American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Emergency Service

A Software Engineering Project Submitted
By

Semester: Summer_21_22		Section:	Group Number:	
SN	Student Name	Student ID	Contribution (CO3+CO4)	Individual Marks
1	Moinul Hasan	21-45777-3	23.35	
2	Ahsanul Haque Joar	22-47183-1	19.20	
3	Fatima Adon	21-45840-3	19.17	
4	Sadia Afrose	21-45820-3	19.18	
5	MD. Naimul Islam	21-45801-3	19.10	

The project will be Evaluated for the following Course Outcomes

CO3: *Select* appropriate software engineering models, project management roles and their associated skills for the complex software engineering project and evaluate the sustainability of developed software, taking into consideration the societal and environmental aspects	Total Marks	
Appropriate Process Model Selection and Argumentation with Evidence	[5 Marks]	
Role and responsibilities in your group for the selected process model	[5Marks]	
Evaluate the sustainability of the developed software in terms of both society and the environment (Impact identification)	[5Marks]	
Submission, Defense, Completeness, Spelling, grammar and Organization of the Project report	[5Marks]	
CO4: *Develop* project management plan to manage software engineering projects following the principles of engineering management and economic decision process	Total Marks	
Develop the project plan, its components of the proposed software products using WBS and testcases	[5Marks]	
Identify all the activities/tasks related to project management and categorize them within Project estimation, and schedule of the tasks using appropriate resources	[5Marks]	
Identify all the potential risks in the specific project and prioritizing/categorizing those, and also mitigation plan to overcome the risk factors.	[5Marks]	

Description of Student's Contribution in the Project work

Student Name: Moinul Hasan
Student ID: 21-45777-3
Contribution in Percentage (%): 23.35%

Contribution in the Project:

- Helped generating project idea
- Added multiple functional requirements
- Added multiple nonfunctional requirements
- Created Use Case Diagram
- Helped creating Class Diagram
- Created Sequence Diagram
- Helped creating Activity Diagram
- Designed UI/UX
- Developed Test Cases
- Designed/Developed Timeline 1
- Designed/Developed Timeline 2
- Calculated EVA
- Developed Risk Table

Signature of the Student

Student Name: Ahsanul Haque Joar
Student ID: 22-47183-1
Contribution in Percentage (%): 19.20%

Contribution in the Project:

- Helped generating project idea
- Added multiple functional requirements
- Added multiple nonfunctional requirements
- Created Activity Diagram
- Helped creating Class Diagram
- Helped creating Sequence Diagram
- Developed Test Cases
- Developed Timeline 1
- Developed Timeline 2
- Developed Risk Table
- Calculated EVA

Signature of the Student

Student Name: Fatima Adon
Student ID: 21-45840-3
Contribution in Percentage (%): 19.17%
Contribution in the Project:

- Helped generating project idea
- Added multiple functional requirements
- Added multiple nonfunctional requirements
- Created Class Diagram
- Helped creating Activity Diagram
- Developed Test Cases
- Developed Timeline 1
- Developed Timeline 2
- Developed Risk Table

Signature of the Student

Student Name: Sadia Afrose Student ID:
21-45820-3
Contribution in Percentage (%): 19.18%
Contribution in the Project:

- Helped generating project idea
- Added multiple functional requirements
- Added multiple nonfunctional requirements
- Created Class Diagram
- Helped creating Activity Diagram
- Developed Test Cases
- Developed Timeline 1
- Developed Timeline 2
- Developed Risk Table

Signature of the Student

Student Name: MD. Naimul Islam Student
ID: 21-45801 - 3
Contribution in Percentage (%): 19.10%
Contribution in the Project:

- Helped generating project idea
- Added multiple functional requirements
- Added multiple nonfunctional requirements
- Created Class Diagram
- Helped creating Activity Diagram
- Developed Test Cases
- Developed Timeline 1
- Developed Timeline 2
- Developed Risk Table

_____
Signature of the Student

1. PROJECT PROPOSAL

1.1 Background to the Problem

In recent decades, emergency services like fire service, providing ambulance, law enforcement service etc have become a necessity for everyday human life. Emergency services have evolved

quite a lot in the past years by pushing the boundaries and incorporating technological innovations. Emergency services are captivating due to their potential to save lives. The challenge lies in minimizing the response time, providing these services in the least amount of time from the time of requesting the service and managing these services simultaneously. The problem arises from the great amount of time emergency services are provided from the time of the request and people facing consequences because of that. Ease of use, automation and a solid software like interface is essential to overcome these challenges and provide emergency services in the least amount of time. The problem is super important because it affects our ability to be safe and to have faith in cases of emergencies. Emergency services can teach us how important someone's time can be in such cases. When we create an efficient and innovative system and deploy it to help people to have fire, police and ambulance services at the palm of their hands we learn more about software, computer, human computer interaction and how they can help us everywhere, not just in the emergency segments. In simpler words, solving this problem helps us to save lives more efficiently than ever before.

1.2 Solution to the Problem

The objective is to develop a software to manage the incoming requests, minimize the response time, give the users an easier way to ask or request for the emergency services with a simpler user interface which will connect the requests to the emergency responders faster than the traditional emergency system. The proposed solution aims to improve the control of the requests and manage the data. On board data tools and integrated AI will be included to minimize the communication gap between the service provider and requester. Which will eventually reduce the required time to get the service. This solution enhances efficiency, reliability, and security in providing these services. Key functionalities include advanced data analytics, real-time communication, global positioning system (gps) to get to the location of the requester as fast as possible. This creative solution not only enhances efficiency and safety but also contributes to the trust we have on our emergency service providers and government. Our project will allow the system to operate faster using previous data samples of any unusual and rare requests, artificial intelligence and global positioning system (gps). There are other system and studies done on this topic, but our system will be more robust, easier to use, readability will be much higher and the size of the designed software will be much smaller so that it can efficiently run on any machine. This will also decrease the cost of the machine on the service provider's side or the government's side for managing the whole system and providing the service without any interruption. Lesser human interaction will be necessary.

The traditional emergency system of Bangladesh can currently take 100 calls at a time and 30,000 calls per day. Which limits it to not having more than 100 calls at a time increasing the waiting time on the queue for the emergency service seekers. That completely defeats the purpose of the service being an emergency service if the requesters cannot have the service provided to them in the shortest possible time. Our system completely eliminates this constraint. Calling will not be necessary to receive the service. Using AI and previous data all of requests will be processed through the software except for the new and unusual cases. Using onboard data

analysis, enhanced autonomy and efficiency our system aims to benefit users and be the bridge between life and death in times of emergencies.

2. SOFTWARE DEVELOPMENT LIFE CYCLE

2.1 Process Model

For the following reasons, we have chosen the Waterfall approach for the development of our emergency service application:

1. Well-Defined needs: We want our project's needs to be steady and unambiguous, considering the importance of emergency services and the need for accurate position tracking. This trait is best suited by the Waterfall model's linear, step-by-step methodology.

2. Safety and Regulatory Compliance: Strict adherence to safety standards and regulations is required for emergency services, particularly those that involve real-time location services. The Waterfall methodology guarantees adherence to these crucial requirements by emphasizing comprehensive documentation, stringent testing, and traceability.

3. Thick Documentation: For the aim of designing, verifying, and validating software for emergency services, thorough documentation is essential. The Waterfall approach facilitates accurate documentation by encouraging the production of comprehensive documents at every level.

Motives for Dismissing Alternative Models:

1. Agile Models: Scrum and other agile approaches are flexible enough to adjust to changing needs, but they might not have the amount of formality and documentation required for emergency services, which frequently deal with systems that are vital to public safety.

2. Iterative Models: While plan-driven models are preferred in safety-critical domains, iterative models may not provide the same level of predictability and regulatory compliance, making them appropriate for projects that are changing quickly.

3. Hybrid Models: Although hybrid models draw on the best features of several techniques, they can also add complexity and necessitate close supervision to guarantee traceability and compliance.

2.2 Project Role Identification and Responsibilities

Important roles and duties in our plan-driven Waterfall approach are as follows:

Project Manager: Ensures that the project's scope, timeline, and budget are followed by supervising planning, scheduling, and general coordination.

Business Analyst: Works closely with stakeholders to ensure clear understanding while gathering, analyzing, and documenting specific project needs.

System Architect/Designer: Develops the overall system design and interfaces, designing the system architecture and components in accordance with requirements.

The Development Team: Made up of engineers and programmers, is in charge of creating the emergency service software and putting the plan into practice.

Test Team and Quality Assurance: Throughout the software development process, the test team does comprehensive testing and quality assurance to make sure the program satisfies requirements.

The Documentation Team: In charge of producing technical papers, user manuals, and specifications as well as other project material.

Client/Customer: Gives project milestones approval.

Experts in Safety and Compliance: Make sure the project complies with all applicable safety and legal regulations.

These responsibilities prioritize vital safety and legal requirements while working together to successfully develop our emergency service application.

Rubric for Project Assessment (CO1)

Marking Criteria	Marks Distribution (Maximum 3X5=15)				Acquired Marks
	Inadequate (1-2)	Satisfactory (3)	Good (4)	Excellent (5)	
Background Analysis	No background information regarding the project is given; project goals and benefits are missing.	Insufficient background information is given; project goals and benefits are poorly stated	Sufficient background information is given; the purpose and goals of the project are explained.	Thorough and relevant background information is given; project goals are clear and easy to identify.	
Analysis the impact of societal, health, safety, legal and cultural issues	Student vaguely discuss the impact of societal, health, safety, legal and cultural issues in their project	Student provided with partial relevance to the impact of societal, health, safety, legal and	Student fairly provided the analysis to the impact of societal, health, safety, legal and	Student comprehensively provided the analysis to the impact of societal, health, safety, legal and cultural	

		cultural issues in their project	cultural issues in their project	issues in their project	
Existing Studies and Relevant Example	Ambiguous representative example.	Partially identify / indicate towards real-life example.	Real-life example is fairly connected towards the definition.	Comprehensively defend with real life example.	
				Acquired Marks:	
				CO Pass / Fail:	

Rubric for Project Assessment (CO2)

Criteria	Marks distribution (Max 3X5= 15)				Acquired Marks
	Inadequate (1-2)	**Satisfactory (3)**	**Good (4)**	**Excellent (5)**	
Argumentation of Model selection with Evidence of Argumentation	Does not articulate a position or argument of choosing appropriate model. Does not present any evidence to support the arguments for the choice of the model	Articulates a position or argument for choosing models that is unfocused or ambiguous. Presents incomplete/vague evidence to support argument for model choice	Articulates a position or argument of choosing models that is limited in scope. Does not present enough evidence to support the argument for the choice of the model	Clearly articulates a position or argument for the choosing software engineering models. Presents sufficient amount of evidence to support argument for the model selection	
Role identification and Responsibility Allocation	The project has poor project management plans for identifying roles and assigning the responsibilities	Identify few roles in the project management where some of the roles are left alone with any project responsibilities	Identify most of the roles in the project management and assign their responsibilities	Well planned project with proper role identification and responsibility allocation in the project management activities	
Submission, Completeness, Spelling, grammar and Organization of the Project report	Project report is not complete and Several errors in spelling and grammar. Present a Confusing organization of concepts, supporting arguments, and real-life example. Sentences rambling, and details are repeated.	Some errors in spelling and grammar. Some problems of organizing the answer in a logical order of defining, elaborating, and providing real-life examples.	Few errors in spelling and grammar. Presents most of the details in a logical flow of organization in definition, details, and example.	Project report is complete and No errors in spelling and grammar. Consistently presents a logical and effective organization of definition, details, and real-life example of the topic.	
				Acquired marks:	
				CO Pass / Fail:	

Functional Requirements

1. Software Login

1.1 The software shall allow users to login with their phone number and password or with their fingerprint.

1.2 The login credentials (phone number and password/ fingerprint) will be verified within database records of registered users.

