## **Online Exchange Using iOS**

## Software Requirements Specification

## <iOS>

## <05/02/2024>

## Akshay Kumar Gupta

## Namruth Potla Bala Venkata Sai Kishore Vattumilli Kumar Chandu Mallireddy Nithish Kumar Madavali

## Submitted in partial fulfillment

## Of the requirements of

## CSIS 44-691 Graduate Directed Project 1

## 

## 

## 

## 

## 

## 

## 

# 

# Revision History

## 

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Author | Comments |
| <date> | <Version 1> | <Your Name> | <First Revision> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Document Approval

## The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Signature | Printed Name | Title | Date |
|  | Akshay Kumar Gupta Attuluru |  |  |
|  | Namruth Potla |  |  |
|  | Kumar Chandu Mallireddy |  |  |
|  | Bala Venkata Sai Kishore Vattumilli |  |  |
|  | Nithish Kumar Madavali |  |  |

## 

## Table of Contents Page Number

## 1. Introduction

## 1.1.Purpose

The documentation seems to encompass the key features and benefits of an online exchange service with inventory tracking and bill generation capabilities. It highlights the system's ability to streamline inventory management, track sales transactions, and generate invoices, thereby improving overall efficiency and customer experience. The mention of integration with other services for shipping and payment processing suggests a comprehensive solution for businesses looking to optimize their operations. The architecture outlined, including a backend database and online user interface, indicates a well-structured system design. Overall, the document effectively communicates the system's value proposition and functionality.

## 1.2.Scope

The scope of the project includes developing a comprehensive sales and inventory management system. Sales transaction information, including the products sold, quantities, prices, and dates, will be recorded by the system. Using this data, bills with a thorough analysis of the clients' purchases will be generated. The system will also provide real-time inventory-level management. This will guarantee that companies have current, accurate stock level information.

The system will also provide sales performance statistics, which will include insights into overall sales, best-selling products, and long-term trends. Businesses may use this information to make well-informed decisions about inventory control and promotions. Additionally, the system will include report-generating features that will let users create a variety of reports, including profit margins, inventory levels, and sales summaries. The system will also manage product characteristics, such as pricing, stock levels, and descriptions, and customer data, which includes contact information and purchase history.

## 1.3.Definitions, Acronyms, and Abbreviations

## IOS: Apple's mobile operating system.

## Swift: Apple's programming language for iOS development.

## API: Application Programming Interface.

* Xcode: Xcode is an integrated development environment (IDE) developed by Apple for macOS. It is primarily used for developing software for macOS, iOS, iPadOS, watchOS, and tvOS.

## 

## 

## 

## 

## 1.4.References

## Xcode : <https://developer.apple.com/documentation/xcode>

Swift : <https://developer.apple.com/documentation/swift>

SwiftUI Animation : <https://developer.apple.com/documentation/SwiftUI/Animations>

## 

## 1.5.Overview:

The online sales tracking and inventory system with bill generation aims to provide businesses with a comprehensive solution for managing their inventory, tracking sales, and generating bills. The system will streamline operations and improve efficiency by automating manual tasks and providing real-time data and insights. Through its various features and capabilities, the system will help businesses optimize their inventory levels, improve customer service, and enhance overall business performance.

## 2. General Description

## 2.1.Product Perspective

## The online exchange platform and inventory management system with bill generation is a versatile tool that can work independently or alongside other business software. It's designed to fit smoothly into existing systems, like accounting or customer relationship management (CRM) software, to make inventory management and sales tracking more efficient.

## 2.2.Product Functions

## Log sales transactions: The system will capture details of each sales transaction, including the items sold, quantities, prices, and dates. All these details will be stored in Firebase

## Generate invoices: Invoices will be automatically generated based on sales transactions, providing clients with a detailed breakdown of their purchases.

## Provide sales performance data: The system will offer insights into sales performance, including total sales, top-selling items, and trends over time.

## Maintain customer data: The system will store customer information, including contact details and purchase history.

## Manage product details: The system will store information about products, including descriptions, prices, and stock levels.

