

LEGACY SYSTEM: HAILO TAXI HAILING APP

SOFTWARE ENGINEERING ASSIGNMENT 1

April 5, 2024

Omphemetse Senna

University of Pretoria

1. INTRODUCTION

1.1 BACKGROUND INFORMATION

Hailo was a British tech-taxi hailing business[2][5] that connected taxi drivers and customers(passengers) through an Android and iOS mobile app[6], providing safe and convenient transportation[1]. Founded in late 2010, it debuted in London in 2011 and then moved to North America[3] by the end of 2012. The software enabled real-time tracking, secure payment processes, and wheelchair-accessible features.

Hailo began as a taxi booking service that brought together customers with drivers in urban areas. However, by the end of 2014, it had ceased operations in North America due to heavy competition and a failure to create a strong relationship with New York taxi drivers. Other competitors, like Uber, dominated in more expensive cities and towns, while Hailo was unable to adjust to the market, eventually leading to its collapse.

Hailo revolutionized[4] transportation by providing real-time tracking, safe payment methods, and wheelchair accessibility[2]. The Hailo mobile application was popular in the transportation industry, notably among taxi drivers who supplied transportation to consumers.

1.2 PROBLEM STATEMENT

During its existence, the Hailo app encountered challenges including competition from other transportation companies, difficulty adapting to national market needs and regulations, and dealing with legal issues such as lawsuits. These difficulties restricted the mobile application's expansion and durability in the sector of transportation, emphasising the importance of innovation, careful planning, and adherence to legislation for market success.

1.3 DEFINITION OF SCOPE

This project aims to improve the Hailo app by adding features such as a cloud database, machine learning, geolocation services, social media platforms, and in-app messaging. Cloud databases and machine learning will make it possible for continuous updates, data analysis for performance monitoring, and recommendations on where to focus and what to improve in order to advance further in the market. Social media will be used for marketing and advertising the application and the geolocation services that enable precise pick-up and drop-off locations, real-time traffic information for informed decision-making, and messaging within the app

for customer-driver contact.

2. USER CHARACTERISTICS

The intended users for the Hailo app are diversified, including customers, drivers, administrators, data scientists(analytics):

- **Customers:**

- **Education:** Customers are assumed to have smartphone expertise and technical skills because the mobile application is designed to be simple to use and accessible to a wide range of users. The customer would use the system to request services (rides), follow driver locations, and make payments.
- **Age:** Customers must be at least 12 years old, unless a parent requests services on their behalf. In such circumstances, the parent should show their registered ID to the driver.
- **Physicality:** The Hailo app should be designed to be accessible and user-friendly for all, including people with disabilities. This includes wheelchair options, audio descriptions, adaptable text sizes, and voice command capabilities, which ensure that every customer can travel simply, communicate with drivers, and navigate their journeys. Prioritising inclusion in app design can help build a more accessible transport service for individuals with disabilities.
- **Gender:** Any gender can request services.

- **Drivers:**

- **Education:** Drivers must have an educational foundation that includes knowledge of geography and basic finance, as well as a basic understanding of mobile devices and navigation systems.
- **Experience:** Drivers should have at least three years of driving experience, as they will be using the system to handle trip requests, find pick-up and drop-off locations, manage their schedules, and track payments, particularly cash.
- **Age:** expected should be above 18 and below 60.
- **Physicality:** The Hailo app should provide accessibility features such as voice commands, larger text sizes, and colour contrast adjustments to ensure that all drivers, regardless of impairment, can efficiently use the platform for tasks such as service requests, navigation, and calendar management.

- **Gender:** Any gender can request services.

- **Admin:**

- **Education:** Administrators are expected to have higher technical expertise, as they would use technological tools to administer accounts for users and monitor system operations. To increase efficiency, monitor performance data and apply updates and enhancements.
- **Age:** should be below 65.
- **Physicality:** Admin with disability are allowed to apply.
- **Gender:** Both Male and Females are allowed.