1.3 If correct credentials have been provided, an one time password (OTP) will be sent to the users registered phone number to verify the ownership of the account.

1.4 If successfully logged in, the interactive home page of the user account will be displayed.

1.5 If wrong login credentials (phone number and password/ fingerprint) have been inserted, then a randomized captcha will appear to make sure the attempt was not made by an AI.

1.6 If 5 times wrong login credentials (phone number and password/ fingerprint) have been inserted, any further login attempt from the machine in question will be locked for an hour. The time will multiply by 3 on every set of wrong login attempts.

Priority Level: High
Precondition: User have valid and registered phone number and password/ fingerprint.

2. Software Signup

2.1 The software shall allow users to register with their phone number, password and their fingerprint(optional) along with other information(name,dob,nid no. etc).

2.2 The software shall ask the users to accept their terms and conditions.

2.3 The software shall verify the ownership of the phone number through an one time password (OTP) sent to the registered phone number.

2.4 If successfully verified, the interactive home page of the user will be displayed.

2.5 The software shall prompt the users to give permission of the location service or the global positioning system (gps) of the user's machine.

2.6 If location permission is given, the software shall let the users choose whether they want to add their insurance papers now or later, so that the charged amount for the emergency services gets covered by insurance agencies.

2.7 The software shall let the users add any available payment method(Bank transfer, mobile banking, online banking etc). The users can choose to add payment method later, then the default payment method will be added to the users account which is cash on delivery (COD).

2.8 If location permission is rejected, the software shall prompt the users that location

access is necessary in order to receive the emergency services.

Priority Level: High
Precondition: User have valid phone number.

3. Generate Probable Causes of Requests

3.1 The software shall fetch the information from the databases of the emergency services (fire brigade, health department, law enforcement department).
3.2 The software shall record all of the causes of the requests.
3.3 The software shall use all the fetched information and the recorded data generate multiple probably causes of the requests based on the geographical location of the users by using data analytics and artificial intelligence (AI).

Priority Level: High
Precondition: User has been logged in successfully into the software.

4. Request Fire Brigade

4.1 The software shall let the users to request for a fire brigade unit.
4.2 The software shall generate probable causes of the request.
4.3 The software shall let the users choose the cause of their request from the generated probable causes.
4.4 The software shall let the users call the emergency number if the generated probable causes doesn't match with the causes of the users' requests to let the emergency responders know the details about the emergency.
4.5 If the users select any one from the probable causes then the software shall let the users choose how severe the emergency is from three predefined options (high, medium, low).
4.6 The software shall then ask the users for the final confirmation of the request showing the estimated time of arrival.
4.7 If confirmed, the fire brigade unit will be notified immediately with the users' information and based on the severity of the emergency one or multiple units will be assigned.
4.8 The software will keep on showing the real time location of the fire brigade and the location of the incident inside a well-coordinated map until the fire brigade units have successfully reached the location of the incident.
4.9 If cancelled, the software shall return the users to the home page.

Priority Level: High
Precondition: User has been successfully logged into the software.

5. Request Health Care Service

5.1 The software shall allow the users to request for health care services.

5.2 The software shall then ask the users to choose between two options (nearby ambulance, emergency consultant).

5.3 If nearby ambulance has been selected by the users, The software shall allow the users to choose from the category of ambulances (premium, moderate, basic).

5.4 The software shall then ask the users to enter the name of the hospital they want to go.

5.5 The software shall then select the closest ambulance of the selected category within 1km radius from the location of the users.

5.6 The software shall assign the request to the ambulance driver.

5.7 The software shall then show the real time location of both the ambulance and the requester along with the estimated time of arrival and the charged amount.

5.8 If emergency consultant is selected in 5.2, the software shall allow the users to choose from the available specialists.

5.9 The software shall then connect the requester with the selected specialist through a video call.

Priority Level: High

Precondition: User has been successfully logged into the software.

6. Request Law Enforcement Unit

6.1 The software shall let the users to request for law enforcement unit(s).

6.2 The software shall generate probable causes of the request.

6.3 The software shall let the users choose the cause of their request from the generated probable causes.

6.4 The software shall let the users call the emergency number if the generated probable causes doesn't match with the causes of the users' requests to let the emergency responders know the details about the emergency.

6.5 If the users select any one from the probable causes then the software shall let the users choose how severe the emergency is from three predefined options (high, medium, low).

6.6 The software shall then ask the users for the final confirmation of the request showing the estimated time of arrival.

6.7 If confirmed, the nearest law enforcement unit will be notified immediately with the users' information and based on the severity of the emergency one or multiple units will be assigned.

6.8 The software will keep on showing the real time location of the law enforcement unit and the users' location inside a well-coordinated map until the law enforcement unit(s) have successfully reached the users' location.

6.9 If cancelled, the software shall return the users to the home page.

Priority Level: High

Precondition: User has been successfully logged into the software.

7. After Service Review

7.1 After the service have been provided successfully, the software shall ask the users to give a review of the responders and vice versa.

7.2 This review will be added to the accounts of the users.

7.3 False requests will be penalized and an amount will be charged on the next request of the users in question.

7.4 The penalty will increase 5 times on each false request.

7.5 If no false request of fraudulent activity were found within 3 years then the amount will be set to the initial amount.

Priority Level: High

Precondition: User has a verified account registered on the software.

8. Track Last and Realtime Location

8.1 The software shall acquire the permission to access the global positioning system embedded into the user's system.

8.2 The software shall continuously monitor real-time location during active emergency services.

8.3 The software shall provide live map interface for the users and emergency responders.

8.4 The software shall handle any inaccuracies due to factors like signal interference.

8.5 The software shall retrieve and display last known location to the emergency responders if any connection problem occurs.

Priority Level: High

Precondition: User has a verified account registered on the software and has given location access to the software.

9. **Generate Penalties**

9.1 The software shall ask the users to give their honest feedback or review of the service provider after the successful completion of the service.

9.2 The software shall then take the review and send it to the emergency service provider's client and store the information into the database for future evaluation.

9.3 The software shall then ask the service providing units to give their honest feedback and comment on the legitimacy of the users.

9.4 These reviews will be added together with the users accounts and points will be generated based on the review (Negative points for negative reviews and positive points for positive reviews).

9.5 Based on these points, if enough negative point has been added to the users accounts, the software shall incorporate an pre stated amount for penalizing the users.

9.6 The software shall stop the user from requesting the services before the penalized amount has been paid.

Priority Level: High
Precondition: User has a verified account registered on the software.
User has successfully taken any listed service.

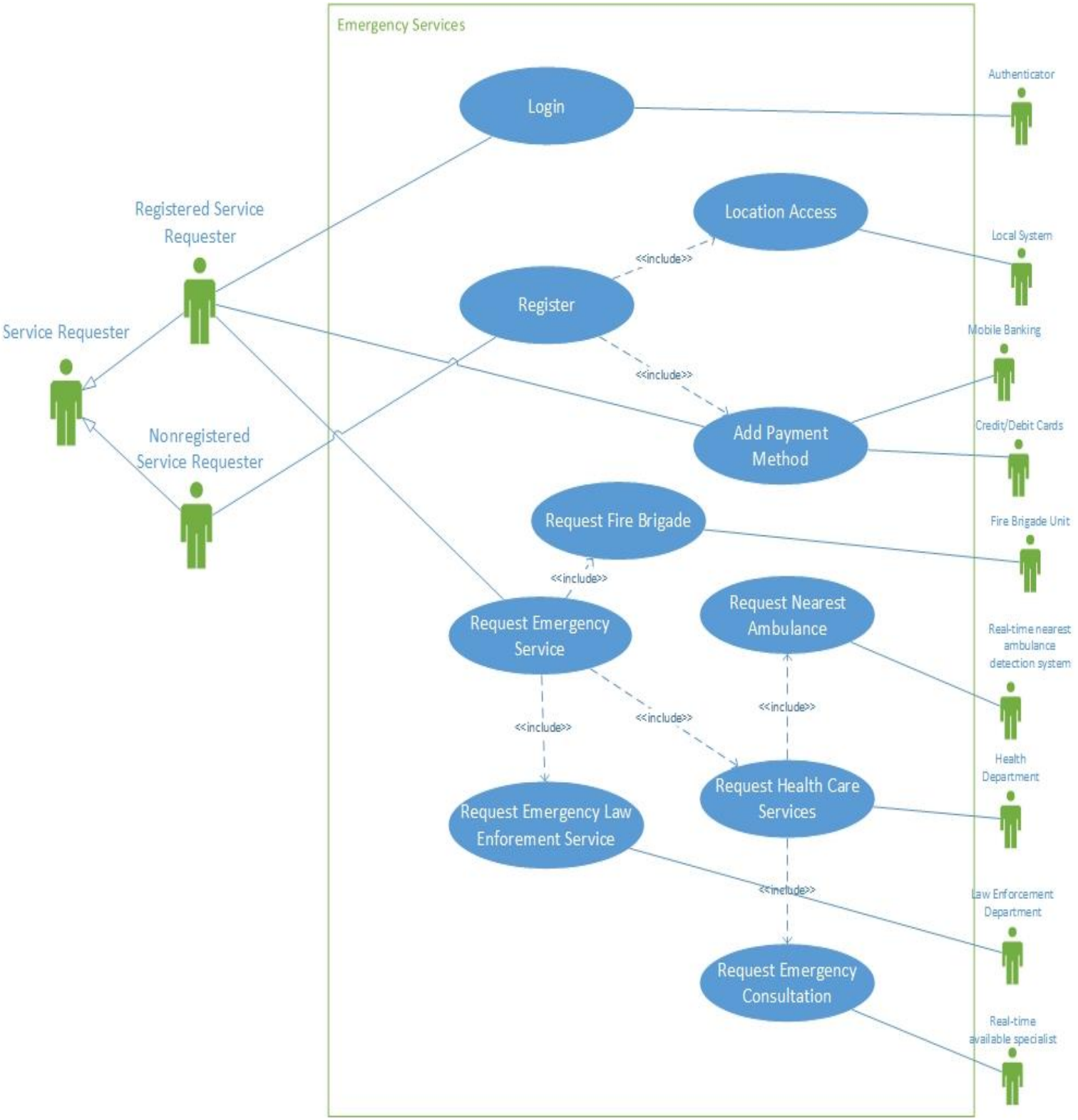
Nonfunctional Requirements

1. Readability: User shall be able to read every instruction clearly without any eye or brain strain and to be able to read in their preferred language.
2. Availability: User shall be able to use the services provided by the system at any given time without any unusual interruptions.
3. Performance: User shall be able to use the system without facing noticeable lag or delay under heavy load.
4. Efficiency: User shall be able to use the system on a low end machine without having to compromise the loading time.
5. Integrity: User shall be able to securely register and login into the system without having any data breach.
6. Interoperability: The system shall be able to communicate with the databases of the integrated emergency services(fire brigade, law enforcement department, medical department) in real time.
7. Reliability: User shall be able to complete a full request of service without facing any