## 2.3.User Characteristics

## The system is designed for use by businesses of all sizes, from small retailers to large enterprises. Users may include sales staff, inventory managers, accounting personnel, and business owners. Users should have basic computer skills and familiarity with inventory management and sales tracking concepts.

## 

## 2.4.General Constraints

## The system must be able to handle a large volume of transactions and data.

## The system should be secure and comply with relevant data protection regulations.

## The system should be scalable to accommodate future growth and changes in business requirements.

## 2.5.Assumptions and Dependencies

## The system assumes that users will have access to a stable internet connection for real-time data updates.

## The system may depend on third-party services or APIs for certain functionalities, such as payment processing or shipping integration and some other.

## The system assumes that users will receive adequate training and support to effectively use the system.

## 3. Specific Requirements

## 3.1.External Interface Requirements

## 3.1.1.User Interfaces

* The system shall have a user-friendly iOS app interface optimized for iPhone and iPad devices.
* The app shall allow users to log in securely using username and password credentials.
* Upon logging in, the app shall display a dashboard showing key metrics such as sales summaries, inventory levels, and top-selling items.
* Users shall be able to navigate through different sections of the app, including sales, and inventory.
* The app shall allow users to create, edit, and delete sales transactions, invoices, inventory items, and customer records.

## 3.1.2.Hardware Interfaces

The app shall be compatible with iPhone and iPad devices running iOS 13 or later.

## 3.1.3.Software Interfaces

## The app shall interact with a backend server for data storage and processing.

## The app shall utilize iOS frameworks and libraries for functionalities such as networking, data persistence, and user authentication.

## 3.1.4.Communications Interface

## The app shall use secure communication protocols for communication between the app and the backend server.

## The app shall support show-up alerts for important updates, such as order confirmations, invoice generation, and low inventory alerts.

## 3.2. Functional Requirements

1. User Authentication:
   * Description: User authentication is crucial for securing user accounts and ensuring that only authorized individuals have access to the app.
   * Requirements:
     1. Users should be able to sign up using their email address or social media accounts.
     2. Users should be able to sign in securely with their credentials.
     3. Implement encryption protocols to protect user credentials and personal information.
2. Browse Listings:
   * Description: The browse listings feature is essential for users to explore available items or services for exchange.
   * Requirements:
     1. Design an intuitive interface for users to navigate through listings.
     2. Include options for sorting listings by category, popularity, or proximity.
     3. Each listing should display relevant details such as images, descriptions, current bid status, and remaining time for bidding.
3. Create Listings:
   * Description: Empower users to create their listings for items or services they wish to exchange.
   * Requirements:
     1. Develop a user-friendly listing creation process with options to upload images, write detailed descriptions, and set starting bid prices.
     2. Include validation checks to ensure that listings adhere to platform guidelines and standards.
4. Bidding System:
   * Users should be able to place bids on products listed by owners.
   * The platform should display the current highest bid for each product.
   * Owners should be able to accept or reject bids.
5. Transaction Management:
   * The platform should facilitate secure transactions between owners and customers.
   * Payment processing should be integrated into the platform, supporting multiple payment methods.
   * The platform should provide notifications to users regarding the status of their transactions.
6. Search Functionality:
   * Users should be able to search for products based on various criteria, such as category, price range, and location.
   * The search results should be displayed in a clear and organized manner.
7. User Feedback and Ratings:
   * Users should be able to leave feedback and ratings for each other after completing a transaction.
   * The platform should display the average rating for each user based on their feedback.
8. Wishlist:
   * Users should be able to create and manage a wishlist of items they are interested in.
   * The platform should provide recommendations based on the items in the user's wishlist.
9. Responsive Design:
   * The platform should be responsive and work well on a variety of devices, including smartphones, tablets, and desktop computers.
   * The user interface should adapt to different screen sizes and resolutions.
10. Messaging System:
    * Users should be able to communicate with each other through a messaging system.
    * The messaging system should support text messages, images, and attachments.
    * Messages should be stored securely and accessible only to the users involved in the conversation.
11. Admin Panel:
    * Description: Provide administrators with a dedicated panel to manage users, listings, and transactions.
    * Requirements:
      1. Allow administrators to add, edit, and remove users and listings.
      2. Provide tools for managing transactions, resolving disputes, and enforcing platform policies.
12. User Profile Management:
    * Description: Allow users to manage their profiles, including updating personal information and viewing transaction history.
    * Requirements:
      1. Users should be able to update their profile information such as name, email, and profile picture.
      2. Display a summary of transaction history, including past purchases and sales.
13. Notifications:
    * Description: Keep users informed about important events such as new messages and bid updates.
    * Requirements:
      1. Send notifications for new messages, bid updates, and transaction status changes.
      2. Allow users to customize notification settings based on their preferences.
14. Privacy and Security:
    * Description: Ensure that user data is protected and that the platform complies with relevant privacy regulations.
    * Requirements:
      1. Implement secure data storage and transmission practices.
      2. Provide users with control over their privacy settings and data-sharing preferences.
15. Feedback and Ratings:
    * Description: Allow users to leave feedback and ratings for each other after completing a transaction.
    * Requirements:
      1. Users should be able to rate their experience with other users based on the transaction.
      2. Display average ratings for each user based on their feedback.

