



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Community First: Efficient Utilities & Primary Health Suite

A Software Requirement Engineering Project Submitted
By

Semester: Fall_23_24		Section: C	Group Number: 08	
SN	Student Name	Student ID	Contribution in % (CO+CO2)	Individual Marks
20	Satyajit Das	21-44374-1		
09	Syed Foysal	20-42505-1		
17	Md Akramul Islam	20-43345-1		
07	Md Minhajur Rahman Khan	19-41777-3		

The project will be Evaluated for the following Course Outcomes

EVALUATION CRITERIA	Total Marks (50)	
Introduction, Format, Submission, Defense	[10 Marks]	
System Overall Description & Functional Requirements	[10 Marks]	
System Quality Attributes and Project Requirements	[10 Marks]	
UML and E-R Diagram with Data Dictionary	[10 Marks]	
UI/UX Prototyping	[10 Marks]	

Software Requirements Specification

for

<Community First: Efficient Utilities & Primary Health Suite>

Community First_V_1.0

**Prepared by <Das, Satyajit, Foysal, Syed, Islam, Md Akramul, Khan, Md
Minhajur Rahman>**

<American International University-Bangladesh (AIUB)>

<December 24, 2023>

Table of Contents

Revision History	3
1. Introduction.....	4
1.1 Purpose.....	4
1.2 Document Conventions.....	5
1.3 Intended Audience and Reading Suggestions	5
1.4 References.....	6
2. Overall Description	6
2.1 Product Perspective.....	6
2.2 Product Functions	7
2.3 User Classes and Characteristics.....	8
2.4 Operating Environment.....	9
2.5 Design and Implementation Constraints	9
2.6 User Documentation	10
3. System Requirements.....	12
3.1 System Features	12
3.2 Non-Functional/Quality Requirements.....	14
3.3 Project Requirements	16
4. Design and Interface Requirements	17
4.1 UML Diagrams	17
4.2 Data Dictionary	21
4.3 UI/UX Design Specification	23

Revision History

Name	Date	Reason for Changes	Version
N/A	N/A	N/A	N/A

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document details the requirements for “Community First: Efficient Utilities & Primary Health Suite” application, referred to as 'Community First'. This document pertains to the initial release, Version 1.0, of the software.

The scope of this SRS encompasses the complete functionality of the Community First software. It covers the application as a whole, including its integrated utility management solutions and primary health information services, as well as the associated database systems. This SRS does not encompass external systems but focuses exclusively on the software components of Community First.

‘Community First’ is a software solution designed to address critical urban and municipal challenges, namely utilities outages (electricity, gas, and water) and the provision of primary health-related information and services. The software aims to provide users with timely and accurate information on utilities outages, allowing for efficient resource planning and usage. Additionally, it serves as a platform for essential health-related news, notifications, and affordable consultation services with medical professionals. This dual-focus approach aligns with corporate goals of enhancing community living standards through technological innovation and support.

The software aligns with broader corporate objectives of leveraging technology for societal benefit. By providing a reliable and user-friendly platform to manage utility outages and health services, 'Community First' contributes towards improving the quality of life in urban and municipal areas. This initiative is part of our commitment to sustainable development and community welfare.

The business requirements for "Community First: Efficient Utilities & Primary Health Suite" cater to both customer needs and the company's objectives. For customers, the software must provide real-time updates on electricity, gas, and water outages, accompanied by alerts and historical usage data analysis for efficient resource planning. Additionally, it should offer health-related news, affordable consultation services with medical professionals, and appointment scheduling. Flexible subscription plans, including monthly, half-yearly, and yearly options with corresponding benefits, are imperative. The platform should feature an intuitive design, customizable preferences, and accessibility across devices.

For the company side of things, "Community First: Efficient Utilities & Primary Health Suite" needs to make money through subscription plans to keep investors and the company happy. It's important to pay consultants fairly while making sure they provide good service. To keep customers happy and coming back, the software needs smart ways to reach them through ads or messages and offer great customer service. Building relationships with utility companies and healthcare providers is key to making the software better and reaching more people. By doing all these things, the software can make money, grow well, and keep everyone involved satisfied.

1.2 Document Conventions

The document conventions outlined for the SRS of "Community First: Efficient Utilities & Primary Health Suite" aim to maintain consistency and clarity throughout:

Font Style and Size:

- The entire document adheres to Times New Roman font with a 12-point size for uniformity and readability.

Headings:

- Major headings are presented in bold text for clear identification.
- Subheadings are written in bold using sentence case to delineate different sections effectively.

Prioritization of Requirements:

- Each requirement statement in the document is assigned a specific priority level as determined by the customer.
- The priority levels for higher-level requirements do not automatically transfer to detailed requirements, ensuring distinct prioritization.

Use of Formatting:

- Important terms, system components, user interface elements, and significant system functions are emphasized using bold font for clarity.
- References to external documents or standards are denoted in italics.
- Hyperlinks to external resources or related documents within the SRS are underlined for easy identification.

Numbering of Requirements:

- Each requirement is structured hierarchically, employing a numbering system (e.g., Major Requirement, Minor Requirement, Sub-Minor Requirement - formatted as 1.1, 1.1.1) for better organization and referencing.

These conventions are aimed at enhancing readability, organization, and the overall understanding of the SRS for stakeholders involved in the project.

1.3 Intended Audience and Reading Suggestions

Intended Audience for the document:

1. **Developers:** This document serves as a guide for developers, enabling them to comprehend the system's requirements thoroughly and implement the necessary functionalities accordingly.
2. **Project Managers:** Project managers will utilize this document to grasp the project's scope, estimate required resources, and oversee the project's progression.
3. **Testers:** Testers will rely on this document to design and execute test cases that validate whether the software aligns with the specified requirements.

4. **Documentation Writers:** Writers responsible for user manuals and other user-oriented documentation will refer to this document to accurately portray the system's features and functions.
5. **Users:** Users will have access to this document to gain insights into the system's capabilities, features, and intended use, fostering a better understanding of the software.

Suggestions for Reading the Document:

1. **Overview Sections:** Begin with the introductory sections that provide an overview of the software, including its purpose, goals, and intended audience. This gives a high-level understanding of the project.
2. **Functional Requirements:** Developers should delve into the functional requirements section to understand the specific functionalities and features the software must have.
3. **Non-Functional Requirements:** This section outlines performance, security, and other non-functional aspects critical for developers, testers, and project managers.
4. **Project Management and Resource Estimation:** Project managers can focus on sections related to project scope, resource estimation, and progress tracking to effectively manage the project.
5. **User-Facing Documentation:** Documentation writers and users should refer to sections outlining the software's intended purpose, features, and functionalities to create user-oriented materials and gain a deeper understanding of the system's capabilities.

