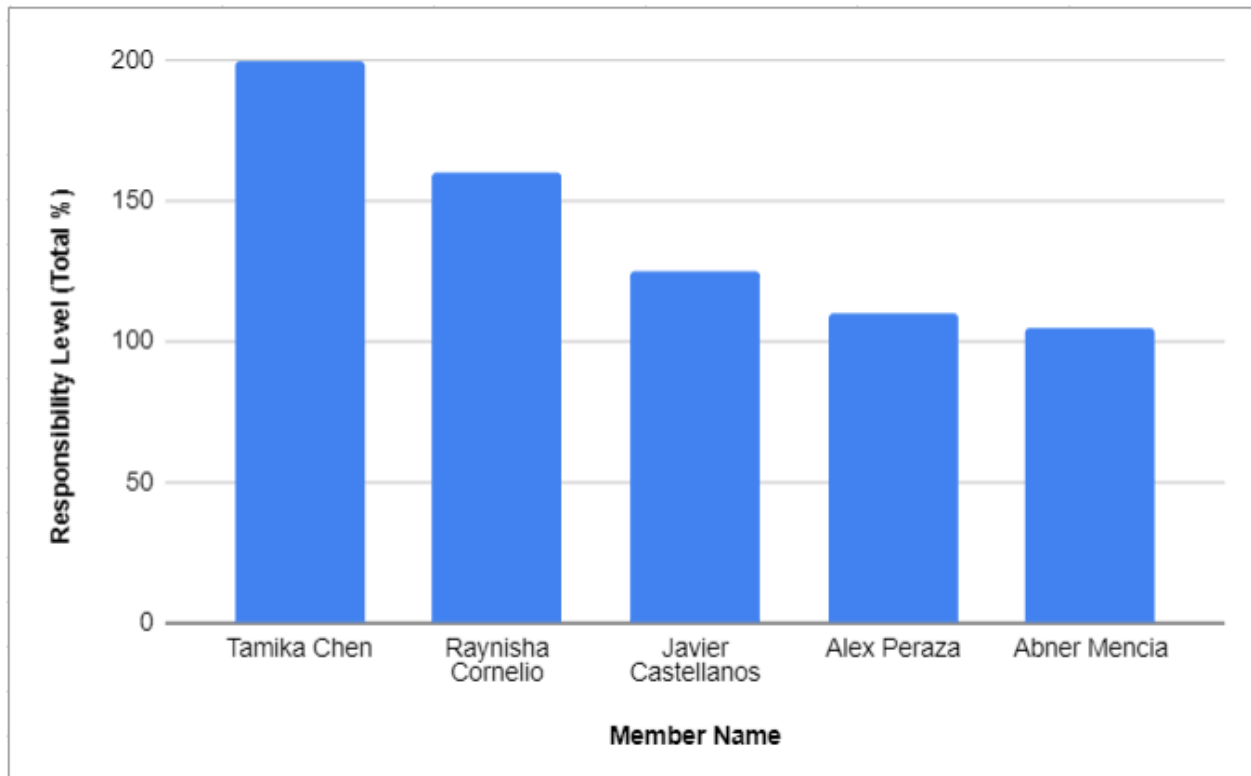
## Contribution Breakdown

Signature Block			
Statement	I did my share of the work, and I have a general understanding of the contents of the assignment.		
Team Member	Contribution	Signature	Date
Tamika Chen			28/02/24
Raynisha Cornelio			28/02/24
Javier Castellanos		Javier Castellanos	28/02/24
Abner Mencia	contributed to Casual Description, Traceability Matrix, Actors & Goals, Architecture Styles and Plan of work	Abner Mencia	28/02/24
Alex Peraza	Contributed to System Architecture, System Requirement, User Interface Specs, and Customer Statement of Requirements	alexPeraza	28/02/24

## Responsibility Matrix

Note: See point allocations in the table below.		Team Member Name				
		Tamika C	Raynisha C	Javier C	Abner M	Alex P
Responsibility	Project management (10 points)	50%	50%			
	Sec.1: Customer Statement of Requirements (9 points)	70%	20%	5%		5%
	Sec.2: System Requirements (6 points)	40%	20%	20%	10%	10%
	Sec.3: Functional Requirements Specification (30 points)		50%	50%		
	Sec.4: User Interface Specs (15 points)	40%	20%	35%		5%
	Sec.5: System Architecture (15 points)			15%	5%	80%
	Sec.6: Plan of Work (19 points)				90%	10%

## Responsibility Allocation Chart



## Customer Statement Requirements

### Problem Statement

The University of Belize grapples with a multifaceted challenge regarding the safety and security of its faculty, staff, and student body due to the absence of an integrated and user-friendly campus safety application. While the university employs certain safety precautions, they are not optimized to meet its community's dynamic and evolving safety needs. The existing measures lack the seamless ease of use, instantaneous information sharing, and personalized support that a comprehensive safety app could provide. Consequently, there is a growing demand among stakeholders for a centralized platform that enables swift responses to crises, disseminates pertinent safety information, and delivers timely notifications.

The current safety landscape at the University of Belize exhibits several deficiencies. Firstly, the absence of a centralized safety application hinders the efficient dissemination of critical safety information during emergencies. Without a unified platform, the communication channels become disjointed, leading to potential delays and confusion in emergency response efforts. Moreover, the lack of real-time alerts and location-based support further exacerbates the challenges of proactive threat mitigation and timely assistance to individuals in distress.

In response to these challenges, there is an urgent need for the development and implementation of a comprehensive campus safety app tailored to the specific needs of the University of Belize. The envisioned app should serve as a centralized hub for accessing essential safety services, including emergency contact information, crisis management protocols, and campus security resources. It should facilitate real-time communication and collaboration among stakeholders, enabling swift responses to safety threats and emergencies.

The campus safety application must provide real-time alerts and notifications to users concerning various safety incidents, weather advisories, and emergency protocols. This feature is pivotal in ensuring that individuals within the university community are promptly informed about potential threats or hazardous situations, allowing them to take necessary precautions or actions. By delivering timely alerts, the app enhances situational awareness and empowers users to make informed decisions in response to emergencies. Additionally, real-time alerts enable swift dissemination of critical information, facilitating coordinated response efforts and minimizing the impact of adverse events. Whether it's notifying users about safety hazards on campus, impending severe weather conditions, or outlining emergency procedures, the real-time alerts feature plays a crucial role in enhancing overall safety and security within the university.

The application should incorporate location-based support through the utilization of GPS technology. This feature enables users to accurately pinpoint their exact location in the event of

emergencies, facilitating swift and effective response from emergency responders. By leveraging GPS technology, the app enhances the ability of users to communicate their precise whereabouts, ensuring that assistance can be promptly dispatched to the relevant location. Moreover, location-based support empowers responders with critical information to assess the situation and provide appropriate assistance in a timely manner. Whether it's guiding individuals to safety during crises or coordinating rescue efforts in emergency scenarios, the integration of location-based support enhances the overall efficacy and responsiveness of the campus safety application, thereby bolstering the safety and security of the university community.

It should encompass comprehensive crisis management tools to aid users in navigating emergency situations effectively. This includes providing intuitive tools and resources such as emergency contact information, evacuation routes, and medical assistance guidelines. By offering accessible and user-friendly interfaces, the app equips individuals with the necessary information and protocols to respond swiftly and appropriately during crises. Emergency contact information ensures that users can quickly reach out to relevant authorities or support services in times of need, facilitating timely assistance and intervention. Additionally, the inclusion of evacuation routes enables users to navigate safely to designated assembly points or shelters during emergencies, minimizing potential risks and hazards. Moreover, the provision of medical assistance guidelines equips users with essential knowledge and procedures for administering first aid or seeking medical attention when faced with medical emergencies. In essence, the integration of crisis management tools within the campus safety application enhances preparedness, responsiveness, and overall safety within the university community.

The campus safety application should incorporate features that allow users to create personalized safety plans and profiles tailored to their individual needs and preferences. This includes the ability for users to input essential medical information, emergency contacts, and specific safety preferences within their profiles. By enabling users to customize their safety plans, the app empowers individuals to proactively prepare for potential emergencies and tailor responses based on their unique circumstances. Including medical information ensures that responders have access to critical health details in the event of medical emergencies, facilitating prompt and informed care. Moreover, the inclusion of emergency contacts allows users to quickly reach out to designated individuals for assistance or support during crises. Additionally, accommodating specific safety preferences ensures that users can adhere to their preferred safety protocols and procedures in various situations. Overall, the provision of customized safety plans enhances user preparedness, facilitates efficient emergency response, and contributes to a safer and more secure campus environment.

The campus safety application should seamlessly integrate with the existing array of campus security systems, encompassing surveillance cameras, access control points, and emergency call boxes. This integration ensures a cohesive and interconnected approach to campus security,

enabling the app to leverage the capabilities of these systems for enhanced monitoring, access control, and emergency response. By synchronizing with surveillance cameras, the app can provide real-time visual monitoring of campus activities, aiding in the detection and prevention of potential security threats. Integration with access control points enables the app to regulate entry and exit points across campus, bolstering security measures and mitigating unauthorized access. Moreover, integration with emergency call boxes facilitates swift and direct communication between users and emergency responders, expediting response times and ensuring timely assistance during emergencies. Overall, the seamless integration of the campus safety application with existing campus resources enhances the overall efficacy and responsiveness of the university's security infrastructure, fostering a safer and more secure environment for all members of the campus community.

