Cure-Well Project

Assignment 3

Names: Layan RaydaaGhaya
Group: 3I

Dept. of Computer Science
Faculty of Computer and Information Systems
Umm Al-Qura University KSA

This project report is submitted to the Department of Computer Science at Umm Al-Qura University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Engineering/Computer Science/Information Systems.

Author(s):
Layan Alharbi
Raydaa Almaabadi
Ghaya Alhazmi

Supervisor(s):
Hanadi Merdah
Computer Science Department

October 2024

Intellectual Property Right Declaration

This is to declare that the work under the supervision of Dr. Hanadi Merdah, titled "CURE WELL," carried out in partial fulfillment of the requirements of Bachelor of Science in Computer Science, is the sole property of Umm Al-Qura University and the respective supervisor and is protected under intellectual property right laws and conventions.

Date: 10-18-2024

Author(s):
Name: Layan Alharbi Signature: _____
Name: Ghaya Alhazmi Signature: _____
Name: Raydaa Almaabadi Signature: _____

Supervisor(s):
Name: Hanadi Merdah Signature: _____

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the supervision of Dr. Hanadi Merdah, having the title "CURE WELL," is the sole contribution of the author(s) and no part hereof has been reproduced illegally (cut and paste) which can be considered as plagiarism. All referenced parts have been used to argue the idea and have been cited properly.

Date: 10-18-2024

Author(s):

Name: Layan Alharbi Signature: _____

Name: Ghaya Alhazmi Signature: _____

Name: Raydaa Almaabadi Signature: _____

Acknowledgments

This work is dedicated to Umm Al-Qura University, an institution that has nurtured my academic and personal growth through its commitment to excellence. I am deeply thankful for the resources, support, and knowledge provided throughout my journey, especially by the Department of Computer Science

I would like to express my deepest gratitude to Dr. Hanadi Merdah, whose guidance, mentorship, and encouragement have been invaluable throughout the course of this project. Her insightful advice and constant support have been a source of inspiration, and her dedication to fostering academic excellence has greatly contributed to the success of this work.

This project reflects the knowledge and experience I have gained under the tutelage of Dr. Merdah and the esteemed faculty at Umm Al-Qura University. I am honored to have had the opportunity to learn from such remarkable educators

Abstract

The Cure Well Pharmacy App is designed to streamline access to pharmaceutical services for users by integrating multiple pharmacies into a single platform. This application enhances the customer experience by offering features such as product price comparison, real-time pharmacist consultations, and secure online purchasing. The primary goal is to facilitate easy browsing of medications, swift ordering, and delivery from local pharmacies while ensuring a reliable and efficient service.

Keywords: Cure Well, Pharmacy App, Pharmaceutical Services, User Experience

Contents

0.1 Software Overview and Purpose

The Cure Well Pharmacy App is designed to streamline access to pharmaceutical services for users by integrating multiple pharmacies into a single platform. It aims to enhance the customer experience by offering features such as product price comparison, real-time pharmacist consultations, secure online purchases, and delivery tracking. The app facilitates easy browsing of medication, consultation with professionals, and swift ordering and delivery from local pharmacies. Additionally, pharmacy administrators and pharmacists can manage inventory, process orders, and provide professional consultations directly through the app.

The app's core objectives are:

- **Accessibility:** Offering 24/7 access to a wide range of pharmaceutical services.
- **Efficiency:** Reducing the time and effort required to search for and order medicines.
- **Reliability:** Ensuring timely delivery, secure payment processing, and trusted medical advice.
- **Customer Empowerment:** Allowing users to compare prices, receive professional consultations, and make informed decisions about their health and purchases.

0.2 First Draft Purpose

The purpose of the first draft is to provide a detailed overview of the app's development process, features, and technical specifications. Its primary purpose is to ensure that all stakeholders, including developers, project managers, and end-users, have access to a comprehensive resource for the system.

The document supports:

- **Knowledge Transfer:** Facilitating effective collaboration across development, management, and operational teams.
- **Future Development:** Laying a solid foundation for enhancements and new features in subsequent versions.
- **Project Continuity:** Ensuring the project aligns with its goals and can be maintained efficiently over time.

This documentation includes system requirements, architectural design, user personas, journey maps, and future improvements. By offering a detailed roadmap and technical guidance, this document promotes the long-term success of the Cure Well Pharmacy App.

0.3 Software Features

- **Price Comparison:** Users can browse and compare prices of medications from different pharmacies.
- **Pharmacist Consultations:** The app offers real-time chat functionality where users can consult with pharmacists.
- **Order and Delivery Management:** Users can place orders, choose delivery options (personal or to friends), and track their deliveries.
- **Secure Payment Processing:** The app integrates payment methods like Apple Pay and credit cards, with features like credit card scanning for easy payment.
- **Login and User Authentication:** Secure login for users using email and password, with enhanced security for administrators.
- **Product Browsing and Search:** Users can search for medications and sort results by various filters like price or pharmacy.
- **Push Notifications:** Users receive updates and notifications about their orders or consultations.
- **GPS and Location-based Services:** The app uses GPS to help users find nearby pharmacies and ensure accurate delivery addresses.
- **Cart and Checkout:** Users can review items in their shopping cart, adjust quantities, and proceed to checkout.
- **User Feedback:** Customers can rate the service they received from pharmacists.

0.4 Software Audience

1. **Customers:** General users who access the app to purchase medications, consult with pharmacists, and manage their orders.

2. **Pharmacy Owners and Administrators:** Users responsible for managing inventory, tracking orders, and overseeing pharmacy operations within the app.

3. **Pharmacists:** Healthcare professionals using the app to provide consultations and manage prescriptions.

4. **Development Team:** Developers, system architects, and quality assurance teams who are responsible for creating, testing, and maintaining the app.

5. **Project Managers and Stakeholders:** Individuals overseeing the development, ensuring the app meets its goals and requirements.

0.5 Documentation Assumptions

1. **User Technical Knowledge:** Users are assumed to have basic proficiency in using mobile apps, including navigation, form completion, and basic security practices (e.g., logging out from shared devices).

2. **System Access Requirements:** The app assumes users will have reliable internet access and GPS functionality enabled for location-based services, such as delivery and nearby pharmacy recommendations.

3. **Security & Privacy:** Users are expected to use strong passwords (minimum 8 characters for customers and 12 characters for admins) and may employ two-factor authentication (2FA) for secure access. Payment details are handled through secure methods (e.g., Apple Pay).

0.6 Validate Your Assumptions (Gap Analysis)

1. **Internet Connectivity:** The assumption about constant internet access may not be valid in regions with poor network infrastructure, impacting usability.

    - **Recommendation:** Consider offline modes or caching essential data to allow users access to limited features when connectivity is intermittent.

2. **Security & Privacy Gaps:** While strong password policies and encryption are mentioned, no specific details are provided about how payment data is protected.

    - **Recommendation:** Enhance security documentation by specifying encryption standards, data storage policies, and compliance with relevant security certifications.

3. **User Roles Flexibility:** The manual mentions customer, pharmacy owner, and administrator roles but does not clarify if role-based access control is implemented to prevent unauthorized access to sensitive data (e.g., pharmacy inventory).

    - **Recommendation:** Specify how different user roles are restricted in the system and whether audit trails are kept for administrative actions.

0.7 Definitions and Acronyms

WORD	MEANING
Persona:	Fictional characters representing user types, helping teams understand user behaviors and needs.
User Story	brief description of product features from the user's perspective.
User Story Map	A visual tool organizing user stories along the user's journey to help prioritize feature development.
GBS	system used to identify the nearest pharmacies.
iOS	Apple's mobile operating system.
API	Software intermediary enabling communication between apps and external services.
User Authentication	system ensuring only authorized users access the app.
Test Plan	Instructions for testing the application, tracking defects and their severity.
SCD	Documentation explaining the source code's structure and functionality.
QAD	Document outlining testing and quality management processes.

Figure 1: Terms and it denftitions

0.8 Software Requirements

0.8.1 Major System Requirements

- The app should be developed using **JAVA**.
- To use the application, you must have internet access as it uses GPS to locate the nearest pharmacies.
- The system must be available online 24/7.
- It must support electronic payments.