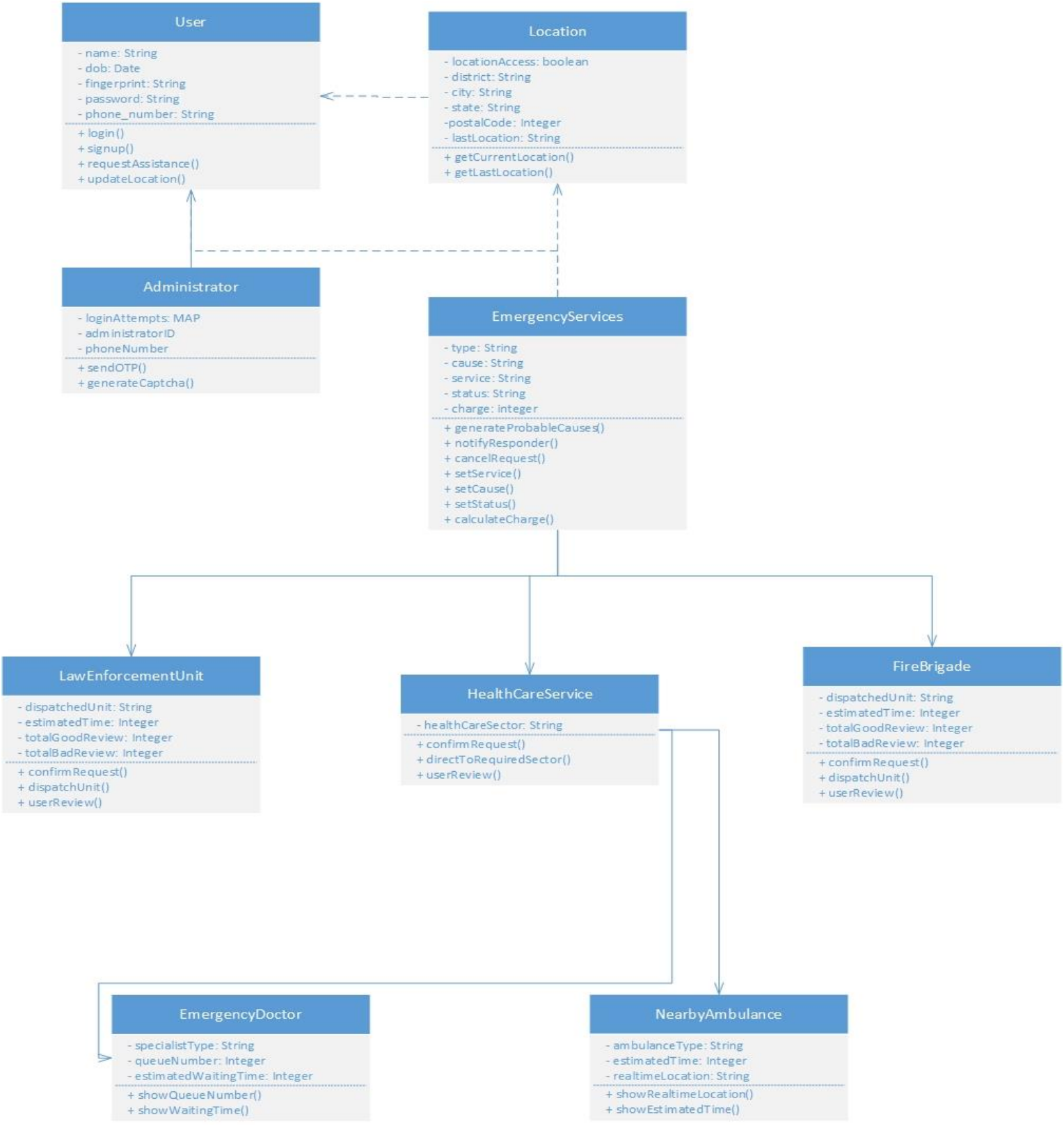
errors

8. Robustness: User shall be able to use the system even after giving invalid inputs or any possible errors thrown by hardware components.
9. Usability: User shall be able to complete a listed request within maximum 1 minute and a non-listed request within minimum 2 minutes.