1.4 References

- *IEEE Standards Association. IEEE Recommended Practice for Software Requirements Specifications. (IEEE Std 830-1998). Available at: [830-1998 - IEEE Recommended Practice for Software Requirements Specifications / IEEE Standard | IEEE Xplore](#)*
- *Software Requirement Specification (SRS) Format. Available at: <https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>*
- *How to Write a Software Requirements Specification (SRS) Document. Available at: [SEI Digital Library \(cmu.edu\)](#)*

2. Overall Description

2.1 Product Perspective

The software specified in this SRS, "Community First: Efficient Utilities & Primary Health Suite," is a new, self-contained product designed to address critical urban and municipal challenges. It stands as an independent software solution, rather than being part of an existing product family or a replacement for any current systems. This software aims to amalgamate

utility outage management and primary health-related services within a single platform, enhancing community living standards through technological innovation.

Context and Origin:

The software, as envisioned, is not a component of a larger system but serves as an all-inclusive tool catering to two vital aspects: utilities outages management (electricity, gas, water) and provision of primary health-related information and services. It is a standalone system providing users with real-time utility outage information and access to health-related news, consultations, and appointment scheduling.

Fig: Overall System

2.2 Product Functions

1. Registration:

- Firstly, enable people to sign up using their email account verification code and personal details.
- Use a random code supplied to the user's email to verify.
- Create a strong password and a distinctive username.
- After a successful registration, the login page will appear.

2. Subscription:

- Offer different subscription tiers (e.g., monthly, half-yearly, yearly) with varying benefits.
- Each plan can provide access to specific features such as uninterrupted outage information, priority consultations, or discounts.
- Allow users to choose, upgrade, downgrade, or cancel subscriptions based on their preferences.
- Automated billing and renewal processes for seamless subscription management.

3. Utilities Outage Management:

- Provide real-time updates on electricity, gas, and water outages.
- Distribute alerts and notifications regarding scheduled maintenance and outage duration.
- Analyze historical data to assist in resource planning and usage optimization.

4. Primary Health Services:

- Deliver health-related news, articles, and information to users.
- Enable affordable consultation services with medical professionals.
- Facilitate appointment scheduling with healthcare providers.

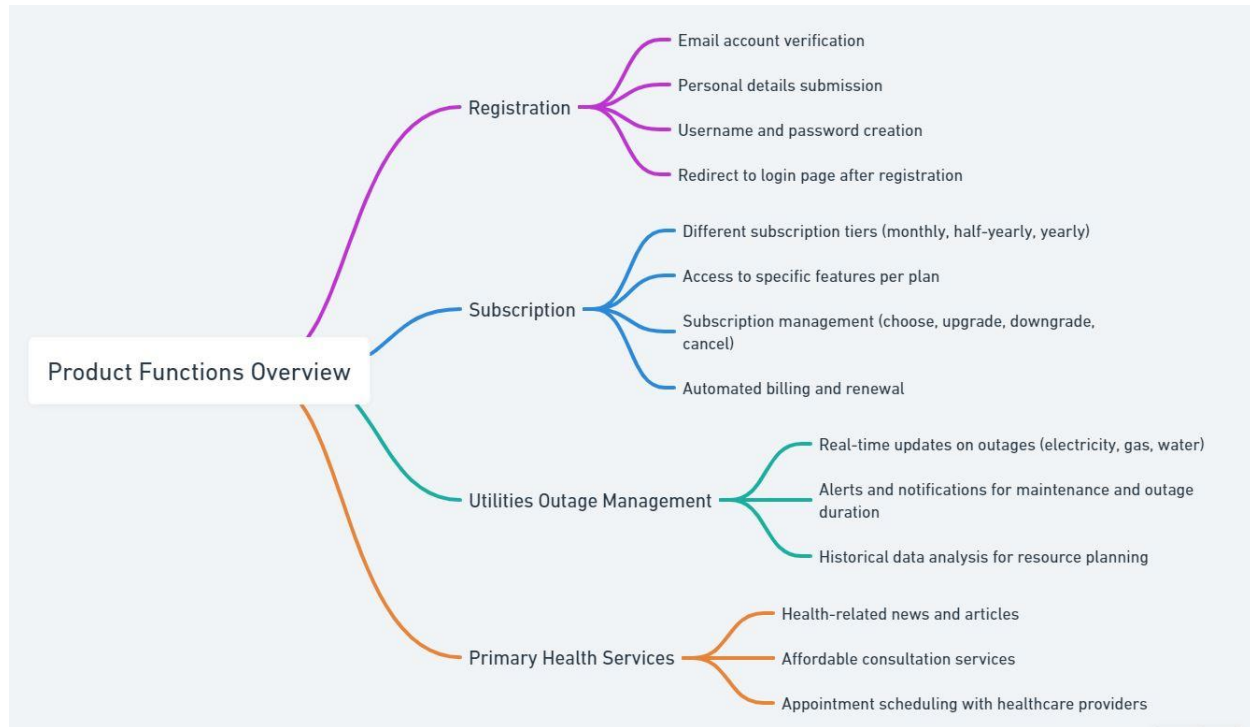


Fig: Major groups of related requirements

2.3 User Classes and Characteristics

1. General Users:

Frequency of Use: Regular access to outage information and health-related services.

Characteristics: Varied technical expertise, diverse educational backgrounds, and varying levels of experience in using similar applications.

Privilege Levels: Standard access to basic functions like viewing outage updates and accessing general health information.

2. Healthcare Professionals:

Frequency of Use: Occasional use for providing consultations or accessing specific health-related data.

Characteristics: High expertise in healthcare, potentially less technical expertise in software systems.

Privilege Levels: Access to features for consultations, patient management, and specialized health resources.

3. Support & Developer Team:

Frequency of Use: Regular interaction for system development, maintenance, and technical support.

Characteristics: High technical expertise, proficiency in software development, and system implementation.

Privilege Levels: Access to system development tools, backend functionalities, and debugging resources.

4. Administrators:

Frequency of Use: Intermittent use for overseeing system functionalities and user management.

Characteristics: Technical proficiency, managerial skills, and experience in overseeing software systems.

Privilege Levels: Admin access for managing users, overseeing subscriptions, and system settings.

Since general users are the main target audience and their satisfaction with the system is crucial to the product's success, they represent the most significant user class for this particular product. Even though they are significant, the other user classes can be added to in a later version of the system.

2.4 Operating Environment

The 'Community First: Efficient Utilities & Primary Health Suite' software system will primarily operate on mobile devices, compatible with both iOS and Android operating systems. The mobile devices used must have internet connectivity and meet the minimum operating system requirements of iOS 10 and Android 08 or above. The software doesn't necessitate any specific hardware components beyond those typically included in standard mobile devices.