In order to cultivate a culture of safety and collaboration within the university community, the campus safety application should incorporate a range of community engagement features. These features include anonymous tip reporting, safety forums, and peer support networks, all aimed at promoting active participation and communication among users. Anonymous tip reporting allows individuals to confidentially report safety concerns or suspicious activities, empowering users to contribute to campus safety without fear of reprisal. Safety forums provide a platform for open dialogue and discussion regarding safety-related topics, facilitating information sharing, awareness-raising, and community-driven initiatives. Additionally, peer support networks foster mutual assistance and solidarity among users, allowing individuals to seek and provide support to one another during times of need. By embracing community engagement features, the app not only strengthens bonds within the university community but also enhances collective efforts towards maintaining a safe and inclusive campus environment.

Ensuring accessibility and usability are paramount considerations for the campus safety application. It should be designed to accommodate all members of the university community, including individuals with disabilities, ensuring equitable access to its features and functionalities. This entails implementing features such as screen reader compatibility, alternative navigation options, and adjustable font sizes to cater to diverse needs and preferences. Moreover, the app should feature an intuitive interface characterized by clear navigation pathways, logical organization of information, and user-friendly controls. By prioritizing accessibility and usability, the app endeavors to empower all users to effectively utilize its resources, engage in safety protocols, and contribute to a safer campus environment.

By developing and deploying a robust campus safety app, the University of Belize aims to enhance the overall safety and security environment, fostering a conducive atmosphere for learning, working, and personal well-being. The app represents a proactive step towards meeting the evolving safety needs of the university community and ensuring a resilient response to emergent safety challenges.



## Glossary of Terms

Terms	Definition
Emergency Alert System	A system designed to quickly disseminate critical information to users during emergency situations, such as natural disasters, security threats, or medical emergencies.
Location-Based Services (LBS)	Services that utilize geographical location data to provide users with relevant information, such as nearby safety resources, emergency contacts, and real-time alerts based on their current location.
Crisis Communication Tools	Tools and technologies used to facilitate communication and coordination during crisis situations, including messaging platforms, push notifications, and emergency broadcast systems.
User Authentication	The process of verifying the identity of users accessing the safety app through secure authentication mechanisms, such as passwords, biometrics, or two-factor authentication.
Incident Reporting	The process of documenting and reporting safety incidents, suspicious activities, or emergencies through the safety app, enabling prompt response and follow-up actions by campus security or authorities.
Accessibility Standards	Guidelines and standards set forth by organizations like the World Wide Web Consortium (W3C) to ensure digital content and applications are accessible to users with disabilities, including requirements for screen readers, keyboard navigation, and alternative text.
Data Encryption	The process of encoding sensitive information transmitted or stored within the safety app to prevent unauthorized access or interception by malicious entities, enhancing data security and privacy.

Push Notifications	Messages or alerts sent to users' devices from the safety app, providing timely updates, reminders, or emergency notifications even when the app is not actively in use.
User Interface (UI) Design	The design and layout of the safety app's graphical user interface (GUI), including elements such as menus, buttons, icons, and navigation bars, aimed at enhancing usability and user experience.
Compliance Regulations	Legal and regulatory requirements governing the collection, storage, and processing of user data within the safety app, ensuring adherence to privacy laws and protection of user rights and information.

## System Requirements

### Enumerated Functional Requirements

ID	PW	REQ-x
REQ1	3	The app should allow users (students, faculty, staff) to register with their university credentials and authenticate their identity securely.
REQ2	4	The app must be equipped with an emergency alert system capable of sending real-time notifications to users in the event of safety threats, natural disasters, or campus emergencies.
REQ3	3	The app should provide location-based services to users, allowing them to share their current location with campus security or emergency responders during emergencies.
REQ4	2	Users should have access to comprehensive safety resources, including emergency procedures, evacuation routes, and safety guidelines tailored to various scenarios
REQ5	4	The app should enable users to report safety concerns, incidents, or suspicious activities directly to campus security or relevant authorities, with options for anonymous reporting if desired.

REQ6	4	The app should facilitate two-way communication between users and campus security, allowing users to request assistance, seek clarification, or provide updates during emergencies.
REQ7	4	The app should include a safety check-in feature that allows users to confirm their safety status during campus-wide emergencies, enabling authorities to track and monitor user well-being in real-time.
REQ8	4	Users should receive push notifications and alerts on their mobile devices for important safety updates, campus announcements, and emergency protocols.
REQ9	3	The app should enable users to create personalized safety plans based on their specific needs, including medical conditions, mobility limitations, and emergency contacts.
REQ10	2	The app should integrate with existing campus services, such as campus shuttles, medical facilities, and counseling resources, to provide users with seamless access to support services during emergencies.
REQ11	2	The app should feature interactive training modules and educational resources on safety best practices, emergency preparedness, and risk mitigation strategies for users to access at their convenience.
REQ12	1	The app should be compatible with multiple platforms, including iOS, Android, and web browsers, to ensure accessibility for all users regardless of their device preferences.
REQ13	2	The app must adhere to strict data security and privacy standards, encrypting user data, and ensuring compliance with relevant regulations to safeguard user information and maintain trust.

Most Important = 4 | Least Important = 1

## Enumerated Nonfunctional Requirements

ID	PW	REQ-x
NONREQ1	2	The app should be responsive and provide quick access to safety information and features, even during peak usage times or in areas with poor network connectivity.
NONREQ2	3	The app must be reliable and available 24/7, ensuring uninterrupted access to emergency services and safety resources for users at all times.

NONREQ3	1	The app should be designed to accommodate potential increases in user traffic and data volume as the university community grows, without compromising performance or reliability.
NONREQ4	1	The app must comply with accessibility standards to ensure that users with disabilities can navigate and utilize all features effectively, including support for screen readers and alternative input methods.
NONREQ5	1	The app must implement robust security measures to protect user data, prevent unauthorized access, and mitigate risks of cyber threats, including encryption of sensitive information and secure authentication mechanisms.
NONREQ6	2	The app should prioritize user privacy by collecting and storing only necessary data for safety purposes, obtaining explicit consent for data collection, and adhering to applicable privacy laws and regulations.
NONREQ7	3	The app should feature an intuitive user interface with clear navigation, consistent design elements, and user-friendly controls, catering to users of varying technological proficiency and familiarity with the platform.
NONREQ8	2	The app should be compatible with a wide range of devices, operating systems, and screen sizes, ensuring seamless user experience across different platforms and devices.
NONREQ9	2	The app should be built using modular, well-documented code and adhere to coding best practices to facilitate future updates, maintenance, and enhancements by developers and administrators.
NONREQ10	2	The app must comply with relevant industry standards, legal requirements, and university policies related to data protection, safety regulations, and software development practices, ensuring accountability and adherence to institutional guidelines.

Most Important = 3 | Least Important = 1

## On Screen Appearance Requirements

ID	PW	REQ-x
AR1	3	The app interface should feature a clean and organized layout with clearly defined sections for different safety features and resources, ensuring ease of navigation and user understanding.

AR2	3	Text displayed on the app interface should be legible and accessible to users of all ages and visual abilities, with adjustable font sizes to accommodate individual preferences.
AR3	3	The app should utilize a color scheme that enhances readability and visual clarity, with appropriate contrast between text and background elements to aid users with visual impairments.
AR4	3	The use of intuitive icons and graphical elements should be incorporated throughout the app interface to visually communicate key safety features, actions, and alerts, enhancing user comprehension and engagement.
AR5	2	The app interface should be designed to adapt seamlessly to different screen sizes and resolutions, ensuring optimal viewing and interaction experiences across a wide range of mobile devices and tablets.

Most Important = 3 | Least Important = 1

### FURPS Table

FURPS (Functionality, Usability, Reliability, Performance, Supportability) is a widely used model for categorizing software requirements. A FURPS table for this Campus Safety management system looks like this:

Functionality	<ul style="list-style-type: none"> <li>I. Allows users to report emergencies, crimes, and suspicious activities on campus.</li> <li>II. Provides real-time alerts and notifications to users about safety issues and incidents.</li> <li>III. Includes a map feature with marked safe zones, emergency exits, and locations of security personnel.</li> <li>IV. Offers a virtual escort service for students walking alone on campus during late hours.</li> </ul>
Usability	<ul style="list-style-type: none"> <li>I. Intuitive user interface with easy-to-access buttons for emergency reporting and assistance.</li> <li>II. Provides clear instructions and guidance on how to use the app effectively in emergency situations.</li> </ul>
Reliability	<ul style="list-style-type: none"> <li>I. Highly reliable in terms of emergency reporting and response, with minimal</li> </ul>

	downtime or crashes II. Ensures data security and privacy of user information. III. Regularly updated to address software bugs and security vulnerabilities.
Performance	I. Fast response times for emergency alerts and notifications. II. Works seamlessly across various devices and operating systems. III. Can handle a large number of simultaneous users during peak times effectively.
Supportability	I. Offers comprehensive user support through in-app FAQs, help guides, and contact information. II. Easily customizable to integrate with existing campus safety infrastructure and protocols.