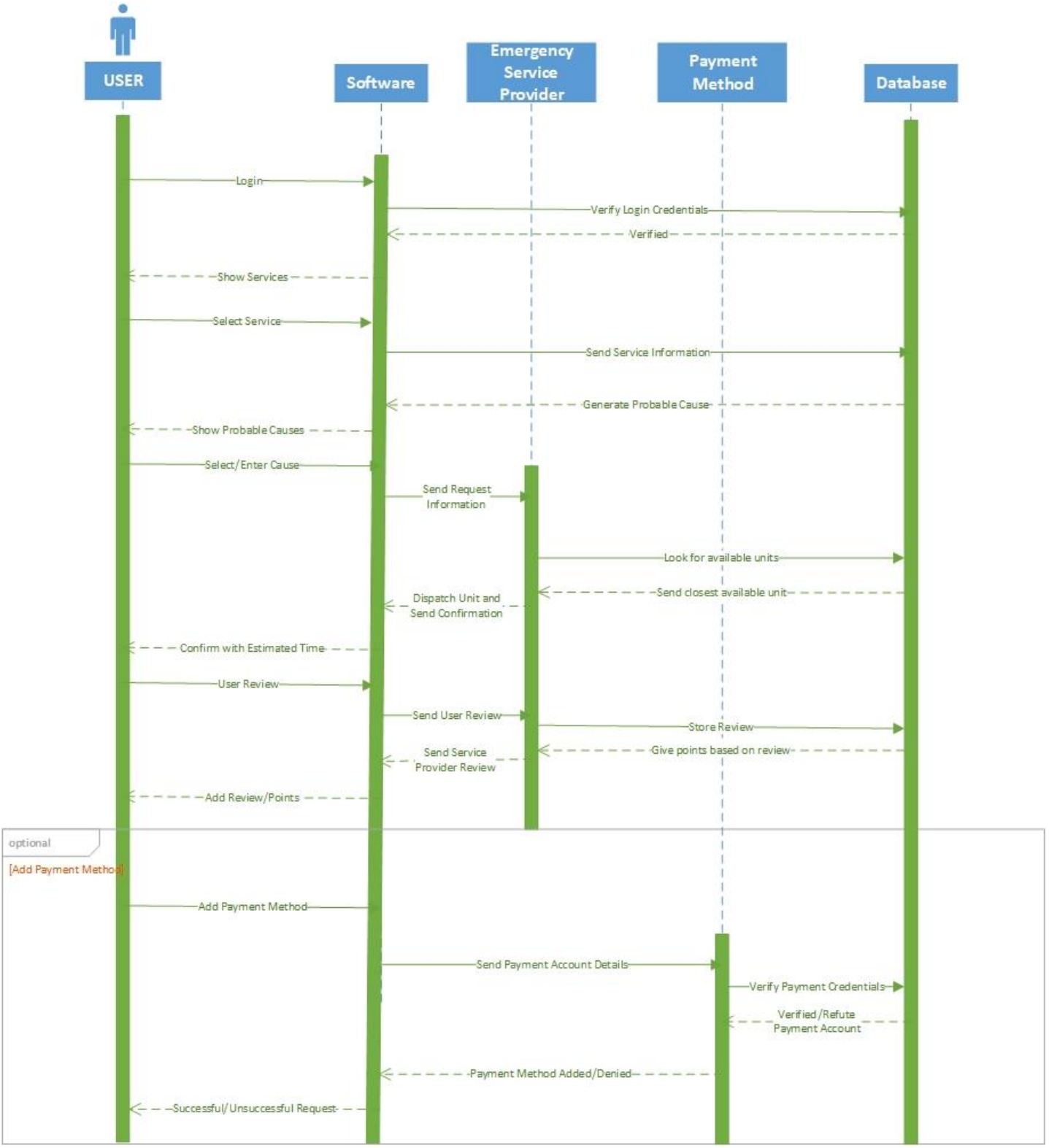
Use Case Diagram



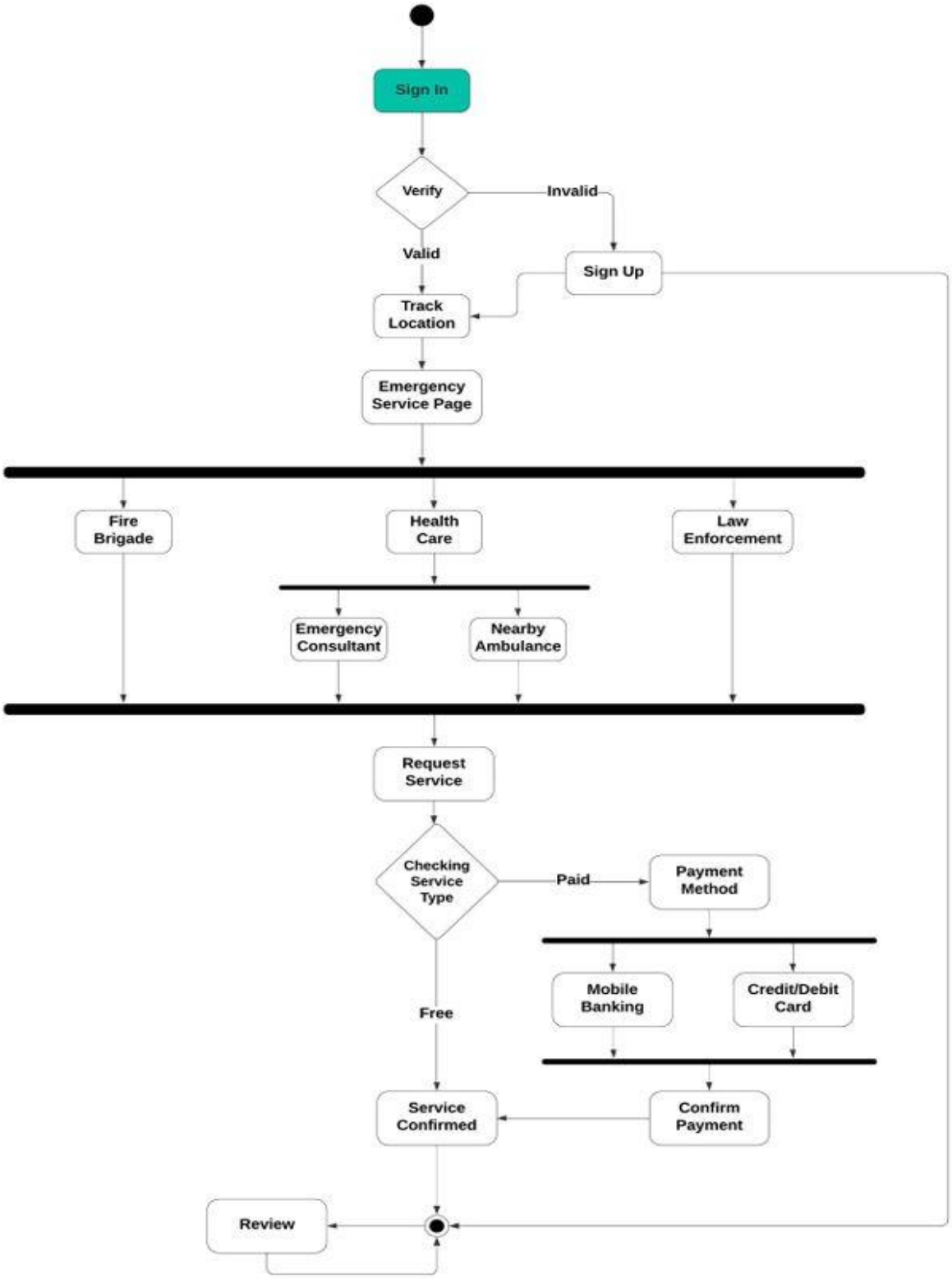
Class Diagram



Sequence Diagram

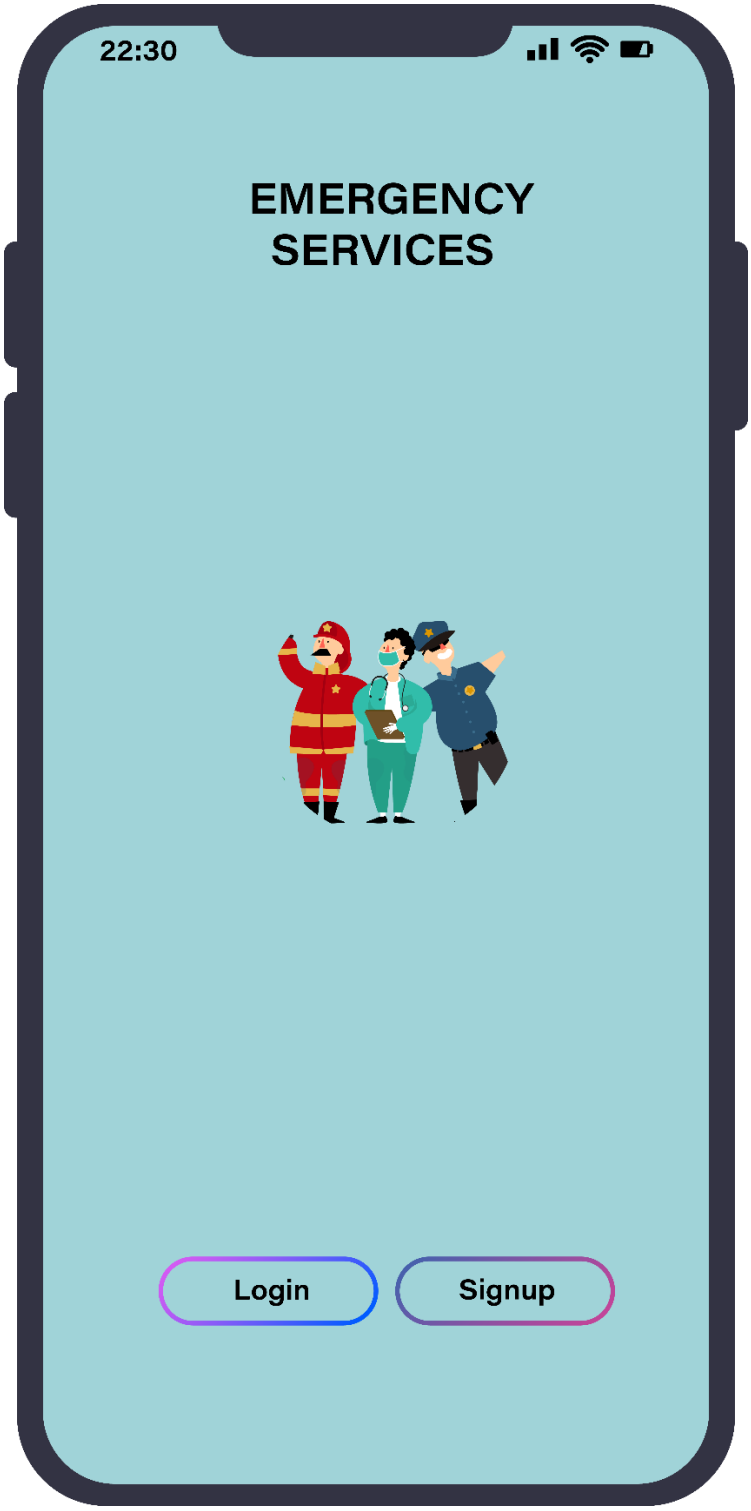


Activity Diagram

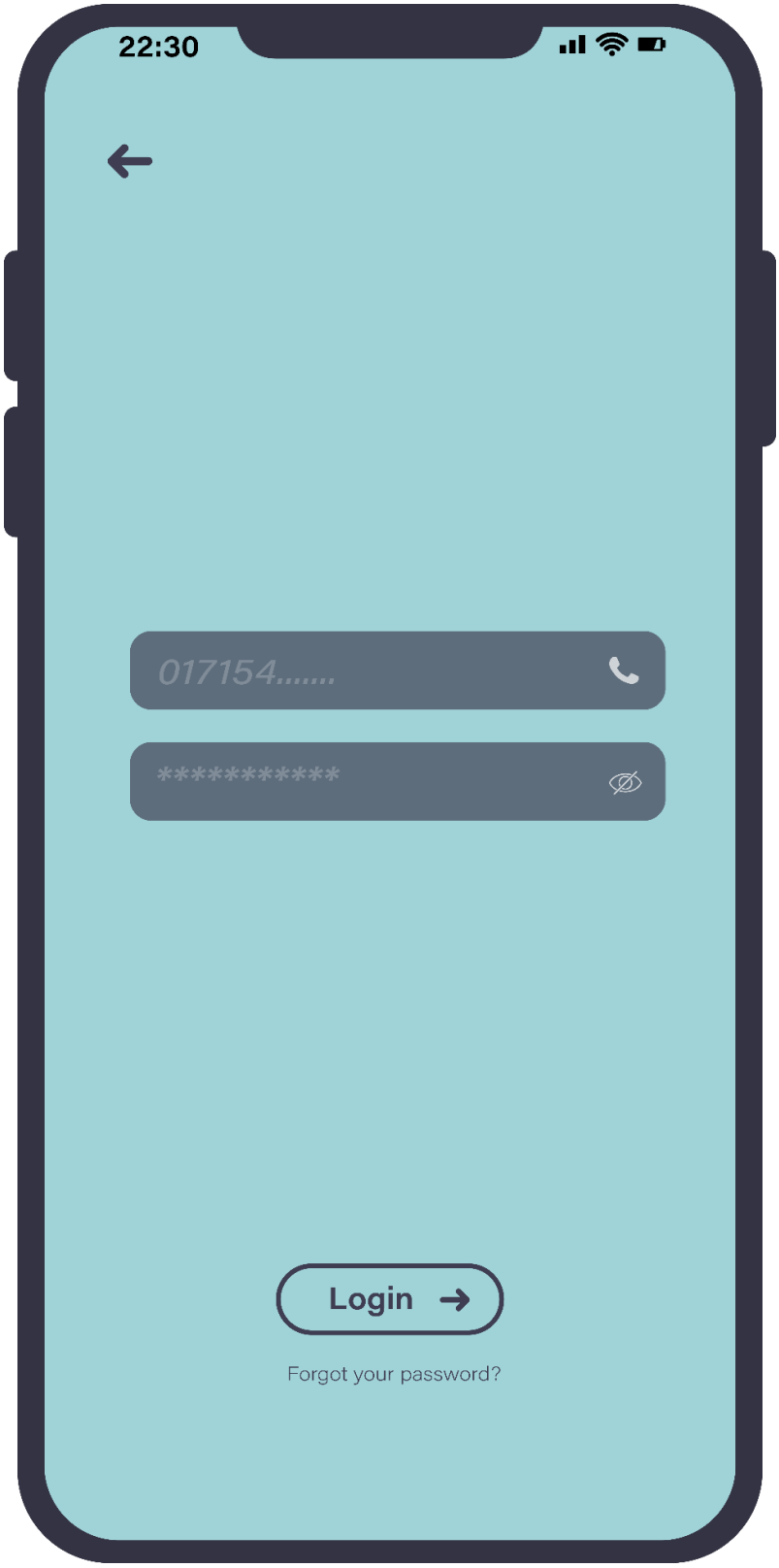


UI / UX

Starting Screen:



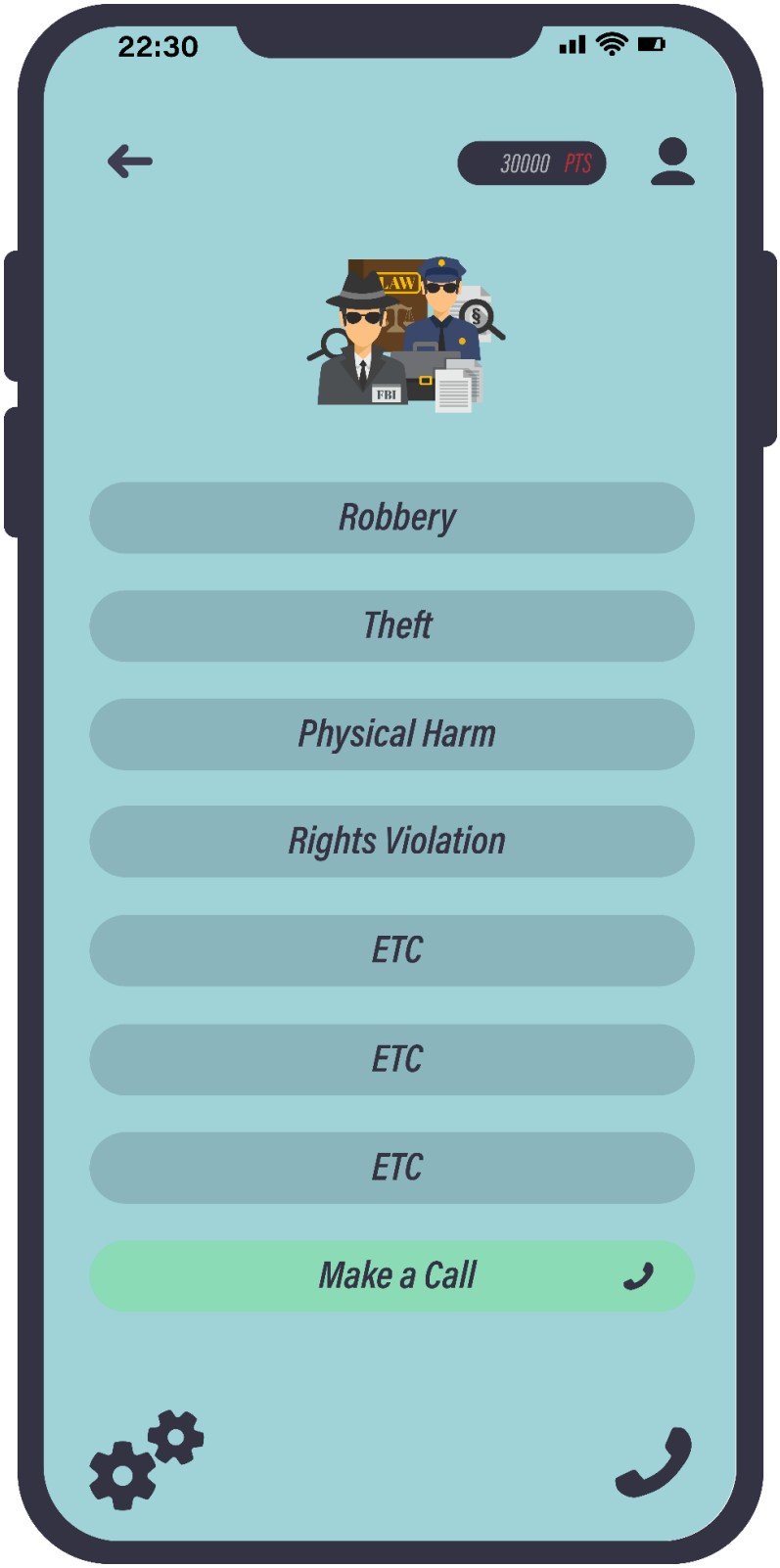
Login:



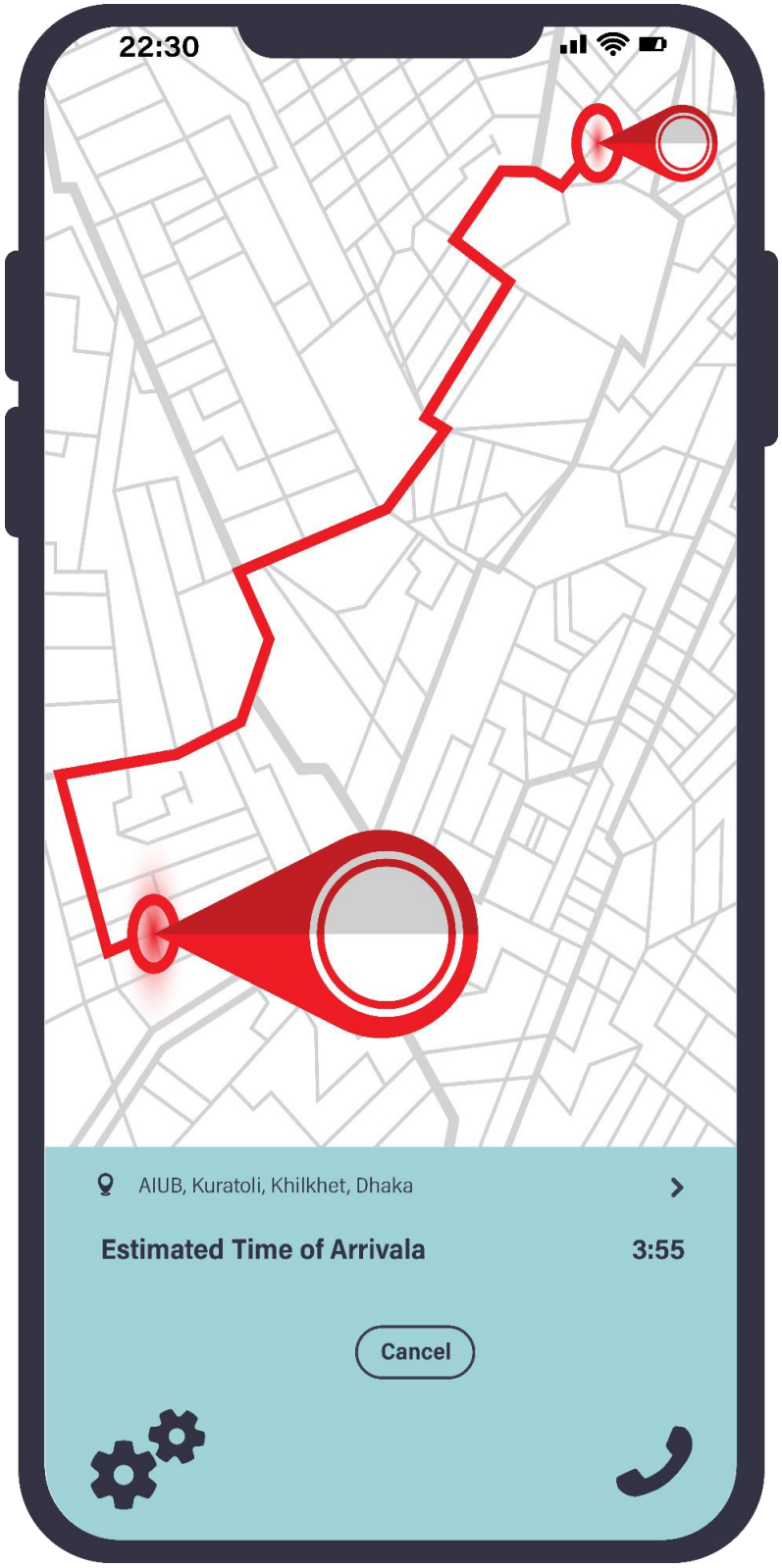
Homepage:


22:30
30000 PTS
AIUB, Kuratoli, Khilkhet, Dhaka
Law Enforcement
Health
Fire Brigade
YOUR SAFETY IS OUR RESPONSIBILITY

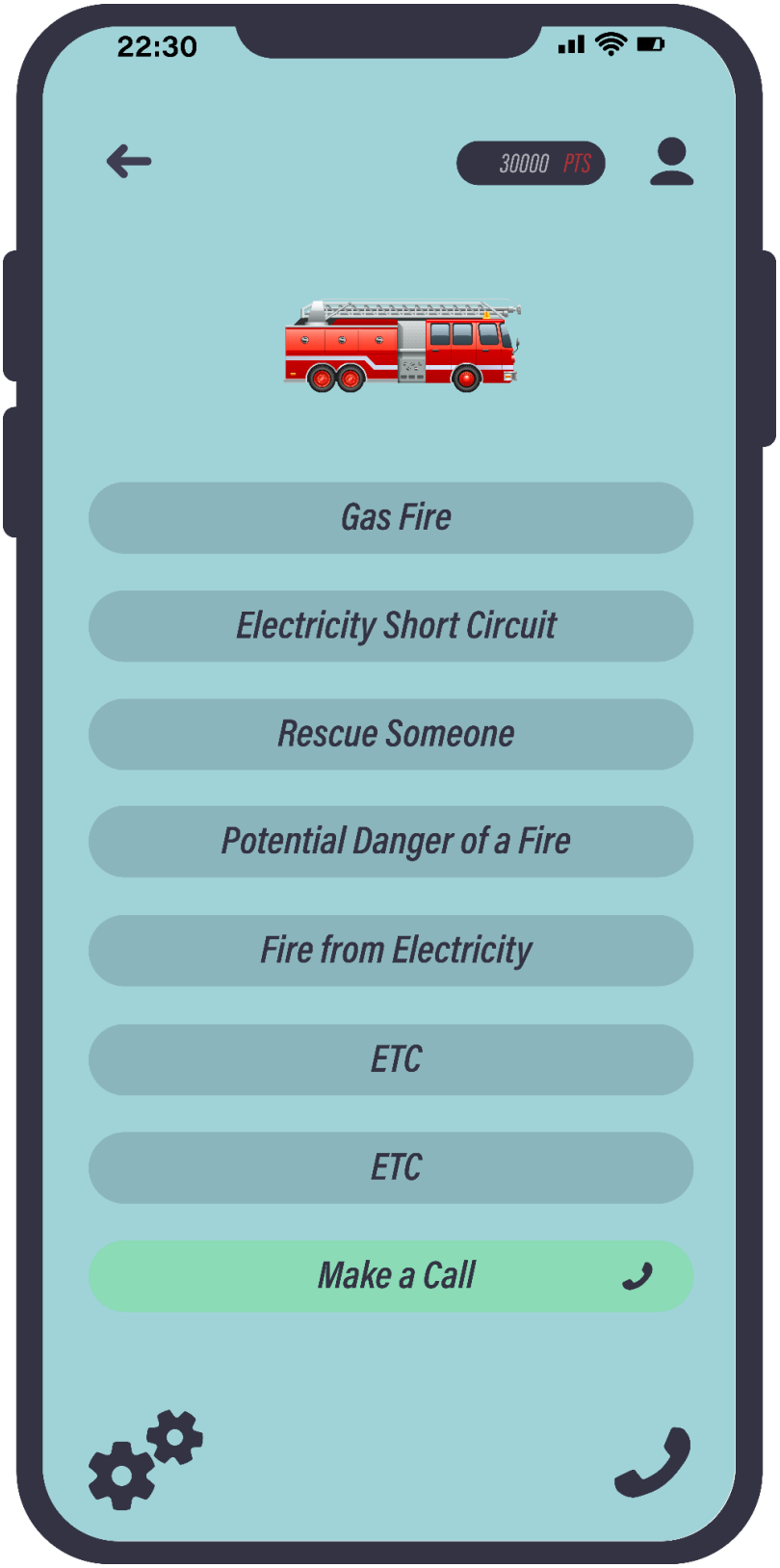
Law Enforcement:



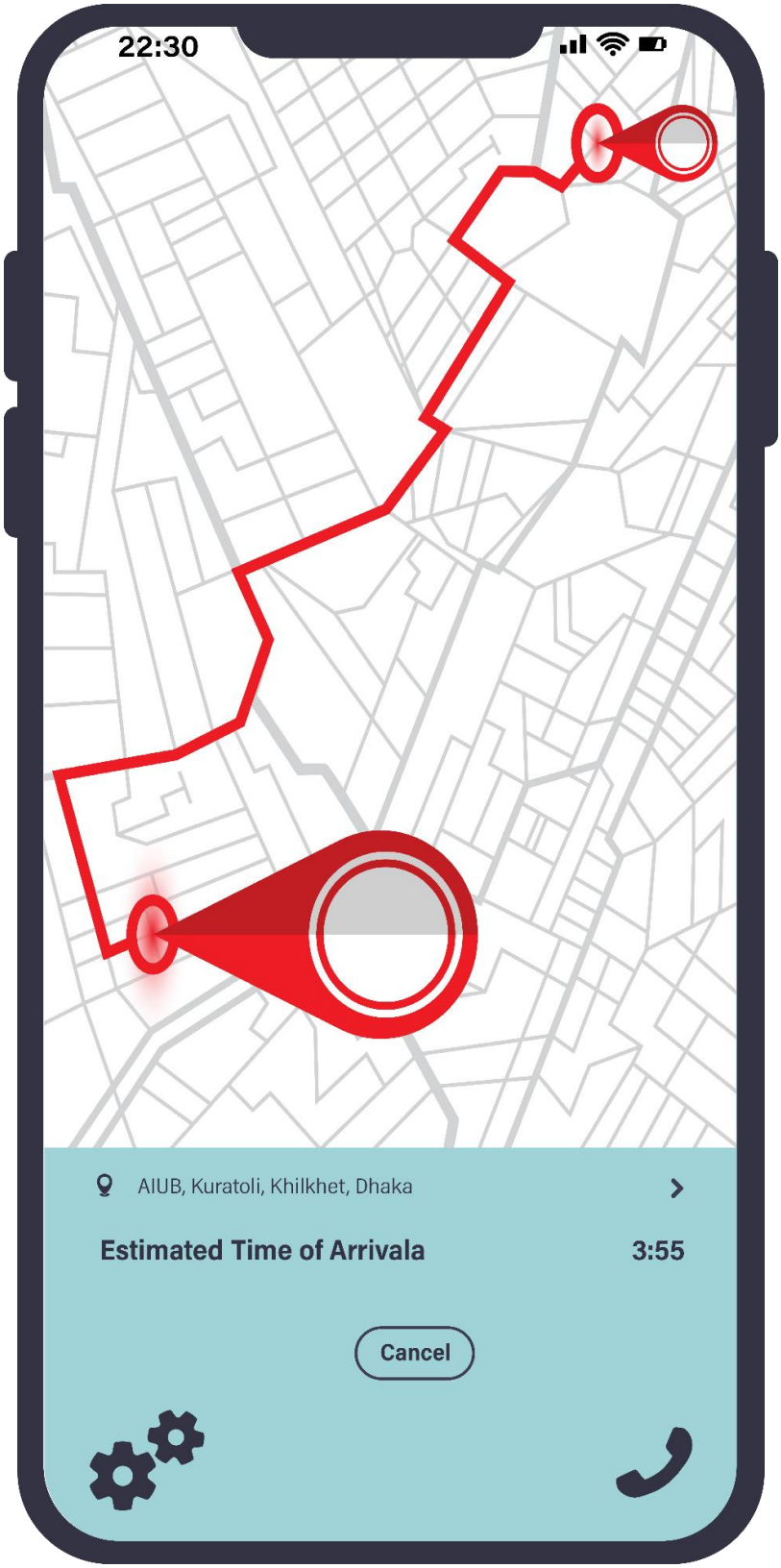
Realtime Location of Law Enforcement:



Fire Brigade:

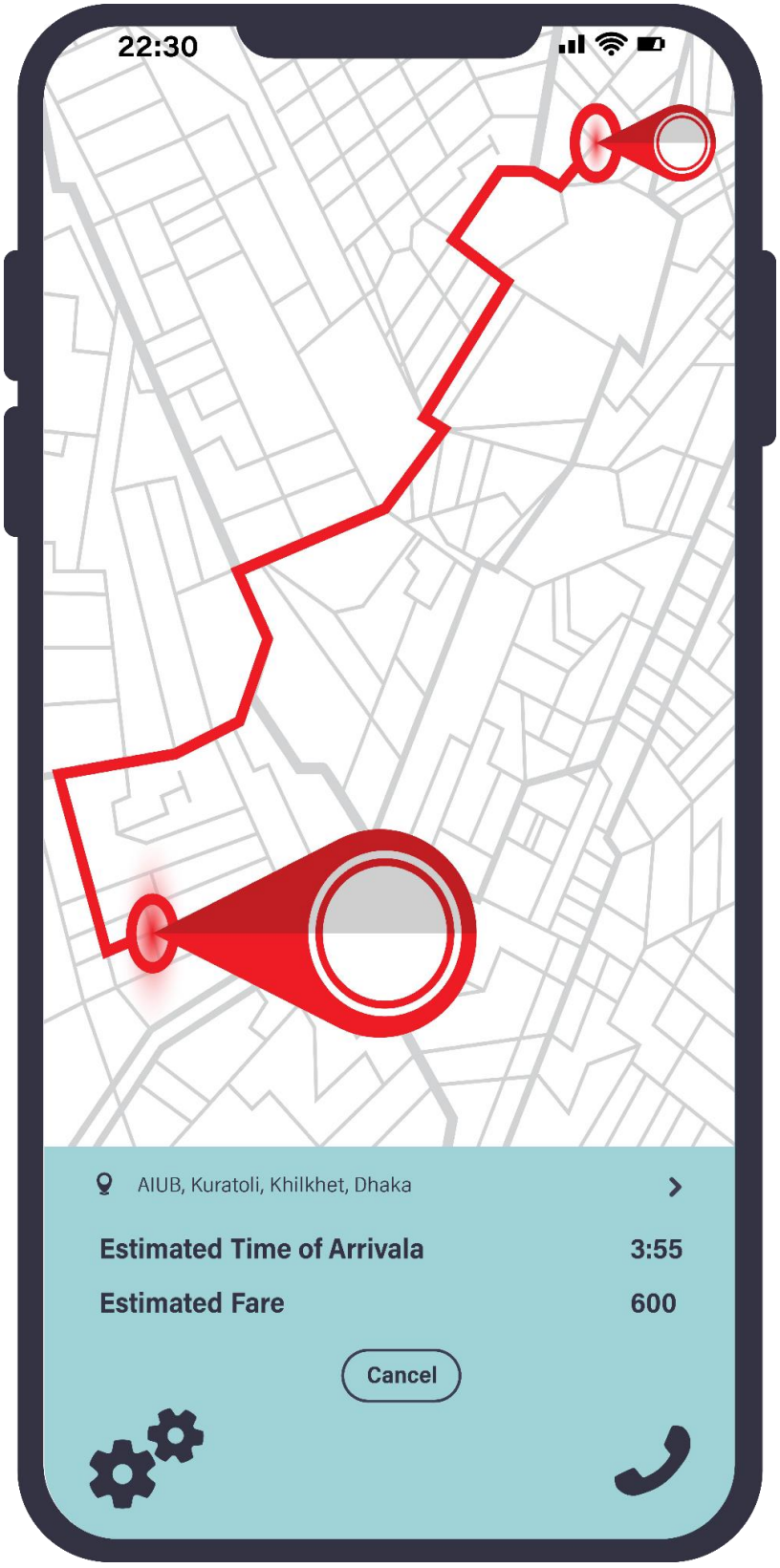


Realtime Location of Fire Brigade:

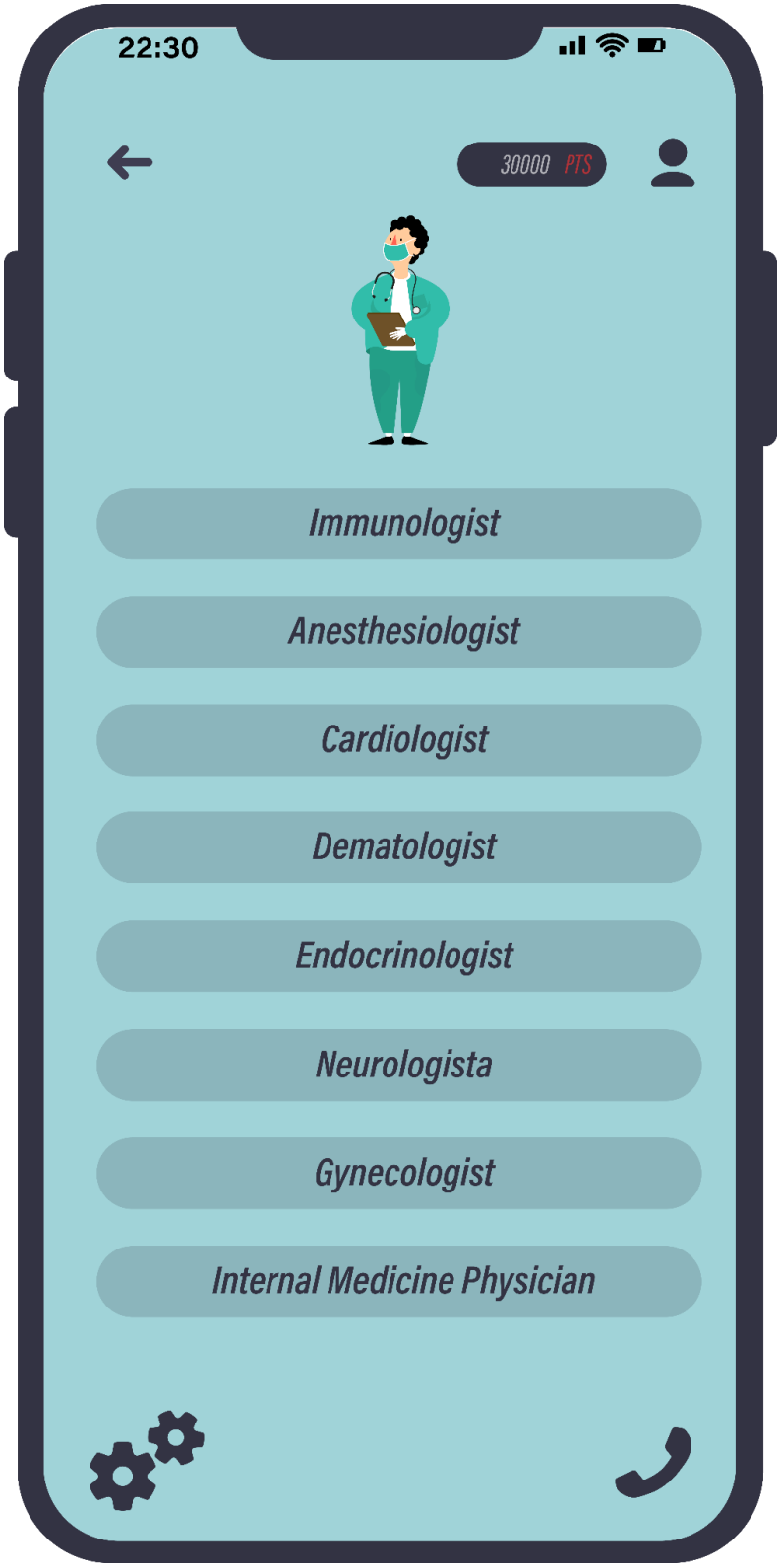


Health Care Services:

Nearby Ambulance:



Consult Specialist:



Test Cases

Project Name: Emergency Services			Test Designed by: Sadia Afrose	
Test Case ID: FR_1 Test Priority (Low, Medium, High): Medium			Test Designed date: 3/ 30/ 2024 Test Executed by:	
Module Name: Software Login			Test Execution date:	
Test Title: Verify Valid User Login				
Description: This test case ensures that the user can successfully login into the account with valid details in the Emergency Services app.				
Precondition (If any): User must have valid phone number and password (or fingerprint).				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the app 2. Enter phone number 3. Enter password 4. Click login	Phone Number: 0134567890 Password: 321	User should login into the application		
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

Project Name: Emergency Services			Test Designed by: Sadia Afrose	
Test Case ID: FR_2 Test Priority (Low, Medium, High): Medium			Test Designed date: 3/ 30/ 2024 Test Executed by:	
Module Name: Software Login			Test Execution date:	
Test Title: Verify Valid User Login				
Description: This test case ensures that the user can successfully login into the account with valid details in the Emergency Services app.				
Precondition (If any): User must have valid phone number and password (or fingerprint).				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the app 2. Enter phone number 3. Enter password 4. Click login	Phone Number: 0134567890 Password: 321	User should login into the application		
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

Project Name: Emergency Services			Test Designed by: Fatima Adon	
Test Case ID: FR_3 Test Priority (Low,Medium,High): Medium			Test Designed date:3/ 30/ 2024 Test Executed by:	
Module Name: User Registration			Test Execution date:	
Test Title: Verify Valid User Registration				
Description: This test case ensures that the user can successfully register an account with valid details in the Emergency Services app.				
Precondition (If any): 1. Valid internet connection				
Test Steps 1.Launch the Emergency Services app. 2.Navigate to the "Register" section. 3. Enter a valid phone number in the number field. 4. Create a username that meets the app's criteria. 5. Enter a password that meets the app's security requirements. 6.Confirm the password by re-entering it in the designated field. 7.Tap the "Register" button.	Test Data 1.Number:01********* 2.Username: TestUser2024 3.Password: ValidPassword123!*	Expected Results Information's are saved in the database.	Actual Results	Status (Pass/Fail)
				a
Post Condition: The user account is created and can be used to login. The database updates with the new user's information.				

Project Name: Emergency Services			Test Designed by: Fatima Adon	
Test Case ID: FR_4 Test Priority (Low,Medium,High): Medium			Test Designed date:3/ 30/ 2024 Test Executed by:	
Module Name: User Registration			Test Execution date:	
Test Title: Verify Invalid User Registration				
Description: This test case aims to ensure that the system properly handles registration attempts using an invalid email format, preventing the registration from proceeding.				
Precondition (If any): 1. User has valid internet connection				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Open the Emergency Services app on the device. 2. Navigate to the "Register" or "Sign Up" section of the app. 3. Enter a username that meets the app's requirements. 4. Enter an invalid number address in the number field (e.g., "012**"). 5. Choose a password that meets the app's security criteria. 6. Re-enter the password for confirmation. 7. Click or tap the "Register" button.	Username: NewUser2024 Number: “012**” (Invalid format) Password: SecurePass!123	No changes occur. User is prompted to do registration again.		
Post Condition: No new user account is created in the system database. The user is informed about the invalid email format and given the opportunity to correct it.				

Project Name: Emergency Services			Test Designed by: Sadia Afrose	
Test Case ID: FR_5 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Fire Brigade Session			Test Execution date:	
Test Title: Dispatch fire brigade unit				
Description: Test fire brigade dispatch				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select Fire Brigade 3. Select cause of request 4. Select severity of emergency 5. Assign units according to the severity of the emergency	Request: Fire Incident Severity: High	Fire brigade units should be dispatched to the user's location based on the severity of the emergency		
Post Condition: 1. The request is validated and stored in the database. 2. Dispatched units reached the location of the user.				

Project Name: Emergency Services			Test Designed by: Fatima Adon	
Test Case ID: FR_6 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Fire Brigade Session			Test Execution date:	
Test Title: Request fire brigade unit with an unlisted cause.				
Description: Request fire brigade unit when the cause is not listed in the probable cause list.				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select Fire Brigade 3. Call the hotline 4. Tell and explain the new cause. 4. Select severity of emergency	Request: Explain unlisted cause Severity: Medium	Fire brigade units should be dispatched to the user's location based on the severity of the emergency		
Post Condition: 1. The request is validated and stored in the database. 2. Dispatched units reached the location of the user.				

