**Project Title: Online Medical Store App**

**Maawiah - L1F21BSCS0078**

**~~~~~~~~~~ Phase I ~~~~~~~~~~**

**Introduction and Background:**

Our project revolves around developing an online medical store app, aiming to simplify the digital experience of purchasing healthcare products. In response to the increasing demand for convenient and efficient healthcare solutions, our app is designed to meet evolving consumer needs.

**Product (Problem Statement):**

We're crafting an online medical store app with an intuitive interface for seamless browsing, selection, and purchase of healthcare products across diverse categories.

**Background:**

The healthcare sector has witnessed a surge in online interactions, and our app seeks to contribute to this trend by providing a user-friendly and efficient platform for purchasing medical supplies.

**Scope:**

This document delineates the functionalities and features of our online medical store app, outlining its comprehensive scope.

**Objective(s)/Aim(s)/Target(s):**

- Develop a user-friendly online medical store app.

- Enhance the user experience in the healthcare product procurement process.

- Ensure secure and efficient transaction processes for medical supplies.

**Challenges:**

- Address security concerns in online transactions related to medical products.

- Ensure compatibility across various devices for a diverse user base.

**Learning Outcomes:**

- Gain insights into developing healthcare-related applications.

- Implement secure transaction protocols for medical supply transactions.

**Nature of End Product:**

The final product will be a fully functional online medical store app with an intuitive interface, secure transactions, and a comprehensive range of healthcare products.

**Completeness Criteria:**

The project will be considered complete when the app meets predefined criteria, including successful transactions, positive user feedback, and adherence to security standards specific to medical supplies.

**Business Goals:**

- Enhance customer satisfaction in the healthcare product procurement process.

- Expand the reach of medical supplies through digital channels.

**Related Work/ Literature Survey/ Literature Review:**

We've conducted a thorough review of existing online medical store apps and healthcare e-commerce platforms to inform our development process.

**Document Conventions:**

Consistent font styles and highlighting have been employed for clarity and uniformity throughout the document.

**Overall Description:**

**Product Features:**

- User account creation and management

- Comprehensive product browsing and searching

- Shopping cart functionality tailored for medical supplies

- Robust and secure payment processing for healthcare transactions

**User Classes and Characteristics:**

- Regular users: individuals requiring frequent medical supplies with varying technical expertise

- Guest users: occasional shoppers for urgent medical needs without user accounts

**Operating Environment:**

The app will be compatible with both iOS and Android platforms, necessitating an internet connection for optimal performance.

**Design and Implementation Constraints:**

- Compliance with healthcare data protection regulations

- Integration with secure payment gateways specializing in medical transactions

**Assumptions and Dependencies:**

Assumptions include a reliable internet connection for medical supply orders, and dependencies involve third-party APIs for secure payment processing in the healthcare domain.