## Functional Requirements Specification

### Stakeholders

- University of Belize
- Security Guards
- Students/Faculty/Staff
- Emergency Responders

### Actors and Goals

Actor	Roles	Type	Goals
System	Responsible for:  1.) Promptly disseminating emergency alerts and notifications to users regarding safety threats, natural disasters, or campus emergencies.	Participating	1.) Deliver timely alerts to users regarding safety incidents, weather advisories, and emergency protocols.  2.) Utilize GPS technology to provide precise location

	<p>2.) Accurately pinpoint users' locations and deliver location-based support during emergencies.</p> <p>3.) Enable users to report safety concerns, request assistance, and provide updates to authorities, while also allowing security personnel to coordinate responses and communicate instructions to users.</p> <p>4.) Verify the identity of users accessing the safety app and control access to sensitive features and information based on user roles and permissions.</p> <p>5.) Encrypt sensitive user data transmitted or stored within the app to protect against unauthorized access or interception.</p>	Participating	<p>information during emergencies.</p> <p>3.) Offer intuitive tools for managing crises, including access to emergency contact information, evacuation routes, and medical assistance guidelines.</p> <p>4.) Foster a culture of safety through anonymous tip reporting, safety forums, and peer support networks.</p> <p>5.) Ensure the app is accessible to all members of the university community, including individuals with disabilities.</p>
University of Belize	<p>Responsible for:</p> <p>1.) Initiating the development and implementation of the campus safety app.</p> <p>2.) Provide database of student IDs</p> <p>2.) Actively promote the app among students, faculty, and staff.</p>		<p>1.) Stay updated about all the incidents, reports that happen on campus.</p>
Security Guards	<p>Responsible for:</p> <p>1.) Campus security officers should be the first responders</p>	Participating	<p>1.) Security guards should swiftly respond to alerts and incidents reported through the app.</p>

	<p>during emergencies. They play a pivotal role in managing crisis situations, and evacuations, and providing immediate assistance.</p> <p>2.) Security personnel should actively monitor the app for incoming alerts, notifications, and incident reports.</p> <p>3.) Work closely with local emergency services (police, fire department, medical responders).</p> <p>4.) Provide escort services for students or staff who feel unsafe. Regular safety patrols contribute to a visible security presence.</p>	Initiating	<p>2.) Continuously monitor the app for incoming alerts and notifications.</p> <p>3.) Utilize the app's GPS functionality to locate individuals in distress.</p> <p>4.) Allocate resources effectively and document incidents.</p> <p>5.) Familiarize themselves with the app's features and its integration with existing security systems.</p>
Students/Faculty/Staff	<p>Responsible for:</p> <p>1.) Students play a crucial role in ensuring their own safety and that of their peers. They should actively engage with the app and utilize its features.</p> <p>2.) Students can use the app's location-based features to share their whereabouts during late-night study sessions, off-campus trips, or other situations where safety concerns them.</p> <p>3.) Encourage students to use the app for anonymous tip reporting. Reporting concerns about safety hazards, potential threats, or</p>	Initiating	<p>1.) Actively participate in safety measures and initiatives implemented by the university.</p> <p>2.) Increase their awareness of safety issues and potential hazards on campus.</p>



	suspicious behavior contributes to a safer campus.		
Emergency Responders	Responsible for:  1.) Work closely with the security guards  2.) Provide first aid or medical care as needed.  3.) Investigations	Initiating	1.) Assess the situation, coordinate with other responders, and reach the scene swiftly.  2.) Prioritize actions based on urgency and allocate resources accordingly.

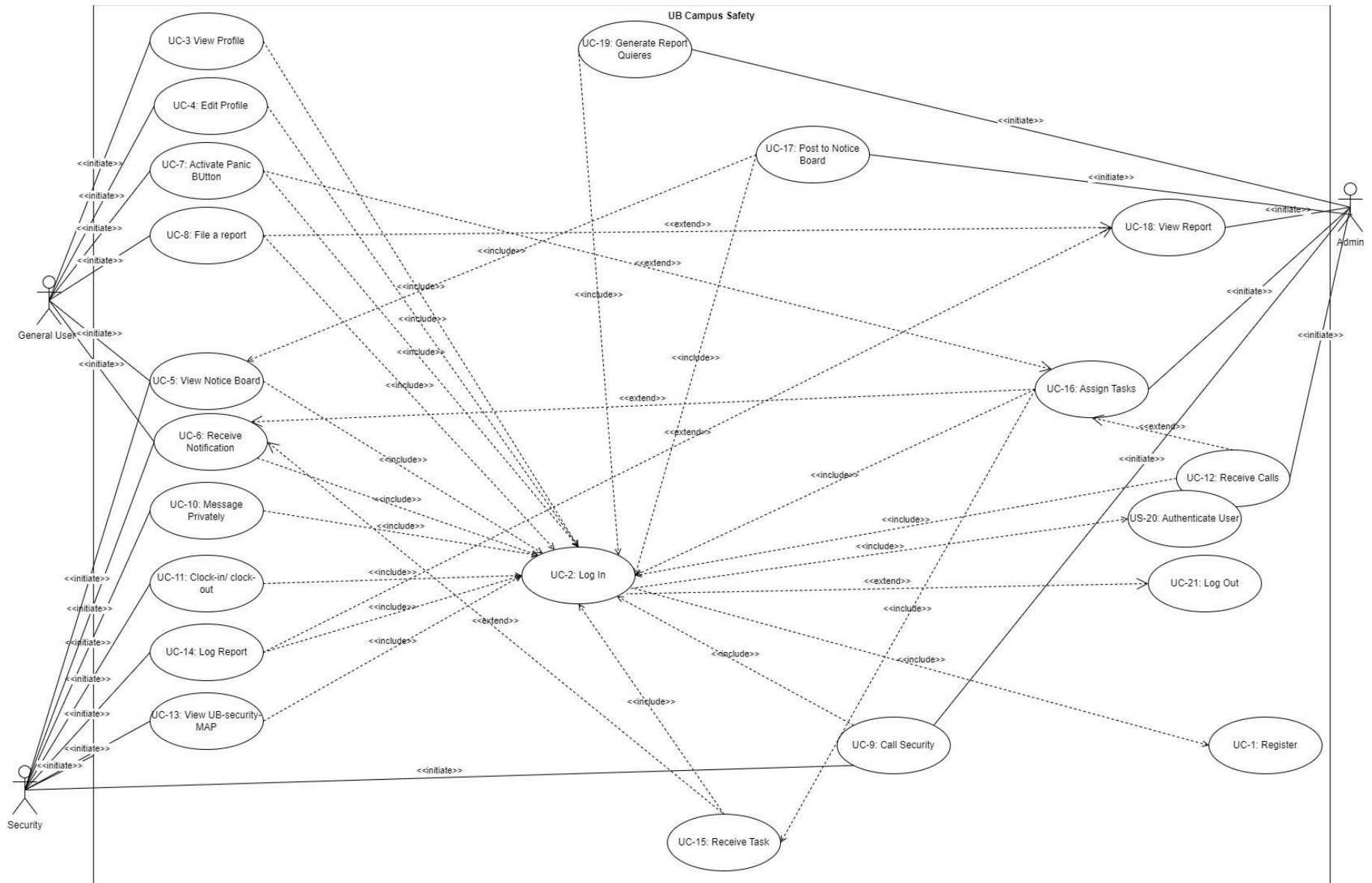
## Casual Description

Name	Description	Requirements Covered
<b><u>GENERAL / ALL Users</u></b>		
UC-1: Register	New users first register using their organizational credentials to use the app.	Functional Requirements: REQ 1
UC-2:Log In/ Log-out	Registered users use credentials to access application features	Functional Requirements: REQ1
UC-3:View Profile	Users have the ability to view their profile information etc...	Functional Requirements:
UC-4:Edit Profile	Users can edit/ update their profile information and set safe status during emergency	Functional Requirements: REQ7
UC-5:View Notice Board	Users Get a daily Board section with events of the Day at the Campus	Functional Requirements: REQ2, REQ4
UC-6:Receive Notification	Users get Notification Whenever an emergency/ event etc... is taking place within the UB campus	Functional Requirements: REQ2, REQ8
UC-7:Activate Panic Button	Users get an emergency	Functional Requirements:

	section where they click this button in case of an emergency which alerts security, health ect.. Staff to move. This button also activates location of device to better help staff locate user within the UB campus	REQ3
UC-8:File a report	A form which allows for reports to be made to staff either anonymous or not. This is used to help in investigation ect... when needed	Functional Requirements: REQ3, REQ5
UC-9:Call Security	A feature which can be an alternative to the panic button based on 'user evaluated emergency level' this can provide a sense of security when crossing the campus at night ect...	Functional Requirements: REQ6
<b><u>SECURITY ONLY</u></b>		
UC-10:Message Privately	Feature that allows security staff to stay in touch with each other throughout the day and during their shifts.	Functional Requirements: REQ6
UC-11: Clock-in/clock-out	This feature is part of a MAP which is planned to be integrated. This is for security staff to be able to see which security boot within the campus have active personnel and which not during their shifts.	Functional Requirements: REQ3
UC-12: Receive Calls	Receive calls from the user side.	Functional Requirements:
UC-13: View UB-security-MAP	MAP integration such that, whenever a user clicks the panic button, a pushpin appears in the map	Functional Requirements: REQ2, REQ3

	pinpointing location of the device within the UB campus. This is to allow security staff to move faster and take action more effectively than in a call.	
UC-14: Log Report	Security staff log reports on happenings during the day, shift they did, emergencies received, patrols taken ect.. This is to allow for investigations when needed	Functional Requirements: REQ5
UC-15: Receive Task	For Security staff to receive orders from administration when needed.	Functional Requirements: REQ10
<b><u>ADMIN ONLY</u></b>		
UC-16: Assign Tasks	Assign tasks to different security personnel throughout the day when needed. This is for better efficiency throughout the day.	Functional Requirements: REQ10
UC-17: Post to Notice Board	This is to add events taking place in the campus/ send emergency alert to campus/ ect... to allow all users to stay updated on what's taking place on campus	Functional Requirements: REQ11, REQ8
UC-18: View Report	Allows for the review of reports made by students for eg. 'investigation' when needed	Functional Requirements:
UC-19: Generate Report	Make a query using filters on the reports table using the GUI provided. This will then allow admin users to filter information which can then be viewed for investigation purposes ect...	Functional Requirements:

## Use Cases Diagram



## Traceability Matrix

In this matrix, the importance of each requirement for each use case is graded on a scale of 5, with 5 being the most important and 1 being the least important. An empty cell indicates that the requirement is not relevant to that particular use case.

Use Case	REQ1	REQ2	REQ3	REQ4	REQ5	REQ6	REQ7	REQ8	REQ9	REQ10	REQ11	REQ12	REQ13
UC-1	3												
UC-2	4												
UC-3													
UC-4							3						
UC-5		4		4									
UC-6		4						4					
UC-7			5										
UC-8			5		5								
UC-9						5							
UC-10						4							
UC-11			3										
UC-12													
UC-13		4	5										
UC-14						4							
UC-15										3			
UC-16										3			
UC-17								4			3		
UC-18													
UC-19													

## Fully Dressed Descriptions

### Fully Dressed Description of Register

Use Case UC - 1	Register
Related Requirement:	Functional Requirement(s): REQ1
Initiating Actors:	Students, Faculty/Staff, Security Guards
Actor Goal:	<ul style="list-style-type: none"> <li>• Login using their UB default credentials (ID, Email)</li> </ul>
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>• User must use their officially given UB email</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>• Systems stores account in database</li> </ul>
Flow of Events for main Success Scenario:	→ Logs into the systems using default UB credentials ← Prompts the user to change password ← Updates password ← Welcomes user to application
Flow of Events for Extensions:	

### Fully Dressed Description of Login/Logout

Use Case UC - 2	Login/Logout
Related Requirement:	Functional Requirement(s): REQ1
Initiating Actors:	Students, Faculty/Staff, Security Guards
Actor Goal:	<ul style="list-style-type: none"> <li>• Login using their UB credentials</li> </ul>
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>• User must have a valid account</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>• System displays the home page</li> </ul>
Flow of Events for main Success Scenario:	→Types username & password ← Welcomes user back
Flow of Events for	

Extensions:	
-------------	--

### Fully Dressed Description of View Profile

Use Case UC - 3	View Profile
Related Requirement:	Functional Requirement(s): REQ
Initiating Actors:	Students, Faculty/Staff
Actor Goal:	Uses system to view profile
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>User must have a valid account</li> <li>User must be logged in</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>System displays profile information</li> </ul>
Flow of Events for main Success Scenario:	→ Clicks the profile icon on the navigation bar ← System shows the users profile
Flow of Events for Extensions:	

### Fully Dressed Description of Edit Profile

Use Case UC - 4	Edit Profile
Related Requirement:	Functional Requirement(s): REQ7
Initiating Actors:	Students, Faculty/Staff
Actor Goal:	Edits emergency/safety features
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>User must have a valid account</li> <li>User must be logged in</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>Changes made from user is updated by the system</li> </ul>
Flow of Events for main Success Scenario:	→ Clicks profile icon ← System shows the users profile → Updates emergency features

	←System prompts user to save changes →User clicks Save ←System prompts “Changes saved”
Flow of Events for Extensions:	

Fully Dressed Description of View Notice Board

Use Case UC - 5	View Notice Board
Related Requirement:	Functional Requirement(s): REQ 2, REQ4
Initiating Actors:	Students, Faculty/Staff, Security Guards
Actor Goal:	To view any updates posted on the notice board
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>User wants to check if there are any updates on the notice board. User receives notification</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>User gets informed with the up to date information</li> </ul>
Flow of Events for main Success Scenario:	→User selects notice icon from menu ←User gets presented with news from the app →User can click on a specific news headline to view more details
Flow of Events for Extensions:	

Fully Dressed Description of Receive Notification

Use Case UC - 6	Receive Notification
Related Requirement:	Functional Requirement(s): REQ 2, REQ8
Initiating Actors:	Students, Faculty/Staff, Security Guards
Actor Goal:	To view receive any notifications



Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>User must have notifications turned on for app</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>System displays notification on phone</li> </ul>
Flow of Events for main Success Scenario:	→ System sends notification to users ← Notification is displayed on the users phone
Flow of Events for Extensions:	

### Fully Dressed Description of Activate Panic Button

<b>Use Case UC - 7</b>	<b>Activate Panic Button</b>
Related Requirement:	Functional Requirement(s): REQ 2, REQ8
Initiating Actors:	Students, Faculty/Staff
Actor Goal:	To help for a serious emergency
Participating Actors:	Security Guards, Admin
Preconditions:	<ul style="list-style-type: none"> <li>User requires immediate assistance in case of an emergency</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>Admin receives alert and sends security guards for assistance</li> </ul>
Flow of Events for main Success Scenario:	→ Enters app ← If already logged in, press and hold panic button → User receives prompt that button has been activated
Flow of Events for Extensions:	

### Fully Dressed Description of File a Report

<b>Use Case UC - 8</b>	<b>File a report</b>
Related Requirement:	Functional Requirement(s): REQ 3, REQ5
Initiating Actors:	Students, Faculty/Staff, Security Guards,

Actor Goal:	To write a report of any suspicious activity or unusual occurrence
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>User can fill out a form to write a report</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>Report is logged and sent to the database where it can be viewed by an admin</li> </ul>
Flow of Events for main Success Scenario:	<p>← If already logged in, User select the File a report item from the menu.</p> <p>→ User fill out form</p> <p>← UserConfirm submission of form</p> <p>→ System stores form in database</p>
Flow of Events for Extensions:	

Fully Dressed Description of Call a Security

<b>Use Case UC - 9</b>	<b>Call a Security</b>
Related Requirement:	Functional Requirement(s): REQ6
Initiating Actors:	Students, Faculty/Staff
Actor Goal:	To call security for assistance
Participating Actors:	System, Security
Preconditions:	<ul style="list-style-type: none"> <li>Security is available</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>Security is notified of incident</li> <li>Call date and time is logged</li> </ul>
Flow of Events for main Success Scenario:	<p>← User selects Call icon</p> <p>→ User press the Second Call icon to activate a call</p> <p>← User gets redirected to an available security</p> <p>→ System logs time and date of call</p> <p>← Security accepts call</p>
Flow of Events for	

Extensions:	
-------------	--

Fully Dressed Description of Message Privately

<b>Use Case UC - 10</b>	<b>Message Privately</b>
Related Requirement:	Functional Requirement(s): REQ6
Initiating Actors:	Admin, Security
Actor Goal:	Communicate with other security and or admin
Participating Actors:	System, Admin, Security
Preconditions:	<ul style="list-style-type: none"> <li>• Messaging Service is working</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>• Message is successfully sent to other personnel</li> </ul>
Flow of Events for main Success Scenario:	← User selects Call icon → User press the Messaging ← User types out message to be sent → User confirms submission of message
Flow of Events for Extensions:	

Fully Dressed Description of Clock in/ Clock out

<b>Use Case UC - 11</b>	<b>Clock in/ Clock Out</b>
Related Requirement:	Functional Requirement(s): REQ6
Initiating Actors:	Security
Actor Goal:	To notify system that they are available/ present for today
Participating Actors:	System, Admin
Preconditions:	<ul style="list-style-type: none"> <li>• Security is available to work</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>• Security notifies personnel that they are available to work</li> </ul>

Flow of Events for main Success Scenario:	<b>Clocking In</b> ← Security selects Profile icon → Security presses the clock in icon ← System marks security as available and logs the time stamp
Flow of Events for Extensions:	<b>Clocking Out</b> ← Security selects Profile icon → Security presses the clock in which is shown as clock out if the user to already clocked in icon ← System marks security as unavailable and logs the time stamp

