Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Sign Up		
Created By:	Matthew Heng Yu Jie	Last Updated By:	Matthew Heng Yu Jie
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	User
Description:	System allows a user to create a new account using their email address, password, and personal details.
Preconditions:	1. User has a valid email address. 2. User is not already registered in the system.
Postconditions:	1. User account is created in the system. 2. User receives a confirmation email/message.
Priority:	High
Frequency of Use:	Once per user.
Flow of Events:	1. User selects "Sign Up" on the landing page. 2. User enters their email address, password, and personal details. 3. User uploads a profile picture if they choose. 4. User indicates if they own a car. 5. If the user owns a car, they enter car details. 6. User submits the registration form. 7. System validates the user details and creates a new account. 8. System sends a confirmation to the user.
Alternative Flows:	AF-S2: The user attempts to register with an email that is already in use. 1. The system displays the message “email is already in use”. 2. Return to step 2. AF-S6: The user does not fill out all required fields. 1. The user is prompted to complete them. 2. Return to Step 2.
Exceptions:	-
Includes:	Verify the account availability
Special Requirements:	System must validate the uniqueness of the email address
Assumptions:	Users will provide accurate personal details during registration.
Notes and Issues:	Security considerations for storing user passwords must be addressed.

Use Case ID:	2		
Use Case Name:	Sign In		
Created By:	Matthew Heng Yu Jie	Last Updated By:	Matthew Heng Yu Jie
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	User
Description:	System authenticates user login through email and password.
Preconditions:	User has already created an account.
Postconditions:	User gains access to their account dashboard.
Priority:	High
Frequency of Use:	Potentially daily, depending on user engagement.
Flow of Events:	1. User navigates to the Sign In page. 2. User enters their registered email address and password. 3. System verifies credentials against the database. 4. System authorises and grants access to the user.
Alternative Flows:	AF-S3: User enters incorrect credentials. 1. The system displays the message “Incorrect credentials”. 2. Return to step 2.
Exceptions:	EX-AF-1: User's account is locked after too many failed login attempts.
Includes:	Verify Login Credentials
Special Requirements:	Ensure secure transmission and handling of credentials.
Assumptions:	Users remember their login details or use a password reset function.
Notes and Issues:	Implement safeguards against brute-force attempts

Use Case ID:	3		
Use Case Name:	Validate the account availability		
Created By:	Matthew Heng Yu Jie	Last Updated By:	Matthew Heng Yu Jie
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	System
Description:	System to validate the account availability, send verification code to the registered email address, and create the user account
Preconditions:	1. User account must not already exist in the database.
Postconditions:	1. User account is successfully created in the system's database. 2. User is able to login using their registered user account
Priority:	Moderate
Frequency of Use:	As needed
Flow of Events:	1. System will validate the user account availability in the database. 2. System will create the user account in the database.
Alternative Flows:	AF-S1: System detects if the user account already existed in the database. 1. System will display an error message "Account already exists." 2. User will enter a new username. 3. User selects the register button to re-attempt registration again. 4. Return to Step 1.
Exceptions:	-
Includes:	1. Validate the account availability.
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Verify Login Credentials		
Created By:	Matthew Heng Yu Jie	Last Updated By:	Matthew Heng Yu Jie
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	System
Description:	System to validate the account availability, send verification code to the registered email address, and create the user account
Preconditions:	1. User account must not already exist in the database.
Postconditions:	1. User account is successfully created in the system's database. 2. User is able to login using their registered user account
Priority:	Moderate
Frequency of Use:	As needed
Flow of Events:	1. User tap login via username. 2. User enters their credentials in the login interface. 3. User select the login button. 4. System validates the account by checking the user's credentials with the database. 5. System authenticates the user to login successfully.
Alternative Flows:	AF-S3: System detects empty username or password fields. 1. System displays an error message "Username or password fields cannot be empty." 2. User filled up the required field(s) for username and password. 3. User selects the login button to re-attempt login again. 4. Return to Step 2. AF-S4: User enters the wrong credentials. 1. System displays an error message "Invalid username or password. Please re-enter." 2. Return to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Edit profile		
Created By:	Matthew Heng Yu Jie	Last Updated By:	Matthew Heng Yu Jie
Date Created:	12 Feb 2024	Date Last Updated:	13 Feb 2024

Actor:	User
Description:	A registered user adjusts their profile information.
Preconditions:	User is authenticated and on their profile page.
Postconditions:	User profile is updated with new information.
Priority:	Moderate
Frequency of Use:	As needed
Flow of Events:	1. User access their profile page. 2. User selects "Edit Profile." 3. System displays editable fields of the user profile. 4. User changes necessary fields and selects "Save Changes." 5. System saves updated information and reflects changes on the profile page
Alternative Flows:	AF-4: User does not save changes. 1. System retains original profile details. 2. Return to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	The user's email address should remain unique.
Assumptions:	Users want to update their profile information occasionally.
Notes and Issues:	Provide an intuitive, user-friendly interface for editing profile information.