**Project Management:**

**Work Breakdown Structure (WBS):**

1. Requirements analysis for medical store app features

2. Design and prototyping tailored for healthcare products

3. Front-end development with a focus on medical supply interface

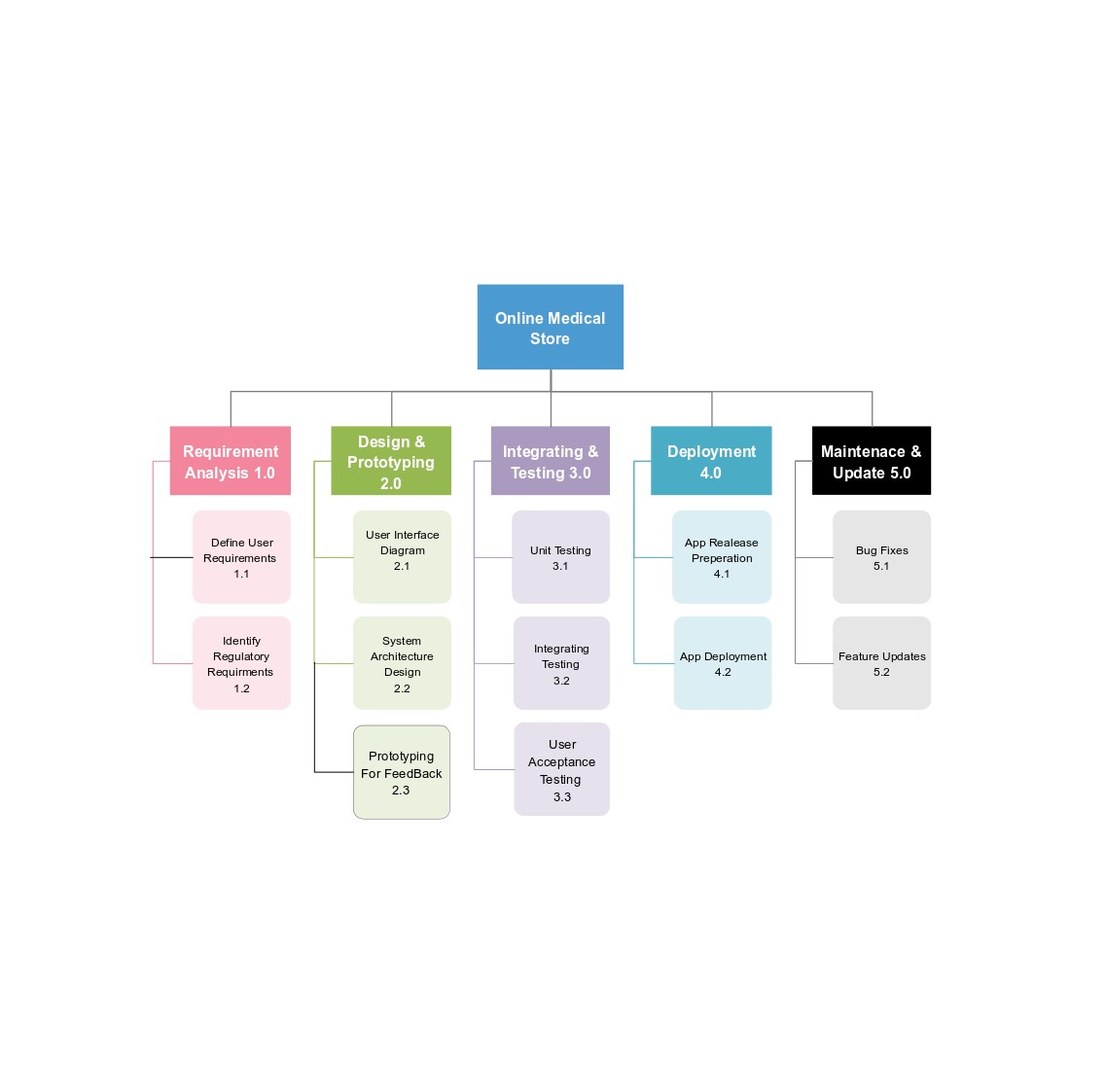
4. Back-end development specific to medical transactions

5. Integration and testing of healthcare-centric functionalities

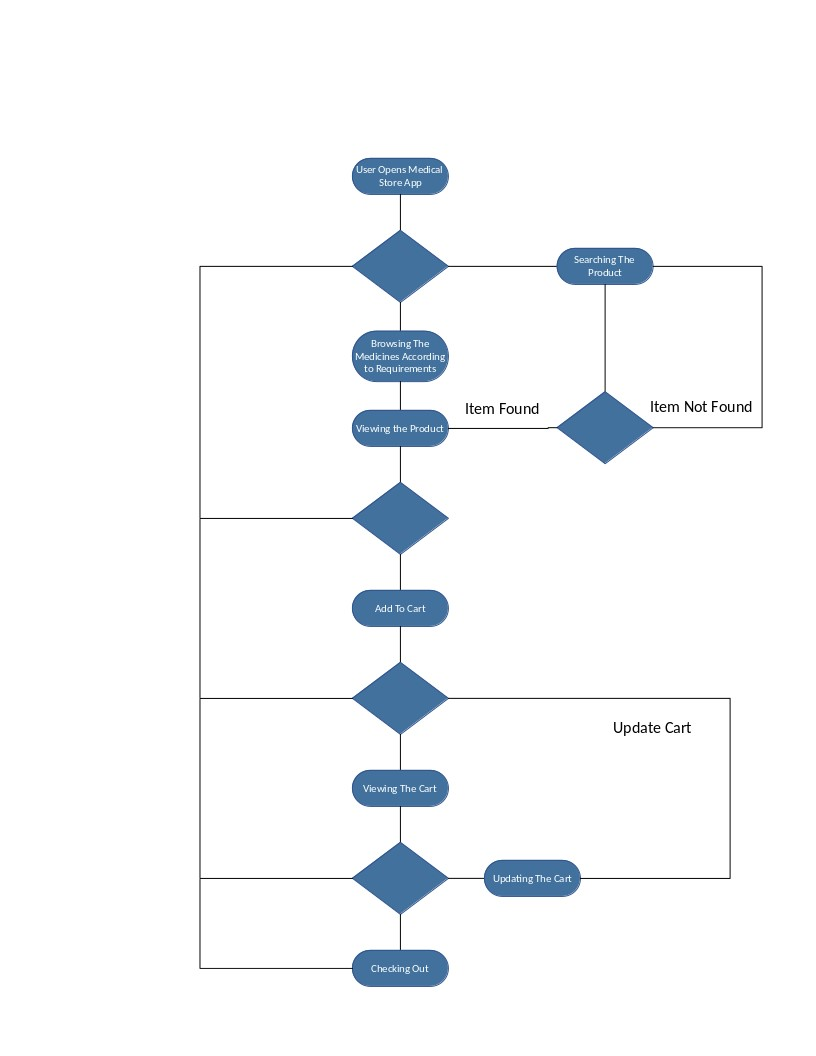
6. Deployment of the online medical store app