### Fully Dressed Description of Receive Calls

Use Case UC - 12	Receive Calls
Related Requirement:	Functional Requirement(s): REQ3
Initiating Actors:	Security
Actor Goal:	Accept calls that have been redirected to security
Participating Actors:	System, Admin
Preconditions:	<ul style="list-style-type: none"> <li>Security is available to work</li> <li>A call is pending from the admin</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>Security is now in a call</li> </ul>
Flow of Events for main Success Scenario:	← Receive Notification from app → Open app ← Select call icon from menu → Accept pending calls
Flow of Events for Extensions:	

### Fully Dressed Description of UB-security-MAP

Use Case UC - 13	View UB-security-MAP
------------------	----------------------

Related Requirement:	Functional Requirement(s): REQ2, REQ3
Initiating Actors:	Security
Actor Goal:	View where tasks/ assistance is located
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>• Security has locations turned on</li> <li>• Security has tasks to accomplish</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>• System display a map with tasks located on it</li> </ul>
Flow of Events for main Success Scenario:	→ Open app ← Select map icon from menu → System display map with tasks
Flow of Events for Extensions:	If there are no tasks for the security → System will still display map

### Fully Dressed Description of Log Report

Use Case UC - 14	Log Report
Related Requirement:	Functional Requirement(s): REQ5
Initiating Actors:	Security
Actor Goal:	To write and log their daily report
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>• Report is written at the end of their shift</li> <li>• Security must be clocked in</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>• Report is logged and stored in the database</li> </ul>
Flow of Events for main Success Scenario:	→ Open app ← Select Log report item from menu → Fill out form for daily report ← Confirm submission → Report is stored in system database
Flow of Events for Extensions:	If security forgot to write report while clocked in → System sends notifications as a reminder. If security attempts to clock out without report → System displays warning that users must write a report while

	<p>giving them the option to clock out anyways.</p> <p>If security is not present today → System marks security as absent and ignore reminder for said security</p>
--	---

### Fully Dressed Description of Receive Task

Use Case UC - 15	Receive Task
Related Requirement:	Functional Requirement(s): REQ10
Initiating Actors:	Security
Actor Goal:	To receive task that must be accomplished, these tasks are sent out by the admin
Participating Actors:	System, Admin
Preconditions:	<ul style="list-style-type: none"> <li>• Security must be clocked in</li> <li>• Security must be available</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>• Task is sent to security(s) with varying priorities</li> </ul>
Flow of Events for main Success Scenario:	<p>← Receive Notification</p> <p>→ Open app</p> <p>← Select Task item from menu</p> <p>→ Give option to mark option as complete, incomplete or ongoing</p>
Flow of Events for Extensions:	

### Fully Dressed Description of Assign Task

Use Case UC - 16	Assign Task
Related Requirement:	Functional Requirement(s): REQ10
Initiating Actors:	Admin
Actor Goal:	Assign tasks to security personnel
Participating Actors:	System, Admin

Preconditions:	<ul style="list-style-type: none"> <li>• Security must be available</li> <li>• Assistance from general user services is being required</li> </ul>
Post Conditions:	When needed, admin will assign tasks to security guards
Flow of Events for main Success Scenario:	← Admin open tasks tab → Admin can select security(s) to assign task to ← Admin gives information on task to selected security(s) → System sends security(s) a notification and task
Flow of Events for Extensions:	

### Fully Dressed Description of Post to Notice Board

<b>Use Case UC - 17</b>	<b>Post to Notice Board</b>
Related Requirement:	Functional Requirement(s): REQ11, REQ8
Initiating Actors:	Admin
Actor Goal:	Post events, emergency alerts that are taking place at campus for students/staff
Participating Actors:	Admin, System
Preconditions:	Admin posts all necessary info on notice board
Post Conditions:	Info is updated on notice board and all users can view
Flow of Events for main Success Scenario:	→ Admin enters the app → Clicks Notice Board Icon ← System shows the Notice Board → Clicks “Add Post” ← System shows a text box → Admin types/adds info → Admin Clicks “Post” ← System then prompts “Post was successfully uploaded”
Flow of Events for Extensions:	

Fully Dressed Description of Post to View Report

Use Case UC - 18	View Report
Related Requirement:	Functional Requirement(s):
Initiating Actors:	Admin
Actor Goal:	Admin views report of the users
Participating Actors:	System, Admin
Preconditions:	Admin reviews all the reports made by the users
Post Conditions:	Admin takes info and investigates report
Flow of Events for main Success Scenario:	← Notification is sent to admin → Admin opens notification ← System opens report for admin to view → Admin views report and if checks if investigation is required
Flow of Events for Extensions:	

Fully Dressed Description of Post to Generate Report

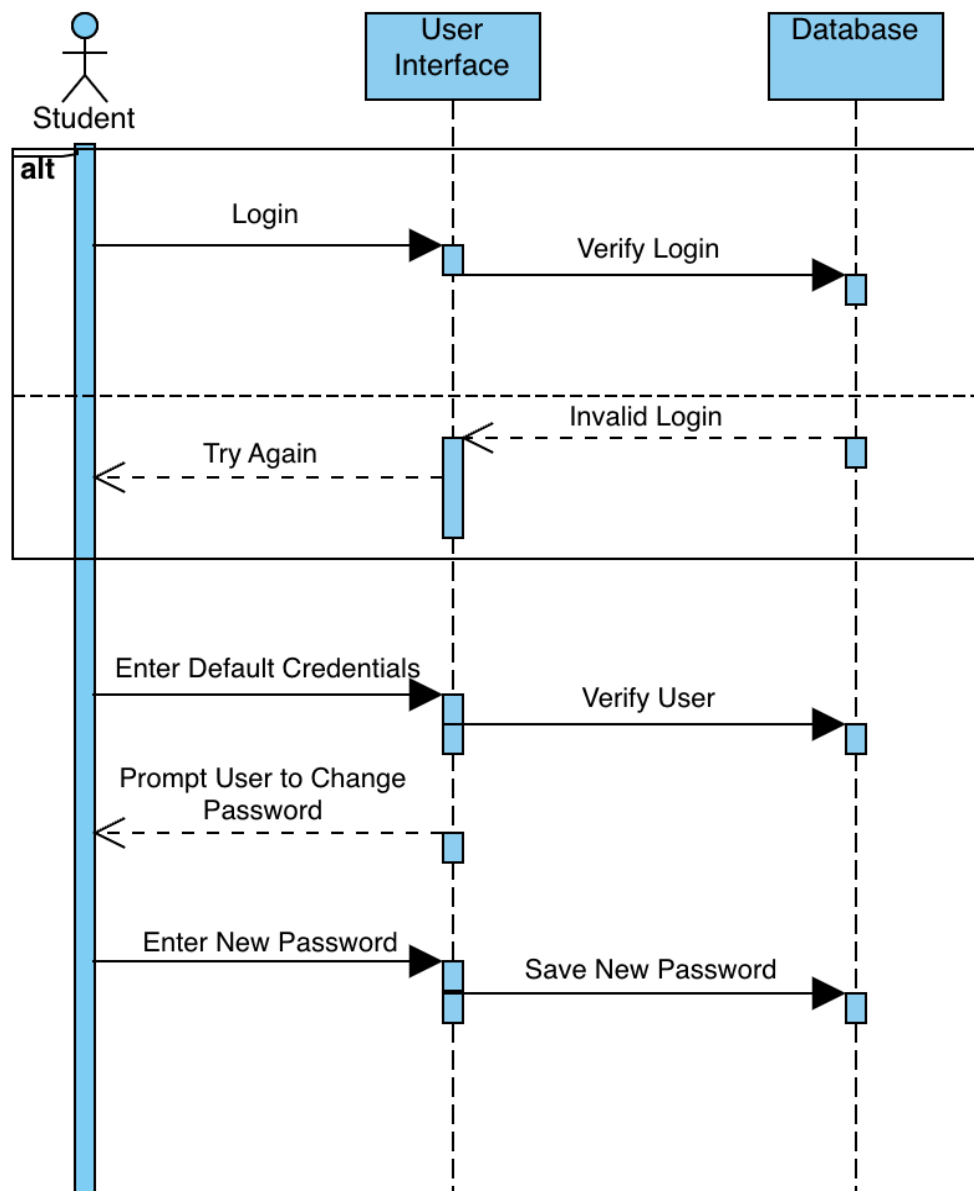
Use Case UC - 19	Generate Report
Related Requirement:	Functional Requirement(s):
Initiating Actors:	Admin
Actor Goal:	Generate Queries of Reports submitted by Users
Participating Actors:	System
Preconditions:	<ul style="list-style-type: none"> <li>Have reports in database from both general users and security</li> </ul>
Post Conditions:	<ul style="list-style-type: none"> <li>Queries of reports is displayed to admin</li> </ul>
Flow of Events for main Success Scenario:	→ Enter reports tab on administrator panel ← Admin enters filters in order to create queries → System display queries of reports to Admin

