Software Requirements Specification (SRS) Document

Cart Genie : Team 29
Team Members: Kushagra Dhingra , Hardik Kalia, Samagra Bharti, Chinmay Sharma

Brief problem statement

The project addresses the problem of tracking deliveries and returns across multiple e-commerce platforms. This problem stems from growth of online shopping and emergence of various e-commerce platforms to fulfill demands. Our application will address this problem by scraping details from text messages and provide an organized infographic.

System requirements

1. Android Version 14+
2. Technologies used:
    a. Frontend - Flutter
    b. Backend - NodeJS
    c. Database management - MongoDB.

Users profile

1. **System Administrator**

    Mode of Access: Developer Mode

    Profile: High level of proficiency with database management and administration of the whole application architecture.

2. **End User**

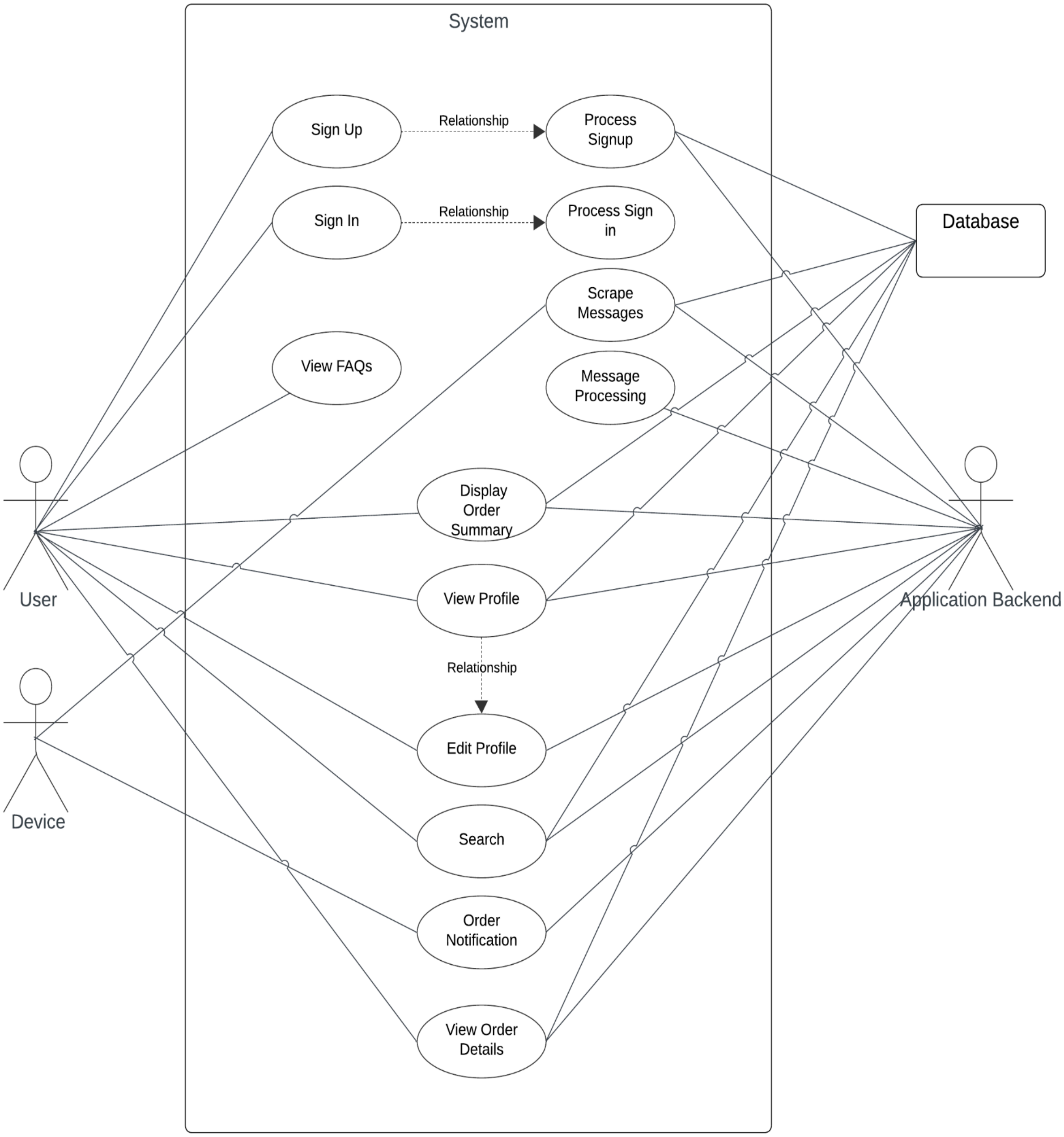
    Mode of Access: User mode

    Profile: Basic familiarity with the android OS and application software usage is assumed.

Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	Sign up	User will sign up using phone number and email id	R1
2.	Process sign up	Add the ‘new user’ information to database	R1
3.	Sign in	Existing user will enter details to sign in	R1
4.	Process sign in	Authenticate user by verifying entered information with database	R1
5.	Scrape messages	Scrape user’s messages and store relevant messages in database	R2
6.	Processing of Messages	Run an algorithm on the messages and store relevant ones to extract information	R2
7.	Display Order summary	Fetch information about orders from the database and display them	R2
8.	View detailed order information	Display details of the selected order	R2
9.	Filter relevant orders	Filter orders on the basis of multiple parameters	R2
10.	View FAQs	User can view FAQs about the app	R1
11.	View Profile	Users can view details about their profile	R1
12.	Edit Profile	Users can edit their profile	R1

Use case diagram



Use case description

Use Case Number:	UC-01
Use Case Name:	Sign Up
Overview:	User creates an account to access the application
Actors:	User, Application Backend
Pre condition:	- Internet Access.
Flow:	Main (success) Flow: 1) User opens the application and is prompted with the “Sign Up” page. 2) The User enters the requisite information - Full Name, Email address, phone number, password. 3) The User submits details by clicking a "Sign Up" button. 4) User enters OTP sent to their mobile number
	Alternate Flows: 1) If the information validation fails due to invalid input e.g - invalid phone number or invalid email address, user is redirected to the form and prompted to fill it again. 2) If the user credentials are not unique i.e the user already exists in the database, the user is shown the appropriate message (E-mail ID or phone number already exists) and is directed to the sign in page.
Post Condition:	The User's account is successfully created and linked to the provided email and phone number.

Use Case Number:	UC-02
Use Case Name:	Process Sign Up
Overview:	The system validates the information given by the User and creates a new account.
Actors:	Application Backend
Pre condition:	The User must have requested for “Sign-Up” and have entered all relevant details.
Flow:	Main (success) Flow: 1) The system validates the information provided by the User, ensuring that all required fields are filled correctly and that the email address is unique. 2) If the validation is successful, the system creates a new account for the User and 3) The User is redirected to the home page.
	Alternate Flows: Incase of any server errors or connectivity problems during the process, the system should display appropriate messages and ask for re-signup.
Post Condition:	The system creates a User account.

Use Case Number:	UC-03
Use Case Name:	Sign In
Overview:	The User logs in to an existing account.
Actors:	User, Application Backend
Pre condition:	The user has an account and active credentials.
Flow:	Main (success) Flow: 1) The User navigates to the Sign In page. 2) The User enters the email address and password in the designated fields. 3) The User clicks the “Sign In” button.
	Alternate Flows: 1) If the entered email address or password is incorrect, the system displays an error message indicating that the credentials are invalid. 2) The User is prompted to re-enter their email address and password. 3) The User also gets an option to login using their mobile number and OTP instead.
Post Condition:	Once the sign-in is successful, the User gains access to their CartGenie account

Use Case Number:	UC-04
Use Case Name:	Process Sign In
Overview:	The system validates and redirects the User to their account.
Actors:	Application Backend
Pre condition:	- The User has an existing account in the system and has requested to log into it. - Internet Access.
Flow:	Main (success) Flow: 1) The system verifies the entered credentials against the stored data. 2) If the email address and password match a valid account or the phone number and OTP are valid, the system grants access to the User's account. 3) The User is redirected to the account's default home page.
	Alternate Flows: 1) If the entered email address or password is incorrect, the system displays an error message indicating that the credentials are invalid. 2) If there is no internet connection the application tries to send the request for 1 minute and times out after that, displaying the appropriate message.
Post Condition:	The system redirects them to their account.