Use Case ID:	6		
Use Case Name:	Offer Ride/Delivery		
Created By:	Matthew Heng Yu Jie	Last Updated By:	Matthew Heng Yu Jie
Date Created:	12 Feb 2024	Date Last Updated:	13 Feb 2024

Actor:	User
Description:	A registered user offers a ride or delivery service for earning money.
Preconditions:	1. The user is authenticated. 2. The user has necessary information about their route and timing. 3. The user has an added car if they offer a ride.
Postconditions:	System lists outbound rides/deliveries for passenger or requester consideration.
Priority:	High
Frequency of Use:	As often as the user wants to earn money.
Flow of Events:	1. User chooses "I Want to Earn Money." 2. User inputs ride/delivery details like start point, end point, departure time. 3. System lists out potential deliveries or passenger pickups in the route. 4. User accepts a ride/delivery request. 5. System updates the ride/delivery status.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	System should accommodate real-time changes in the user's location, schedule, and availability.
Assumptions:	Users will offer rides/deliveries with genuine willingness and capability to fulfil the services.
Notes and Issues:	System will need to ensure the safety and reliability of rides and deliveries.

Use Case ID:	7		
Use Case Name:	Request a Ride		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	12 Feb 2024	Date Last Updated:	13 Feb 2024

Actor:	User
Description:	A registered user requests a ride to their destination.
Preconditions:	User is authenticated
Postconditions:	User's ride request is registered in the system available for drivers to accept.
Priority:	
Frequency of Use:	As often as user needs a ride.
Flow of Events:	1. User chooses "I Want To Go Somewhere." 2. User inputs pick-up location and destination. 3. User selects ASAP or schedules a pick-up time. 4. System registers the ride request for drivers to accept.
Alternative Flows:	AF-1: System can't find any available drivers, after searching for a set period of time. 1. The user asks to try searching again and return to step 3. 2. The user asks to cancel searching and return to step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	User will be ready for pick up at the scheduled time.
Notes and Issues:	System will need to ensure the safety and reliability of rides.

Use Case ID:	8		
Use Case Name:	Request Delivery (I Want Something Delivered)		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	12 Feb 2024	Date Last Updated:	13 Feb 2024

Actor:	User
Description:	A registered user requests a delivery service for their items.
Preconditions:	User is authenticated.
Postconditions:	User's delivery request is registered in the system available for delivery people to accept.
Priority:	High
Frequency of Use:	As often as user needs a delivery.
Flow of Events:	1. User chooses "I Want Something Delivered." 2. User inputs pick-up location, drop-off location, and pick-up time. 3. User enters item details. 4. System registers the delivery request for delivery people to accept.
Alternative Flows:	AF-S4: System can't find any available delivery man. 1. The system display there is no any available delivery man at this moment 2. Return to Step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	User's item will be ready for pick up at the scheduled time. User’s items are not illegal in nature.
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	Chat in Real-Time		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	User
Description:	A registered user communicates with their driver or delivery person in real time through the chat feature.
Preconditions:	User is authenticated. User has booked a ride or a delivery service.
Postconditions:	-
Priority:	Moderate
Frequency of Use:	As often as user has questions or instructions for their driver or delivery person.
Flow of Events:	1. User opens their ongoing ride or delivery details. 2. User opens the chat feature. 3. User sends a message. 4. System sends the message to the respective driver or delivery person.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	System should accommodate real-time messaging.
Assumptions:	User's messages will be related to their ride or delivery.
Notes and Issues:	The system needs to ensure the privacy and security of messages.

Use Case ID:	10		
Use Case Name:	Top Up Wallet		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	User
Description:	User increases their wallet balance in the app
Preconditions:	User is authenticated User is on their wallet page.
Postconditions:	User's wallet balance is increased.
Priority:	Moderate
Frequency of Use:	As needed.
Flow of Events:	1. User chooses to top up the wallet. 2. User selects an amount to add to the wallet. 3. System adds the amount to the wallet balance.
Alternative Flows:	An error occurs during the top-up process.
Exceptions:	EX1: An error occurs during the top-up process. 1. The system displays the reason why the error occurred such as “Insufficient balance in bank of origin”.
Includes:	-
Special Requirements:	System should accurately add the specified amount to the user’s wallet.
Assumptions:	Users will want to periodically top up their wallet balance.
Notes and Issues:	The system must handle potential errors during the top-up process.