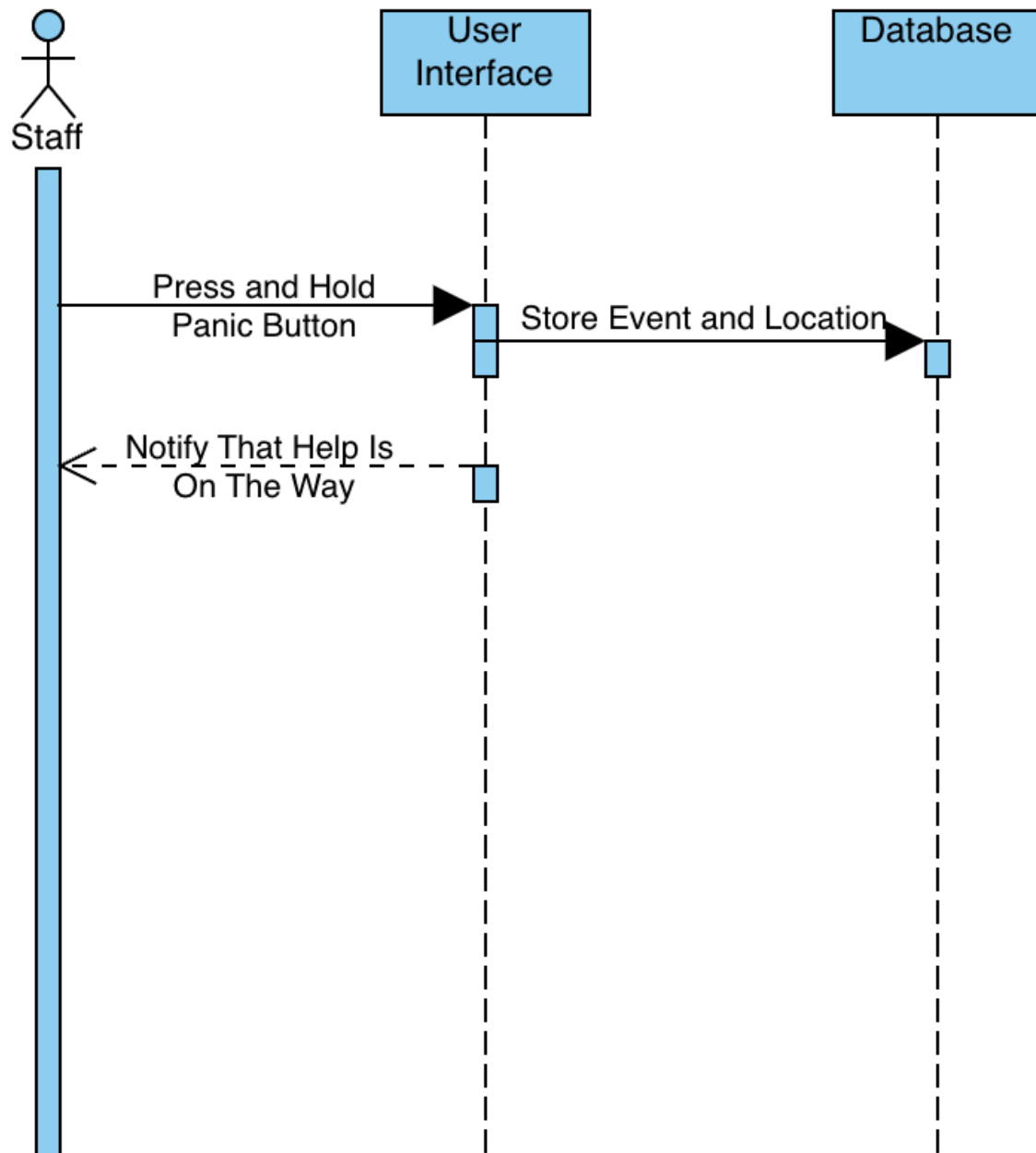


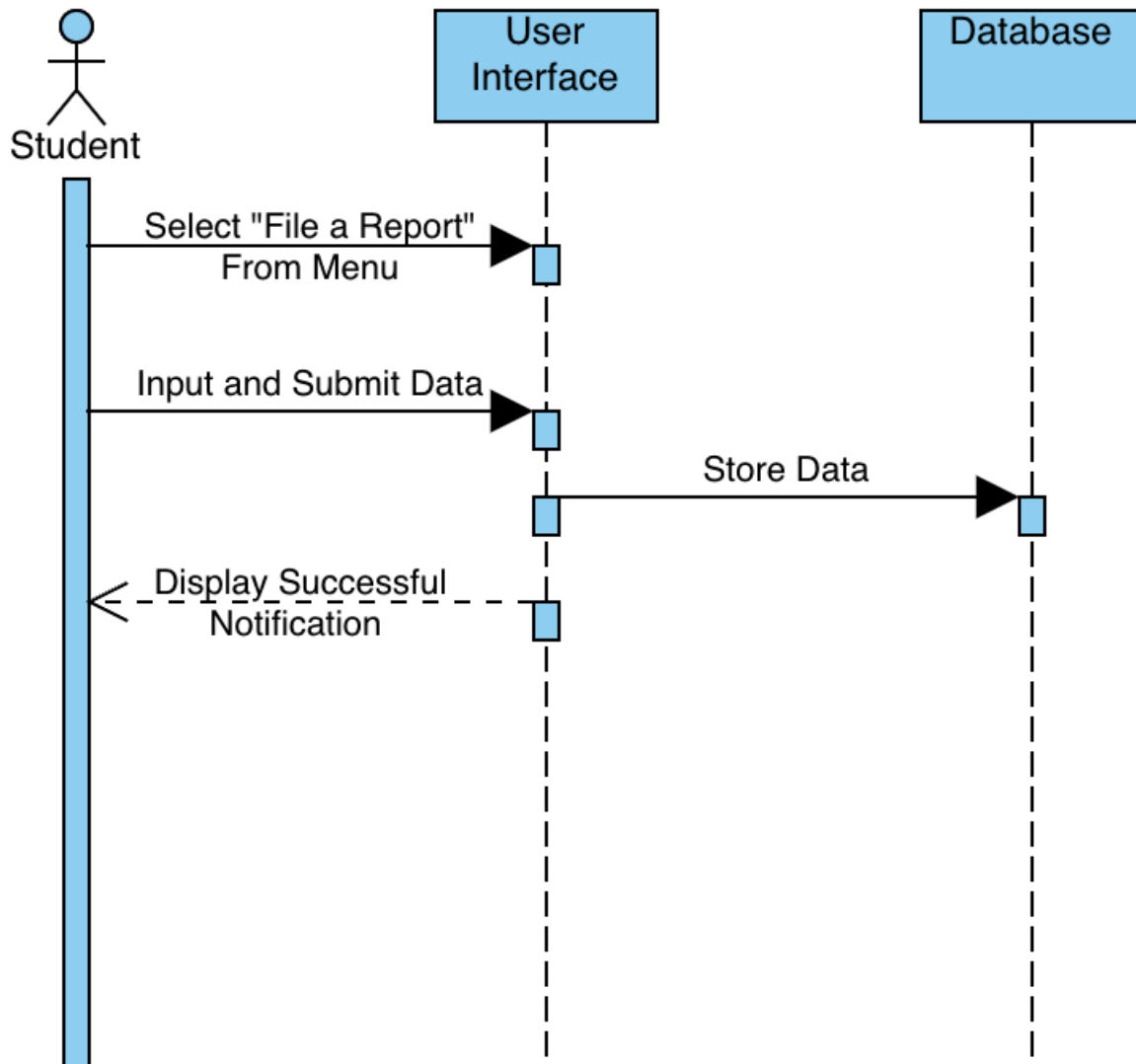
Flow of Events for Extensions:	
--------------------------------	--