0.8.2 External Interface Requirements

Requirement Type	Description
User Interface Requirements	The app will feature an intuitive, easy-to-navigate interface with consistent design elements and a user-friendly layout.
Hardware Interface Requirements	The app relies on hardware interfaces such as a credit card camera scanner for capturing and processing credit card information during transactions. The specific communication methods between components are managed by the iOS operating system.
Software Interface Requirements	The app integrates with the Wallet application for Apple Pay transactions, communicates with the GPS for location services, interacts with the database for product and staff information, and interfaces with the iOS operating system.
Communication Interface Requirements	Communication between different parts of the system is crucial as they depend on each other. However, the method of achieving this communication is not critical to the system and is therefore handled by the underlying operating systems of the iOS mobile phone application.

0.8.3 Functional Requirements

1. **User Log-in**
    - **DESC:** Given that a user has registered, they should be able to log in to the mobile application. The log-in information will be stored on the phone, and in the future, the user should be logged in automatically.
    - **RAT:** In order for a user to register on the mobile application.
2. **View Orders**
    - **DESC:** To view the orders, users can access this service through the home page by selecting "view orders." They will then see details of their current and past orders, such as the delivery date or expected delivery date, number of products, order number, and order price.
    - **RAT:** To display the user orders list.
3. **Cart**
    - **DESC:** The cart allows the user to view the products that have been added. The system should provide options to manage the cart, such as adding, removing, or updating quantities of selected products.
    - **RAT:** In order for the user to complete the ordering process.
4. **Check Out**
    - **DESC:** The user is able to choose the payment methods either Apple Pay or by Credit Card, and then the system should be able to determine the location in two options: either the user's location or the user's friend's location after the user enters their friend's information.
    - **RAT:** In order for the user to complete the ordering process.
5. **Get Advice**
    - **DESC:** Upon logging into the application, the user will be presented with the home page containing two options: get advice and Products Page. If they choose to get advice (conversation with a pharmacist), they will be directed to the get advice page. This feature allows users to seek advice and information from pharmacists about their medications or health concerns.

- **RAT:** In order to provide medical consultation for the user.

6. **View Product**

   - **DESC:** Upon logging into the application, the user will be presented with the home page containing two options: Conversation with a pharmacist or Products Page. If they select the Products Page, they will be directed to a page where they can search for medications. Users can also browse through the products. They should have the option to search by entering text.
   - **RAT:** Users can browse through the products.

7. **Answer Customer Conversation**

   - **DESC:** The system should allow the pharmacist to answer customer questions by texting and provide them with advice.
   - **RAT:** In order to provide medical consultation to the user.

8. **Change Language**

   - **DESC:** A user should also be able to choose what language the mobile application should be set to. The different language choices are Arabic and English.
   - **RAT:** In order for a user to download the mobile application.

0.8.4 Non-Functional Requirements

Requirement Type	Description
Security	User passwords must be encrypted during transmission and storage. All credit card numbers transmitted between the app and the payment gateway must be encrypted. The payment methods should be safe and secure. Updates to the app should not affect its security.

Requirement Type	Description
Capacity	The app should be designed to handle a high number of concurrent users without a decrease in performance. For instance, it should be capable of processing multiple transactions, user searches, and consultations simultaneously.
Compatibility	The app should be compatible with the latest versions of iOS and Android operating systems.
Reliability	In the case of a system failure, the app should recover and resume normal operations within 15 minutes. The pharmacy app should be available for use 99.5% of the time during regular operating hours.
Scalability	The system should be scalable to accommodate future growth. As the user base expands, the app should easily scale up in terms of storage, processing power, and network bandwidth to maintain optimal performance.
Maintainability	App updates should not occur at peak hours when customers are using the application. Automated daily backups of the app's database should be stored securely for a minimum of 28 days.
Usability	The user interface should follow accessibility guidelines to support users with disabilities. The update process should be easy for users with disabilities and users who do not have technical experience.
Performance	The app should respond to user interactions (searching for products, processing orders, and payment) within 3 seconds.

0.8.5 Software Requirements List

ID	Software System	Category	Description	Requirement

ID	Software System	Category	Description	Requirement
1	App	Functional	User authentication system	The app must allow users to log in securely using a username and password, with password recovery.
2	App	Functional	User settings management	Users should be able to adjust app settings, such as notifications, privacy options, and account details.
3	App	Functional	Order history and status tracking	Users must be able to view past orders, track current order status, and receive updates.
4	App	Functional	Shopping cart functionality	Users can add products to a cart, view the cart, and modify quantities before checkout.
5	App	Functional	Order checkout process	The app must guide users through the checkout process, including payment and order confirmation.
6	App	Functional	Access to pharmacist consultation	Users should be able to consult with a pharmacist for advice on medications or product usage.
7	App	Functional	Product catalog and details	Users can browse and view detailed information on available products, including price comparisons.
8	App	Functional	Customer service response system	Pharmacists or support staff should be able to answer customer inquiries through the app.
9	App	Usability	Language selection and localization	Users should be able to change the app's language to their preference.