- **Data Scientists/Analytics:**

- **Education:** Degrees in data science, analytics, or similar fields are among the educational prerequisites.
- **Experience or expertise:** Proficiency in data visualisation, machine learning, and database management are among the expertise sets required to analyse large amounts of data, identify trends, develop demand forecasting models, and increase productivity in the on-demand transportation industry, promoting innovation and business growth.
- **Age:** should be below 65 years.
- **Gender:** Any gender can request services.

3. USER STORY AND FUNCTIONAL REQUIREMENTS:

U1 - Use Case Name: Sign up

Actor(s): Customer (Primary) and Driver(Primary)

User Story:

The on-boarding process allows the riders to be able to utilise the system's functionalities. During this initial phase, user identities are created and are unique for each new user, and security protocols are put in place to prioritise safety.

In the system, there are two types of sign-ups: for customers (U1.1) and for drivers (U1.2).

- **For U1.1,** customers have the option to set up an account by entering their personal information. This allows them to request services (ride) and ensures data accuracy.

- **For U1.2**, drivers have the option to register by submitting their personal information, vehicle details, and licence information. The details entered during registration will be stored in a cloud-based database.

Input data:

1. Username.
2. Surname.
3. First name.
4. Gender.
5. Email Address.
6. Cell Phone number.
7. Password.
8. Date of birth

Input data - Extension for Drivers:

1. Licence Information.
2. Vehicle Information.
3. Address.
4. Documents(Licence documents, Proof of address).

Functional Requirements:

- **FR1.1:** If the username, email address, and phone number already exist in the database, the system should not register the new user account.
- **FR1.2:** The system should be able to create a new account for the user and validate the provided information.
- **FR1.3:** The system should allow the users to register an account on the system using their phone number, email address, or Google account.
- **FR1.4:** The system should be able to send an account confirmation link or code on the email or on text message.

- **FR1.5:** The system should be able to input user passwords and other personal information and be able to confirm the password entered.
- **FR1.6:** In order to guarantee account security and authenticate user credentials, the system must be able to incorporate a verification process.
- **FR1.7:** The system should have an option to enter payment details for easy client transactions.
- **FR1.8:** The system should be able to upload the documents from the user (Driver).

Extensions:

- **FR1.9:** If the customer enters invalid information, the system should prompts for correction.
- **FR1.10:** If the verification code or link on the email or cell phone message is not sent, the user can resend to receive one.

Pre-Condition:

- The user must have access to the internet before interacting with the system.
- The user must have or know their cellphone number, email, username to use, and date of birth to register an account.

Post- Condition:

- The user have successfully created the new account on the system up.
- The user received an email (or optionally text message) for confirmation of the email account.
- The user's account is verified, and they can start using the system.
- The user's(driver) account is verified and the background checks have been made.

U2 - Use Case Name: Sign in

Actor(s): customer (Primary) and Driver (Primary)

User Story: There are two sign-ins in the system (U3 and U3):

Registered users (drivers and customers) can log in to their accounts using their credentials.

- Allows the customer-user to access their account, which can be used by the customer to login to the app and request services such as booking a ride, viewing history, rating drivers, and making payments.

- Allows the driver-user to access their account, which can be used by the driver to login to the app and provide service. Drivers can accept or reject ride requests from customers and can also view history, ratings, and update their availability.

Input data:

1. Email Address/Username/cellphone number/Google account
2. Password

Functional Requirements:

- **FR2.1:** The system must have "Remember Me" functionality to facilitate faster logins for returning and frequent users.
- **FR2.2:** An instant notification system for important updates on the system.
- **FR2.3:** In a situation where login credentials are forgotten, the system needs to include a link for redirecting the users to a page of recover passwords.
- **FR2.4:** Secure authentication techniques, including two-factor authentication or password-based login, must be implemented in the system.
- **FR2.5:** Verification process to verify user credentials and ensure compliance with regulations.
- **FR2.6:** The system must allow the user to have access to their service history upon successful login.
- **FR2.7:** The system must have the ability to set availability status (online or offline) for receiving ride requests from the user (driver).
- **FR2.8:** The system prompts the user to enter their email address, cellphone number or user name and passwords.
- **FR2.9:** Logging in date and time is recorded upon successful login.