7. Ongoing maintenance and updates with a healthcare emphasis

**Work Breakdown Structure (WBS):**



**Activity Graph:**



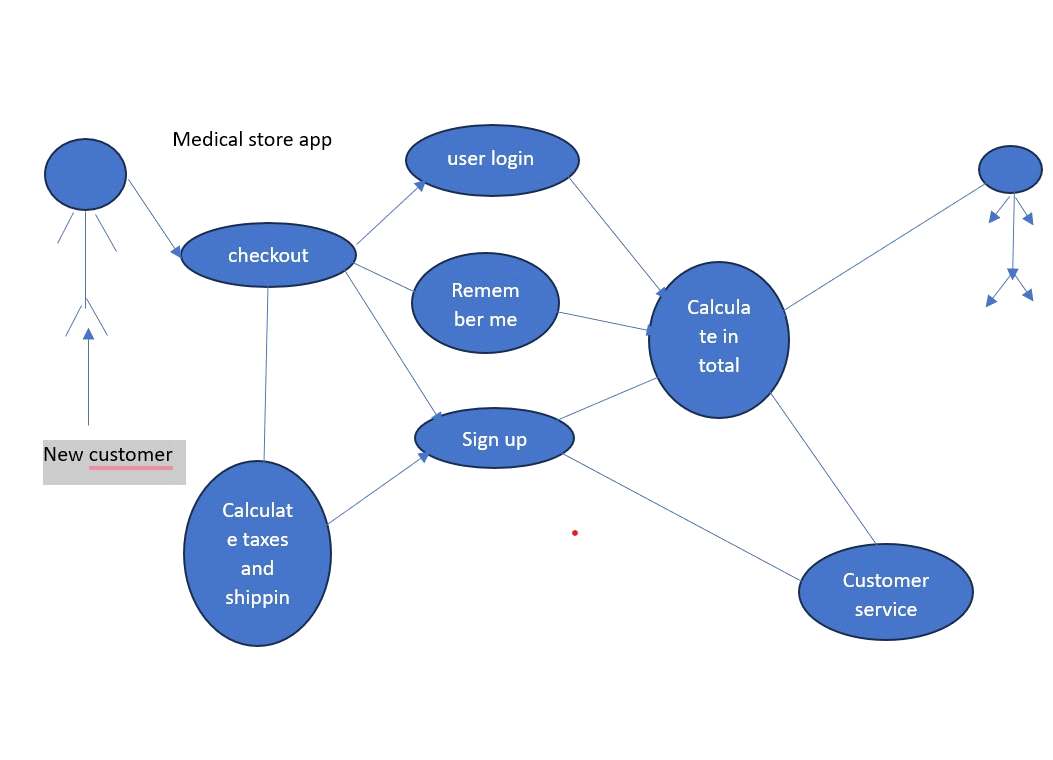
**~~~~~~~~~~ Phase II ~~~~~~~~~~**

* **Functional Requirements**
* **Name of Use-Case 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | | | UC-1 | |
| **Purpose** | | | Allow users to create, modify, and manage their accounts | |
| **Priority** | | | High | |
| **Pre-conditions** | | | User is not logged in. | |
| **Post-conditions** | | | User has a registered account. | |
| **Typical Course of Action** | | | | |
| **S#** | **Actor Action** | | | **System Response** |
| **1** | User registers with required details | | | System validates and creates the user account |
| **2** | User logs in using credentials. | | | System authenticates and grants access. |
| **3** | User requests password recovery. | | | System sends a recovery email. |
| **…** |  | | |  |
| **Alternate Course of Action** | | | | |
| **S#** | | **Actor Action** | | **System Response** |
| **1** | | User enters invalid credentials. | | System prompts for re-entry. |
| **2** | | User cancels account creation. | | System aborts account creation process. |