0.9 Persona

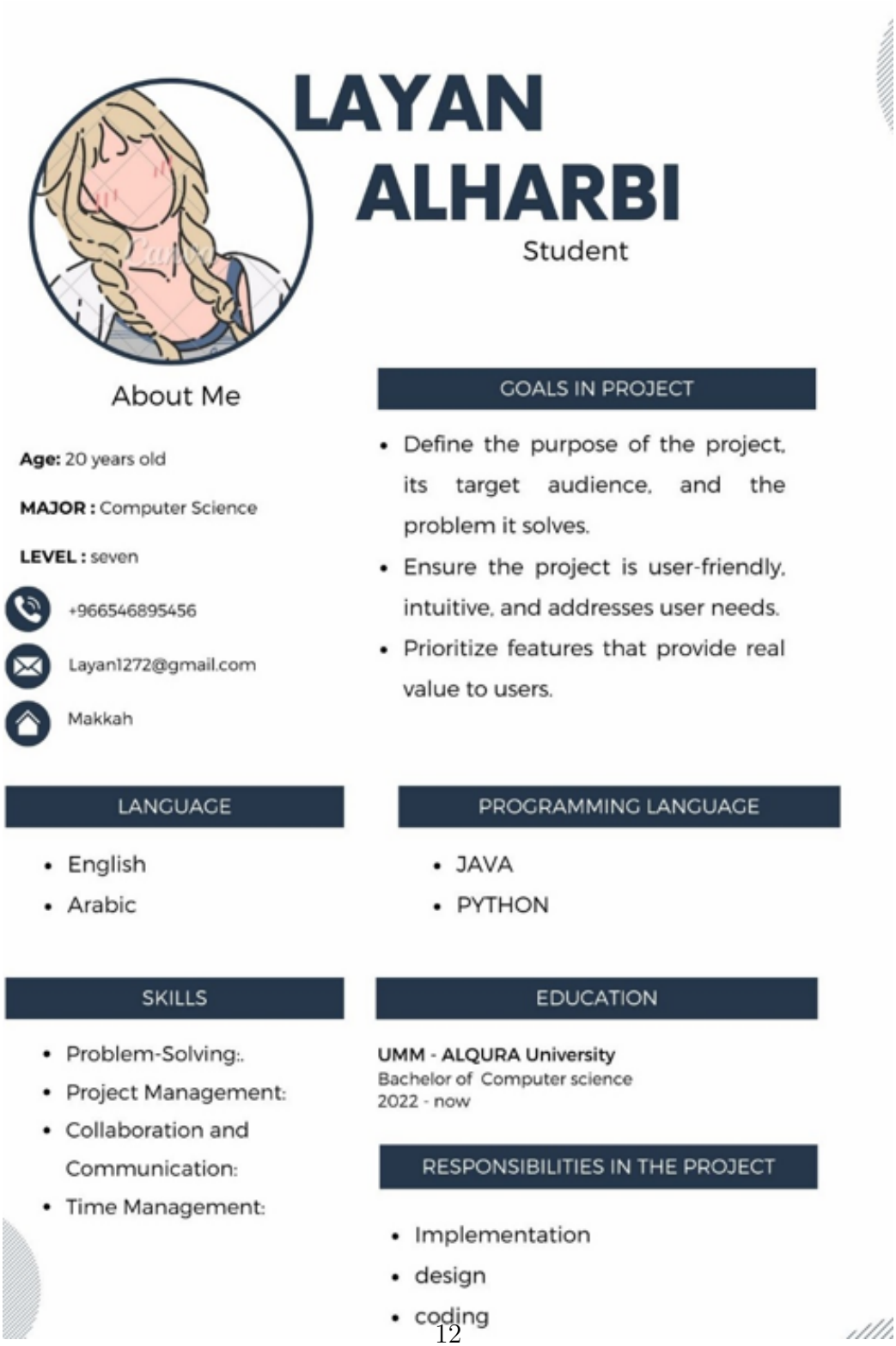


Figure 2: Layan's person

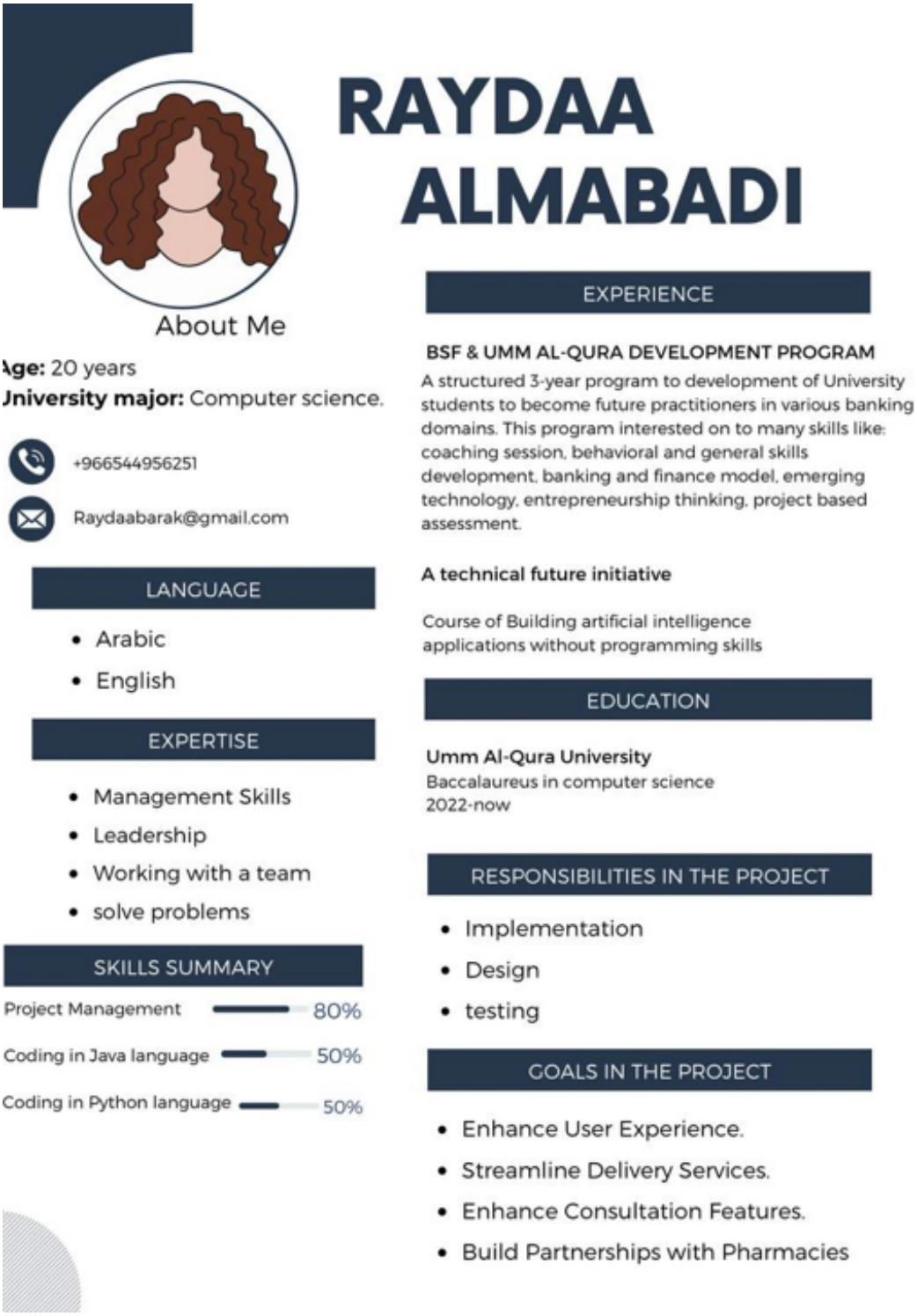


Figure 3: Raydaa's person

Figure 4: Ghaya's person

0.10 User Stories

- As a customer, I want to search for medicine across multiple pharmacies, so that I can find the best price and availability quickly.
- As a customer, I want to have my medicine delivered from the nearest available pharmacy, so that I can receive it as soon as possible.
- As a customer, I want to access the app at any time, so that I can order medicine even outside of regular pharmacy hours.
- As a customer, I want to receive reliable descriptions and information about the medicines, so that I can understand their benefits and risks before purchasing.
- As a customer, I want to easily consult with a pharmacist through the app, so that I can get professional advice about my medication.
- As an administrator, I want to list my pharmacy's products on the app, so that I can reach more customers and increase sales.
- As an administrator, I want to manage orders from the app efficiently, so that I can fulfill them promptly and improve customer satisfaction.
- As an administrator, I want to track sales and inventory through the app, so that I can keep my stock updated and manage my business effectively.
- As an administrator, I want to organize promotions, so that I can attract more customers.
- As a pharmacist, I want to provide advice to customers through the app, so that I can help them make informed decisions about their medication.
- As a pharmacist, I want to update medicine descriptions and information in the app, so that customers always have access to accurate and up-to-date details.
- As a pharmacist, I want to respond to customer inquiries quickly, so that I can assist them in a timely manner and build trust.

0.11 user story map

Figure 5: User story map

0.12 Software Design Documentation (SDD)

0.12.1 System Architecture

Data Platform:

- **Database:** A cloud-based database such as Firebase and MySQL will store customer data, product information, orders, and pharmacist consultations.
- **Storage:** Data will be stored in secure, encrypted storage with automated daily backups.

Hardware Requirements:

- **Specialized Hardware Requirements:**
  - **Mobile Device:** The app is designed to run on iOS devices only, so customers must have an iPhone or iPad.
  - **GPS Module:** GPS-enabled devices are required for accurate location-based services such as finding nearby pharmacies.
  - **Credit Card Camera Scanner:** The app will utilize the mobile device's camera to scan credit card information for payments.

Software Requirements:

- **Specialized Software Requirements:**
  - **iOS Operating System:** The app is built for iOS only and requires iOS 13 or later.
  - **Development Tools:** Xcode for iOS development, Swift as the primary programming language, and Java for backend APIs.
  - **Payment Gateway Integration:** The app integrates with Apple Pay and third-party payment gateways.
  - **Map and GPS Services:** The app uses Apple Maps for location tracking and pharmacy display, as well as GPS services for delivery coordination.
  - **Push Notifications:** Integrates with Apple Push Notification Service (APNs) to send notifications to users.

Programming Languages and Tools:

- **Programming Languages and Tools:**

- **Frontend:** Swift (iOS app development).
- **Backend:** Java for APIs.
- **Version Control:** GitHub for version control.
- **Database:** Firebase for user authentication and MySQL for storing customer, pharmacy, and product data.

Operating System and Network Requirements:

- **Network/Operating System:**
  - **iOS:** The app will run on iOS 13 or later.
  - **Network:** The app requires an Internet connection (Wi-Fi or cellular) for all functionalities, including product browsing, orders, and consultations.
  - **GPS Connectivity:** For location tracking to find nearby pharmacies.

