

Swinburne University of Technology Sarawak Campus

School of Information and Communications Technology

ASSIGNMENT AND PROJECT COVER SHEET

Subject Code: SWE30003

Unit Title: Software Architectures and Design

Assignment number and title: 2, Object Design Project Group No.: Group 21

To be completed as this is a group assignment

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number	Name	Signature
102777306	Kenneth Yang Sheng KUAN	
102777759	Alvin Yi Tung TAN	
102778574	Kylie Sze Ting LAU	
104384397	Joey Yi Xin THIEN	

Marker's comments:

Total Mark: _____

Extension certification:

This assignment has been given an extension and is now due on 9 May 2025 _____

Signature of Convener: _____

Executive Summary

To enhance and modernize the ticketing experience for users, Kuching's Autonomous Rapid Transit (ART) system aims to expand beyond its current physical on-site ticketing operations by developing both a mobile application and a website. This report presents the design of a comprehensive ART online ticketing system, allowing users to conveniently purchase ART tickets without needing to visit a physical station.

The system is designed to enhance both user and admin experience by facilitating seamless transactions, effective reward management and streamlined merchandise stock control. For users, the system provides intuitive functionality for account top-ups, viewing accumulated reward points and redeeming those points for promotional vouchers or merchandise. These processes are supported by clear workflows that ensure both successful and unsuccessful transactions are handled gracefully.

For the admin perspective, the system empowers staff to efficiently manage merchandise inventory by monitoring stock levels, identifying items that fall below a set threshold, adjusting quantities and removing out-of-stock products as needed. To support system clarity and maintainability, UML sequence diagrams have been developed for twelve core processes. These diagrams visually represent the interactions among system components and serve as a foundation for a scalable, user-centric design.

The report also includes a detailed analysis of the background problems that prompted this initiative and outlines the structure and features of the proposed system. Core components covered include candidate class specifications, design quality and design patterns. Additionally, the bootstrap process explains, detailing how users interact with the system from account creation to ticket purchase. A verification section is included to validate the system design and ensure that all functional requirements are met.

Overall, the proposed solution aims to improve accessibility, convenience, and operational efficiency for the ART system by leveraging modern web and mobile technologies.

Table of Contents

Executive Summary	1
1.0 Introduction	5
1.1 Outlook of Solution	5
1.2 Trade-offs and Object Design.....	6
1.2.1 Consistent User Interface	6
1.2.2 Efficient Route Navigation.....	6
1.2.3 Cross-Platform Compatibility	6
1.2.4 Accessibility and Inclusivity.....	6
1.2.5 Object Model Simplification.....	7
1.3 Documentation and Guidelines for Interface	7
1.4 Definitions, Acronyms, and Abbreviation	7
2.0 Problem Analysis.....	8
2.1 Assumptions	9
2.2 Simplifications.....	11
2.3 Design Justification	12
2.4 Discarded Class List	13
3.0 Candidate classes.....	15
3.1 Candidate Class List	15
3.2 UML Diagram	16
3.3 CRC Cards	22
3.3.1 Account.....	22
3.3.2 Top-up.....	23
3.3.3 Point	23
3.3.4 Feedback	24
3.3.5 Sale	24
3.3.6 Line of Sale	25
3.3.7 Promotion.....	25
3.3.8 Admin.....	26
3.3.9 Merchandise.....	26
3.3.10 Trip.....	27
3.3.11 Notification.....	28

3.3.12 Voice Prompt.....	29
3.3.13 Statistic	29
4.0 Design Quality.....	30
4.1 Design Heuristics.....	30
5.0 Design Patterns.....	32
5.1 Creational Patterns	32
5.2 Behavioural Patterns.....	32
5.3 Structural Patterns	32
6.0 Bootstrap Process.....	33
7.0 Verification	34
7.1 Registers and Verifies Account	34
7.2 Performs Top-up.....	35
7.3 Books a Trip with Promotion	36
7.4 Reschedules a Trip.....	37
7.5 Cancels a Trip and Requests Refund.....	38
7.6 Submits Feedback.....	39
7.7 Views Points and Redeems Reward	40
7.8 Reviews Feedback	41
7.9 Manages Merchandise Stock.....	42
7.10 Views Statistic	43
7.11 Manages Promotions	44
7.12 Updates Account Status	45
8.0 Appendix.....	46

1.0 Introduction

This Object Design Document describes the problem analysis conducted, which identified classes, interfaces, and guidelines required to form the system. This document serves as a reference for developers, managers, and stakeholders in the system to exchange information about the system design and future testing.

1.1 Outlook of Solution

The proposed solution for the ART online ticketing system involves the development of two platforms, web-based application and mobile-based application. Both platforms are designed to provide users with a seamless, flexible and accessible way to manage their travel needs with the ART system.

The web-based platform is optimized for desktop and laptop users, offering a wider interface and detailed layout that enhances readability, especially for elderly users or those with visual impairments. It allows users to register, browse ART routes and schedules, make bookings, view past trips, and track transaction history without visiting a physical station. Tickets are confirmed instantly and can be saved as PDFs, with optional printing available at ART stations. While digital tickets also sync to the mobile app for added convenience, users who prefer not to install the app can still access all essential features directly through the web. This platform is ideal for those who prefer a larger screen experience or need to manage multiple bookings easily.