The software should peacefully coexist with standard mobile device applications, ensuring no conflicts with common user applications like browsers, messaging apps, or other utility software.

The backend infrastructure includes:

- A scalable database system, preferably leveraging cloud-based solutions like AWS RDS or similar, ensuring robust data management and scalability.
- Web server compatibility with Apache or Nginx on a Linux OS, equipped with adequate resources for handling concurrent user interactions and data processing.

The system interfaces with external mapping and geolocation services via the internet, necessitating compatibility with the system's API for accurate location-based information.

2.5 Design and Implementation Constraints

1. Platform constraints: The software should be designed and implemented in such a way that it can run smoothly on both Android and iOS platforms.

- 2. Language Requirements:** The software should be developed using languages compatible with mobile app development for iOS (Swift or Objective-C) and Android (Java or Kotlin).
- 3. Security constraints:** Implementation should align with industry-standard security protocols to ensure data encryption, secure user authentication, and protection against potential cybersecurity threats.
- 4. Compliance/Permission constraints:** The software must comply with all relevant laws and regulations, such as data protection and privacy laws.
- 5. User interface constraints:** The software must have a simple and intuitive user interface that is easy to use for all classes of users.
- 6. Compatibility and Integration constraints:** The software needs to integrate seamlessly with external services, necessitating adherence to specific APIs and protocols for mapping, geolocation, and other essential functionalities.
- 7. Parallel Operations:** Ensuring the system can perform parallel operations efficiently, especially when handling outage updates and health-related information simultaneously.

2.6 User Documentation

The system can be used with the help of this user manual, which also includes the following step-by-step instructions:








User Manual

Community First: Efficient Utilities & Primary Health Suite

Community First is an innovative application designed to enhance your urban living experience by providing real-time utility outage information and primary health services at your fingertips.

!! System Requirements: To use the Community First app, you need a mobile device such as a smartphone or tablet with GPS capabilities and internet connectivity. The application is compatible with iOS 10+ and Android 08+ operating systems.

Using the system:

1	Download and install the Community First application on your mobile device from your respective app store.	
2	Open the application and create your user profile to start using the utility and health suites.	
3	Customize your subscription plan and manage your account settings to suit your preferences.	
4	Set up alerts for utility outages in your area by enabling location services and selecting your utility providers.	
5	Access health-related news and information tailored to your interests and location.	
6	Schedule appointments with medical professionals and manage your health service bookings.	
7	For real-time updates and notifications, ensure your device allows push notifications from the Community First app.	

Overall:

Community First is your go-to solution for managing utilities and health services efficiently. With a user-friendly interface and a variety of subscription options. Download now and join us in our mission to improve community living standards.

Contact Support:

For any assistance or feedback, please reach out to us at

support@communityfirstapp.com



Figure: User Manual

3. System Requirements

3.1 System Features

1. Registration

Functional Requirements (FRs)

- 1.1 Users can start the registration process by entering their verified email account and personal details, including their name, date of birth, and gender.
- 1.2 To confirm their identity, the system will generate a random verification code and send it to the user's email address.
- 1.3 After verification, users will be prompted to set up their username and password.
- 1.4 Users must create a strong password and select a unique username.
- 1.5 If the chosen username is already in use, the system will display an alert message, and the user will need to choose a different one.
- 1.6 Upon successful registration, users will be redirected to the login page for their account.
- 1.7 To register as a Healthcare professional user, need to additionally provide his/her bank details and professional certificate number.

Priority Level: High

Precondition: The user has a valid email account.

Cross-references: N/A

2. Subscription:

Functional Requirements (FRs)

- 2.1 Offers flexible subscription options (monthly, half-yearly, yearly).
- 2.2 Allows users to modify, upgrade, downgrade, or cancel subscriptions.
- 2.3 Automates billing processes for subscription payments.
- 2.4 Integrates secure payment gateways for transaction processing.

Priority Level: High

Precondition: The user has registered in the system.

Cross-references: N/A

3. Utilities Outage Management:

Functional Requirements (FRs)

- 3.1 Provides real-time outage details for electricity, gas, and water services.
- 3.2 Enables users to report localized outages.
- 3.3 Sends notifications regarding scheduled maintenance or resolved outages.

3.4 Displays estimated restoration times for outages.

3.5 Allows users to report emergency outages.

Priority Level: Medium

Precondition: The user has already subscribed a plan.

Cross-references: 2

4. Health Information Updates:

Functional Requirements (FRs)

4.1 Displays health-related news and campaigns.

4.2 Notifies users about vaccination or blood donation campaigns.

4.3 Provides a database of general health articles and resources.

4.4 Allows users to search for health-related information.

Priority Level: High

Precondition: The user has already subscribed a plan.

Cross-references: 2

5. Consultations

Functional Requirements (FRs)

5.1 Enables users to schedule appointments with medical experts.

5.2 Provides video or chat-based consultations.

5.3 Allows users to review consultation history and recommendations.

Priority Level: High

Precondition: The user has already subscribed a plan.

Cross-references: 2

6. Patient Management (For Healthcare Consultant)

Functional Requirements (FRs)

6.1 Records and organizes patient health information securely.

6.2 Supports diagnosis, treatment planning, and prescription management.

6.3 Enables consultants to communicate securely with patients.

Priority Level: High

Precondition: The user is a Healthcare Professional.

Cross-references: 1.7

7. Emergency Services

Functional Requirements (FRs)

7.1 Offers emergency contact information and guidelines.

7.2 Enables users to request emergency medical services.

Priority Level: High

Precondition: The user has registered in the system.

Cross-references: N/A

8. Community Engagement

Functional Requirements (FRs)

8.1 Provides a platform for user feedback and suggestions.

8.2 Facilitates community discussions or forums.

Priority Level: Low

Precondition: The user has registered in the system.

Cross-references: N/A

9. User Profile Management

Functional Requirements (FRs)

9.1 Enables users to update personal information.

9.2 Customization of notification settings.

9.3 Preferences for language or display settings.

Priority Level: Low

Precondition: The user has registered in the system.

Cross-references: N/A

3.2 Non-Functional/Quality Requirements

QA1: Usability: A registered user shall navigate through critical functionalities, such as outage information access and health consultations, with minimal interaction steps, achieving task completion within an average of three to five clicks.

Priority Level: Medium

Precondition: Users possess a valid user ID and password.

Cross-references: QA9.

QA2: Performance: The system should respond to user interactions (e.g., outage reporting, health consultation requests) within an average of two seconds under regular operational load.

Priority Level: High

Precondition: Users have completed the login process.