**Table 1: UC-1**

**UC-1**



* **Requirements Analysis and Modeling**

*<Include the following analysis models:* ***use-case diagram,******state diagram****,* ***decision table****, event table etc.*>

Certainly! Below are the requirements modified to align with a medical store application:

**Performance Requirements**

**5.1.1 Response Time:**

- The system should respond to user interactions within 2 seconds for 95% of the requests.

- The response time for critical transactions, such as updating patient records or processing prescriptions, should be less than 1 second.

**5.1.2 Concurrent Users:**

- The system must support a minimum of 500 concurrent users during peak hours without significant degradation in performance.

- Performance should be monitored and tested regularly to ensure scalability, and adjustments should be made to accommodate increased user loads.

**5.1.3 Throughput:**

- The system should be capable of handling at least 500 transactions per minute during peak usage.

- Database queries and data retrieval processes should have a throughput of at least 250 queries per second.

**5.1.4 Availability:**

- The medical store app should have an availability of at least 99.5% during regular business hours.

- Scheduled maintenance or updates should be performed during off-peak hours to minimize disruptions to users.

\*\***5.1.5 Scalability:**

- The system architecture should be designed to scale horizontally to accommodate an increasing number of users.

- Performance tests should be conducted periodically to assess the scalability of the system and make necessary adjustments.

**Safety Requirements**

**5.2.1 Secure Transactions:**

- All financial transactions, including payment processing and insurance claims, must be encrypted using industry-standard protocols (e.g., TLS) to prevent unauthorized access.

- Compliance with relevant healthcare data security standards is mandatory.

**5.2.2 User Authentication:**

- The medical store app must implement robust user authentication mechanisms to protect sensitive patient data.

- Multi-factor authentication (MFA) should be considered for enhanced security.

**5.2.3 Data Privacy:**

- The app must comply with relevant healthcare data protection laws and regulations, ensuring the privacy and confidentiality of patient data.

- Users must have control over their health information, with clear consent mechanisms for data collection and processing.

**5.2.4 Fraud Detection and Prevention:**

- Implement algorithms and mechanisms for real-time fraud detection during transactions.

- Monitor user activities and employ anomaly detection to identify and prevent fraudulent behavior related to medical records.

**5.2.5 Emergency Notification:**

- In the event of a security breach or suspected unauthorized access, the system must have mechanisms in place to notify users promptly.

- Clear and concise communication channels should be established to inform users about actions they should take to secure their health records.

**5.2.6 Compliance with Regulatory Standards:**

- The medical store app must comply with relevant safety and security standards specified by healthcare regulatory bodies.

- Regular audits and assessments should be conducted to ensure ongoing compliance with safety regulations.

**Security Requirements**

**5.3.1 User Authentication and Authorization:**

- User authentication must be enforced before granting access to the medical store app.

- Role-based access control (RBAC) must be in place to ensure that users have appropriate permissions based on their roles within the system.

**5.3.2 Data Encryption:**

- All sensitive data, including patient health records and financial transactions, must be encrypted during transmission and storage using industry-standard encryption algorithms.

- End-to-end encryption should be implemented for communication between the app and any external healthcare services or APIs.

**5.3.3 Session Management:**

- Secure session management techniques, such as session timeouts and secure session tokens, must be implemented to prevent session hijacking and unauthorized access.

- Session data should be securely stored and managed to avoid data leakage.

**5.3.4 Secure APIs:**

- Any APIs used by the medical store app must be secured with proper authentication and authorization mechanisms.

- API endpoints should be protected against common security vulnerabilities, such as injection attacks and cross-site scripting (XSS).

**5.3.5 Logging and Monitoring:**

- Comprehensive logging mechanisms must be implemented to record and monitor user activities, system events, and security-related events.