The mobile-based application is designed to leverage the widespread use of mobile devices which provide a convenient and flexible way for users to manage their ART travels activities. It offers all the core functionalities of the web platform which including account registration, trips booking, tickets purchasing and secure payments through the integrated e-wallet system from their phones. Unique features enhance the mobile experience such as real-time push notifications for schedule changes, booking confirmations, delays or cancellations as well as voice command and voice navigation options to support users with disabilities or those seeking greater convenience. The tickets are stored digitally as PDFs file with embedded QR codes to support a completely paperless and streamlined boarding process.

Both platforms are built with cross-platform synchronization, allowing users to switch seamlessly between web and mobile without losing access to their bookings, transaction history or account information. Accessibility is a priority in both designs,

with support for multiple languages and user-friendly interfaces optimized for responsiveness and touch interactions.

Overall, this solution offers a modern, efficient and inclusive digital experience for ART travellers. By integrating features like real-time notifications, e-wallet transactions, voice assistance and cross-platform accessibility, the system aims to transform and enhance the public transportation ticketing experience in Kuching.

1.2 Trade-offs and Object Design

The object design of the ART online ticketing system involves several trade-offs aimed at optimizing user experience, system efficiency, and maintainability.

1.2.1 Consistent User Interface

A key design consideration is the use of consistent layouts across the web and mobile platforms. This ensures that users encounter a familiar and easy-to-use interface, regardless of the device they use. While this approach may require more effort in the initial design phase, it simplifies user training and reduces support costs in the long run.

1.2.2 Efficient Route Navigation

To facilitate efficient trip planning, the system incorporates search bars and filters, allowing users to find routes by date, time, and destination quickly. Although this feature increases usability, it calls for continuous search processing in the backend and a more complex data retrieval technique.

1.2.3 Cross-Platform Compatibility

The system is designed to allow users to switch between web and mobile platforms without losing their data or booking information. This cross-platform compatibility enhances user convenience but requires a strong synchronization mechanism to maintain data consistency across devices.

1.2.4 Accessibility and Inclusivity

The inclusion of multiple language options is a crucial design decision aimed at accommodating non-native speakers and elderly users. This promotes inclusivity but adds complexity to development, testing, and ongoing content management.

1.2.5 Object Model Simplification

During object design refinement, several classes were merged or the responsibilities were combined to reduce the redundancy. For example, the Payment class will merge into the Sale class, and ticket generation responsibilities were moved into the Trip class. These trade-offs reduce the system complexity and improve the maintainability, but slightly limit the future extensibility of separate payment and ticketing modules.

1.3 Documentation and Guidelines for Interface

This section will list all the documentation and guideline.

Identifier	Rules	Examples
Classes	Names for classes are to be descriptive, simple and related to ART software system. Class names are to be capitalised.	class Account class Admin class Trip
Attributes	Describe the behaviours or attributes that own by a class.	Trip ID Date, time Account status
Operations/Methods	The functions that a class has to perform or trigger by other class.	bookTrip() cancelTrip() rescheduleTrip()

1.4 Definitions, Acronyms, and Abbreviation

This section lists all the word definitions and their short forms, which will be used in the following sections.

Definition	Short Form
Autonomous Rapid Transit	ART
User Interface	UI
Object-Oriented Design	OOD
Responsibility-Driven Design	RDD
Unified Modelling Language	UML

2.0 Problem Analysis

The Software Requirements Specification included a requirements analysis, which revealed a range of functional and quality requirements to build an effective solution from now on. The requirements analysis revealed a list of key functionalities that the program needs to perform to meet the specifications of the various use cases. The key functionality of the application includes:

- Account Management
 - Account registration
 - Secure user login and logout
 - User profile update
 - Account verification
- Trip Management
 - Trip booking
 - Trip rescheduling
 - Trip cancellation
 - View booking history
 - Display real-time schedule
- Payment and Financial Transactions
 - E-wallet top-up
 - Secure online payment processing
 - Refund processing
 - Transaction history
- Merchandise Management
 - Merchandise purchase
 - Inventory management
 - Purchase history for reordering
- Promotions
 - Promotion creation and management by the admin
 - Each user can only apply a discount voucher once
- Points and Rewards
 - Get points per purchase for rewards
 - Merchandise redemption
 - Voucher redemption

- Notifications
 - Send transaction confirmations
 - Send booking updates
 - Send booking confirmation
 - Send promotions alert
 - Send expire date of voucher
 - Send reminder to user about their trip
 - Send feedback confirmation
 - Store notifications history for review
 - Send redemption confirmation
 - Send new merchandise notification
- Feedback and Support
 - User feedback submission
 - Feedback review and management by admin
- Ticketing
 - Digital ticket generation
 - One-time ticket printing
- System Administration
 - Admin handling account issues
- Voice prompt assistance
 - Assist user with disabilities or inconvenience to book a trip
- Statistic
 - Generate the most frequently booked trip routes
 - Analyse peak travel times and dates
 - Generate reports about daily ticket sales
 - Summarize user activity trends
 - Generate monthly or weekly trip statistics
 - Generate data visualisation for the admin review

2.1 Assumptions

Below are the assumptions made by the team during the design of the ART online ticketing system. Each assumption is based on the Sarawak cultural context, user behaviours, system functionality requirements, and operational expectations.