## 3.3. Use Cases

Online Exchange describes the various interactions between users and the system to achieve specific goals related to buying, selling, and managing items or services. These interactions are essential for the functionality and usability of the system. Use cases help to define the requirements of the system and provide a clear understanding of how users will interact with it

* User Registration: The user wants to create a new account on the platform.
* User Login: The user wants to log in to their account.
* Browse Listings: The user wants to view available items or services for exchange.
* Create Listing: The user wants to create a new listing for an item or service they wish to exchange.
* Place Bid: The user wants to place a bid on a listed item.
* Accept Bid: The user wants to accept a bid on their listed item.
* Manage Inventory: The user wants to view, add, edit, or remove items from their inventory.
* View Transaction History: The user wants to view their past transactions.
* Contact Seller: The user wants to communicate with a seller about a listing.
* Generate Invoice: The system generates an invoice for a completed transaction.

## 3.4. Class/Objects

1. User
   1. Attributes:
      1. Username
      2. Email
      3. Password
      4. Methods:
         1. Register()
         2. Login()
         3. UpdateProfile()
2. Listing
   1. Attributes:
      1. Title
      2. Description
      3. Price
      4. Category
      5. Methods:
         1. Create()
         2. Edit()
         3. Delete()
3. Bid
   1. Attributes:
      1. ID
      2. Amount
      3. Bidder (User)
      4. Listing (Listing)
      5. Methods:
         1. PlaceBid()
4. Transaction
   1. Attributes:
      1. ID
      2. Listing (Listing)
      3. Buyer (User)
      4. Seller (User)
      5. Amount
      6. Status
         1. Methods:
            1. ProcessTransaction()
5. Message
   1. Attributes:
      1. ID
      2. Sender (User)
      3. Receiver (User)
      4. Content
      5. Timestamp
      6. Methods:
         1. SendMessage()
6. Invoice
   1. Attributes:
      1. ID
      2. Transaction (Transaction)
      3. TotalAmount
      4. DateGenerated
      5. Methods:
         1. GenerateInvoice()

## 3.5. Non-Functional Requirements

## 3.5.1. Performance

* The system should be responsive and provide quick access to listings and user actions.
* The app should load listings and images quickly, even with a large number of concurrent users.

## 3.5.2. Reliability

* The system should be reliable, with minimal downtime or disruptions.
* It should handle user interactions, transactions, and data storage reliably.

## 3.5.3. Availability

## The system should be available 24/7, allowing users to access it at any time.

## It should have mechanisms in place to handle spikes in traffic and maintain availability..

## 3.5.4.Security

## The system should ensure the security of user data, including passwords, personal information, and transaction details.

## It should use encryption for data transmission and storage and implement secure authentication methods.

## 3.5.5.Portability

* The app should be compatible with a range of iOS devices, including iPhones and iPads.
* It should be designed to work smoothly on different screen sizes and resolutions.