Cross-references: QA3, QA4, QA5, QA6.

QA3: Security: All user data, including personal information and health-related details, should be encrypted and stored securely in compliance with industry-standard encryption protocols (e.g., AES 256-bit encryption).

Priority Level: High

Precondition: User data transmission or storage.

Cross-references: QA2.

QA4: Reliability: The system should maintain an uptime of at least 99.9%, ensuring minimal service disruptions or outages, barring scheduled maintenance or unforeseen critical incidents.

Priority Level: Medium

Precondition: System operational hours.

Cross-references: QA2.

QA5: Scalability: The system should be scalable to accommodate an increasing user base without compromising performance, maintaining responsiveness even during peak loads without exceeding a specified server load threshold.

Priority Level: High

Precondition: High user influx during peak hours.

Cross-references: QA2, QA4.

QA6: Interoperability: The system should integrate seamlessly with external systems (e.g., mapping services, healthcare databases) via standardized APIs (Application Programming Interfaces) to ensure smooth data exchange and functionality.

Priority Level: High

Precondition: Integration with external systems.

Cross-references: QA2, QA7.

QA7: Integrity: All data transactions and storage operations within the system should maintain integrity, ensuring accurate and consistent data entry, processing, and retrieval across all modules and databases.

Priority Level: High

Precondition: Data manipulation and storage processes.

Cross-references: QA1, QA6, QA8.

QA8: Reusability: The system architecture and design should follow modular structures to promote code reuse, enabling easy extraction and integration of components for future enhancements or new functionalities.

Priority Level: Medium

Precondition: System design and development phases.

Cross-references: QA4, QA7.

QA9: Testability: The system should have comprehensive test coverage to validate functionality, performance, and security aspects.

Priority Level: Medium

Precondition: N/A

Cross-references: QA2, QA3.

3.3 Project Requirements

Software Process Model – Scrum

Why Scrum?

The adoption of the Scrum software development process model for the "Community First: Efficient Utilities & Primary Health Suite" project stems from its agile and adaptable nature. Scrum's iterative approach allows for incremental development, fostering frequent feedback loops and accommodating evolving requirements. Its flexibility in embracing changes ensures a more responsive development cycle, crucial for a project dealing with utilities and health services, subject to continuous updates and user feedback. Scrum's collaborative team dynamics and focus on risk management align well with the project's need for constant adaptation and quick responses to stakeholder needs.

Constructive cost Model (*COCOMO*):

Let's assume Source Line of Code is 4000.

So, effort need to be, $PM = 2.4 (4000/1000)^{1.05} = 10.289$

Development time, $DM = 2.5 * (PM)^{0.38} = 6.0623 = 6$

Required number of people, $ST = PM/DM = 1.697 = 2$

That means we need to work for $(4*6) = 24$ weeks.

Resources Needed for the Project:

Human Resources:

Product owner: 1 person

Scrum Master: 1 person

Developers (Frontend, Backend): 2 persons (As per COCOMO)

UI/UX Designers: 1 person

Quality Assurance/Testers: 1 person

Hardware and Networking Devices: High-performance servers, Storage Systems, Cloud Services, Mobile Devices (iOS and Android), Computers/Workstations for Development and Testing.

Reusable Components: Frameworks, Libraries, and Modules for User Authentication, Payment Gateways, Data Encryption, API Integrations (Mapping, Healthcare Services), and UI Components to expedite development.

Timeline Chart:

Weeks Person	Pregame Phase						Development												Postgame Phase					
	Planning			Architecture			Sprint1				Sprint2				Sprint3									
	We ek1	We ek2	We ek3	We ek4	We ek5	We ek6	We ek7	We ek8	We ek9	We ek10	We ek11	We ek12	We ek13	We ek14	We ek15	We ek16	We ek17	We ek18	We ek19	We ek20	We ek21	We ek22	We ek23	We ek24
A																								
B																								
C																								
D																								
E																								
F																								
G																								
H																								
I																								

A: Project initiation, Scoping, requirements gathering and planning

B: Design, product backlog creation and sprint planning

C: Requirements for each sprint

D: Analysis for each sprint

E: Development for each sprint

F: Testing for each sprint

G: Integration testing

H: System testing

I: Release preparation and launch

4. Design and Interface Requirements

4.1 UML Diagrams

Use case diagram: Use case diagram for the system "Community First: Efficient Utilities & Primary Health Suite" includes several actors: General User, an Admin who "Manager All user", Health Consultant, the system itself as "Community First" and Development & Support Team. The use cases depicted are User Profile Management, Do Registration, Sign in, Sign out, Utilities Outage Management, Health Information Update, Consultations, Emergency Service, Patient