Project Name: Emergency Services			Test Designed by: Moinul Hasan	
Test Case ID: FR_7 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Law Enforcement Session			Test Execution date:	
Test Title: Dispatch law enforcement unit				
Description: Test law enforcement dispatch				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select Law Enforcement 3. Select cause of request 4. Select severity of emergency 5. Assign units according to the severity of the emergency	Request: Law Enforcement Cause: Life Threatening situation Severity: Low	Law enforcement units should be dispatched to the user's location based on the severity of the emergency		
Post Condition: 1. The request is validated and stored in the database. 2. Dispatched units reached the location of the user.				

Project Name: Emergency Services			Test Designed by: Moinul Hasan	
Test Case ID: FR_8 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Fire Brigade Session			Test Execution date:	
Test Title: Request law enforcement unit with an unlisted cause.				
Description: Request law enforcement unit when the cause is not listed in the probable cause list.				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select Law Enforcement 3. Call the hotline 4. Tell and explain the new cause. 4. Select severity of emergency	Request: Law Enforcement Cause: Explain unlisted cause Severity: Medium	Law Enforcement units should be dispatched to the user's location based on the severity of the emergency		
Post Condition: 1. The request is validated and stored in the database. 2. Dispatched units reached the location of the user.				

Project Name: Emergency Services			Test Designed by: MD. Naimul Islam	
Test Case ID: FR_9 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Generate Probable Causes			Test Execution date:	
Test Title: Generate probable causes using previous data				
Description: When requesting a service, the probable causes of the service will be generated in a list.				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled 4. User selected one of the three emergency services				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select an emergency service 3. Get previous causes from database 4. Show those causes	Emergency Service: Fire Brigade	Multiple probable causes will be displayed in order to make the experience smoother for the user.		
Post Condition: User selects a cause from the generated list of causes				

Project Name: Emergency Services			Test Designed by: MD. Naimul Islam	
Test Case ID: FR_10 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Health Care Session			Test Execution date:	
Test Title: Request emergency consultant				
Description: Request emergency consultant after choosing from listed specialists.				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select Heath Care services 3. Select Emergency Consultant 4. Choose one from the list of specialists 5. Confirm your request	Specialist: Dermatologist	User is connected with the desired specialist		
Post Condition: 1. The information of the request is stored in the database. 2. A video call is conducted between the user and the selected specialist.				

Project Name: Emergency Services			Test Designed by: Moinul Hasan	
Test Case ID: FR_11 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Health Care Session			Test Execution date:	
Test Title: Request Nearby Ambulance				
Description: Request nearby ambulance within a kilometer of radiusa				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select Heath Care services 3. Select Nearby Ambulance 4. Assign Ambulance to the user's location 6. Select the location of the hospital 5. Show amount of fare. 6. Confirm request	Estimated Time of Arrival: 4 minutes Fare: 400	User is assigned an ambulance and ambulance is reached to the location within 10 minutes of time.		
Post Condition: 1. The information of the request is stored in the database. 2. Patient is provided with health care facilities present inside the ambulance.				

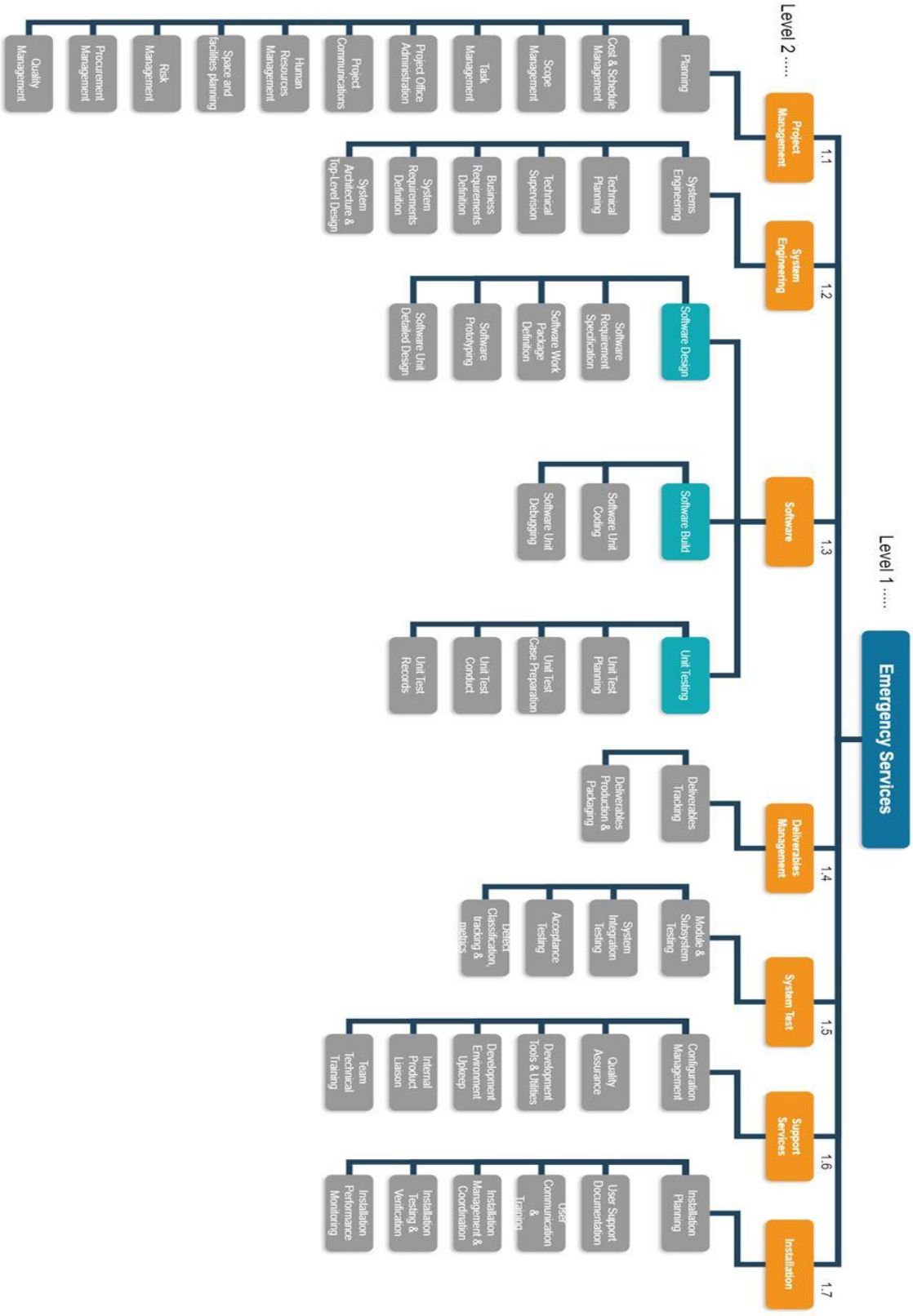
Project Name: Emergency Services			Test Designed by: MD. Naimul Islam	
Test Case ID: FR_12 Test Priority (Low,Medium,High): High			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Health Care Session			Test Execution date:	
Test Title: Nearby Ambulance not found within 5 minutes of waiting time				
Description: Nearby ambulance within a kilometer of radius is not being found				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Select Heath Care services 3. Select Nearby Ambulance 4. Waiting period for the ambulance exceeds 5 minutes	Request: Health Care -> Nearby Ambulance	The request is cancelled and control is sent to the home screen.		
Post Condition: 1. The information of the request is stored in the database for future Improvements.				

Project Name: Emergency Services			Test Designed by: Ahsanul Haque Joar	
Test Case ID: FR_13 Test Priority (Low,Medium,High): Medium			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: After Service Review			Test Execution date:	
Test Title: Set review after receiving or providing a service				
Description: User or service provider provides a review about the each other after the request has been successfully conducted				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled 4. A request is successfully completed				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Request a service 3. Service is completed 4. Enter the review when asked	Review: 4 stars	The review is saved in the database and points are added to the respective accounts.		
Post Condition: 1. The review is stored in the database for points evaluation.				

Project Name: Emergency Services			Test Designed by: Ahsanul Haque Joar	
Test Case ID: FR_14 Test Priority (Low, Medium, High): Medium			Test Designed date: 3/ 30/ 2024 Test Executed by:	
Module Name: Track Last and Realtime Location			Test Execution date:	
Test Title: Track location				
Description: This test case ensures that the user can successfully login into the account with valid details in the Emergency Services app and track last and realtime Location of the user for better service .				
Precondition (If any): User must have valid phone number and password (or fingerprint).				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the app 2. Enter phone number 3. Enter password 4. Click login	Phone Number: 0134567890 Password: 321	User location shared in the database	Location traced	pass
Post Condition: User is validated with database and successfully ready to take emergency services.				

Project Name: Emergency Services			Test Designed by: Ahsanul Haque Joar	
Test Case ID: FR_15 Test Priority (Low,Medium,High): Medium			Test Designed date: 30/ 3/ 2024 Test Executed by:	
Module Name: Penalty Generation			Test Execution date:	
Test Title: Generate penalties based on the points				
Description: Generated penalties based on the points from the review given by the users or the service providers				
Precondition (If any): 1. Valid internet connection 2. User logged in to the system 3. Location service is enabled 4. Point crosses the set limit for penalties				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login to the system 2. Request a service 3. Service is completed 4. Points crosses the limit of penalties	Points: -20000	The software shall not let any further request from the respective users before they pay the penalized amount and clear the negative points from their account.		
Post Condition: 1. No further request can be conducted before the payment of the penalized amount.				

WBS (Work Breakdown Structure)