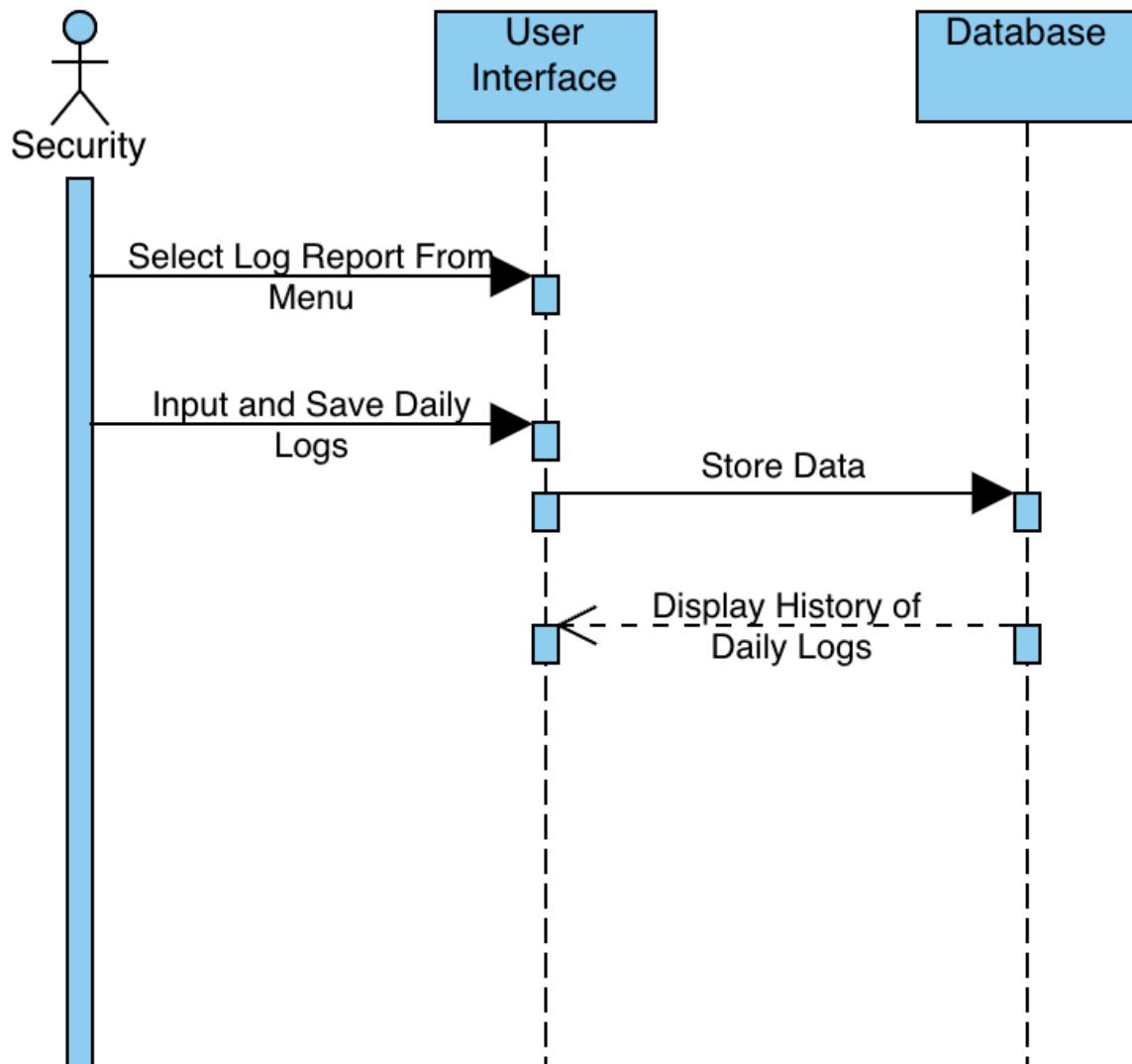
## System Sequence Diagram

### Use Case 1: Register



**Use Case 7: Activate Panic Button**

**UC-8: File a Report**

**UC-14: Log Report**

## User Interface Specification

### Preliminary Designs (MOCKUPS)

**UC-2: Login**



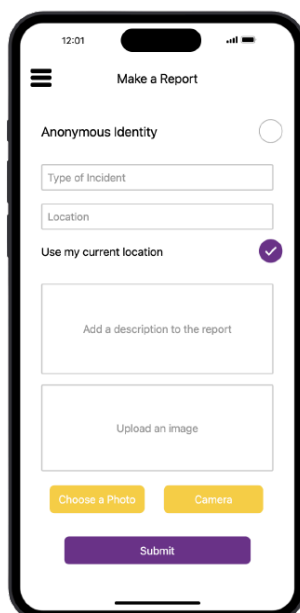
**Menu**



**UC-7 : Activate Panic Button**



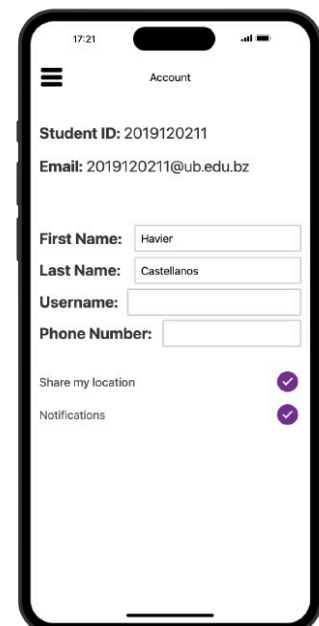
**UC-8: File a Report**



**UC-9: Call Security**



**UC-4: Edit Profile**



## **User Effort Estimation**

### **UC-1: Register**

Effort Estimate: 10 clicks and various keystrokes

1. Click on the registration button (1 click)
2. Enter organizational credentials (5 clicks and keystrokes)
3. Enter other required information (5 clicks and keystrokes)

### **UC-2: Log In/Log-out**

Effort Estimate: 5 clicks and various keystrokes

1. Click on the login button (1 click)
2. Enter username and password (4 clicks and keystrokes)

### **UC-3: View Profile**

Effort Estimate: 3 clicks

1. Click on the profile tab (1 click)
2. View profile information (2 clicks)

### **UC-4: Edit Profile**

Effort Estimate: 8 clicks and various keystrokes

1. Click on the edit profile button (1 click)
2. Update profile information (7 clicks and keystrokes)

### **UC-5: View Notice Board**

Effort Estimate: 2 clicks

1. Click on the notice board tab (1 click)
2. View daily events (1 click)

### **UC-6: Receive Notification**

Effort Estimate: Passive action, no user effort required

### **UC-7: Activate Panic Button**

Effort Estimate: 1 click

### **UC-8: File a Report**

Effort Estimate: 7 clicks and various keystrokes

1. Click on the report form (1 click)
2. Fill in report details (6 clicks and keystrokes)

**UC-9: Call Security**

Effort Estimate: 2 clicks

1. Click on the call security button (1 click)
2. Confirm call (1 click)

**UC-10: Message Privately**

Effort Estimate: 5 clicks and various keystrokes

1. Click on the messaging feature (1 click)
2. Select recipient (2 clicks)
3. Compose and send message (2 clicks and keystrokes)