Extensions:

- **FR2.10:** If the user enters incorrect credentials, the system prompts for correction message to the user to re-enter details.
- **FR2.11:** For faster access, a Google account login option must be included in the system.

- **FR2.12:** reset password when the user forgot it.

Pre-Condition:

- The user must have access to the internet before interacting with the system.

Post-Condition:

- The user have successfully logged in on the system.
- The security measures are in place to secure user's login information upon successful login.
- The user (driver) can access the driver's portal for ride requests.
- The user (customer) can access the customer's portal for service requests.

U3 - Use Case Name: Request password

Actor(s): Driver (Primary), customer (Primary)

User Story:

Registered users (customers and drivers) can reset and update their passwords if they forget them or want to secure their accounts. After resetting the password, the user can gain access to the system.

Input data:

1. Email Address or cellphone number.
2. Identity Number.

Functional Requirements:

FR3.1: The user launches the system and chooses the Forgot Password option.

FR3.2: Accepts email addresses, cell phone numbers, and the user's identification number as inputs.

FR3.3: Authenticates the input credentials to verify if the account exists in the database and matches.

FR3.4: The system has to forward a password reset link to the user's email address or phone number.

FR3.5: Sends the link to the user, and when clicked, the system must route the user to the page where they can input their new password.

FR3.6: Validate the new password and submit it.

FR3.7: A successful password reset requires the system to change the password in the database and notify the user by email or cell phone message.

Extensions:

FR3.8: If the user enters a password that does not meet the system's requirements, the system shows a correction message and asks the user to reenter the password.

Pre-Condition:

- The user must have downloaded and installed the system and additionally possess an active connection to the internet.
- The user forgot the system password.

Post-Condition:

- After successfully resetting their password, the user can log in to the system using their new credentials.
- Users can view the app's services and history.
- user can request rides through the driver site.
- User login information is secure.

U4 - Use Case Name: Request Service (Ride)

Actor(s): customer (Primary), Bank (secondary)

User Story:

User (customers) are given the opportunity to enter the system and request for services (rides). Once a user, which is the customer logs in they can request services, have various payment options available. They are able to see their pick up spot, on a map based on geo-location. After the service is completed users will receive a prompt to rate the driver. Will be sent a payment receipt after service.

The bank is a third-party that accepts payments made by the customer such as EFT, vouchers and using bank card.

Input data:

1. Pick-up location.
2. Drop-off location.

Functional Requirements:

- **F4.1:** The system must allow the user to specify the pick-up and drop-off locations.

- **F4.2:** The system estimates service rates based on location.
- **F4.3:** The system must be able to locate the precise locations specified by the user.
- **F4.4:** The system must indicate available nearby drivers, as well as the drivers' expected arrival times to the customer and the drop-off site.
- **F4.5:** The system must allow customers to select their preferred car and display the amount of service.
- **F4.6:** The system must allow the customer to select the mode of payment.
- **F4.7:** The system displays the details of the driver who accepted the customer's service request.
- **F4.8:** In-app messaging and calls allow customers and users to communicate directly.
- **F4.9:** In-app messaging and in-app call feature for communication between the customer and the user.
- **F4.10:** The system must alert the user (customer) when the driver arrives at the pickup and drop-off locations.
- **F4.11:** The system must allow users to rate the driver and provide tips.