MMH

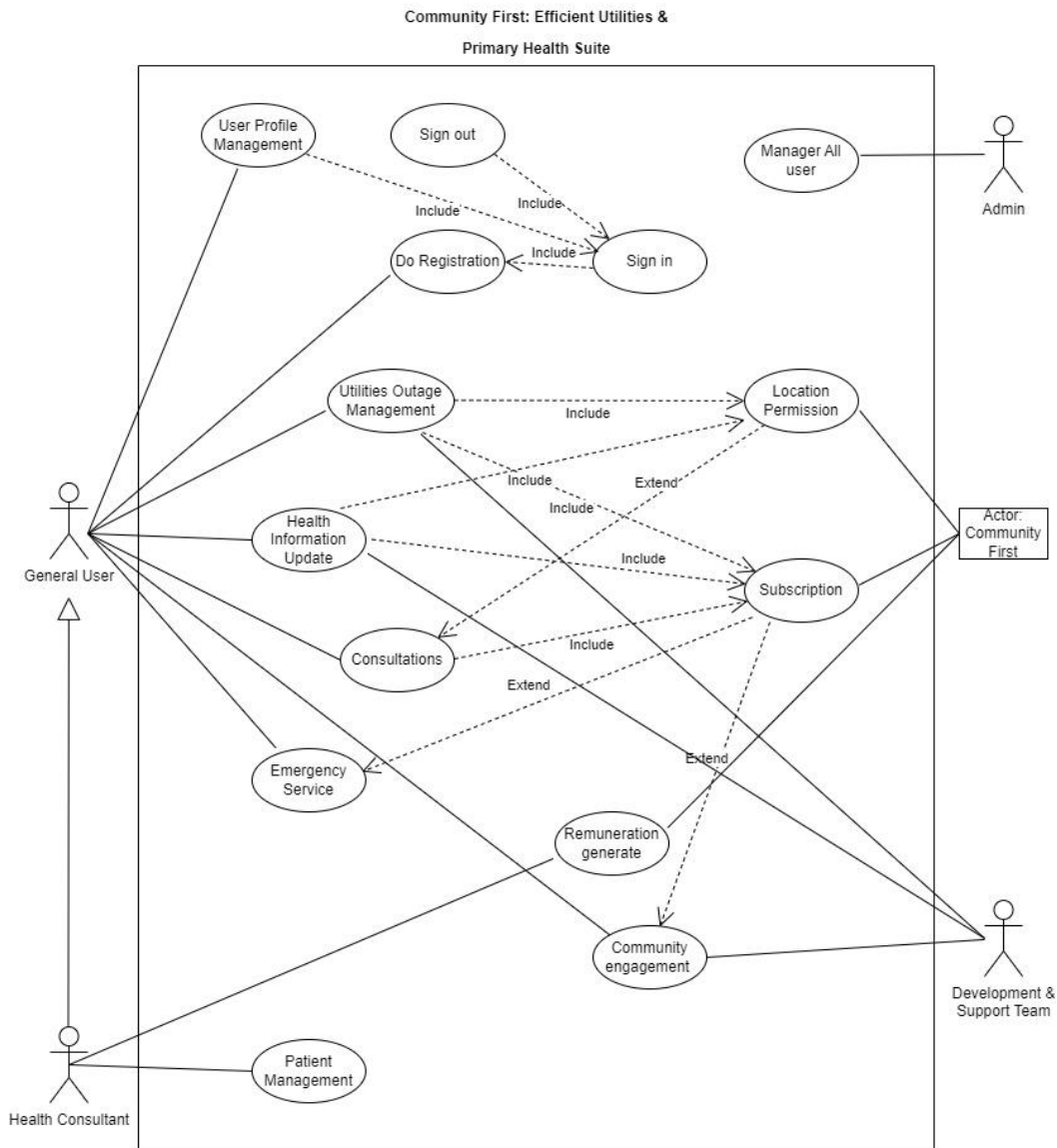


Fig: Use case Diagram

Class diagram: Below image is a class diagram for "Community First, Efficient Utilities & Primary Health Suite" system. It outlines various classes like DevSupportTeam, Admin, User, HealthcareProfessional, Patient, Outage, HealthInfo, Consultation, EmergencyService, Subscription, and CommunityEngagement, each with specific attributes and operations that define their functionality. The relationships among these classes are indicated by lines, suggesting collaboration or interaction, such as DevSupportTeam updating Outage and HealthInfo, Admin managing User, and HealthcareProfessional managing Patient. Subscription

acts as a central node with multiple connections to other services like Outage, Consultation, and EmergencyService, indicating a system where user subscriptions may influence or access various services.

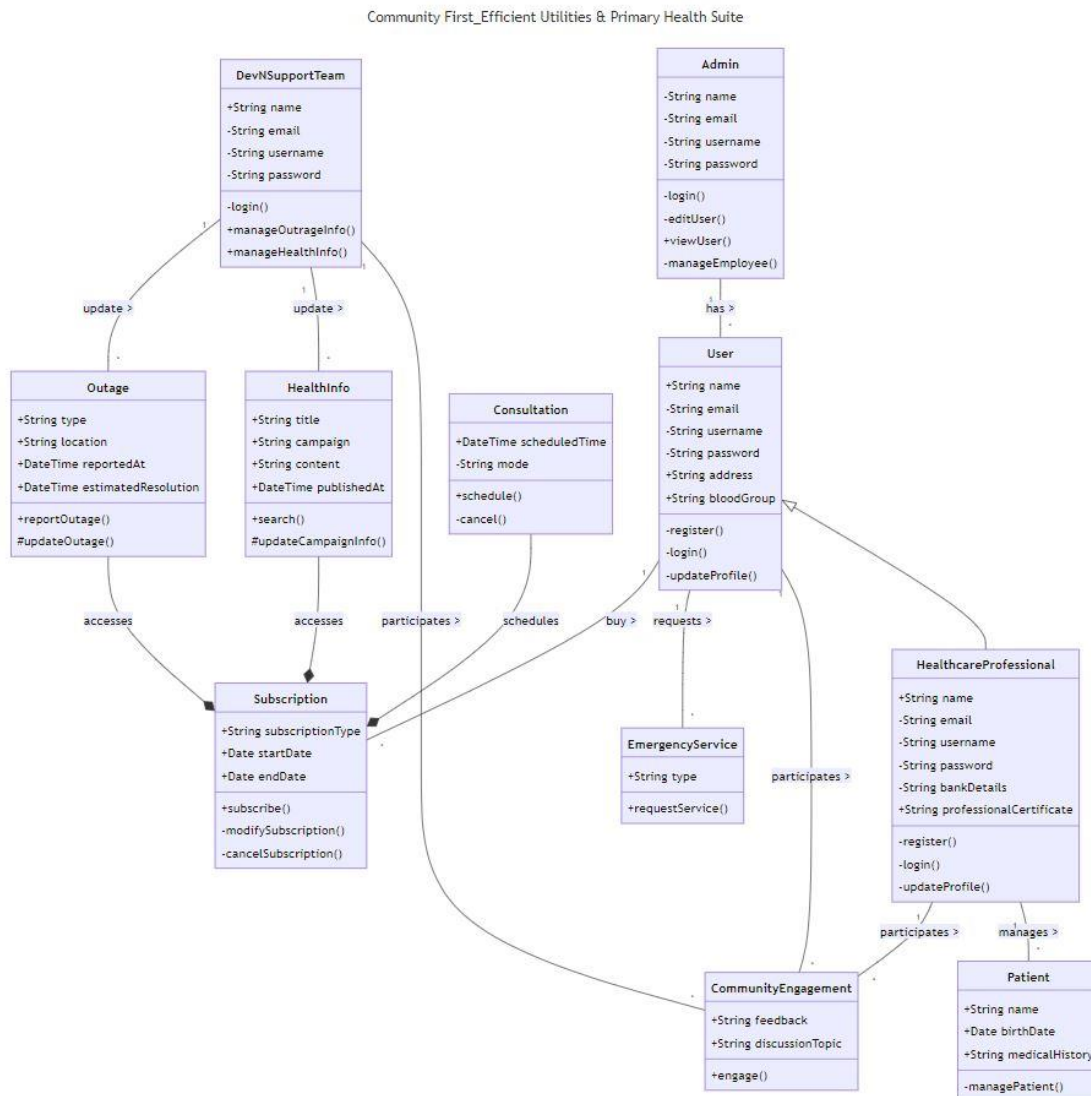


Fig: Class Diagram

Activity diagram: The below image presents an activity diagram for the "Community First: Efficient Utilities & Primary Health Suite" system. It outlines the workflows for four roles: General User, Health Consultant, Development & Support Team, and Admin. Each role starts with authentication processes like Registration and Sign in. Post-authentication, General Users and Health Consultants engage in various services and profile management tasks, while the Development & Support Team focuses on updating information. The admin primarily manages user accounts. All roles eventually reach a decision point that leads to the Sign out process, ending their session. The activities and decision flows are represented in a structured manner, highlighting the sequence of operations for each role within the system.