- A1** ART's management already has a UI/UX guideline for any IT system it develops.
- A2** Users are expected to register only one account per phone number and use their own phone number for account registration.
- A3** Users can remember their own password to login or use a password recovery mechanism.
- A4** The login module is assumed always to be secure, and the system will log changes for security and audit purposes.
- A5** Users are expected to upload their identity card or passport for verification.
- A6** Users have internet access and own a device to book trips at any time and from anywhere.
- A7** Users can cancel the trips within 5 hours before departure, but last-minute cancellations are not allowed.
- A8** Trips can only be rescheduled if there are available seats on a new date or time.
- A9** Users are expected to review past or current bookings to manage their trips.
- A10** Online transactions are assumed to be secure and trustworthy. Users are expected to top-up their accounts before booking.
- A11** User's payment methods are reliable and compatible with the system and the system automatically calculates fees and discounts during checkout.
- A12** System can handle a large number of users as well as transactions.
- A13** Refunds will be processed within 20 minutes if the user requests to cancel a trip.
- A14** Admins will create promotions within the company's feasible scope to make the company profitable. Users are expected to apply the promotional voucher when there is a voucher available to apply.
- A15** Users will be motivated to earn points to redeem the vouchers and check their total points regularly to redeem rewards.
- A16** Users are expected to understand the use of digital tickets.

- A17** Users are assumed to buy merchandise during promotions.
- A18** Admin will always keep the inventory updated to prevent overselling.
- A19** Users will check past deliveries or reorder the items as needed.
- A20** Users will provide useful feedback to help improve the system or services, and station managers will prioritize resolving the critical feedback.
- A21** Admins will handle all account issues and use statistics to optimise services such as ART routes or schedules.
- A22** Users will always check the notification every time they complete a transaction.
- A23** Users will always rely on real-time schedules to plan their trips or activities.
- A24** The system is expected to have a voice prompt to help users with disabilities.
- A25** Users will only be allowed to print the ticket once to prevent cheating by printing multiple tickets.
- A26** Users are expected to purchase two tickets if their trip requires a transit, as there is no direct route to the destination.

2.2 Simplifications

Based on the assumptions made regarding the requirements, there were some simplifications made to the project that allowed us to structure the application more effectively:

- User class will replace with the Account class to better reflect the system's requirement that users must have an account to interact with the system.
- Booking class can be remove from our design diagram and its responsibilities like booking, cancelling and rescheduling trips were integrated into Trip class. This will simplify the design by placing trip management logic within the Tripp class.
- Ticket class will be removed from the design diagram also and the ticket generation will made a reasonability of the Trip class. This eliminates the redundancy as the ticket information is derived from trip details.
- Payment class can be merged into Sale class to simplify the relationship since a payment is inherently a part of sale, combining them streamlines the model.

- Voice Prompt class will be discarded, but its functionality will be considered a method within the Trip class as it is a feature rather than a core entity.
- Use consistent layouts across web and mobile platforms to ensure a unified and intuitive interface design.
- Add search bars and filters, such as by date, time, and destination, to help users find trips quickly.
- Users can switch platforms without losing their data sync, web and mobile bookings.
- Provide language options to accommodate non-native speakers.

2.3 Design Justification

The ART online ticketing system is designed to operate seamlessly across web and mobile platforms, ensuring user accessibility and operational scalability. The system components are structured to allow each platform to present information and handle user interactions independently. Each screen or page focuses only on the specific functions, such as browsing schedules, booking tickets, and making payments. No redundant information is displayed, achieving a clean and efficient interface design.

Both platforms are developed to share a consistent visual language and workflow while adapting to the unique interaction styles of web and mobile users. In addition, user data and activities are synchronized across platforms, allowing users who switch between web and mobile to continue their interactions seamlessly. This ensures a smooth, connected, and intuitive experience when using the ART online ticketing system.

During object design refinement, responsibilities were distributed using Responsibility-Driven Design (RDD) to ensure each class captures a single, focused abstraction. This process was supported by CRC cards and a UML diagram to clearly define the class interactions.

To improve cohesion, redundant classes such as Payment class and Ticket class were merged into Sale class and Trip class respectively. The design patterns like Singleton for the Notification class and Observer for the Trip class were employed to

ensure a clean inter-class communication while maintaining low coupling and high modularity.

Moreover, the Statistic class was introduced to relieve analytics responsibilities from the Trip class, enabling advanced admin insights. This separation of concerns promotes the modularity, improves the maintainability, makes testing more straightforward, and supports future enhancements like visual dashboards or machine learning-based predictions.