Extensions:

- **F4.12:** If the system cannot find the entered location, it should display a possible location and allow the consumer to enter the closest place to the drop-off location.
- **F4.13:** A success notification appears when the customer gets into the driver's car and is ready to depart the starting point, as well as after the ride is concluded once the drop-off location has been reached.
- **F4.14.** The system can display a notification for rating drivers.
- **F4.15.** The system allows user to change the driver drivers.
- **F4.15.** The system allows user to cancel service requested..

Pre-Condition:

- The user must have access to the internet before interacting with the system.

Post-Condition:

- The user successfully requests a ride through the system, and the driver accepts the request.

- The user can track the driver's location to the pick-up site, the ride's progress, and the drop-off location.
- The customer can rate the driver's service and leave a tip.
- The user made payments.

U5 - Use Case Name: Accept and Provide Service (Ride)

Actor(s): Driver (Primary), Google(secondary)

User Story:

Permits the driver to access the system, accept customer requests, and deliver services (riding). Geo-location can optimise routes by displaying traffic and construction, as well as providing quickest pathways to client destinations (pick-up and drop-off locations).

Google will be used as one of the route optimization tool if there is no built in route-optimization.

Functional Requirements:

- **FR5.1:** The user has successfully accessed the system, and the system permits the driver to navigate the service portal or page.
- **FR5.2:** The system allows the driver to identify and update their availability (whether online or offline).
- **FR5.3:** The system must allow the driver to get the customer's trip request notification and report, which includes the amount, pick-up location, projected arrival time to the customer and drop-off destination, and the method of payment.
- **FR5.4:** The system must allow the driver to accept or decline the customer's ride request.
- **FR5.5:** Allows the user to navigate to the pickup location of the customer's geo-location navigator.
- **FR5.6:** The system must allow the driver to alert the customer through the system when they arrive at the pickup location.
- **FR5.7:** When the user selects cash as a payment method, the system must finish the rides in the mobile app and prompt a message informing the driver if he or she took the payment.

Extensions:

- **FR5.8:** When the driver arrives at the drop-off location and completes the transaction, a successful message appears.

- **FR5.9:** The system allows the user to cancel service.

Pre-Condition:

- The driver has logged into the system and is ready to give services to consumers.
- In order to use the system, the user must have a steady network connection.

Post-Condition:

- The driver effectively provides a ride to the consumer through the application.
- The driver refreshes their state so that the system can detect and assign customers.
- The driver either earned rates and tips for the service or did not.

U6 - Use Case Name: Source and Extract Data

Actor(s): Data Scientist/Analyst (Primary)

User Story:

enables the user to acquire raw data on customers and drivers, locations, politics, markets, and competitors from many sources and save it in a database for analysis to help enhance the business.

Functional Requirements:

- **FR6.1:** The user integrates with a database to collect user information.
- **FR6.2:** The system enables data extraction by collecting structured and unstructured data from various data sources.
- **FR6.3** The system allows data to be categorised into categories and saved in a database.
- **FR6.4** The system supports many data pipelines for effective data collection.
- **FR6.5** Allows data encryption and secure storage processes.

Extensions:

- **FR6.6.** Set pipelines or external data sources to enhance the analysis.
- **FR6.7.** Message of success or failure to store data.

Pre-Condition:

- The user possess access and permission to relevant data sources.
- The user have access to appropriate data repositories for storing data.

Post-Condition:

- The user has successfully sourced the data.
- The user successfully stored data across many repositories.

U7 - Use Case Name: Analyse Data

Actor(s): Data Scientist/Analyst (Primary), Stakeholder(secondary)

User Story:

The user evaluates data from customers, drivers, and other sources to improve business operations. It also entails cleaning raw data, analysing customer behaviours, anticipating demand, optimising pricing, and implementing various machine learning models to improve operational efficiency and other aspects of the business that require improvement.