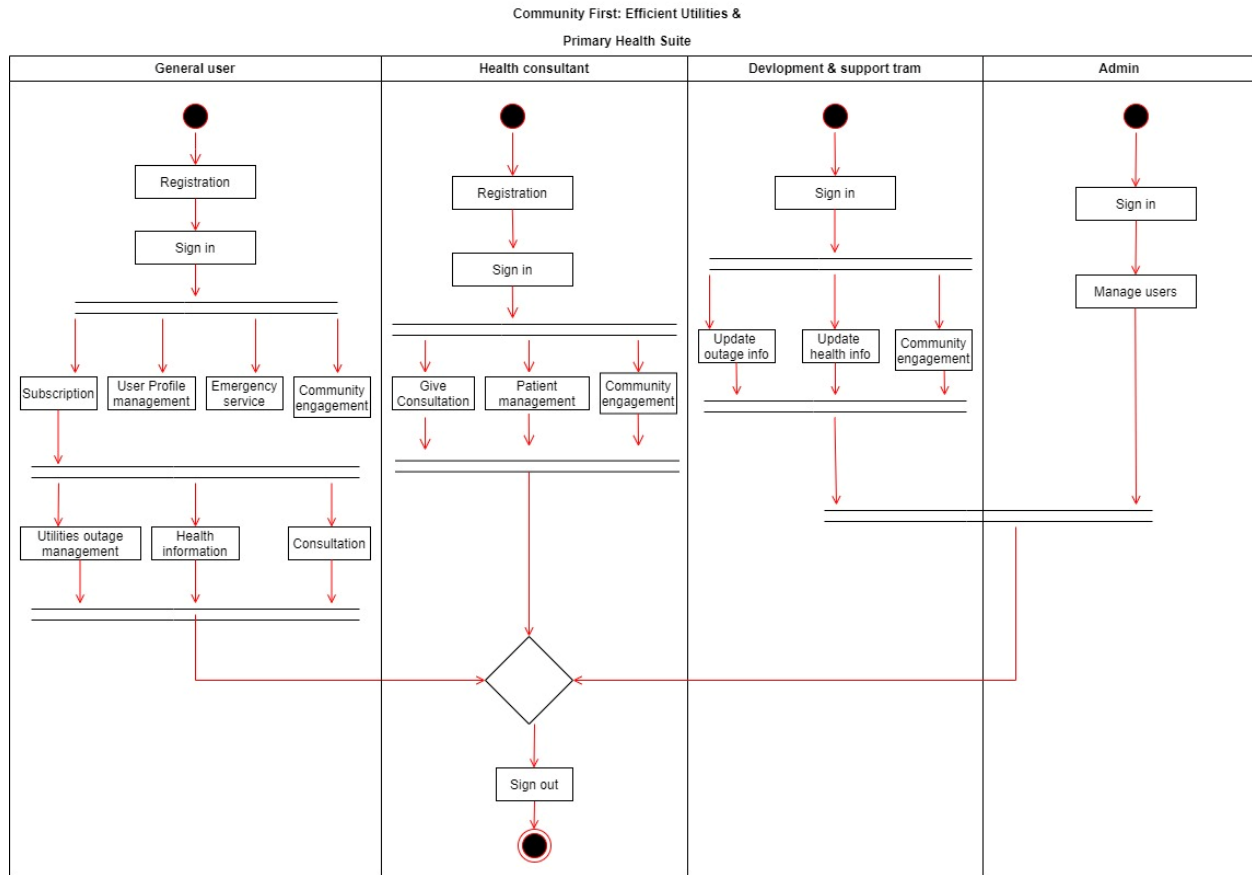


Fig: Activity Diagram

E-R diagram: This ER diagram is a high-level representation of the system's data model, showing how different entities are interconnected and what kind of information might be stored or related within the application.