2.4 Discarded Class List

From the SRS documentation, the team has created a data model for the online ART ticketing system. After the refinement process, the team found that some classes in the data model may be considered a function or method under another class. The team also found that some classes can be discarded due to redundant connections and causing a loop between classes, which is not ideal for the developers and stakeholders. Below is the list of classes that are discarded from the UML data model:

1. User

This class is just replacing the class name to Account class. Since the team is designing a software system for the ART online ticketing system, the User class may not be suitable in this case. The reason is that the team has made an assumption that the user will need an account in order to book a trip or purchase merchandise. Thus, based on this concept, we actually found that the Account class is a more suitable name in terms of constructing the software design.

2. Booking

Initially, our design is to put it as a function class, where it has responsibilities to book, cancel or reschedule a trip. However, we discovered that all these responsibilities actually have the same purpose, which is changing the state of a trip. Thus, instead of creating a function class called Booking, we just integrate all the responsibilities that the Booking class has into the Trip class.

3. Ticket

For the Ticket class, basically it is a function class which will only print the ticket once the user book or purchase a trip. It is not necessary to be put in a class, but can be included in one of the Trip class responsibilities. Moreover, the ticket class has attributes like origin, destination, date, and time, but all these attributes are redundant in the Trip class, so it is not good to repeat all the attributes again. After refining, the Ticket class can be discarded and the generate ticket method can be put into the Trip class.

4. Payment

In our initial redefined data model, we did include the Payment class in it and it connected to a new Sale class. However, the relation between Payment and Sale class is 1 to 1 relationship. It means that the relationship between the classes is actually redundant to each other, or in other words, it is a repeated class. Thus, we eventually discarded the Payment class and put all the Payment class responsibilities into the Sale class. This minor change makes the refined data model more straightforward to understand for those developers or stakeholders who are first time proceed to this software system.

3.0 Candidate classes

After discarding the irrelevant class list in Section 2.4, Section 3.0 will discuss the full list of candidate classes contained in the design solution, a refined UML-type class diagram based on the candidate class list and deploy the CRC cards for each of the classes given in Section 3.1.

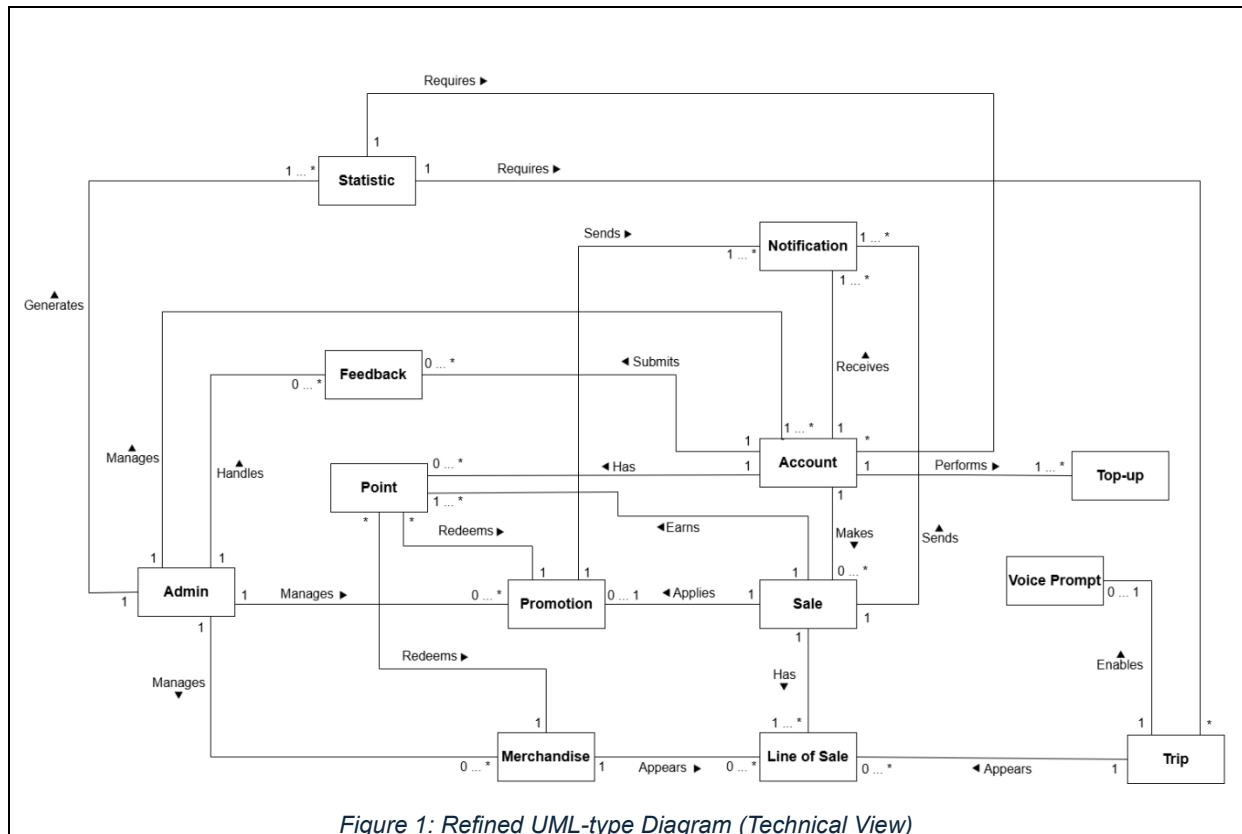
3.1 Candidate Class List

In this section, the team will list out all the candidate class lists based on the online ART ticketing system after refining the data model from Assignment 1. Below are classes that fulfil the software system requirements:

- Account
- Feedback
- Top-up
 - E-wallet
 - Online Banking
 - Credit Card
 - Debit Card
- Point
- Sale
- Line of Sale
- Promotion
- Admin
 - System Admin
 - Feedback Manager
 - Promotion Coordinator
 - Merchandise Coordinator
 - System Analyst
- Notification
- Merchandise
- Trip
- Voice Prompt
- Statistic

3.2 UML Diagram

In this section, the team will discuss the refined UML-type class diagram, which highlights the relationships between the candidate classes. The team also includes a brief justification of the chosen classes and how the relationship is being connected between classes.



In the Figure 1 above, it shows the relationships between the classes. This is slightly different with the final refined UML-type diagram (Figure 2), which the team included Voice Prompt class for the trip booking, reschedule and cancel processes. This is more to technical part for the developers to get more insights on how these UML diagram helped in the implementation. However, the team will focus on the final refined UML-type diagram in the next page.

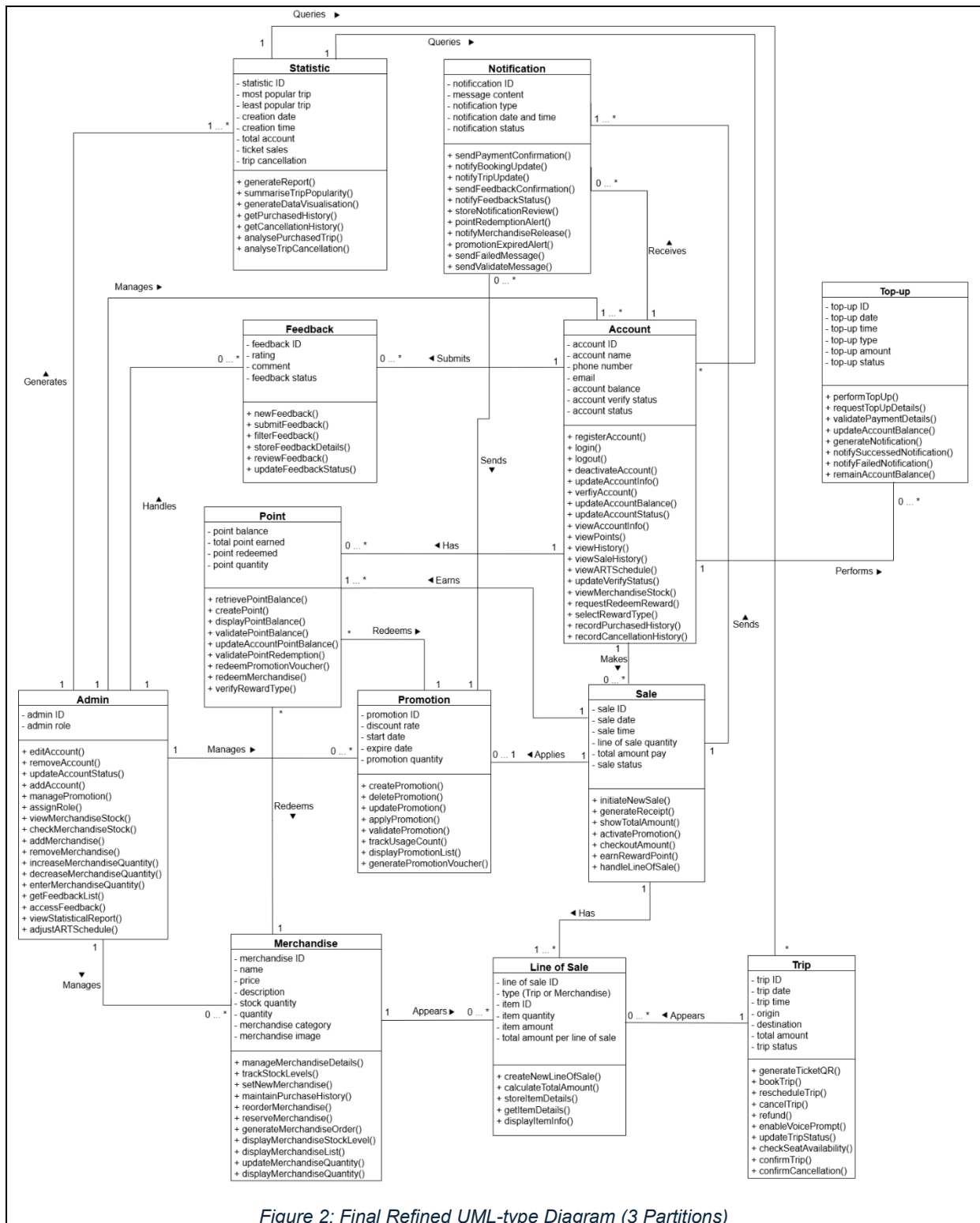


Figure 2: Final Refined UML-type Diagram (3 Partitions)

In this refined UML diagram, the team actually further refined the data model from SRS documentation earlier and split the classes into three partitions. Each class has their own attributes and the methods, and demonstrated how the class going to interact with other classes.