Functional Requirements:

- **FR7.1:** The system provides access of raw data stored from the database.
- **FR7.2:** Clean data is stored in the database and ready for analysis.
- **FR7.3:** The system allows data visualisation and creates dashboards and reports.
- **FR7.4:** The system supports statistical analysis and machine learning methods for predictive modelling and data forecasting.
- **FR7.5:** Improve operational efficiency.
- **FR7.6:** The data scientist employs several models to forecast demand for services at various locations and periods.
- **FR7.7:** Optimise pricing techniques.

Extensions:

- **FR7.8:** The user engage with external partners to provide data and insights.

Pre-Condition:

- The user has access to relevant data repositories, which hold data.
- The user has access to analytical tools.

Post-Condition:

- The results of the analysis are collected, and the user creates reports.
- Businesses can use insights to make better decisions.
- The user reported their findings to the stakeholders.
- The user enhanced the client experience while also advancing the business.

U8 - Use Case Name: MANAGE USERS

Actor(s): Admin (Primary) and Stakeholder (Secondary)

Summary Description:

The user (admin) can manage active accounts, user profiles, and procedures on the system. It also allows the administrator to connect with drivers and customers who are experiencing system troubles. The user can update, add, and delete customer and driver profiles, as well as compile and deliver reports to stakeholders.

Functional Requirements:

- **FR8.1:** permits the user to verify a driver's information and approve their application.
- **FR8.2:** The system enables the user to manage customer and driver complaints and feedback.
- **FR8.3** User account administration (activating and deactivating accounts).
- **FR8.4:** The system enables the user to communicate with clients and drivers as needed and provide support.
- **FR8.5:** Dashboards for transaction and account reporting.
- **FR8.6** Functionality for monitoring and analysing user actions.
- **FR8.7** Generate profile reports.

Extensions:

- **FR8.8.** The system must allow the user to delete, add, and modify users' profiles.
- **FR8.9.** Resolve any issues that arise.

Pre-Condition:

- The user has the required access.
- The user maintains communication channels with both customers and drivers.
- The user understands the company's policies and processes.

Post-Condition:

- User (Customer and Driver) profiles are updated, added, or deleted, and any issues are addressed.
- Drivers have been verified and approved on the site.
- Reports are created or generated.

U9 - Use Case Name: MAINTAIN APPLICATION

Actor(s): Admin (Primary) and Stakeholder (Secondary)

Summary Description:

Users can execute maintenance assignments to maintain optimal system performance. After making changes to the system, the user can update it, add new features, shut the system off when necessary, measure progress, and receive feedback from customers and drivers. Following these steps, the user can create a report and communicate it to stakeholders.

Functional Requirements

- **FR9.1:** Based on input from users, the administrator adds new features.
- **FR9.2:** Update and repair software, servers, configurations, and databases.
- **FR9.3:** Procedures for disaster recovery and backup to ensure data continuity.
- **FR9.4:** For insight, create dashboards and trends.

Extensions:

- **FR9.5:** The administrator adds new features based on user feedback.
- **FR9.6:** Stop application when required.
- **FR9.7:** Track the system's progress after configuration.
- **FR9.8:** Gets feedback from users about the system's efficiency.