- Security monitoring tools should be in place to detect and respond to suspicious activities in real-time.

**5.3.6 Security Audits and Assessments:**

- Regular security audits and assessments must be conducted to identify and address security vulnerabilities.

- Vulnerability scanning and penetration testing should be performed periodically to ensure the ongoing security of the system.

**5.3.7 Compliance with Security Standards:**

- The medical store app must comply with industry-specific security standards relevant to healthcare, and relevant legal regulations.

- Regular assessments should be conducted to ensure ongoing compliance with security standards.

**Additional Software Quality Attributes**

**5.4.1 Usability:**

- The medical store app should achieve a System Usability Scale (SUS) score of at least 80, indicating a high level of user satisfaction.

- Common user tasks, such as updating patient information and processing prescriptions, should be accomplishable within an average time of 3 minutes.

**5.4.2 Availability:**

- The medical store app should have a minimum availability of 99.5% during both regular business hours and peak usage times.

- Downtime for maintenance or updates should be scheduled during off-peak hours to minimize the impact on users.

**5.4.3 Adaptability:**

- The app should be adaptable to various screen sizes and resolutions, providing a consistent and user-friendly experience across different devices (desktop, mobile, and tablet).

- The app should support the latest versions of major web browsers (e.g., Chrome, Firefox, Safari) to ensure compatibility.

**5.4.4 Reliability:**

- The medical store app should demonstrate a Mean Time Between Failures (MTBF) of at least 30 days, indicating a reliable and stable performance.

- The app should automatically recover from common failures, such as server errors or network disruptions, within 2 minutes.

\*\*

**5.4.5 Maintainability:**

- Code maintainability should be measured using a maintainability index, with a target score of at least 70.

- Regular code reviews and documentation updates should be performed to facilitate ongoing maintenance by the development team.

**5.4.6 Performance Efficiency:**

- The app's page load time should be optimized, with a goal of rendering critical content within 3 seconds.

- Images and other media assets should be compressed and optimized to reduce bandwidth usage and improve overall performance.

**Other Requirements**

**6.1 Database Requirements:**

- The medical store app should utilize a relational database management system (RDBMS) to store and manage healthcare data.

- Database backups must be performed regularly, and a disaster recovery plan should be in place to ensure data integrity and availability in case of unforeseen events.

\*\***6.2 External Interface Requirements:**

- The app must integrate with a secure payment gateway to facilitate online transactions related to healthcare services.

- Integration with third-party healthcare services is required for managing patient records and prescriptions.

- External APIs used by the app should be documented, and any changes to these APIs must be communicated in advance.

**6.3 Hardware Requirements:**

- The hardware infrastructure hosting the medical store app should meet or exceed the minimum system requirements specified by the chosen technology stack.

- Scalability considerations should be taken into account, and the infrastructure should be capable of handling increased loads related to healthcare data.

**6.4 Software Requirements:**

- The app should be compatible with the latest stable versions of commonly used web browsers (e.g., Chrome, Firefox, Safari).

- Compatibility with mobile platforms (iOS and Android) should be ensured for a seamless mobile experience.

**6.5 Communication Requirements:**

- Secure communication protocols, such as HTTPS, must be enforced to protect healthcare data during transit.

- Clear communication channels should be established for customer support, including email, chat, and a helpline.

**6.6 Internationalization Requirements:**

- The app should support multiple languages to cater to an international user base seeking healthcare services.

- Cultural considerations, such as date formats and patient information conventions, should be taken into account for a globally inclusive user experience.

**6.7 Legal Requirements:**

- The medical store app must comply with local and international laws and regulations regarding healthcare data, e-commerce, and patient rights.