After conducting an in-depth problem analysis, a list of candidate classes was identified through applying the separation of concerns in order to develop an online ART ticketing system. A few classes were added to this refined diagram, and at the same time, some classes were discarded as well (as mentioned in Section 3.1). The table below will give a thorough justification for how each class is chosen, along with the type of class.

Table 1: Justification of Classes Chosen

Class Name	Class Type	Justification
Account (replace User class)	Data class	Account is clearer and context-appropriate in ART online ticketing system if compared to User class. It stores user-related information such as personal information and account balances, as well as the ability to initiate top-ups and purchases. It replaces User class to better represent real-world use in e-commerce situations.
Top-up (new added)	Function class	The TopUp class is in charge of processing payments added to a user's account. It works with the Account class to validate inputs, make payments, and update account balances.
Sale (new added)	Data class	Sale class is a basic entity that collects payment and transaction information. It contains the date, time, and total amount and represents as a parent class to the Line of Sale class. It also has the ability to generate receipt after a successful purchase.
Line of Sale (new added)	Data class	A Line of Sale refers to individual items either Merchandise or Trip in a sales transaction. It includes the quantity, type, and reference of the item purchased. The design supports itemized billing and transaction tracking.
Notification (new added)	Function class	The Notification class sends real-time alerts and messages for users, such as booking confirmations, trip reminders, promotional updates, and payment notifications. It ensures that users are consistently notified about important system events.

Statistic (new added)	Data class	Statistic class collects and generates analytical data in order to monitor and make predictions about the system. It works with the Admin class to create reports and visualizations such as user activity, revenue trends, and booking frequency.
Voice Prompt (new added but not involved in UML diagram)	Function class	Voice Prompt class manages audio-based user interactions, including voice-assisted navigation, confirmations, and prompts. It improves system accessibility, especially for the elderly or users with disabilities.
Feedback	Data class	Feedback gathers user-submitted reviews, ratings, or remarks about trips, merchandise, or the overall system experience. It allows administrators to monitor and improve the quality of services.
Point	Data class	Point class represents reward points earned through transactions (in our design is under Sale class). It keeps track of the point balance, earning and redemption history, and connects with the Account and Promotion classes.
Promotion	Data class	Promotion class handles discount promotions and special offers. It can be applied to a Sale to adjust the total amount according to eligibility like point redemption, seasonal discounts.
Admin	Data class	Admin class is made up different admin roles, such as system admin, feedback coordinator, promotion manager, and merchandise coordinator and system analyst . It handles the system's backend operations, user moderation, and content management.
Merchandise	Data class	Merchandise class refers to items that are available for purchase within the system. It stores information such as ID, name, price, stock quantity, and links to Line of Sale for purchase tracking.
Trip	Data class	Trip includes travel-related information such as origin, destination, date, time, and seat availability. It allows for trip booking,

		reservation tracking, and ticket production like QR code for digital boarding.
--	--	--------------------------------------------------------------------------------

After understanding the justification of chosen classes, now in the table below will indicate the relationship between each class. Notably, each class is assuming as a single object or class.

Table 2: Relationship between Each Classes

Class Name	Relationship between classes
Account	<ul style="list-style-type: none"> • One Account submits zero to many Feedback • One Account has zero to many Point • One Account performs zero to many Top-up • One Account makes zero to many Sale • One Account receives zero to many Notification • One to many Account manages by one Admin • Many Account data used to generate one Statistic
Top-up	<ul style="list-style-type: none"> • Zero to many Top-up performs by one Account
Point	<ul style="list-style-type: none"> • One to many Point earns by one Sale • Zero to many Point owns by one Account • Many Point redeems one Merchandise • Many Point redeems one Promotion voucher
Feedback	<ul style="list-style-type: none"> • Zero to many Feedback handles by one Admin • Zero to many Feedback submits by one Account
Sale	<ul style="list-style-type: none"> • One Sale earns one to many Point • One Sale has one to many Line of Sale • One Sale applies zero to one Promotion • One Sale sends one to many Notification
Line of Sale	<ul style="list-style-type: none"> • One Line of Sale contains zero to many Merchandise • One Line of Sale contains zero to many Trip • One to many Line of Sale appears in One Sale
Promotion	<ul style="list-style-type: none"> • One Promotion sends zero to many Notification • Zero to one Promotion uses in one Sale • One Promotion voucher redeems by many Point
Admin	<ul style="list-style-type: none"> • One Admin handles zero to many Feedback • One Admin manages one to many Account • One Admin manages zero to many Merchandise • One Admin manages zero to many Promotion • One Admin generates one to many Statistic
Merchandise	<ul style="list-style-type: none"> • One Merchandise appears in zero to many Line of Sale • Zero to many Merchandise manages by one Admin