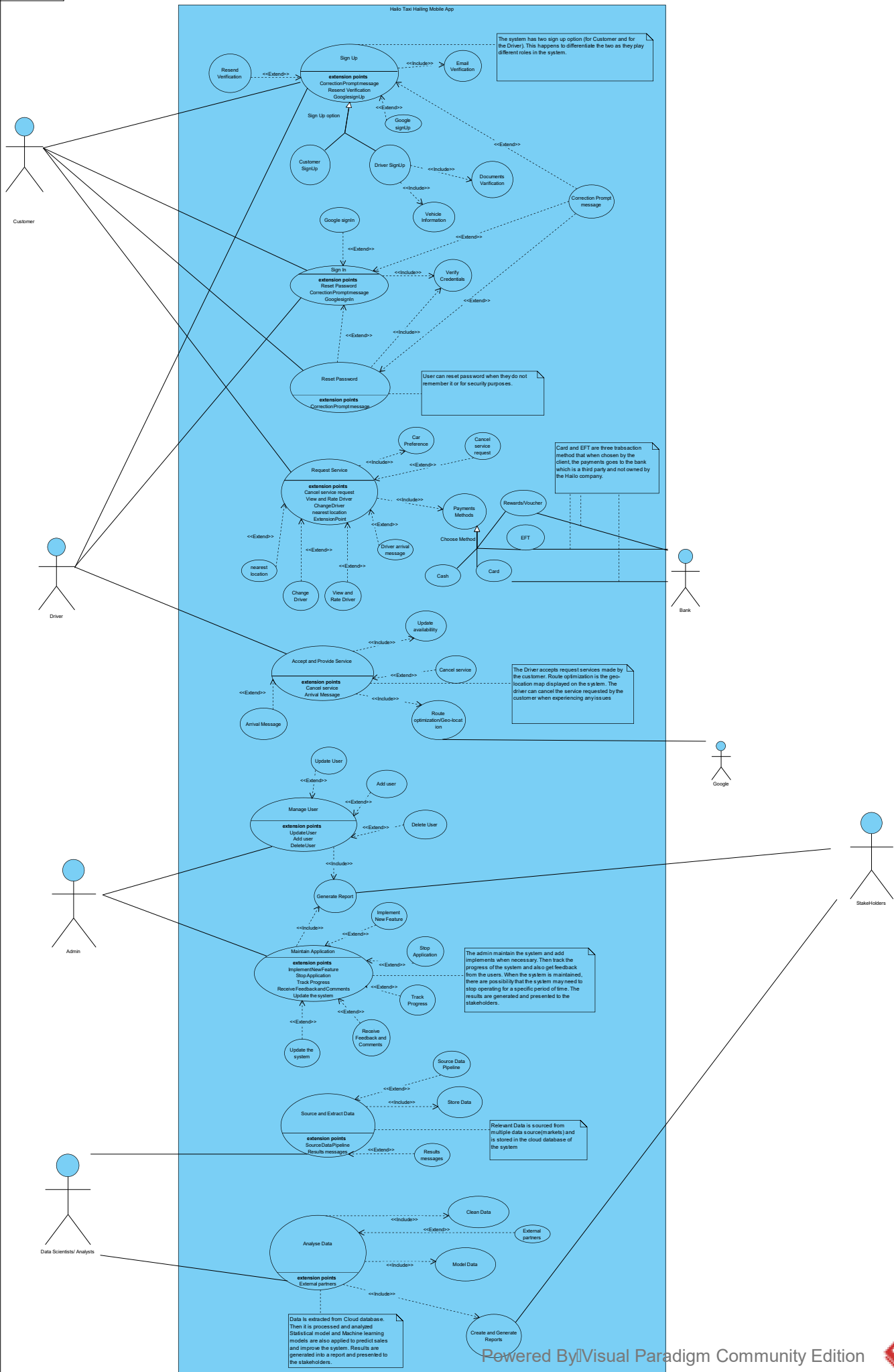
- **FR9.9:** Updates the system.

Pre-Condition:

- The user has the required access to the system.
- Knowledge of the company policies and procedures.

Post-Condition:

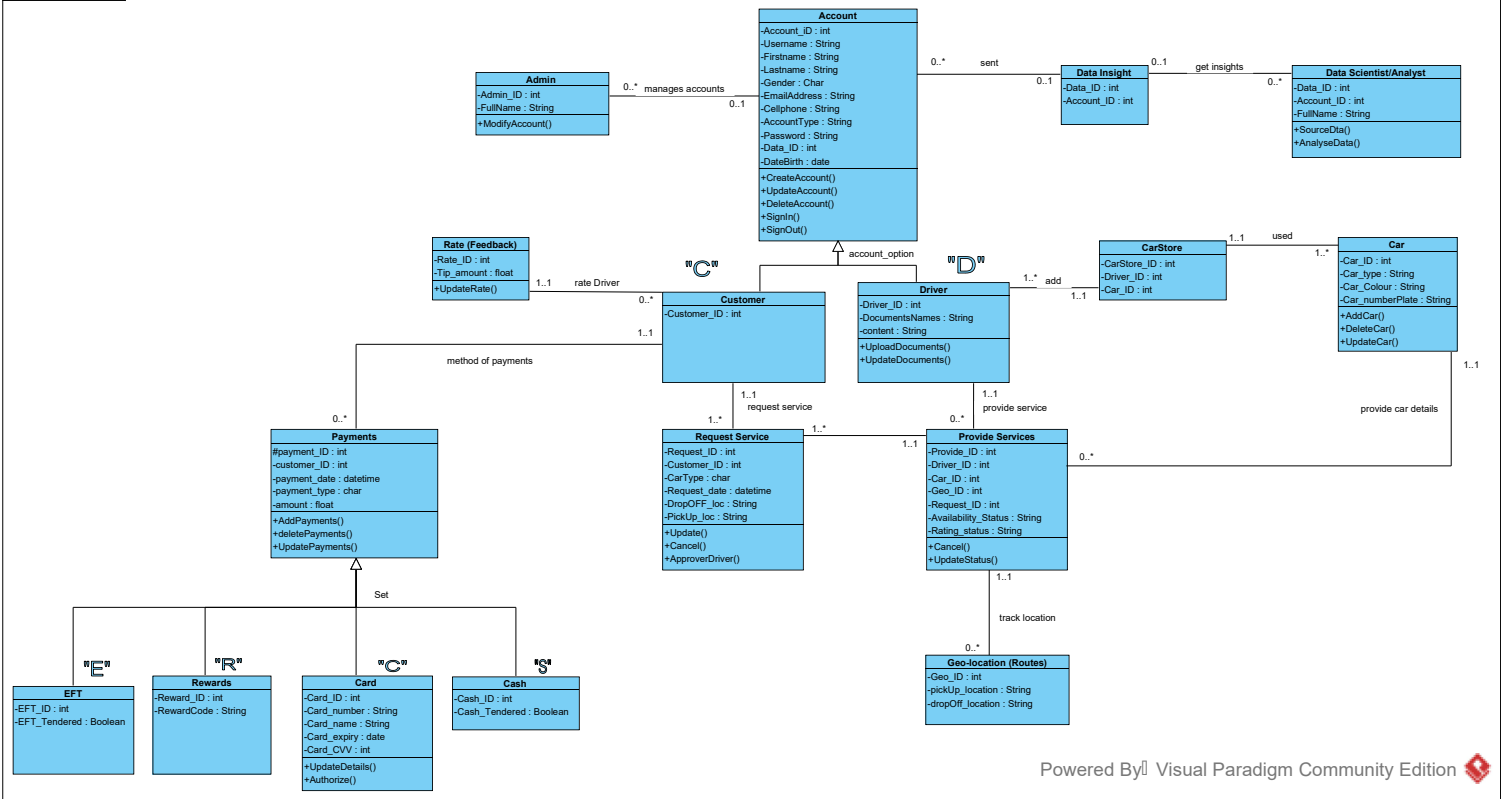
- The system is updated, and any issues are resolved.
- The system is completed successfully, and reports are generated.
- The system is back to its fully functioning state.
- Drivers are updated and notified about the resuming of the application.