**UC-11: Clock-in/Clock-out**

Effort Estimate: 2 clicks

1. Click on clock-in/clock-out feature (1 click)
2. Confirm action (1 click)

**UC-12: Receive Calls**

Effort Estimate: Passive action, no user effort required

**UC-13: UB-Security-MAP**

Effort Estimate: Passive action, no user effort required

**UC-14: Log Report**

Effort Estimate: 8 clicks and various keystrokes

1. Click on log report feature (1 click)
2. Fill in report details (7 clicks and keystrokes)

**UC-15: Receive Task**

Effort Estimate: Passive action, no user effort required

**UC-16: Assign Tasks**

Effort Estimate: 5 clicks and various keystrokes

1. Click on assign tasks feature (1 click)
2. Select personnel (2 clicks)
3. Assign task (2 clicks and keystrokes)

**UC-17: Post to Notice Board**

Effort Estimate: 5 clicks and various keystrokes

1. Click on post to notice board feature (1 click)
2. Enter event details (4 clicks and keystrokes)

### UC-18: View Report

Effort Estimate: 3 clicks

1. Click on view reports tab (1 click)
2. Select report for viewing (2 clicks)

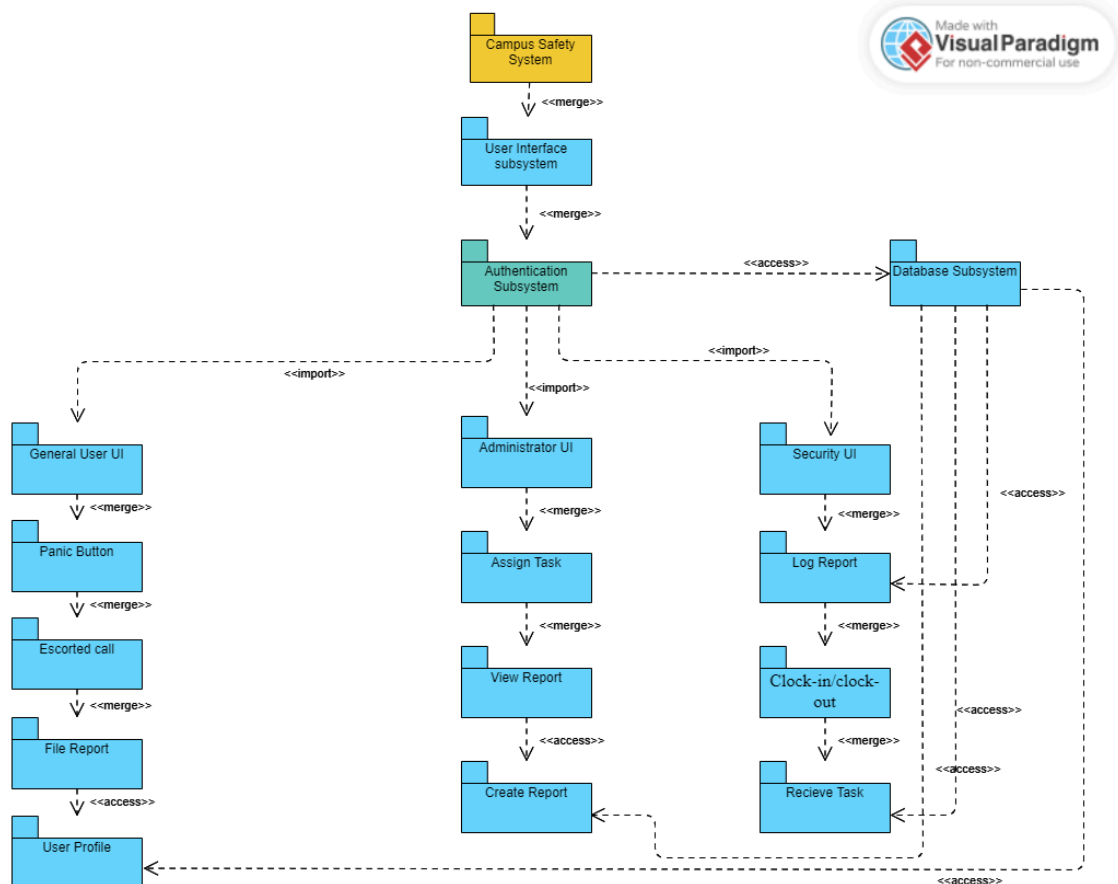
### UC-19: Generate Report Queries

Effort Estimate: 4 clicks and various keystrokes

1. Click on report queries feature (1 click)
2. Select filters (2 clicks)
3. Generate report (1 click)

## System Architecture

### Identifying Subsystems





## Architecture Style

**This campus safety system will implement three architecture styles: Client Server Architecture, Microservices Architecture, and Event-Driven Architecture.**

1. Client-Server Architecture: With this design, the application will be split into two main parts: the server, which handles the business logic and stores and manages data, and the client, which runs on the user's device. When the client asks the server for information or commands, the server answers appropriately. The program is simpler to update and maintain because of this architecture, which enables the concentration of logic and data. By adding new servers as the user base expands, it also permits scalability. The UB campus safety app's client exposes this data to users in an intuitive interface, while the server stores details about the campus's amenities, activities, academic resources, and user profiles.

2. Microservices Architecture: A microservices architecture consists of building an application as a group of loosely linked services, each of which is in charge of handling a particular function. Greater flexibility and agility are made possible by the ability for each service to be designed, implemented, and scaled independently. Different microservices can manage features like course registration, campus maps, event scheduling, library resources, and so on for a university campus safety app. Complex applications, such as university campus safety apps, might benefit from microservices architecture's ability to promote modularity and accelerate development cycles.

3. Event-Driven Architecture: In an event-driven architecture, components of the system communicate with each other by generating and responding to events. Events can be triggered by user actions, system events, or external inputs, and they are handled asynchronously. This architecture promotes loose coupling and scalability, as components can react to events without direct dependencies on each other. For a university campus safety app, event-driven architecture can be utilized for functionalities such as real-time notifications for upcoming events, panic alerts, or changes in campus services.

## Mapping Subsystem to Hardware

### 1. Server Infrastructure:

- ❖ Hardware: High-performance servers with multi-core processors and sufficient RAM to handle concurrent user requests.
- ❖ Functionality: Responsible for storing and managing campus maps, user data, and real-time location tracking information.
- ❖ Requirements: Reliable network connectivity, fault-tolerant architecture, and scalable storage solutions.
- ❖ Components: Load balancers, database servers, and backup systems.

## 2. Geolocation Hardware:

- ❖ Smartphones: Utilize GPS, Wi-Fi, and cellular networks for accurate location tracking of users.
- ❖ Beacons: Install Bluetooth Low Energy (BLE) beacons at key locations across the campus for precise indoor positioning.
- ❖ Requirements: Compatibility with the campus safety app, low power consumption, and minimal maintenance.
- ❖ - Integration: APIs to interface with the mapping subsystem and provide real-time location data.

## 3. User Devices:

- ❖ Smartphones/Tablets: Platforms for accessing the campus safety app and receiving real-time alerts and notifications.
- ❖ Requirements: Compatibility with major operating systems (iOS, Android), sufficient processing power, and network connectivity.

## 4. Networking Equipment:

- ❖ Wi-Fi Access Points: Ensure ubiquitous coverage across the campus for seamless connectivity.
- ❖ Ethernet Switches/Routers: Backend infrastructure for routing data between servers, devices, and external networks.
- ❖ Requirements: High bandwidth, low latency, and support for secure communication protocols (e.g., HTTPS, SSL/TLS).
- ❖ Security: Implementation of firewalls, intrusion detection/prevention systems, and network segmentation to safeguard data.

## 5. Backend APIs and Services:

- ❖ API Gateway: Centralized interface for accessing backend services, ensuring secure and efficient communication.
- ❖ Microservices Architecture: Modular design for scalability, allowing independent development and deployment of services.
- ❖ Authentication/Authorization: Implement robust authentication mechanisms to safeguard user data and access control policies.

## 6. Integration with External Systems:

- ❖ Emergency Services: Interface with local emergency services (police, fire department, medical services) for seamless coordination during emergencies.
- ❖ Campus Facilities Management: Integration with facilities management systems for real-time updates on building access, maintenance activities, and infrastructure status.

## Connectors and Network Protocols

### Connectors:

1. Wi-Fi: Allows devices to connect to the university's wireless network for communication.
2. Bluetooth: Enables short-range communication between devices, useful for features like proximity detection and device pairing.
3. GPS (Global Positioning System): Provides location information for devices, allowing for features like real-time tracking and geofencing.
4. Cellular network: Enables communication via mobile networks, ensuring connectivity even when Wi-Fi is unavailable.
5. Ethernet: For wired connections, typically used for more stable and secure connections in specific areas like offices or security stations.

### Network Protocols:

1. HTTP/HTTPS (Hypertext Transfer Protocol/Secure): For communication between the app and web servers, used for accessing campus safety information, updates, and other online resources securely.
2. TCP/IP (Transmission Control Protocol/Internet Protocol): Fundamental protocols for transmitting data across networks, ensuring reliable communication between devices.
3. UDP (User Datagram Protocol): Provides faster communication suitable for real-time data transmission, such as live video streaming or voice calls.
4. MQTT (Message Queuing Telemetry Transport): Lightweight messaging protocol suitable for IoT devices and sensors, allowing for efficient communication between devices on the network.
5. SNMP (Simple Network Management Protocol): Used for monitoring and managing network devices, helpful for ensuring the reliability and security of network infrastructure supporting the safety app.
6. SIP (Session Initiation Protocol): Used for initiating and terminating multimedia communication sessions, potentially useful for features like emergency calls or video conferencing within the app.
7. LDAP (Lightweight Directory Access Protocol): For accessing directory information, such as user authentication and authorization data, which could be relevant for user management within the safety app.

## Global Control Flow

1. Initialization and Authentication: Upon launching the university campus safety app, the user is prompted to authenticate their identity, typically through a login screen. Authentication may involve entering a username and password or utilizing biometric authentication methods such as fingerprint or facial recognition.

2. **Main Menu Navigation:** After successful authentication, the user is directed to the main menu interface of the app. Here, they are presented with various options and functionalities tailored to campus safety, such as reporting incidents, accessing emergency contacts, requesting assistance, or exploring safety tips and resources.

3. **Incident Reporting:** One of the core functionalities of the app is the ability for users to report incidents or concerns regarding safety on campus. This can include incidents such as accidents, suspicious activities, or emergencies. Upon selecting the incident reporting feature, the user is guided through a form where they can provide details such as the type of incident, location, and any additional information necessary for responders.

4. **Emergency Assistance Request:** In case of emergencies, users can quickly request assistance through the app. This feature typically involves a prominent emergency button on the main menu, which, when activated, immediately alerts campus security or emergency services to the user's location and situation.

5. **Notification and Alert System:** The app includes a notification and alert system to disseminate important safety information to users in real-time. This can include alerts about weather emergencies, campus closures, or security threats. Users may receive these notifications through push notifications, SMS messages, or email, depending on their preferences and the severity of the situation.

6. **Safety Resources and Information:** Additionally, the app serves as a hub for safety resources and information, providing users with access to safety tips, campus maps highlighting safe zones and emergency exits, contact information for campus security personnel and support services, as well as educational materials on topics such as personal safety and emergency preparedness.

7. **Logging Out and Session Management:** Finally, users can log out of the app to end their session securely, especially if accessing the app from shared or public devices. Proper session management is crucial to protect user privacy and ensure that sensitive information is not accessible to unauthorized individuals.

## **Hardware Requirements**

### **Mobile Requirements:**

- OS: Android 8.0 or later; iOS 12.0 or later
- Storage: Minimum of 2GB available space
- Screen Resolution: Minimum screen resolution of 640 x 480
- Network Bandwidth: Stable internet connection with a minimum bandwidth of 60 kbps

### **PC Requirements:**

- OS: Any operating system compatible with the following web browsers:
- Chrome: Version 70 or later
- Firefox: Version 70 or later
- Safari: Version 12 or later
- Opera: Version 60 or later
- Network: Broadband internet connection is required or Physical LAN connection

## Plan of Work

### Gantt Chart