0.12.2 Software Architecture Document (SAD)

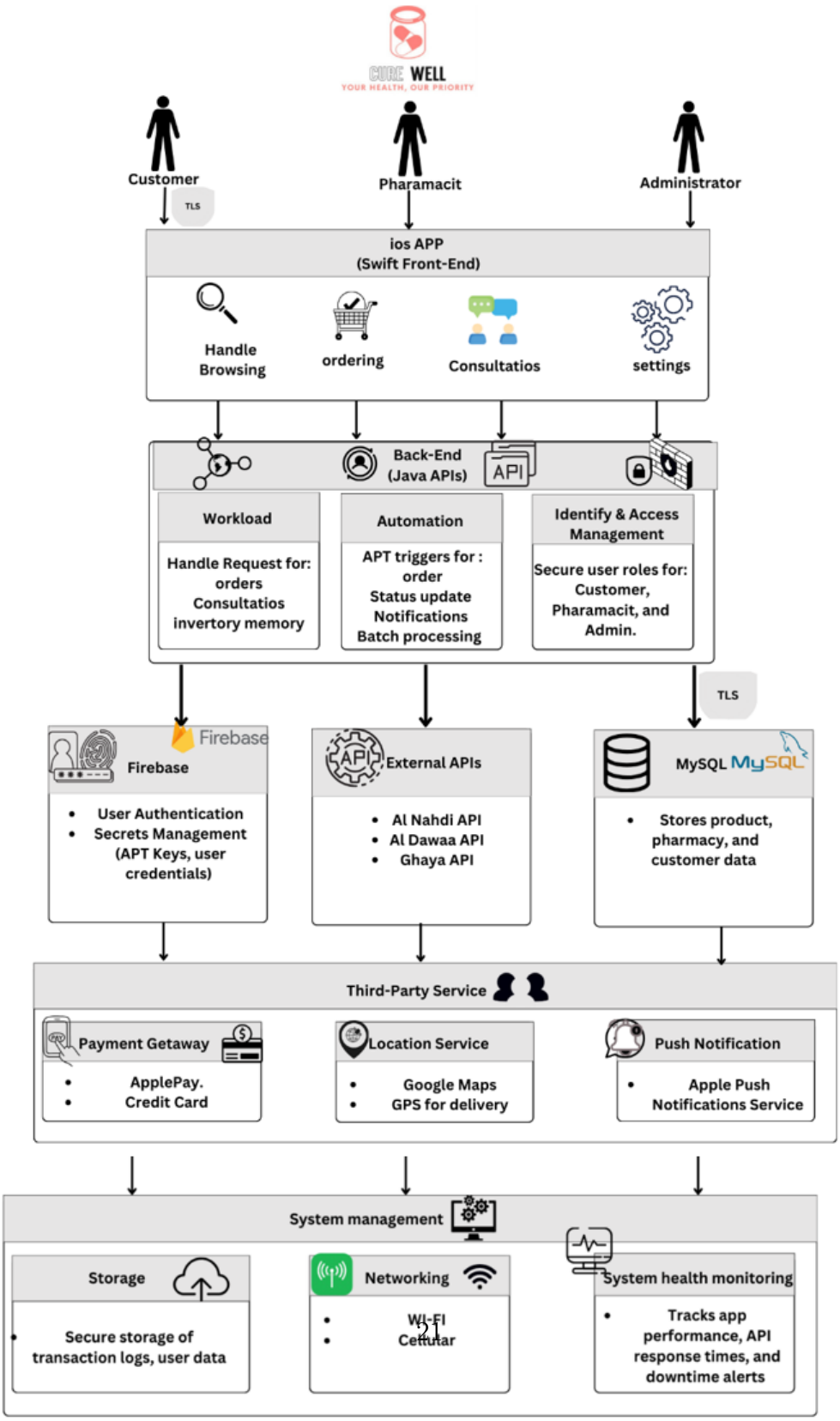
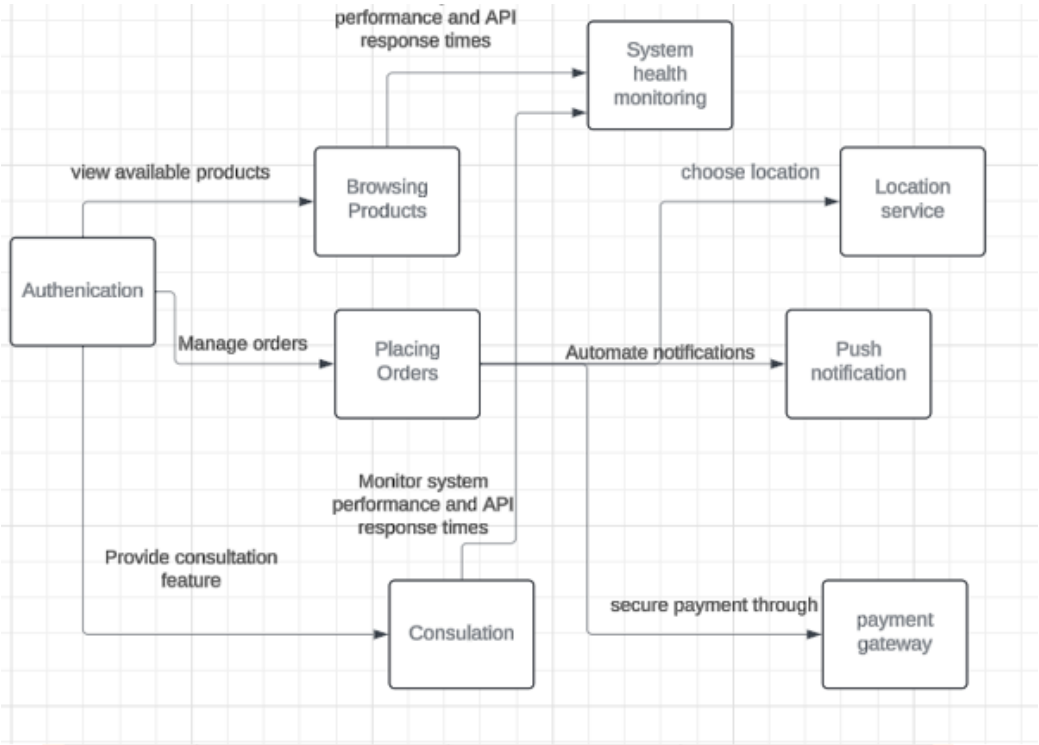


Figure 6: Software Architecture Document (SAD)

0.13 Decomposition Description



Requirement ID	Requirement Description	Design Component	Test-Case
REQ-001	Allow customers to browse pharmacy products	iOS App (Front-End) – Browsing Products	Verify that customers can search and view available products
REQ-002	Enable customers to place orders for pharmacy products	iOS App (Front-End) – Placing Orders	Confirm order placement functionality works correctly
REQ-003	Provide consultation feature between customers and pharmacists	iOS App (Front-End) – Consultation Feature	Ensure customers can send messages and receive responses
REQ-004	Authenticate users (customers, pharmacists, administrators)	Firebase – User Authentication	Test user login/logout functionality with different user roles
REQ-005	Manage orders, consultations, and inventory	Back-End (Java APIs) – Workload Management	Validate that the back-end correctly processes orders and updates
REQ-006	Automate notifications for order status updates	Back-End (Java APIs) – Automation	Test that users receive notifications upon order updates
REQ-007	Ensure secure user access control	Back-End (Java APIs) – Identity & Access Management	Check that only authorized users can access sensitive areas
REQ-008	Store sensitive information securely	Back-End (Java APIs) – Secret Management	Ensure sensitive data such as API keys are stored securely
REQ-009	Provide secure payment through a third-party gateway	Third-Party Services – Payment Gateway	Test the integration of Apple Pay for successful transactions
REQ-010	Track product and order data storage	MySQL Database – Product & Order Data	Validate data storage and retrieval for products and orders
REQ-011	Coordinate delivery using	Third-Party Services –	Test GPS tracking and