COCOMO

$$PM = 2.4 * 6^{1.05}$$

$$= 15.750$$

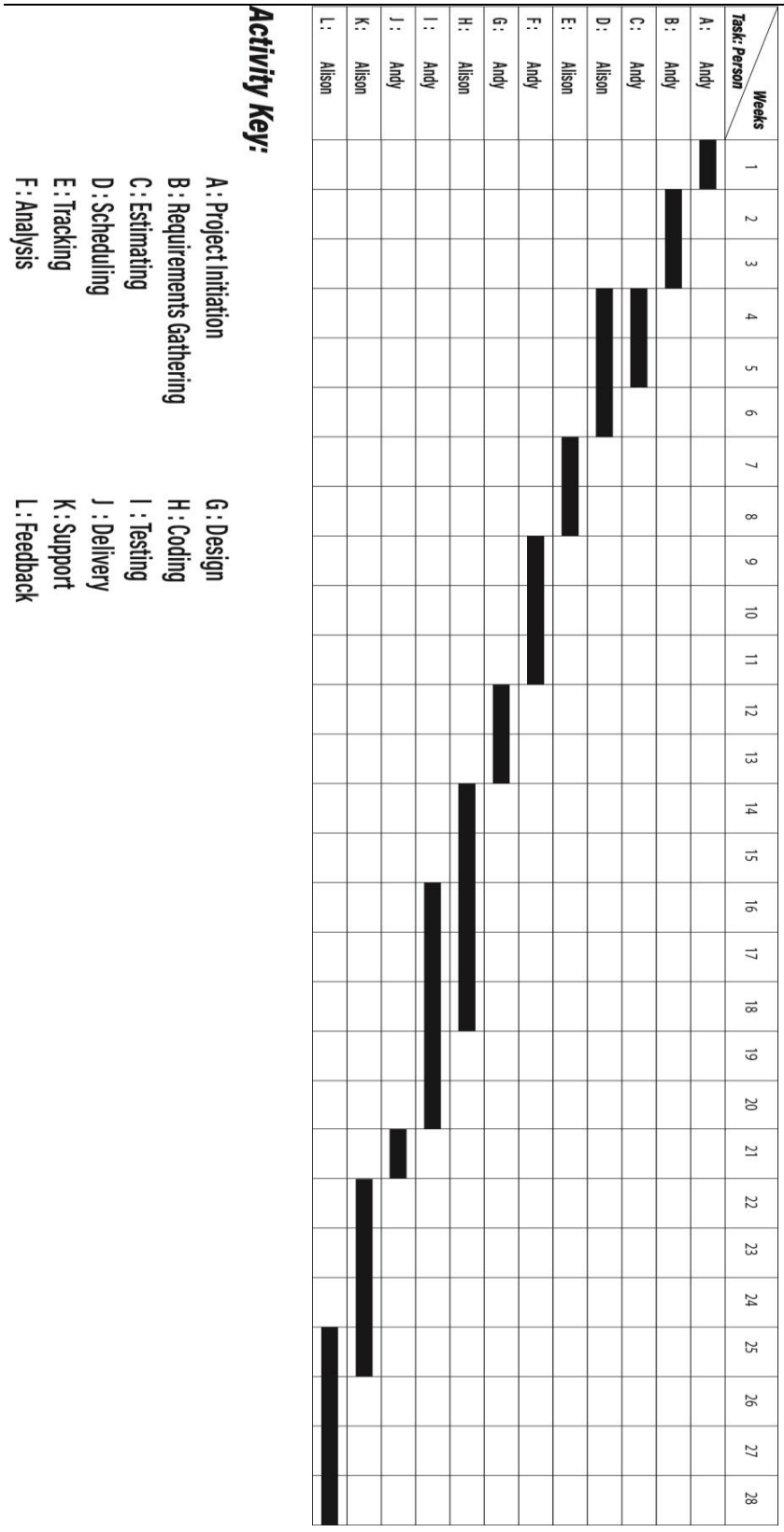
$$DM = 2.5 * (PM)^{0.38}$$

$$= 7.127$$

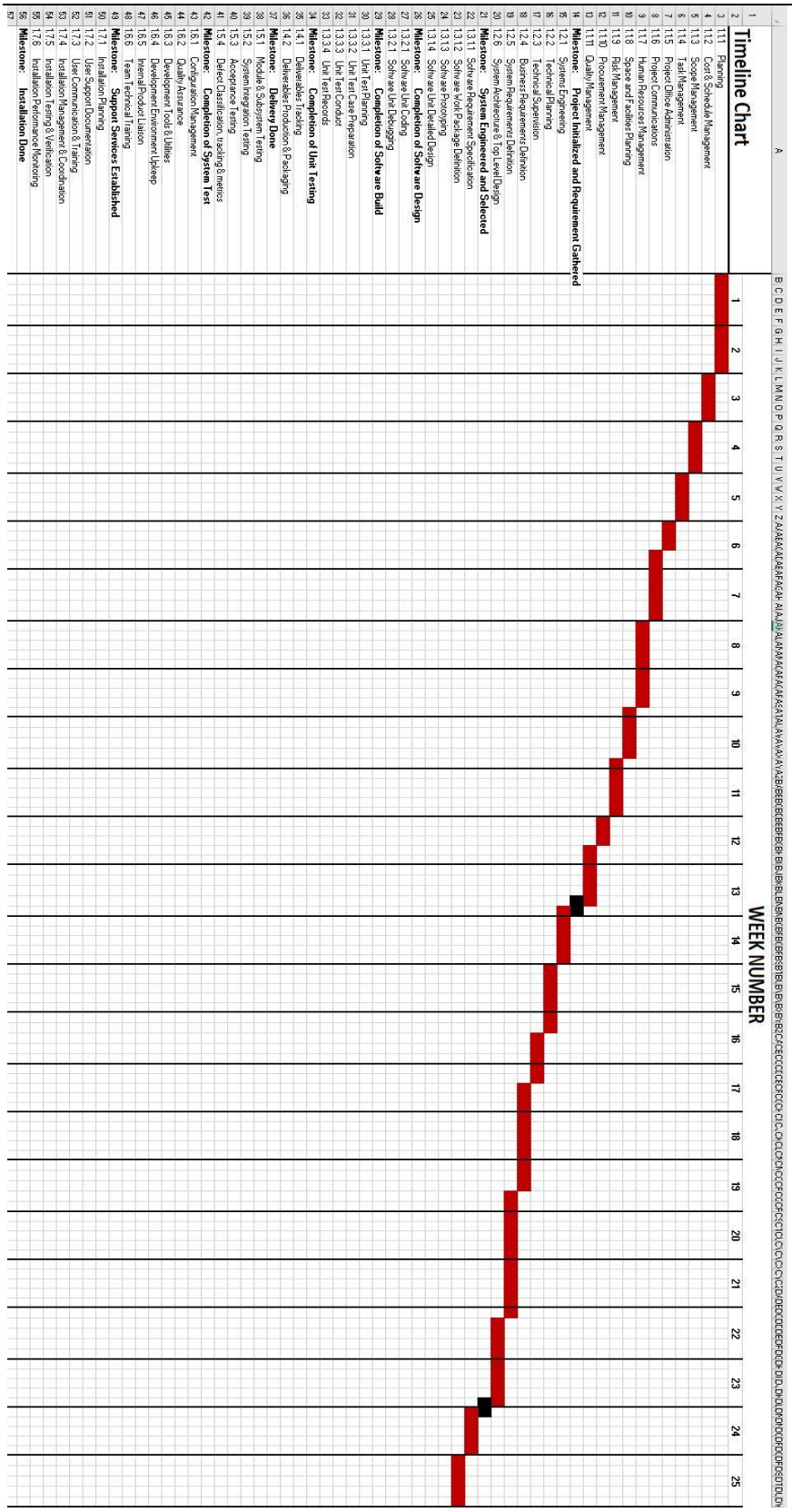
$$ST = \frac{PM}{DM} = 2.210 \approx 2$$

//

Timeline 1


Task: Person
Weeks
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
A: Andy
B: Andy
C: Andy
D: Alison
E: Alison
F: Andy
G: Andy
H: Alison
I: Andy
J: Andy
K: Alison
L: Alison
Activity Key:
A: Project Initiation
B: Requirements Gathering
C: Estimating
D: Scheduling
E: Tracking
F: Analysis
G: Design
H: Coding
I: Testing
J: Delivery
K: Support
L: Feedback

Timeline 2



Timeline Chart
WEEK NUMBER
1.1.1 Planning
1.1.2 Cost & Schedule Management
1.1.3 Scope Management
1.1.4 Task Management
1.1.5 Project Office Administration
1.1.6 Project Communications
1.1.7 Human Resources Management
1.1.8 Space and Facilities Planning
1.1.9 Risk Management
1.1.10 Procurement Management
1.1.11 Quality Management
1.2 Milestone: Project Initialized and Requirement Gathe
1.2.1 Systems Engineering
1.2.2 Technical Planning
1.2.3 Technical Supervision
1.2.4 Business Requirements Definition
1.2.5 System Requirements Definition
1.2.6 System Architecture & Top Level Design
1.3 Milestone: System Engineered and Selected
1.3.1.1 Software Requirement Specification
1.3.1.2 Software Work Package Definition
1.3.1.3 Software Prototyping
1.3.1.4 Software Unit Detailed Design
1.3 Milestone: Completion of Software Design
1.3.2.1 Software Unit Coding
1.3.2.2 Software Unit Debugging
1.3 Milestone: Completion of Software Build
1.3.3.1 Unit Test Planning
1.3.3.2 Unit Test Case Preparation
1.3.3.3 Unit Test Conduct
1.3.3.4 Unit Test Records
1.4 Milestone: Completion of Unit Testing
1.4.1 Deliverables Tracking
1.4.2 Deliverables Production & Packaging
1.5 Milestone: Delivery Done
1.5.1 Module & Subsystem Testing
1.5.2 System Integration Testing
1.5.3 Acceptance Testing
1.5.4 Defect Classification, tracking & metrics
1.6 Milestone: Completion of System Test
1.6.1 Configuration Management
1.6.2 Quality Assurance
1.6.3 Development Tools & Utilities
1.6.4 Development Environment Upkeep
1.6.5 Internal Product Liaison
1.6.6 Team Technical Training
1.7 Milestone: Support Services Established
1.7.1 Installation Planning
1.7.2 User Support Documentation
1.7.3 User Communication & Training
1.7.4 Installation Management & Coordination
1.7.5 Installation Testing & Verification
1.7.6 Installation Performance Monitoring
1.8 Milestone: Installation Done

Timeline Chart
WEEK NUMBER
1.1.1 Planning
1.1.2 Cost & Schedule Management
1.1.3 Scope Management
1.1.4 Task Management
1.1.5 Project Office Administration
1.1.6 Project Communications
1.1.7 Human Resources Management
1.1.8 Space and Facilities Planning
1.1.9 Risk Management
1.1.10 Procurement Management
1.1.11 Quality Management
Milestone: Project Initialized and Requirement Gathe
1.2.1 Systems Engineering
1.2.2 Technical Planning
1.2.3 Technical Supervision
1.2.4 Business Requirements Definition
1.2.5 System Requirements Definition
1.2.6 System Architecture & Top Level Design
Milestone: System Engineered and Selected
1.3.1.1 Software Requirement Specification
1.3.1.2 Software Work Package Definition
1.3.1.3 Software Prototyping
1.3.1.4 Software Unit Detailed Design
Milestone: Completion of Software Design
1.3.2.1 Software Unit Coding
1.3.2.1 Software Unit Debugging
Milestone: Completion of Software Build
1.3.3.1 Unit Test Planning
1.3.3.2 Unit Test Case Preparation
1.3.3.3 Unit Test Conduct
1.3.3.4 Unit Test Records
Milestone: Completion of Unit Testing
1.4.1 Deliverables Tracking
1.4.2 Deliverables Production & Packaging
Milestone: Delivery Done
1.5.1 Module & Subsystem Testing
1.5.2 System Integration Testing
1.5.3 Acceptance Testing
1.5.4 Defect Classification, tracking & metrics
Milestone: Completion of System Test
1.6.1 Configuration Management
1.6.2 Quality Assurance
1.6.3 Development Tools & Utilities
1.6.4 Development Environment Upkeep
1.6.5 Internal Product Liaison
1.6.6 Team Technical Training
Milestone: Support Services Established
1.7.1 Installation Planning
1.7.2 User Support Documentation
1.7.3 User Communication & Training
1.7.4 Installation Management & Coordination
1.7.5 Installation Testing & Verification
1.7.6 Installation Performance Monitoring
Milestone: Installation Done

EVA

Task	Planned Effort	Actual Effort
1	10.0	11.2
2	5.0	4.5
3	5.0	6.5
4	5.0	7.3
5	3.0	2.9
6	7.0	8.5
7	9.0	10.2
8	5.0	8.3
9	6.0	—
10	3.0	—

BCWS = 58.0

BCWP = 49.0

ACWP = 59.4

- BAC = 15.75 x 20 = 315
- SPI = 49/58 = 0.84
- SV = 49-58 = -9 person-day
- CPI = 49/59.4 = 0.82
- CV = 49-59.4 = -10.4 = -10 person-day

- %schedule for completion = 58/315 = 18.41%

- %complete = 49/315 = 15.55%

Risk Table

Risks	Category	Probability	Impact
Size estimate may be significantly low	PS	40%	2
Delivery deadline will be tightened	BU	30%	3
Staff inexperienced	ST	20%	2
Staff turnover will be high	ST	50%	2
Unauthorized Access to the system	ST	10%	1
The client is not available to the dev team	BU	10%	4
Unstable workload	DE	10%	3
Integration of new, unproven technologies	TE	15%	3
Lack of training on tools	DE	70%	3
Technology will not meet expectation	TE	20%	1

Impact Values:

1 – Catastrophic

2 – Critical

3 – Marginal

4 – Negligible