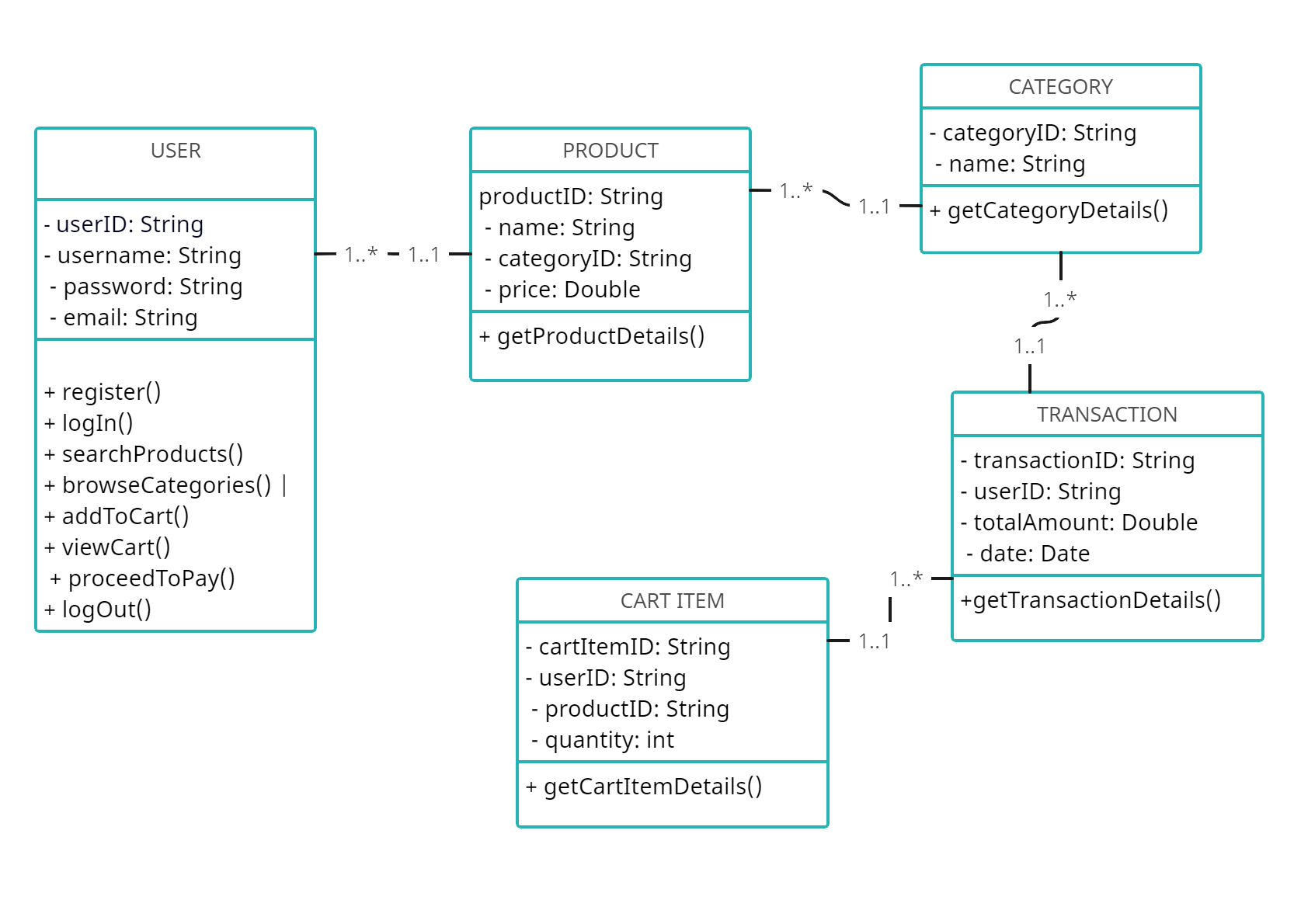
- Clear terms of service and privacy policies must be provided to users, outlining the rules and regulations governing app usage and healthcare data handling.

**6.8 Reuse Objectives:**

- Code modularization and documentation practices should be followed to facilitate code reuse within the project.

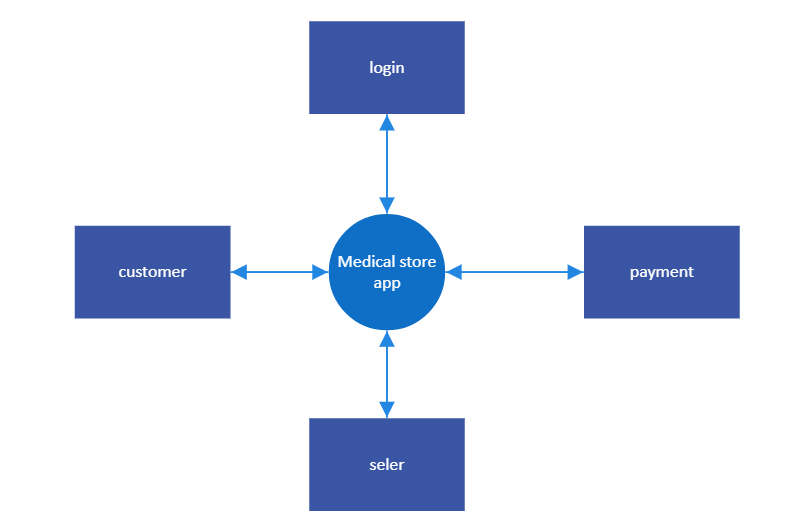
- Where applicable, open-source libraries and frameworks should be used to leverage existing solutions and promote reusability in the context of healthcare application development.