0.14 Human interface design

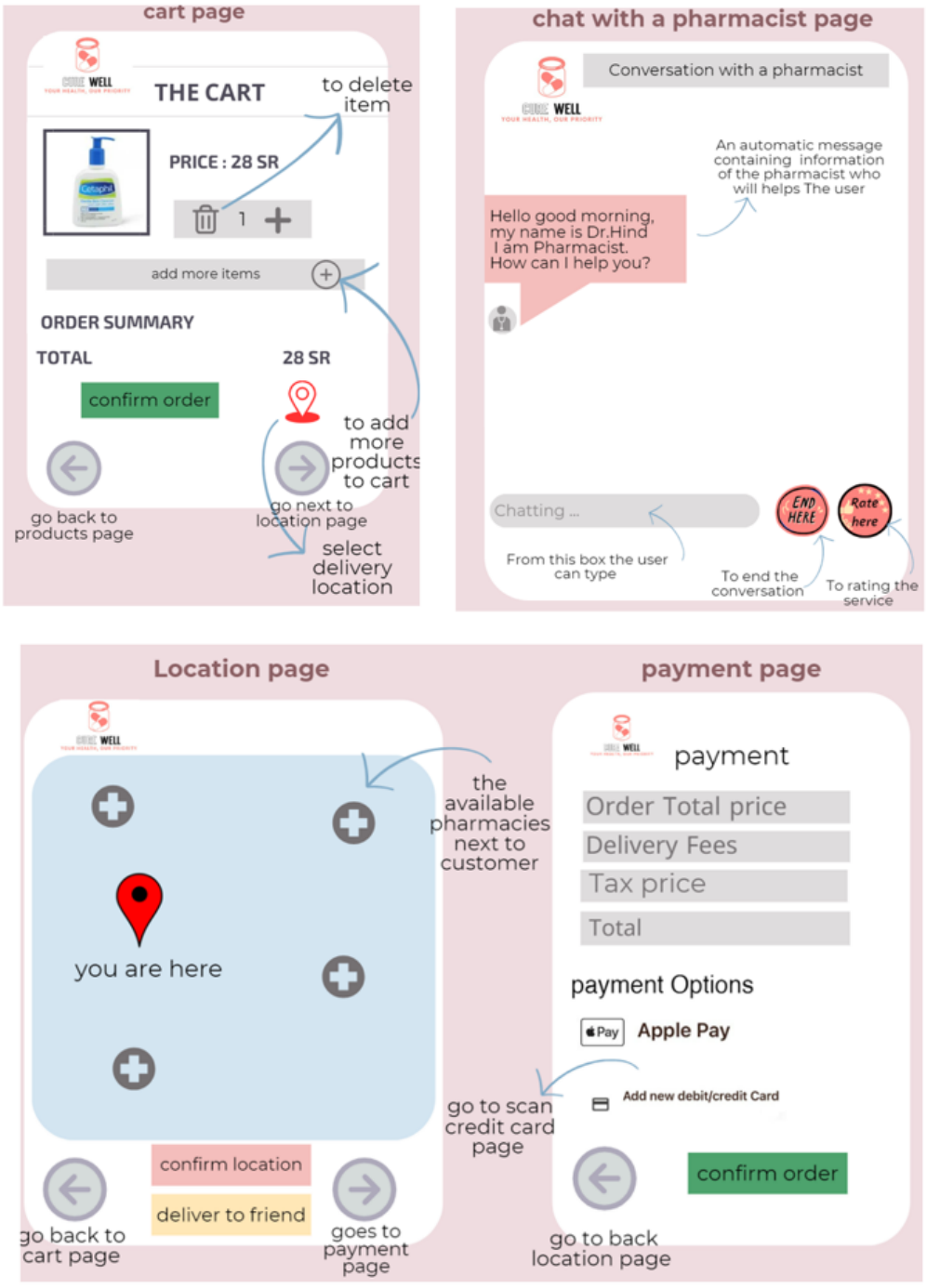


Figure 8: Human interface design

0.15 Documentation Plan

0.15.1 Readme

Figure 9: README

Audience:

1. Customers: General users who access the app to purchase medications, consult with pharmacists, and manage their orders.
2. Pharmacy Owners and Administrators: Users responsible for managing inventory, tracking orders, and overseeing pharmacy operations within the app.
3. Pharmacists: Healthcare professionals using the app to provide consultations and manage prescriptions.
4. Development Team: Developers, system architects, and quality assurance teams who are responsible for creating, testing, and maintaining the app.
5. Project Managers and Stakeholders: Individuals overseeing the development, ensuring the app meets its goals and requirements.

Software Features:

• Price Comparison: Users can browse and compare prices of medications from different pharmacies.

• Pharmacist Consultations: The app offers real-time chat functionality where users can consult with pharmacists.

• Order and Delivery Management: Users can place orders, choose delivery options (personal or to friends), and track their deliveries.

• Secure Payment Processing: The app integrates payment methods like Apple Pay and credit cards, with features like credit card scanning for easy payment.

• Login and User Authentication: Secure login for users using email and password, with enhanced security for administrators .

• Product Browsing and Search: Users can search for medications and sort results by various filters like price or pharmacy.

• Push Notifications: Users receive updates and notifications about their orders or consultations.

• GPS and Location-based Services: The app uses GPS to help users find nearby pharmacies and ensure accurate delivery addresses.

• Cart and Checkout: Users can review items in their shopping cart, adjust quantities, and proceed to checkout.

• User Feedback: Customers can rate the service they received from pharmacists.