	<ul style="list-style-type: none"> • One Merchandise redeems by many Point
Trip	<ul style="list-style-type: none"> • One Trip appears in zero to many Line of Sale • One Trip enables zero to one Voice Prompt • Many Trip data requires to generate one Statistic
Notification	<ul style="list-style-type: none"> • Zero to many Notification sends by one Promotion • Zero to many Notification receives by one Account • One to many Notification sends by one Sale
Statistic	<ul style="list-style-type: none"> • One to many Statistic generates by one Admin • One Statistic queries many Account data • One Statistic queries many Trip data

3.3 CRC Cards

In this section, the team will discuss 12 CRC cards by listing out possible responsibilities and their collaborators along with a brief description of the class.

3.3.1 Account

Class Name: Account	
Account class represents a user who will interact with the online ART ticketing system. It holds the user's personal details, manages account managements like register, login or remove account. It also tracks account balance, views and manages trips, purchases, and notifications.	
Attributes	Collaborators
Knows the account ID	
Knows the account name	
Knows the phone number	
Knows the email	
Knows the account balance	
Knows the account verify status (verified or unverified)	
Knows the account status (active or inactive)	
Responsibilities	Collaborators
Can register a new account	<ul style="list-style-type: none">• Point• Top-up• Sale• Notification
Can sign in account	
Can sign out account	
Can remove or deactivate an account	
Can update account information	
Can verify account	
Can update account balance	
Can update account status	
Can view account information	
Can view and manage point balance	
Can view account balance and top-up history	
Can view sale history	
Can receive and view notification	
Can view ART schedule	

3.3.2 Top-up

Class Name: Top-up	
The top-up class manages the financial operation of adding funds to a user's e-wallet, ensuring the payment is securely processed and users are notified upon success or failure.	
Attributes	Collaborators
Know the top-up ID	
Know the date and time	
Knows the top-up type (e-wallet, online banking, credit card or debit card)	
Knows the top-up amount	
Knows the top-up account ID	
Knows the top-up status (pending, success, failed)	
Responsibilities	Collaborators
Can perform top-up transaction	<ul style="list-style-type: none"> • Account • Notification
Allow user to choose top-up type such as e-wallet, online banking, credit card or debit card	
Can request top-up details from users	
Can validate payment details	
Can update e-wallet balance	
Can remain account balance when the transaction is failed	
Confirm successful or unsuccessful top-up and notify user	

3.3.3 Point

Class Name: Point	
The Point class manages reward points earned by users through Sale. It enables point redemption for promotion vouchers or merchandise and tracks the point balances.	
Attributes	Collaborators
Knows the point balance	
Knows the total point earned	
Knows the point redeemed	
Knows the point quantity	
Responsibilities	Collaborators
Can redeem reward like promotion vouchers or merchandises	<ul style="list-style-type: none"> • Merchandise • Promotion • Account • Notification
Can create points to account	
Can update points balances to account	
Can display point balance to users	
Can verify reward type	
Can validate sufficient points for redemption	

3.3.4 Feedback

Class Name: Feedback	
The Feedback class collects user-submitted reviews, complaints or suggestions about the system. It facilitates feedback resolution by the admin (feedback coordinators) and stores historical data.	
Attributes	Collaborators
Knows Feedback ID	
Knows Account ID	
Knows comment or rating	
Knows the status, like pending or resolved	
Responsibilities	Collaborators
Can create new feedback	<ul style="list-style-type: none"> Account Admin
Can submit feedback	
Can collect user feedback	
Can store feedback details	
Can allow the admin to review and process feedback	
Can update feedback status	

3.3.5 Sale

Class Name: Sale	
Sale class represents a completed payment in the ART online ticketing system. It manages the payment records when an Account book a trip or purchases merchandise, calculates the total amount, applies promotions if available and generate a receipt. Each Sale also earns some points depends on the total amount of the Sale.	
Attributes	Collaborators
Knows the sale ID	<ul style="list-style-type: none"> Line of Sale
Knows the sale date and time	
Know the quantity of line of sale	
Knows the total amount need to pay	
Knows the sale status like completed, refunded or failed	
Attributes	Collaborators
Can initiate new sale	<ul style="list-style-type: none"> Account Promotion Point Line of Sale Notification
Can generate receipt either downloadable or viewable	
Can show total amount payable	
Can activate promotion if available	
Can check amount payable	
Can earn reward points based on the Sale amount (RM10 equals to 10 points)	
Can handle line of sale	