* **Designing**
* **Complete class diagram**

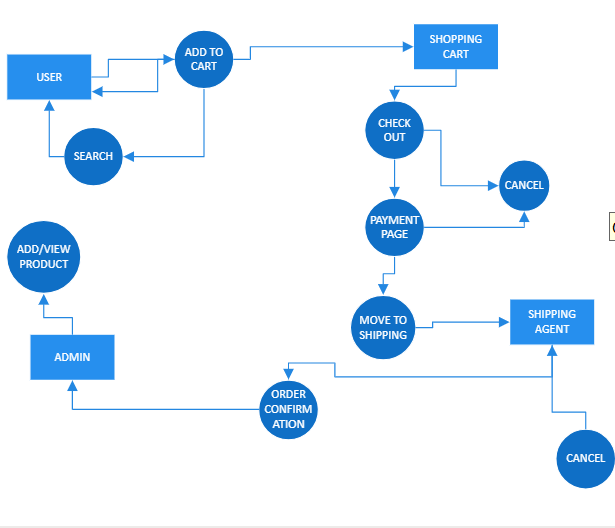


* **Complete Data Flow Diagram (DFD)**

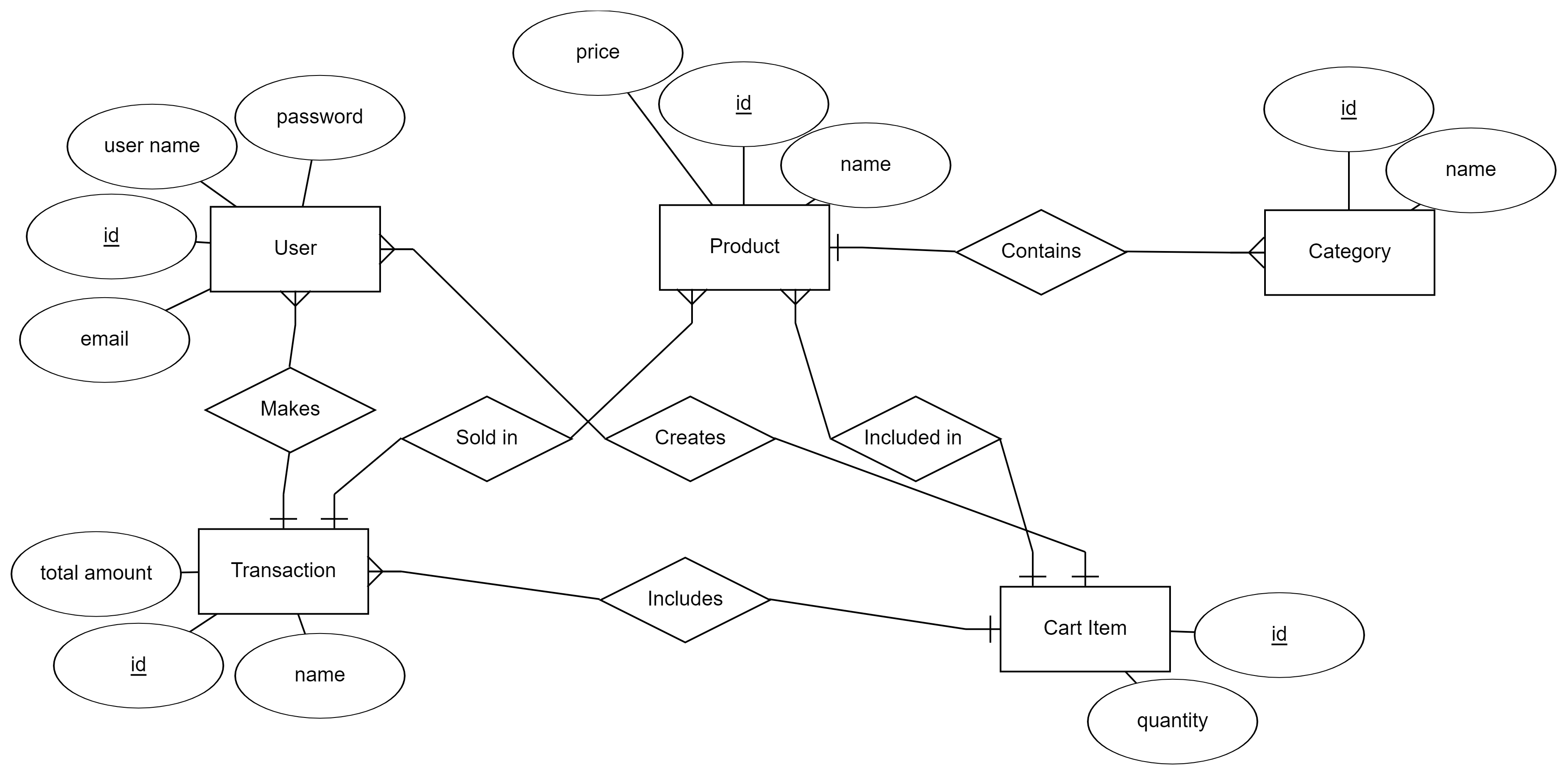
*DFD LEVEL 0*



**DFD LEVEL 1**



* **Complete ER Diagram**



* **Information on use of design patterns while designing the modules**

*<e.g. while making class diagram. Highlight/Mention the relevant classes>*

**Return on Investment (ROI) {1st year}**

1. Eliminate Noise Words:

- 1 year: 24,000 / 30,000 = 0.80

- 5 years: (24,000 \* 5) / (30,000 + (28,000 \* 5)) = 0.6154

- 10 years: (24,000 \* 10) / (30,000 + (28,000 \* 10)) = 0.6316

2. Store Indices in Bins:

- 1 year: 280,000 / 300,000 = 0.93

- 5 years: (280,000 \* 5) / 300,000 = 4.6667

- 10 years: (280,000 \* 10) / 300,000 = 9.3333

3. Add Second Server:

- 1 year: 110,000 / 30,000 = 3.6667

- 5 years: (110,000 \* 5) / 30,000 = 18.3333

- 10 years: (110,000 \* 10) / 30,000 = 36.6667

**% Gain on ROI {1st year}**

1. Eliminate Noise Words:

- 1 year: (24,000 \* 1 - (30,000 + (28,000 \* 1))) / (30,000 + (28,000 \* 1)) \* 100 = -6.67

- 5 years: (24,000 \* 5 - (30,000 + (28,000 \* 5))) / (30,000 + (28,000 \* 5)) \* 100 = -8.07