## 3.6. Inverse Requirements

## The system should not allow unauthorized access to user accounts or data.

## It should not store sensitive information such as passwords or credit card details in plain text.

## 3.7. Design Constraints

## The app must be developed using Swift for iOS development.

## Firebase must be used as the backend database for storing user data, product listings, and transactions.

## The app must follow Apple's design guidelines for iOS apps.

## 3.8. Logical Database Requirements

## The system will utilize Firebase as the backend database.

## Firebase provides real-time database capabilities, crucial for updating listings, transactions, and user information in real-time.

## Firebase offers secure authentication and data storage, ensuring compliance with privacy regulations.

## The database design will be optimized for efficient data retrieval and storage, considering user profiles, listings, transactions, and messaging.

## Firebase's scalability will accommodate future growth and changes in the platform's requirements.

## The system will leverage Firebase's cloud functions for implementing backend logic such as invoice generation, bid management, and transaction processing.

## Firebase's features and capabilities make it a suitable choice for the online exchange platform's database requirements.

## 3.9. Other Requirements

## The system must protect user data and follow privacy laws.

## Data transmission and storage must be secure.

## Users should have control over their privacy settings.

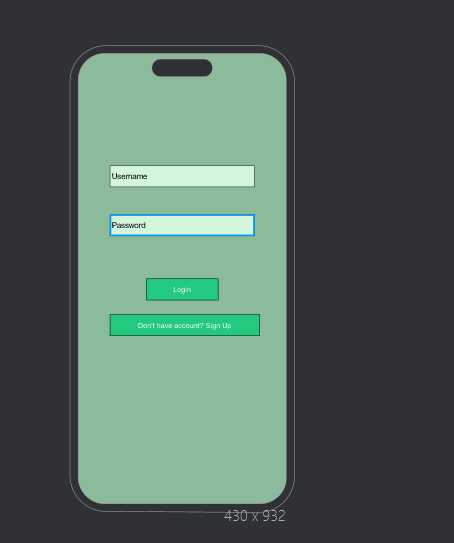
## The system must support various payment methods for transactions.

## 

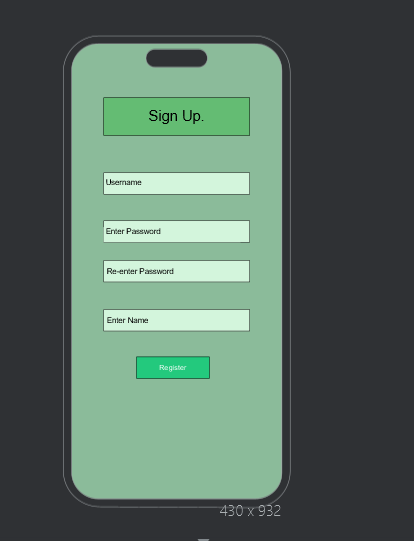
## 

## 3.10. Prototypes (for complete project)

User Interface Prototypes



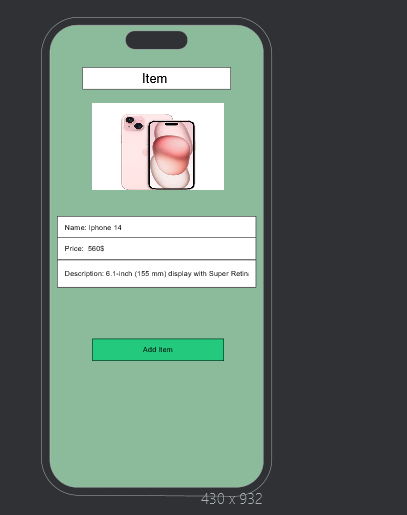
* Login Screen:
  + Input fields for username and password.
  + Login button.



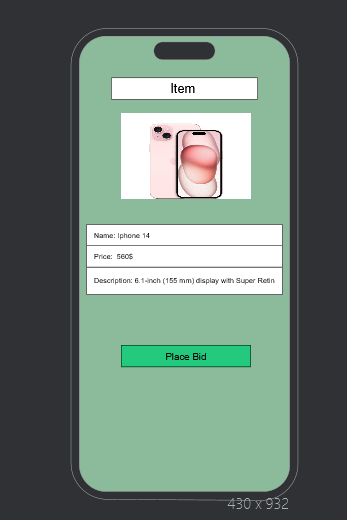
* Registration Screen:
  + Input fields for username, email, password, and any other required information.
  + Register button.