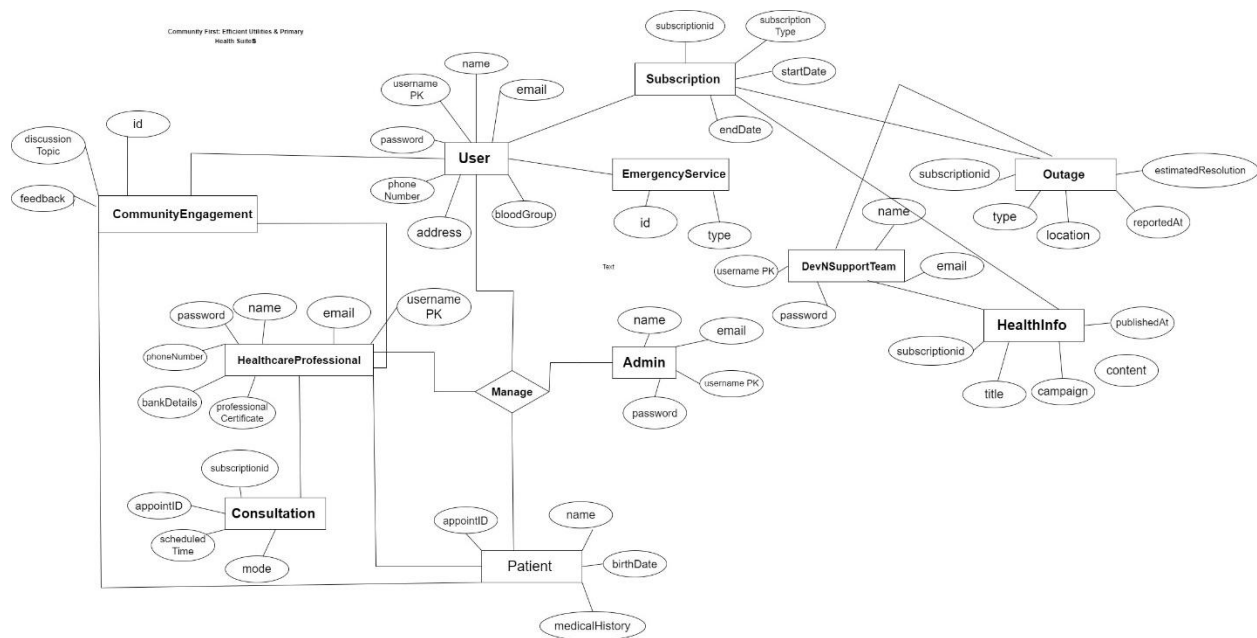


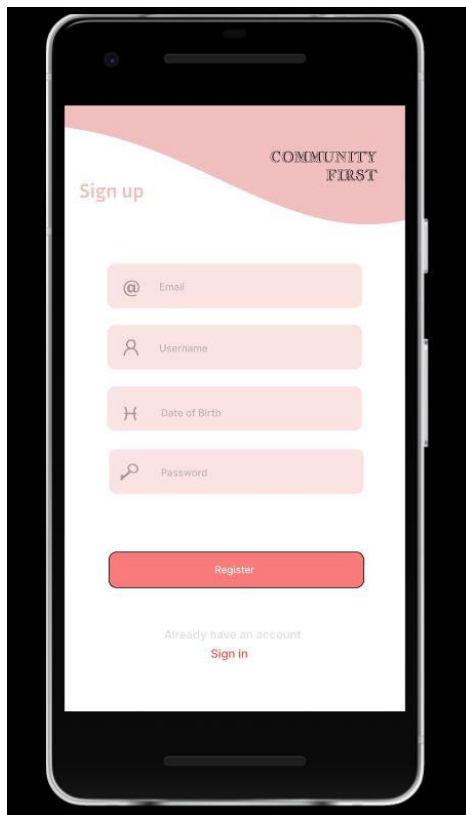
Fig: ER Diagram

4.2 Data Dictionary

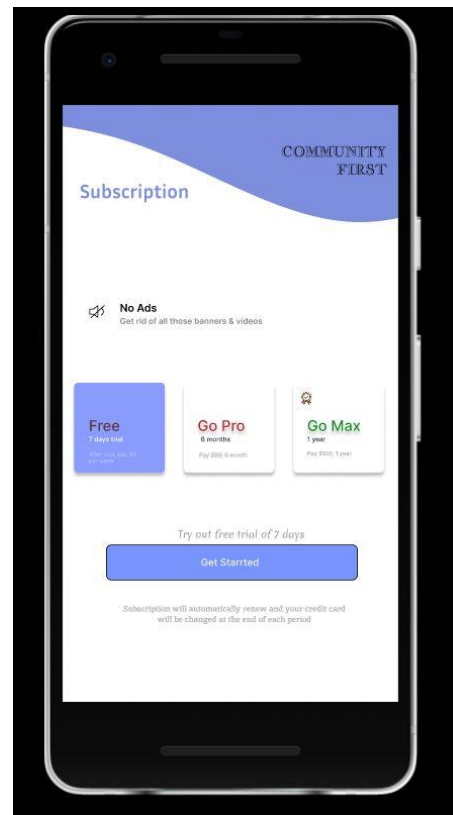
Entity	Attribute	Type/Size	Validation	Key
user	name	string (15)	Required	
user	email	string (20)	Required	
user	dateofBirth	Date (8)	Required	
user	username	string (8)	Required	Primary
user	password	string (8)	Required	
user	phoneNumber	int (11)	Required	
user	address	string (25)	Required	
user	bloodGroup	string (3)	Required	
admin	name	string (15)	Required	
admin	email	string (20)	Required	
admin	username	string (8)	Required	Primary
admin	password	string (8)	Required	
devNSupportTeam	name	string (15)	Required	
devNSupportTeam	email	string (20)	Required	
devNSupportTeam	username	string (8)	Required	Primary
devNSupportTeam	password	string (8)	Required	
healthcareProfessional	name	string (15)	Required	
healthcareProfessional	email	string (20)	Required	
healthcareProfessional	username	string (8)	Required	Primary
healthcareProfessional	password	string (8)	Required	
healthcareProfessional	phoneNumber	int (11)	Required	
healthcareProfessional	bankDetails	string (30)	Required	
healthcareProfessional	profCertificateNo	string (12)	Required	

subscription	subscriptionid	int(3)	001-999	Primary
subscription	subscriptionType	string (6)	Required	
subscription	startDate	Date (8)	Valid Date	
subscription	endDate	Date (8)	Valid Date	
outage	subscriptionid	int (3)	001-999	Foreign
outage	type	string (12)	Required	
outage	location	string (25)	Required	
outage	reportedAt	DateTime	Required	
outage	estimatedResolution	DateTime	Required	
healthInfo	subscriptionid	int (3)	001-999	Foreign
healthInfo	title	string (20)	Required	
healthInfo	campaign	string (20)	Required	
healthInfo	content	string (200)		
healthInfo	publishedAt	DateTime	Required	
consultation	subscriptionid	int (3)	Required	Foreign
consultation	appointID	int (10)	Required	Primary
consultation	scheduledTime	DateTime	Required	
consultation	mode	string (3)	Required	
patient	appointID	int (10)	Required	Foreign
patient	name	string (15)	Required	
patient	birthDate	date (8)	Valid Date	
patient	medicalHistory	string (140)	Required	
emergencyService	Id	int (8)	Required	Primary
emergencyService	type	string (20)	Required	
communityEngagement	id	int (10)	Required	Primary
communityEngagement	feedback	string (120)		
communityEngagement	discussionTopic	string (20)	Required	