5. TRACEABILITY MATRIX

Functional Requirements	Requirements Rates	U1	U2	U3	U4	U5	U6	U7	U8	U9
Totals:		9	5	4	3	3	3	3	3	3
FR1.3	3	X		X						
FR1.5	3	X								
FR1.6	5	X								
FR2.7	4		X		X					
FR2.9	2	X	X							
FR4.1	4				X	X				
FR6.3	4						X	X		
FR7.3	4							X	X	X
CR.1	4	X	X	X	X	X				
CR.2	4	X	X	X						
CR.3	4	X					X			
CR.6	4	X					X	X	X	X
CR.8	4	X	X	X		X			X	X

UML CLASS DIAGRAM



4. HIGH-LEVEL NON-FUNCTIONAL REQUIREMENTS:

PIECES CATEGORIES (RE- QUIREMENTS)	DESCRIPTION
CR1.Performance	The application should respond quickly to administrative activities. When connecting with the database through APIs for payments, sign-ups, sign-ins, or password recovery, the application should listen and respond quickly.
CR2.Security	Validations should occur during sign-up, sign-in, and password recovery. Sensitive data should be preserved and encrypted before being saved to a database or other data repository. There should be access controls in place, as well as regular security audits. When unknown situations occur, the system should be capable of responding promptly.
CR3.Scalability	The programme should be capable of handling vast quantities of data going through databases, listening to data from the database, updating data, and accommodating an increasing number of users, service requests, and accepting service requests.
CR4.Reliability	The system should be reliable and stable, with minimal downtime and no data loss. The system should shut down less frequently, and services should always be available at any time. There should be limited delays when requesting services and no incorrect payments.
CR5.Compatibility	The programme should be compatible with a variety of mobile operating systems, particularly for IOS and Android.
CR6.Compliance	To ensure data protection, the system ought to adhere to the legislation of the country (location) where the services are provided, as well as data privacy regulations.
CR7.User Interface	The system should be user-friendly and simple to use, with typefaces that may be modified for better reading. The system should be simple to use, available across several devices, and include audio.
CR8.Error Handling	The system should have the capacity to handle unexpected errors and record them so that maintainers can locate where the error occurred while minimising hours of downtime.
CR9.Maintainability	The code should be fully documented and written to allow adaptation to subsequent updates and errors, as well as easy navigation. There should be system documentation that serves as a roadmap for changes and maintenance.

PIECES CATEGORIES (RE- QUIREMENTS)	DESCRIPTION
CR10.Accessibility	The system should be accessible to all users with impairments by allowing them to modify the system's colours to assist their eyesight and, if possible, resize the text.
CR11.Localization	The system should use English as a general language, as well as other spoken languages in the area where it will operate. The system should accept the money of the location in which it operates.

Bibliography

- [1] K. Dunken. A mobile app that makes hailing a taxi easier and more efficient. <https://www.entrepreneur.com/science-technology/a-mobile-app-that-makes-hailing-a-taxi-easier-and-more/226684>, 2013. Accessed: Apr. 04, 2024.
- [2] Hailo. "taxi heaven for london's black cab drivers". <https://web.archive.org/web/20120104040211/http://hailocab.com/press-releases/taxi-heaven-for-londons-black-cab-drivers/>, 2011. Accessed: Apr. 04, 2024.
- [3] J. Kasperkevic. "hailo: the taxi app that gets people off the streets". <https://www.theguardian.com/business/2013/dec/15/hailo-taxi-hailing-app-global-growth>, 2017. Accessed: Apr. 04, 2024.
- [4] K. Meghang. "hailo confirms its 30m round from richard branson, union square ventures". <https://venturebeat.com/entrepreneur/hailo-funding-nyc/>, 2013. Accessed: Apr. 04, 2024.
- [5] P. Pessok. "perh(apps) this is the answer". https://web.archive.org/web/20111013063423/http://hailocab.com/docs/Hailo_Taxi_June2011.pdf, 2011. Accessed: Apr. 04, 2024.
- [6] M. Warman. "hailo android app review". <https://www.telegraph.co.uk/technology/mobile-app-reviews/9041116/Hailo-Android-app-review.html>, 2012. Accessed: Apr. 04, 2024.