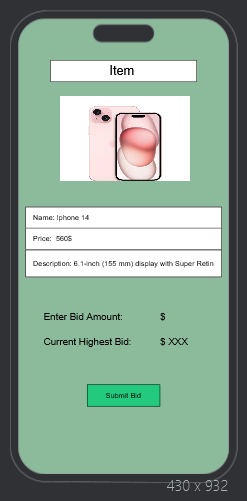
* Browse Listings:
  + List of available items or services for exchange.
  + Search and filter options.
  + Details for each listing including images, descriptions, current bid status, and remaining time for bidding.



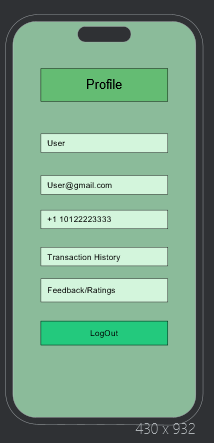
* Create Listing:
  + Form to create a new listing with fields for title, description, price, and category.
  + Option to upload images.



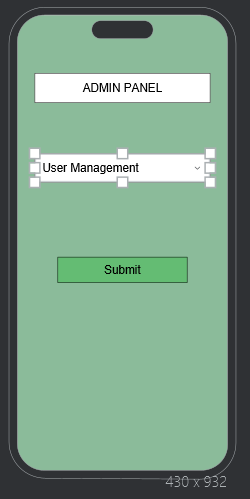
* Product Details:
  + Detailed view of a product including images, description, price, and bidding options.



* Bidding Interface:
  + Interface for placing bids on products.



* User Profile:
  + User's profile information including username, email, and profile picture.
  + Transaction history and feedback/ratings.



* Admin Panel (if applicable):
  + Interface for administrators to manage users, listings, and transactions.