Use Case Number:	UC-05
Use Case Name:	Scrape Messages
Overview:	Scrape the user's messages and send relevant messages (after processing on the frontend) to the backend for further processing.
Actors:	Application Backend, Device ,
Pre condition:	The application must have the permission to access the messages and contacts of the user device.
Flow:	Main (success) Flow: 1) The system reads the messages from the user device. 2) Messages are filtered and the relevant messages are sent to backend for further processing.
	Alternate Flows: 1) If the application doesn’t have permission to access user messages, it prompts the user to grant permissions again.
Post Condition:	Message data is stored in the database.

Use Case Number:	UC-06
Use Case Name:	Processing of Messages
Overview:	Run an algorithm on the messages and store relevant ones to extract information/
Actors:	System
Pre condition:	Database has received the message
Flow:	Main (success) Flow: 1) The message is processed. 2) If it has information pertaining to orders, it is stored in the database, else deleted.
	Alternate Flows: N/A
Post Condition:	Message with order information is stored in the database and is further processed to extract and display it.

Use Case Number:	UC-07
Use Case Name:	Display Order Summary
Overview:	Fetch information about all the orders from the database and display them.
Actors:	Application Backend, User
Pre condition:	The user has visited the “Orders” section and requested to fetch a summary about orders.
Flow:	Main (success) Flow: 1) The User navigates to the "Orders" section of their account. 2) The system retrieves information about all orders from the database. 3) The platform presents the order summary to the User, typically in a structured format containing product name, delivery firm, and the status of the delivery.
	Alternate Flows: Include the post condition for each alternate flow if different from the main flow.
Post Condition:	The User can review the order details, including items purchased and shipping information.

Use Case Number:	UC-08
Use Case Name:	View detailed order information
Overview:	User clicks/taps on an order on home page to view more details.
Actors:	User, Application Backend
Pre condition:	- User is on order summary page - Internet Access.
Flow:	Main (success) Flow: 1) User taps on an order on the home page 2) The application fetches details about the order from the database 3) The application displays data about the order to the user on a separate screen
	Alternate Flows: If there is no internet connection the application tries to send the request for 1 minute and times out after that, displaying an error message.
Post Condition:	A new screen is opened with detailed order information

Use Case Number:	UC-09
Use Case Name:	Filter
Overview:	User can filter messages from on the following parameters: company name, order type, order status and dates.
Actors:	User, Application Backend
Pre condition:	The User is logged in to their account on the system. Messages are stored in the database
Flow:	Main (success) Flow: 1) User clicks on the Search tab in the dashboard which opens up the search page. 2) They type the name of the service used for the order. 3) All orders corresponding to the entered service are displayed.
	Alternate Flows:N/A
Post Condition:	All orders corresponding to the search term are displayed with a back button.

Use Case Number:	UC-10
Use Case Name:	View FAQs
Overview:	Users can view the frequently asked questions about CartGenie.
Actors:	User
Pre condition:	N/A
Flow:	Main (success) Flow: 1) The User navigates to the ‘FAQs” section. 2) The system presents the user with a list of frequently asked questions (FAQs). 3) The user clicks on a specific question to view its answer. 4) The system expands or displays the answer to the selected question below or next to the question.
	Alternate Flows: N/A
Post Condition:	The user has successfully viewed the answer to their question or found information on a topic of interest in the FAQ section. The user can continue exploring other sections of the application or take action based on the information obtained from the FAQ.

Use Case Number:	UC-11
Use Case Name:	View Profile
Overview:	User can view their profile information.
Actors:	User
Pre condition:	The User is logged in to their account on the system.
Flow:	Main (success) Flow: 1) The User navigates to the “Account” section. 2) The application displays information such as Full Name, Contact number, email address, address etc.
	Alternate Flows: N/A.
Post Condition:	The User can view stored information about their profile

Use Case Number:	UC-12
Use Case Name:	Edit Profile
Overview:	User can edit their stored profile information in the system
Actors:	User , Application Backend
Pre condition:	The User is logged into their account and has accessed their profile settings.
Flow:	Main (success) Flow: 1) The system retrieves the User's current profile information and displays it for editing. 2) The User modifies the desired fields of their profile. 3) The User reviews the changes made to the profile. 4) The User saves the changes and the system updates the profile with the edited information.
	Alternate Flows: If there is no internet connection the application tries to send the request for 1 minute and times out after that, displaying an error message.
Post Condition:	The User's profile reflects the changes made, and the updated information is visible.