Figure 10: README

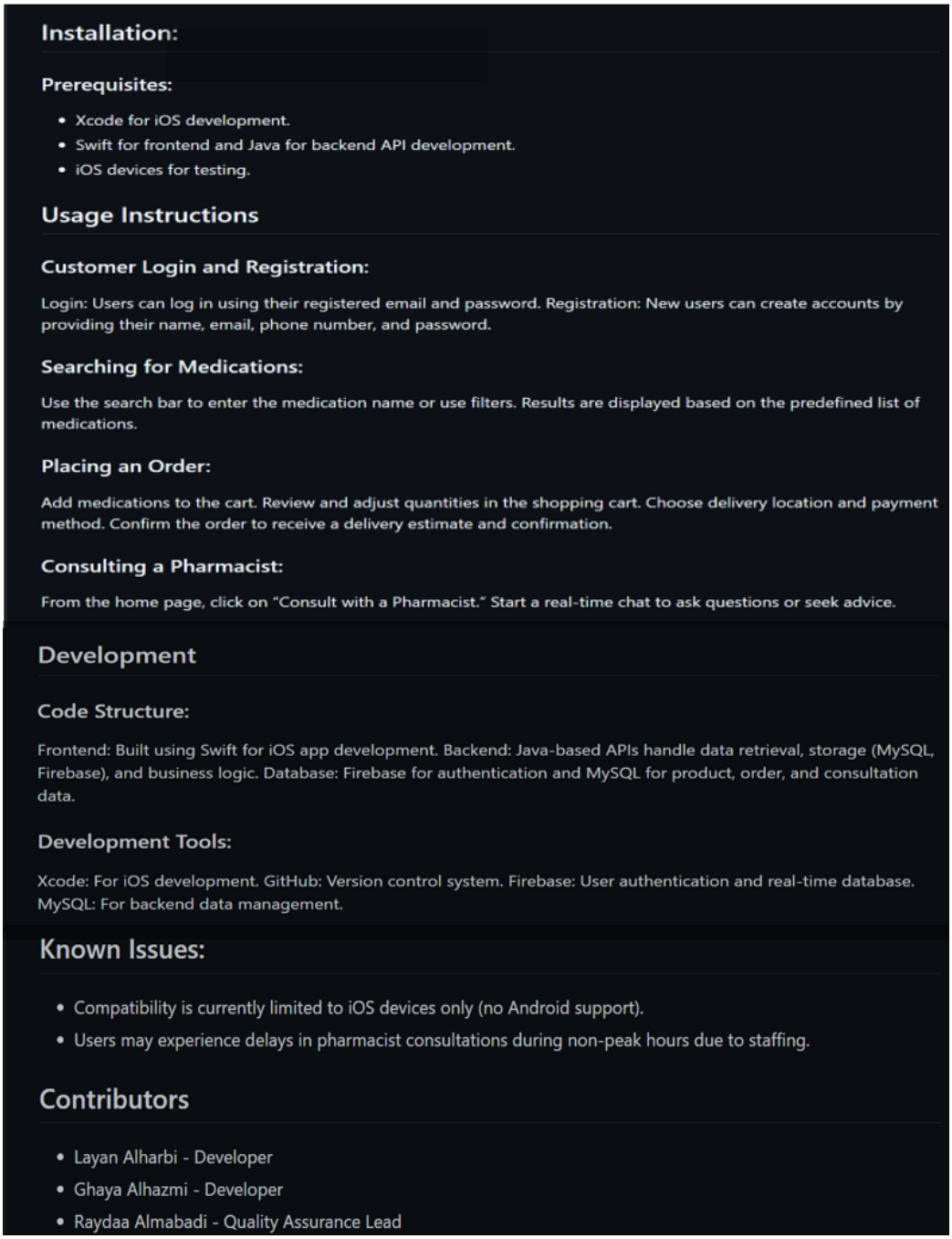


Figure 11: README

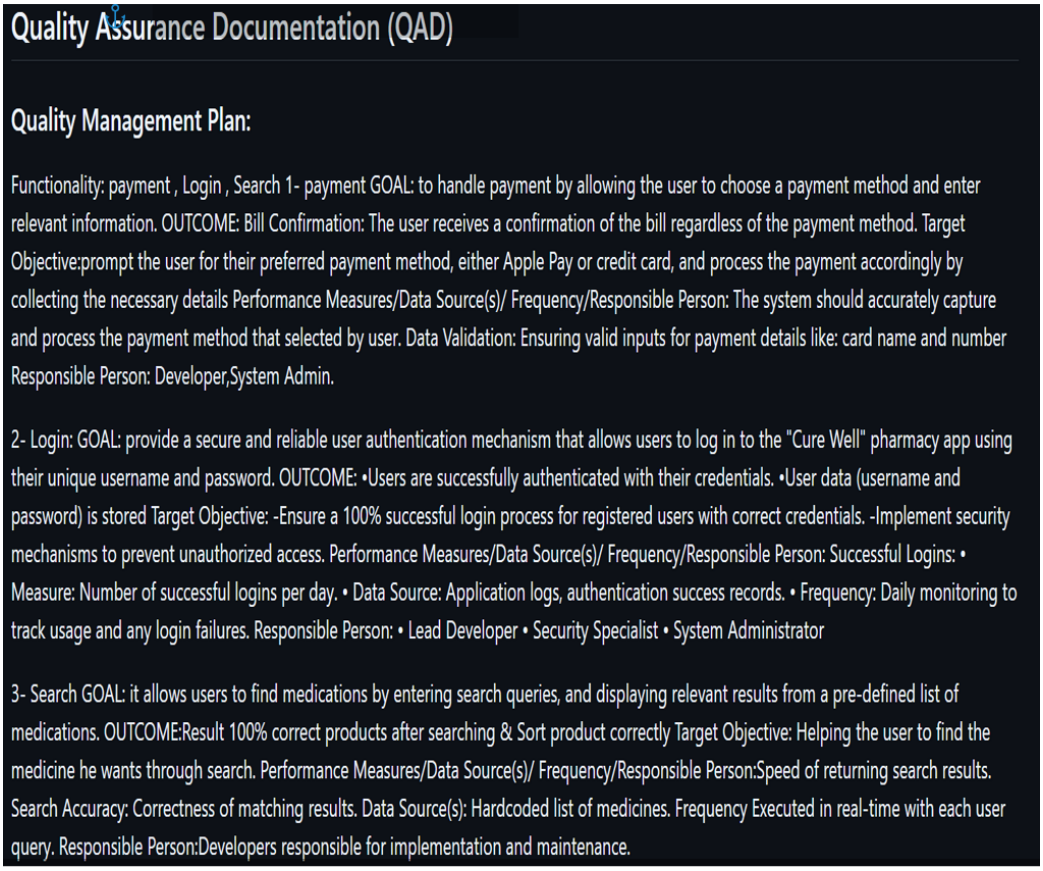


Figure 12: README

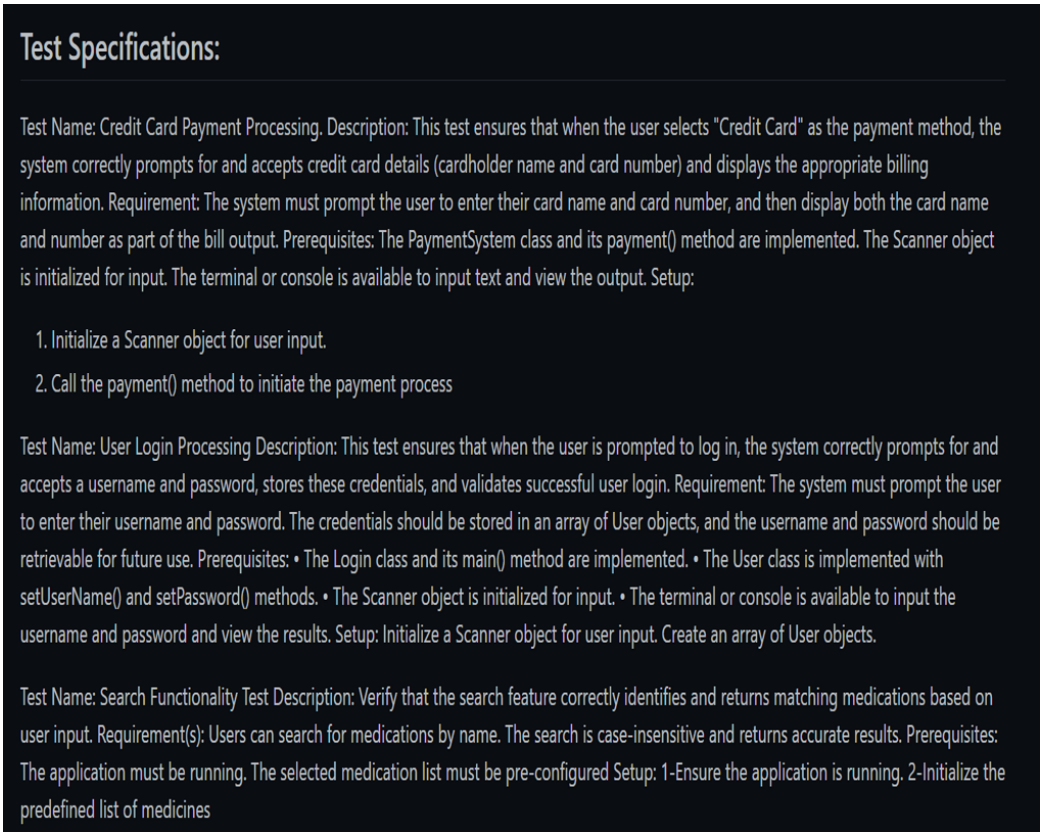


Figure 13: README

0.15.2 Troubleshooting

Troubleshooting:

Purpose:

A dedicated guide to address common technical issues users and developers may face. This is vital for minimizing frustration during the use or setup of the Cure Well app, ensuring faster resolution of problems.

Content:

Login Issues: Covers problems such as incorrect credentials, password reset errors, and account lockouts.

Payment Errors: Steps to troubleshoot failed transactions or issues with payment gateways.

Search Results Problems: Troubleshooting incorrect or no results when searching for medications.

Consultation Delays: Addresses possible causes for delays in pharmacist consultations and how to resolve them (e.g., internet issues, server downtime).

Installation Problems: Guidance on handling dependency issues or setup errors in development environments (Xcode, Swift).

Order Tracking and Delivery Issues: Helps users resolve issues with order status or delivery tracking.

Audience:

- Customers
- Pharmacy Owners/Administrators
- Developers

Figure 14: Troubleshooting

0.15.3 user Guide: *Additional type*

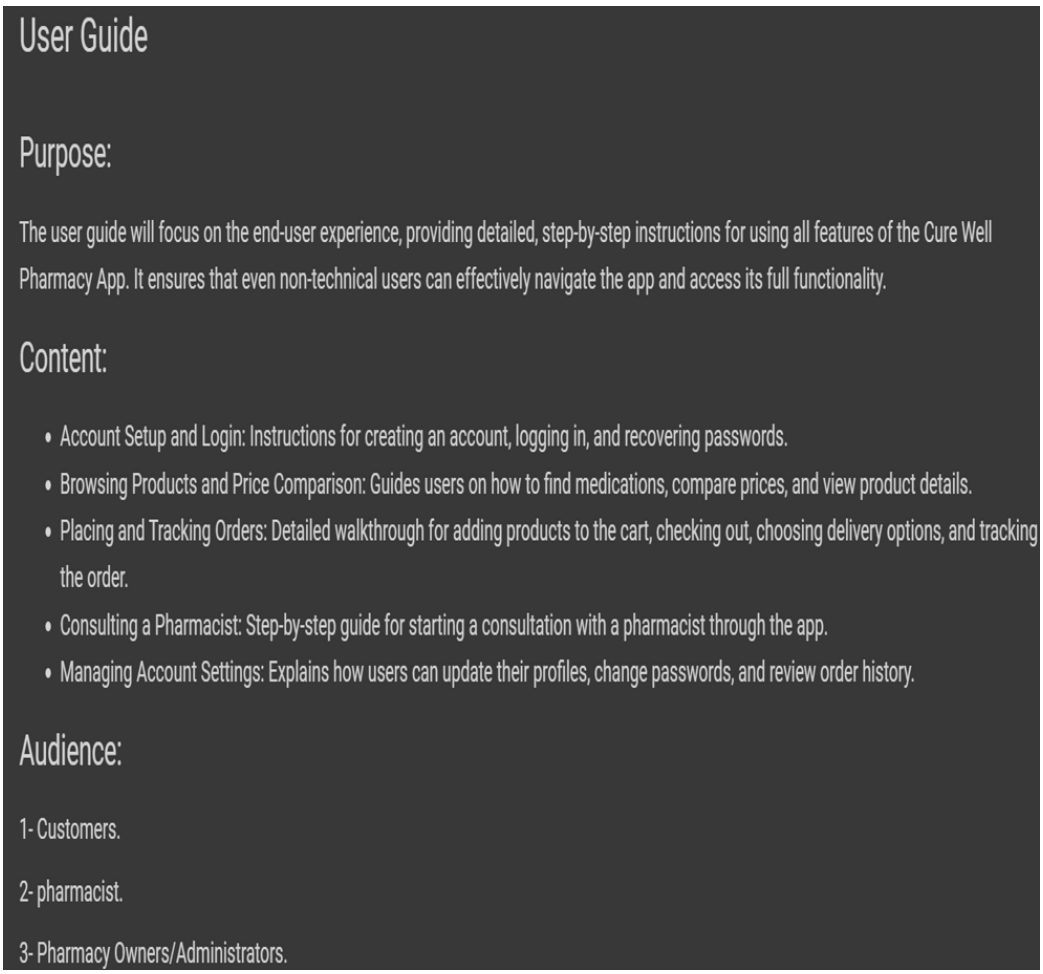


Figure 15: user Guide

0.16 visual content

0.16.1 Use Case

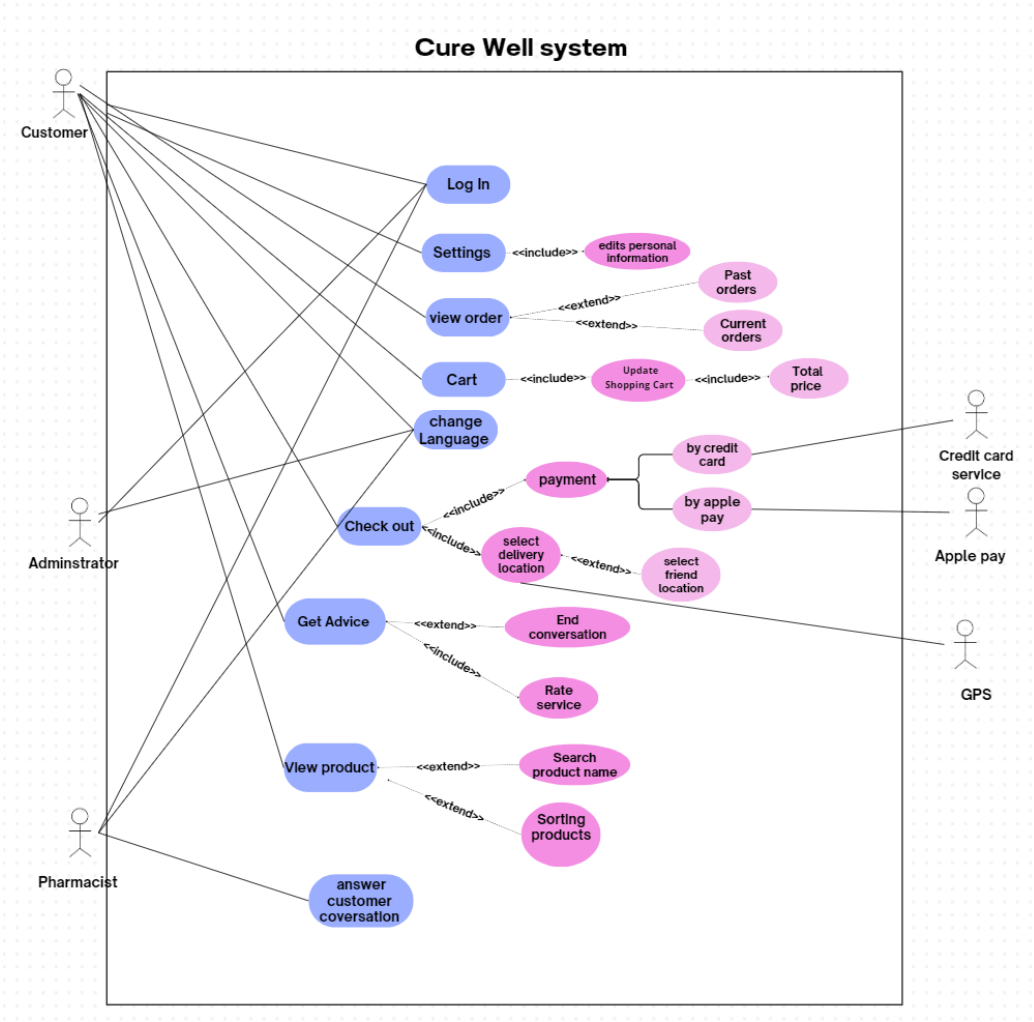


Figure 16: use case

0.16.2 Sequence Diagram

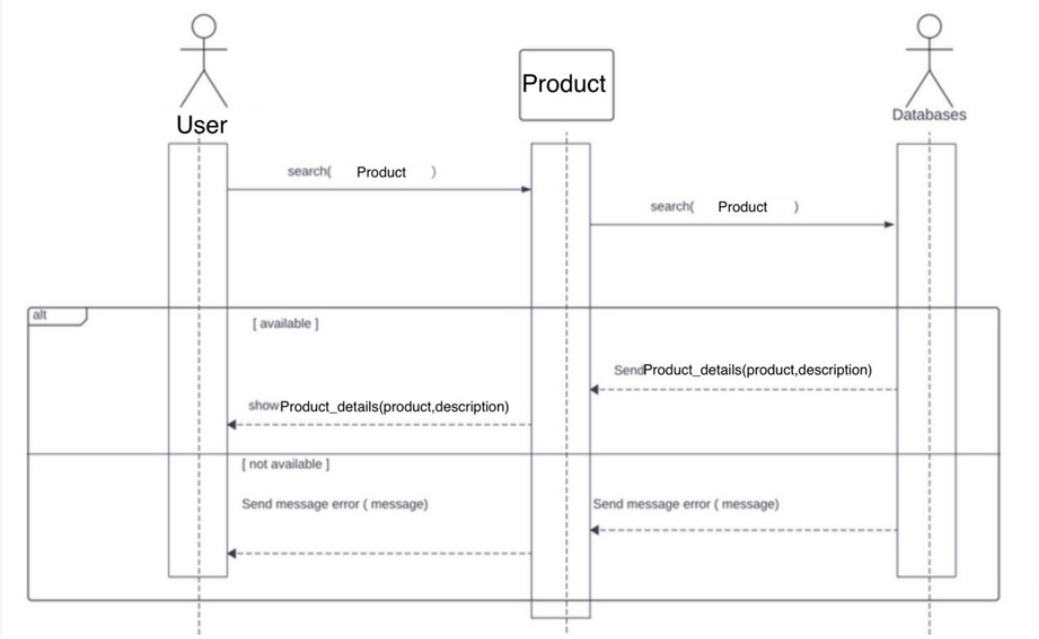


Figure 17: Sequence Diagram

0.16.3 UML Diagram

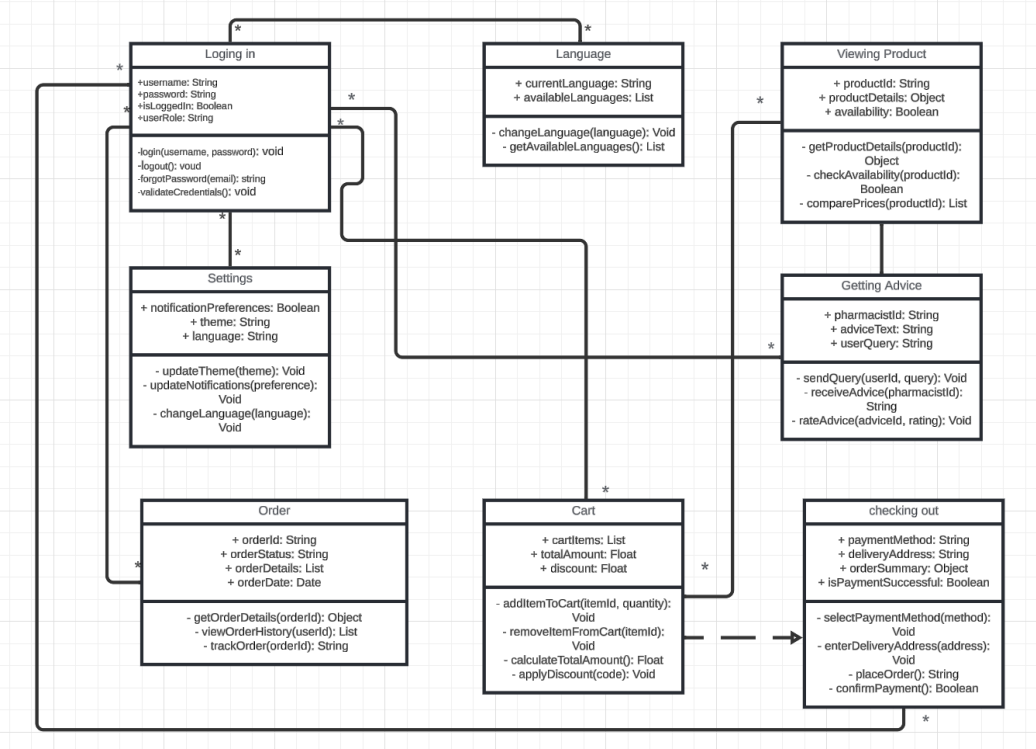


Figure 18: UML Diagram

0.17 Conclusion

The document provided for the Cure Well Pharmacy App outlines a robust framework for the development, quality assurance, and functionality of a mobile pharmacy application. The app's primary goal is to simplify pharmaceutical services, allowing customers to compare prices, consult pharmacists, and place orders efficiently.

It also provides tools for administrators and pharmacists to manage inventory and orders while offering a secure, user-friendly interface.

Key strengths of the documentation include detailed descriptions of the system architecture, user interactions, testing plans, and security measures, such as the integration of payment gateways like Apple Pay and robust authentication mechanisms.

The app is tailored for iOS devices, leveraging key features like GPS for location-based services and Apple's secure infrastructure for payments and

notifications.

The quality assurance strategies in place, including unit and integration testing, ensure that core functions like login, payment processing, and search functionality are reliable.

The system's architecture, based on cloud-based databases (Firebase, MySQL) and secure data encryption, aligns with industry standards for secure healthcare applications.

0.18 Future Work

1. **Android Compatibility:** Currently, the app is only designed for iOS devices. Extending support to Android would significantly increase the app's user base, especially in markets where Android devices are more prevalent.

2. **Offline Mode:** Introducing an offline mode that allows users to browse products and track previous orders when not connected to the internet would enhance the user experience, particularly in regions with unstable internet connectivity.

3. **Advanced AI-Based Consultations:** Implementing AI-driven chatbots could assist users when pharmacists are unavailable, providing basic consultation based on common inquiries or FAQs. This would improve response times and availability.

4. **Multi-Language Support:** Expanding the app to include multiple language options (other than arabic and english) would cater to a more diverse user base, allowing it to serve non-English speaking markets more effectively.

5. **Integration with More Payment Systems:** Expanding the range of payment gateways beyond Apple Pay to include more local payment options, such as Google Pay and popular regional payment methods, would cater to a wider audience and increase user convenience.

0.19 References

Document Name	Document Location and/or URL
Cure-well project for SE	https://www.canva.com/design/DAF2nK7c4L4/3QvJ4OHafuXPnYrQuJWfCg/edit
The report template	In blackboard
Decomposition Description	In SDD Documentation

0.20 Appendix

0.20.1 code sample

```
public class Login {

    public static void main(String[] args) {

  /* Log in method takes username and password then store it */

        User[] arrayUser = new User[n];
        for (int i = 0; i < arrayUser.length; i++) {
            System.out.print("Enter your Username: ");
            String Username = scan.nextLine();
            System.out.print("Enter your password: ");
            String password = scan.next();


            arrayUser[i] = new User();
            arrayUser[i].setUserName(Username);
            arrayUser[i].setPassword(password);

        }
```

Figure 19: Code sample - Log in

```java
package search;
import java.util.ArrayList; // Import ArrayList class
import java.util.List; // Import List interface
import java.util.Scanner; // Import Scanner class for user input

public class Search {
    // List of medicines
    private static List<String> medicines = new ArrayList<>();
    // Method to initialize the list of medicines
    static {
        medicines.add("Paracetamol"); // Add medicine to the list
        medicines.add("Ibuprofen");
        medicines.add("Amoxicillin");
        medicines.add("Cough Syrup");
        medicines.add("Vitamin C");
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Create a Scanner object for input