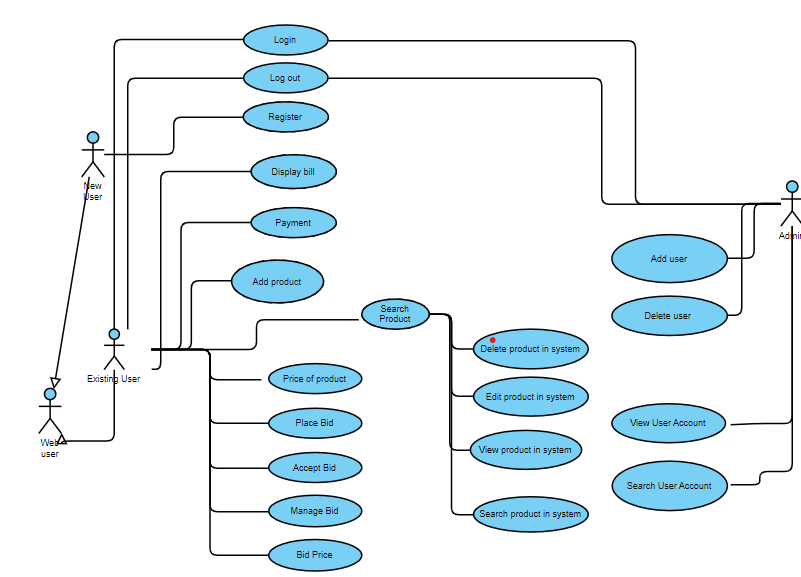
## 

## 

## 

## 

## 3.11. Use Case Diagrams



## 

## 

## 

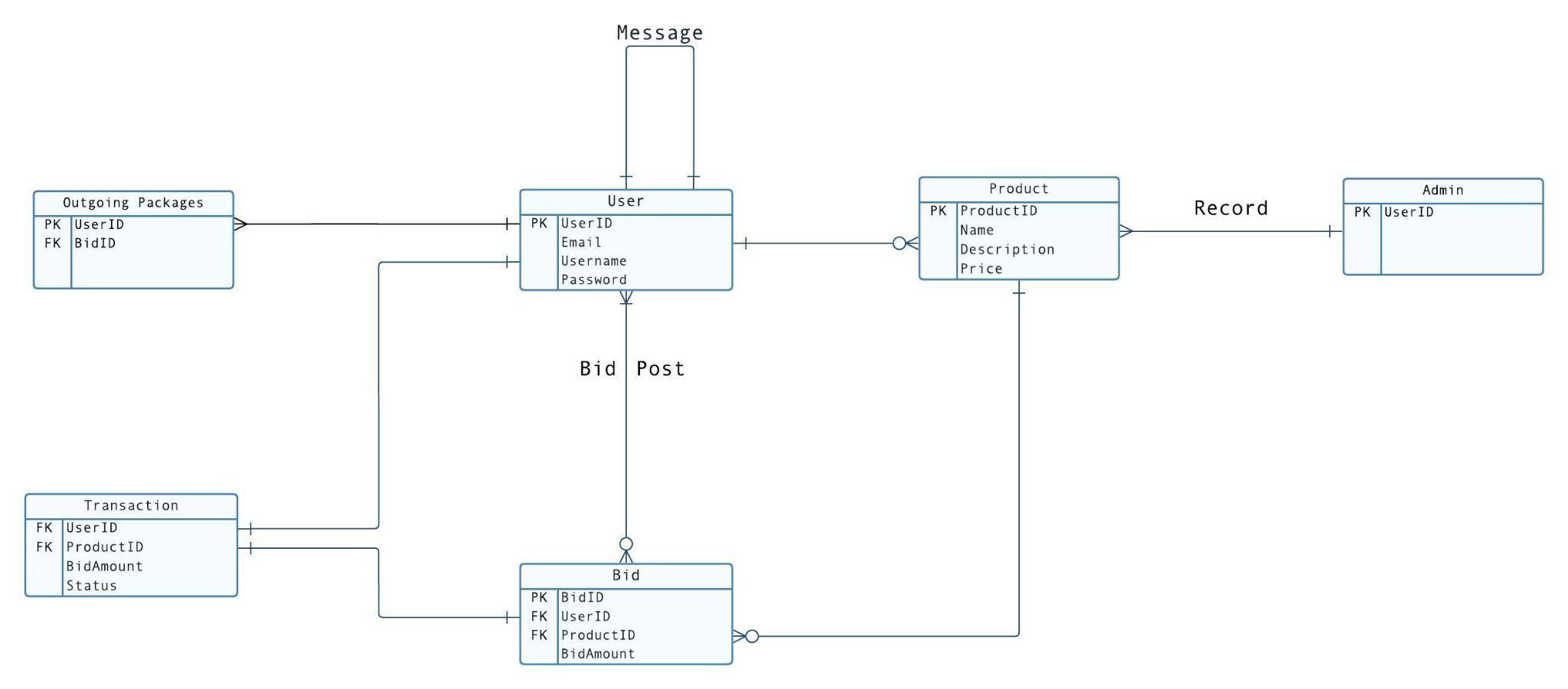
## 

## 

## 

## 4. Design

## 4.1. ER diagram



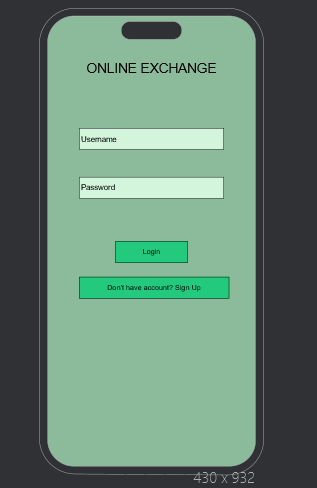
In the above ER diagram:

1. User to Product (One-to-Many):
   * Each User can have multiple Products, but each Product belongs to only one User.
   * This is represented by the 1-to-many notation, meaning one User can be associated with multiple Products, but each Product is associated with only one User.
2. Product to Transaction (One-to-Many):
   * Each Product can be part of multiple Transactions, but each Transaction involves only one Product.
   * This is represented by the many to 1 notation, meaning each Product can have multiple Transactions, but each Transaction is associated with only one Product.
3. Transaction to Invoice (One-to-One):
   * Each Transaction generates one Invoice, and each Invoice is generated from one Transaction.
   * This is represented by the 1 to 1 notation, indicating a one-to-one relationship between Transaction and Invoice.
4. User to Bid (One-to-Many):
   * Each User can create multiple Bids, but each Bid is created by only one User.
   * This is represented by the 1-to-many notation, indicating that one User can be associated with multiple Bids, but each Bid is associated with only one User.
5. Product to Message (One-to-Many):