- 10 years: (24,000 \* 10 - (30,000 + (28,000 \* 10))) / (30,000 + (28,000 \* 10)) \* 100 = -2.31

2. Store Indices in Bins:

- 1 year: (280,000 \* 1 - 300,000) / 300,000 \* 100 = -6.67

- 5 years: (280,000 \* 5 - 300,000) / 300,000 \* 100 = -8.07

- 10 years: (280,000 \* 10 - 300,000) / 300,000 \* 100 = -2.31

3. Add Second Server:

- 1 year: (110,000 \* 1 - 30,000) / 30,000 \* 100 = 266.67

- 5 years: (110,000 \* 5 - 30,000) / 30,000 \* 100 = 733.33

- 10 years: (110,000 \* 10 - 30,000) / 30,000 \* 100 = 1223.33

**Payback Period in Years**

- Eliminate Noise Words: 30,000 / 24,000 = 1.25 years

- Store Indices in Bins: 300,000 / 280,000 = 1.07 years

- Add Second Server: 30,000 / 110,000 = 0.27 years

**FP Based Estimation**

- FP estimation = C.T[(0.65 + (0.01 \* ∑Fi)]

- = 450[0.65 + (0.01 \* 76)]

- = 450[0.65 + 0.76]

- = 634.5

**COCOMO Estimation**

- Productivity = 8 FP/m

- Labour Rate = $905

- Total Number of FP estimation = 30,000

- Cost per FP estimation = Labour Rate / Productivity

- = $905 / 8.0

- = $113.125

- Effort for Project = Total NO. Of LOC / Productivity

- = 30,000 / 500

- = 60 per month