Use Case ID:	11		
Use Case Name:	Transfer out of wallet		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	12 Feb 2024	Date Last Updated:	13 Feb 2024

Actor:	User
Description:	The user transfers the money in the app wallet to his/her personal bank account.
Preconditions:	1. User is authenticated. 2. User has sufficient wallet balance to make the transfer.
Postconditions:	1. User's wallet balance is decreased by the transfer amount. 2. The recipient account is increased by the transfer amount.
Priority:	Moderate
Frequency of Use:	As needed.
Flow of Events:	1. User chooses to transfer money from the wallet. 2. User enters the recipient account details and transfer amount. 3. System checks if the user has sufficient wallet balance. 4. System transfers money from the wallet to the recipient account.
Alternative Flows:	AF-S3: The user does not have sufficient balance. 1. The system displays “Insufficient balance”. 2. Return to step 1.
Exceptions:	-
Includes:	-
Special Requirements:	System should accurately deduct the specified amount from the user's wallet.
Assumptions:	Users will make payments or purchases using the balance in their wallet.
Notes and Issues:	The system must handle potential errors during the redemption process.

Use Case ID:	12		
Use Case Name:	Provide Rating		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	User
Description:	Registered user provides a rating to the driver after a ride or a delivery session.
Preconditions:	1. The user is authenticated. 2. The user has completed a ride or received a delivery.
Postconditions:	1. The driver's or delivery person's rating is updated in the system.
Priority:	Low
Frequency of Use:	After each completed ride or delivery session.
Flow of Events:	1. The user is prompted to rate their experience. 2. The user submits a rating and optional feedback. 3. The system updates the driver's or delivery person's overall rating.
Alternative Flows:	-
Exceptions:	EX1:User closes the program before leaving rating. 1. Allow the user to not leave a rating. 2. Return to main page.
Includes:	-
Special Requirements:	System should accurately calculate and store ratings for each driver or delivery person.
Assumptions:	The user will provide honest and fair feedback.
Notes and Issues:	The system needs to handle situations where users do not want to leave a rating during calculation of ratings.

Use Case ID:	13		
Use Case Name:	Matching algorithm		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	12 Feb 2024	Date Last Updated:	12 Feb 2024

Actor:	System
Description:	System matches between requester and driver/delivery man based on locations, timing, and vehicle capacity.
Preconditions:	User is authenticated. User has inputted their current location and desired destination.
Postconditions:	System shows results of whether a match is successful or not.
Priority:	High
Frequency of Use:	As needed.
Flow of Events:	1. User inputs request for a ride or delivery in the app. 2. System sources for available drivers or deliverymen in the area. 3. System utilises a matching algorithm to pair the user with the most optimal driver or delivery man based on route, timing, and vehicle capacity. 4. System sends out a message to the user indicating a successful match or that no matches were found.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	1. Matching algorithms should take into account real-time traffic data for estimated travel times. 2. System should have a backup plan for when there's a case without any match.
Assumptions:	Drivers and delivery men keep their availability and location information up-to-date.
Notes and Issues:	1. Algorithm development and testing can be complex and time-consuming. 2. Consideration must be given to user privacy and data protection in relation to location data and other personal information.

Use Case ID:	14		
Use Case Name:	Manage Wallet Balance		
Created By:	Tan Jia Hao Isaac	Last Updated By:	Tan Jia Hao Isaac
Date Created:	13 Feb 2024	Date Last Updated:	13 Feb 2024

Actor:	User
Description:	For user interactions with their wallet.
Preconditions:	1. User is authenticated. 2. User is on their wallet page.
Postconditions:	1. User manages their wallet balance according to their needs.
Priority:	High
Frequency of Use:	As needed.
Flow of Events:	1. User chooses to view their wallet balance. 2. If needed, the user chooses to top up their wallet. 3. If needed, the user chooses to transfer money out of the wallet. 4. System reflects changes in wallet balance.
Alternative Flows:	Failure in the top-up or redeem processes.
Exceptions:	-
Includes:	Top-up wallet, Transfer out of wallet
Special Requirements:	System should accurately reflect all transactions made by the user.
Assumptions:	Users will want to periodically manage their wallet balance.
Notes and Issues:	The system must handle potential errors during the top-up and redeem processes effectively.