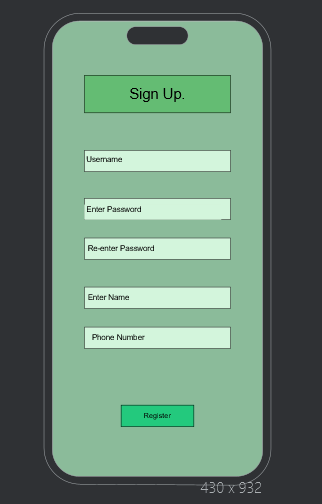
Each Product can have multiple Messages, but each Message is related to only one Product.

## 4.2. GUI

Login Screen:



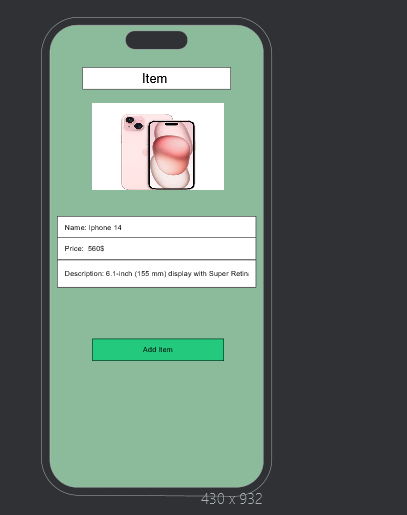
Registration Screen



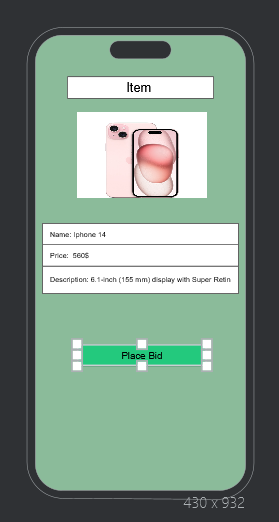
Browse Listings



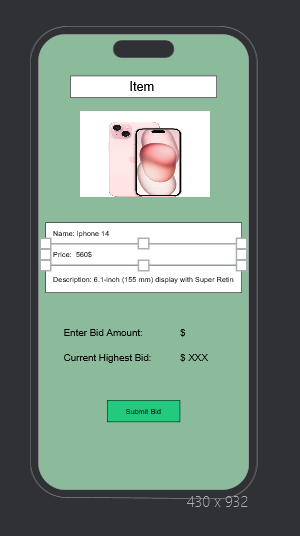
Create Listing



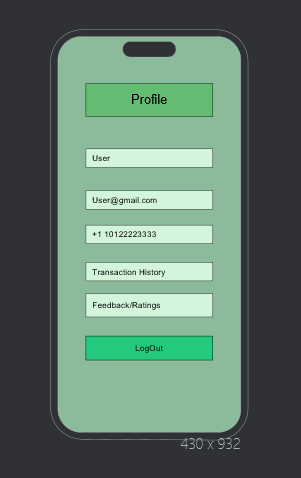
Product Details



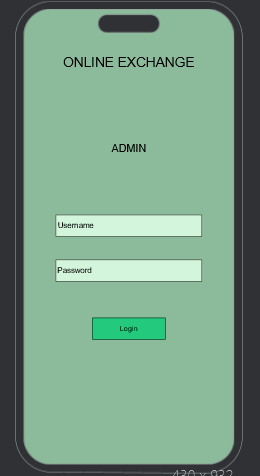
Bidding Interface



User Profile



Admin Panel

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screen shot of a phone

Description automatically generated

## 5. Analysis Models

## 5.1. Data Flow Diagram

A diagram of a product

Description automatically generated

## 5.2. Sequence Diagram

A diagram of a product

Description automatically generated

## 

## 

## 