        System.out.println("Welcome to the Cure Well App"); // Welcome message
        System.out.print("Enter the medicine name to search: "); // Prompt for user input
        String searchQuery = scanner.nextLine(); // Read the search query from user

        List<String> results = searchMedicines(searchQuery); // Perform the search
        // Display results
        if (results.isEmpty()) {
            System.out.println("No medicines found for: " + searchQuery); // No results found
        } else {
            System.out.println("Medicines found:"); // Display found medicines
            for (String medicine : results) {
                System.out.println("- " + medicine); // Print each medicine name
            }}
        scanner.close(); // Close the scanner to prevent resource leaks
    }
    // Method to search for medicines based on the query
    private static List<String> searchMedicines(String query) {
        List<String> foundMedicines = new ArrayList<>(); // List to store found medicines
        for (String medicine : medicines) {
            // Check if the medicine name contains the search query (case insensitive)
            if (medicine.toLowerCase().contains(query.toLowerCase())) {
                foundMedicines.add(medicine); // Add matching medicine to results
            } }
        return foundMedicines; // Return the list of found medicines
    }}
```

Figure 20: Code sample - Search product

```
// this method helps to pay
 public  void payment( ){

   // there is two print statements and "scan.next" to allow the user put the way he wants to pay with
   System.out.println("What the way do you prefer to paying? ");
   System.out.println("we have \n -Apple Pay \n -Cridet Card");
   payType=scan.nextLine();

   // we use if-statement to know what the way the user wants to pay with
  // if the user want to pay by Credit Card he will put his credit information and his personal data

   if(payType.equalsIgnoreCase("Cridet card")){

   System.out.println("plese enter your card name");
   nameOfCard=scan.nextLine();
   System.out.println("plese enter your card number");
   card = scan.nextLong();
   System.out.println("\n\n");
   System.out.println(" Ok that is your bill");
   System.out.println("------------------------------------------------------ \n");
   System.out.println("credit data :\n Card Name:  "+nameOfCard +"\n Card Number: "+card);

   }
   // else if the user want to pay by Apple Pay he will only put his personal data
   else{
       System.out.println("\n\n");
       System.out.println(" Ok that is your bill");
   }
```

Figure 21: Code sample - payemnt

0.20.2 Testing plan

TEST CASE FOR USER LOGIN PROCESSING

Step	Operator	Expected Results	Observed Results	Pass/Fall
1	The operator runs the Login program.	The program starts and prompts the user to enter their username.	The program prompts the user to enter their username.	PASS
2	The operator enters a valid username.	The system accepts the input and prompts for the password.	The system accepts the username and prompts for the password.	PASS
3	The operator enters a valid password.	The system accepts the password input and stores the credentials.	The system accepts the password and stores the credentials.	PASS

Figure 22: Test plan

TEST CASE FOR SEARCH PROCESSING

Step	Operator	Expected Results	Observed Results	Pass/Fall
1	Enter a valid medicine name (e.g., "Paracetamol ") in the search field.	The application returns "Paracetamol" in the search results	displays the medicine	PASS
2	Enter an invalid medicine name (e.g., "XYZ") in the search field.	The application displays a message indicating no results found	There is no medicine	PASS
3	Enter a medicine name with different casing (e.g., "ibuprofen").	The application returns "Ibuprofen" in the search results.	Shows the medicine	PASS

Figure 23: Test plan

TEST CASE FOR PAYMENT PROCESSING

Step	Operator	Expected Results	Observed Results	Pass/ Fall
1	Select Credit Card as payment method	The system prompts for card name and card number, then displays bill with card info	The system will display the full bill with the card info.	Pass
2	Select **Apple Pay** as payment method	The system skips card info prompts and displays the bill directly	The system skipped card prompts and displayed the bill	Pass
3	Enter a string for card number (invalid data)	The system handles the error gracefully, prompting again for valid numeric input	The system threw an error but handled it gracefully, prompting for valid input	Pass

Figure 24: Test plan