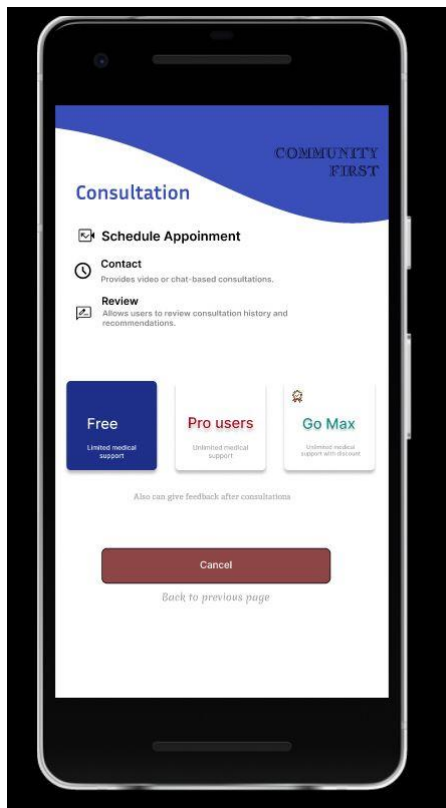
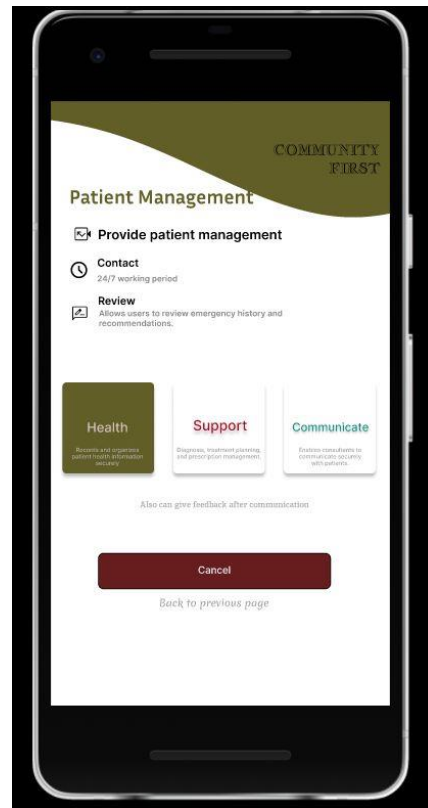
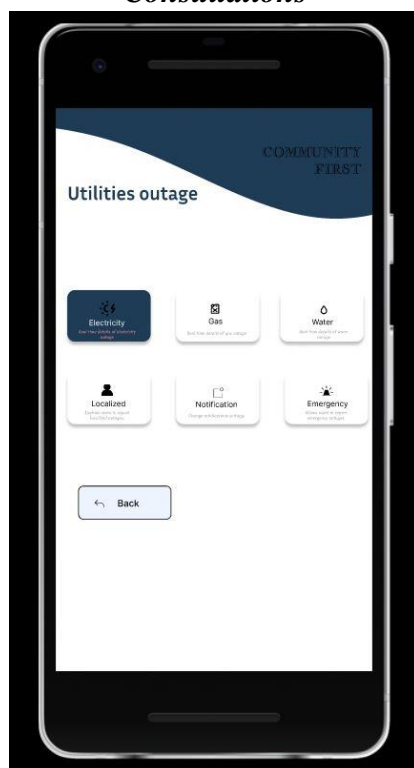
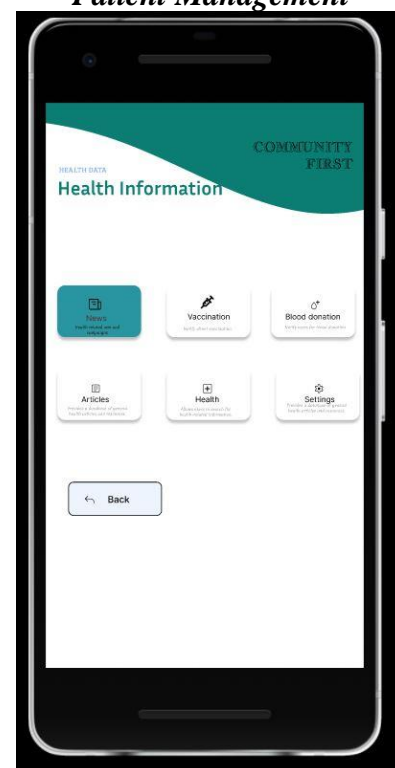
4.3 UI/UX Design Specification

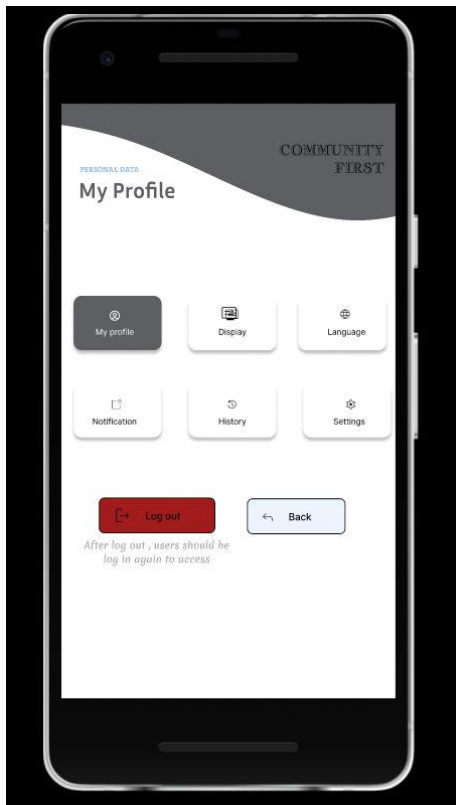


Registration

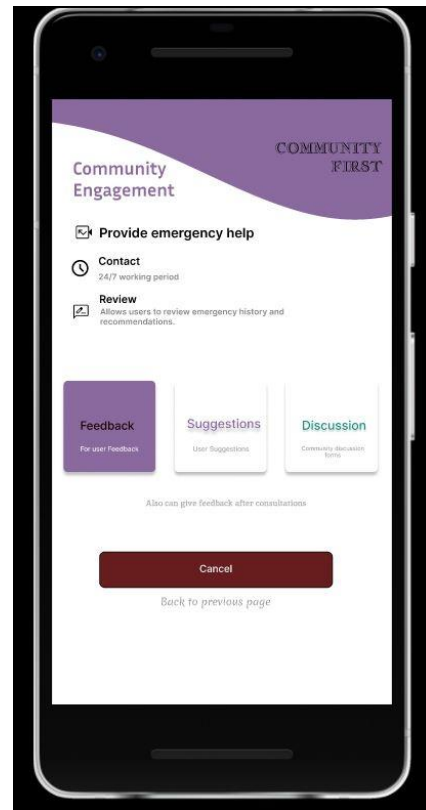


Subscription

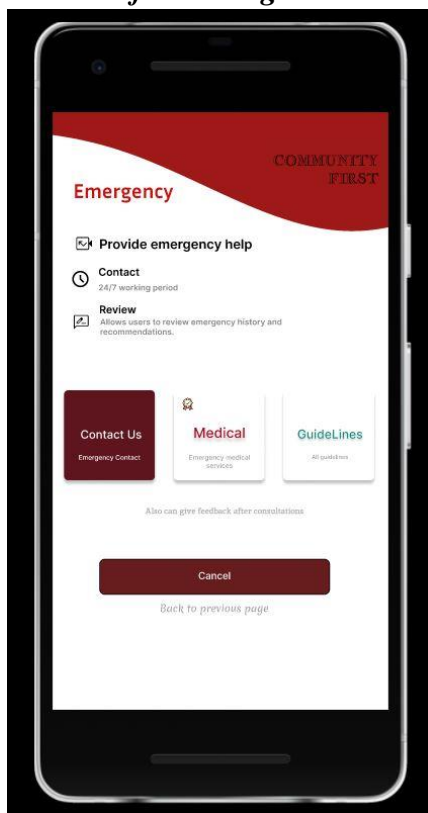
*Consultations**Patient Management**Utilities Outage**Health Information*



Profile Management



Community Engagement



Emergency

Figure: UI/UX Design by Figma tool