3.3.6 Line of Sale

Class Name: Line of Sale	
Parent Class: Sale	
Line of Sale class represents an individual item or service purchased within a single Sale payment in the ART online ticketing system. Each Line of Sale records detailed information about the purchased item, including the quantity, the item type and total amount for that particular item. This provides a cleaner view for a single Sale to manage multiple purchases efficiently, whether they are tickets for trips or merchandise products.	
Attributes	Collaborators
Knows line of sale ID	<ul style="list-style-type: none"> • Sale • Trip • Merchandise
Knows the type (Trip or Merchandise)	
Knows the item ID	
Knows the item quantity	
Knows the amount per item	
Knows the total amount per line	
Knows the sale ID it belongs to	
Responsibilities	Collaborators
Can create new line of sale	<ul style="list-style-type: none"> • Sale • Trip • Merchandise
Can calculate the total amount payable per item or trip	
Can get and store item details for each merchandise or trip	
Can link the purchased item to the Sale	
Can provide item information for receipt generation	

3.3.7 Promotion

Class Name: Promotion	
The promotion class manage all promotional offers within the ART system, including general discounts, special event offers, and voucher codes, which help attract users and encourage them to purchase more trips.	
Attributes	Collaborators
Knows the promotion ID	
Knows the discount rate	
Knows the start date	
Knows the expire date	
Knows the promotion quantity	
Responsibilities	Collaborators
Can create promotion	<ul style="list-style-type: none"> • Trip • Sale • Point
Can delete promotion	
Can update promotion	
Can apply promotion discounts when purchasing ticket or merchandise	
Can validate if a promotion is active based on the current date	

Can apply a promotion voucher from redeemed points	
Can generate promotion voucher	
Can track the usage count	
Can display promotion list	

3.3.8 Admin

Class Name: Admin	
The admin manages and maintains the overall ART system by handling user accounts, promotions, merchandise, and feedback, as well as analyzing statistical reports, ensuring smooth and efficient operations.	
Attributes	Collaborators
Knows the admin ID	
Knows the admin's role	
Responsibilities	Collaborators
Can edit user accounts	<ul style="list-style-type: none"> • Account • Promotion • Merchandise • Feedback • Statistic
Can remove user account	
Can add user account	
Can manage promotion	
Can assign roles	
Can view merchandise stock	
Can add merchandise	
Can remove merchandise	
Can increase merchandise quantity	
Can decrease merchandise quantity	
Can enter merchandise quantity	
Can get the feedback list	
Can access feedback submitted by the user	
Can view the statistical report and adjust the ART trip schedules	

3.3.9 Merchandise

Class Name: Merchandise	
Merchandise class represents products available for purchase through the system, maintaining stock levels, item descriptions and enabling purchases as sales.	
Attributes	Collaborators
Knows merchandise ID	
Knows merchandise details such as name, price and description	
Knows merchandise stock quantity	
Knows category of merchandise	
Knows merchandise image	
Responsibilities	Collaborators
Can manage merchandise details	<ul style="list-style-type: none"> • Admin
Can track stock of levels	

Can link merchandise to line of sale	<ul style="list-style-type: none"> • Line of sale • Account
Can maintain user purchase history	
Can view and reorder from the previous merchandise purchases	
Can reserve merchandise	
Can generate merchandise order	
Can display merchandise list	
Can update merchandise quantity	
Can display merchandise quantity	
Can display merchandise stock level	

3.3.10 Trip

Class Name: Trip	
The Trip class manages travel bookings, including ticket generation, rescheduling, cancellations and refunds. Trip booking can also be initiated via voice prompt assistance.	
Attributes	Collaborators
Knows the Trip ID	
Knows the Trip Date and Time	
Knows the Total Amount	
Knows the origin	
Knows the destination	
Knows the trip status	
Responsibilities	Collaborators
Can generate Ticket QR	<ul style="list-style-type: none"> • Voice Prompt • Line of Sale
Can booking a trip	
Can reschedule a trip	
Can cancel a trip	
Can refund	
Can enable voice prompt	
Can update trip status	
Can check seat availability	
Can confirm the trip once selected by the user	
Can confirm trip cancellation upon user request	

3.3.11 Notification

Class Name: Notification	
Notification class acts as the central messaging service that sends alerts and confirmations to users which ensuring users are kept updated with their transaction statuses, trip schedules, promotions and feedback acknowledgments.	
Attributes	Collaborators
Knows the notification ID	
Knows the account ID	
Knows the message content	
Knows the notification type (booking, payment, promotion, feedback)	
Knows the notification generated date and time	
Knows the notification status (read or unread)	
Responsibilities	Collaborators
Can send payment confirmation	<ul style="list-style-type: none"> • Line of sale • Promotion • Feedback • Account • Point
Can notify booking updates	
Can notify booking confirmation	
Can remind user of their trip departure and arrival times	
Can alert users about promotions	
Can send feedback confirmation	
Can notify feedback status once the admin has read the feedback	
Can notify the admin of trip updates when the trip schedule is adjusted	
Can store notifications for user review	
Can notify user once the points are redeemed	
Can notify user if there is a new merchandise release	
Can notify user when the voucher redeemed expire	
Can send validate message	
Can send fail message to other classes	

3.3.12 Voice Prompt

Class Name: Voice Prompt	
The voice prompt class provides accessibility features for users with disabilities or those preferring voice commands. It converts speech to actionable system commands and delivers audio feedback.	
Attributes	Collaborators
Knows the languages	
Knows the voice output settings like speed and volume	
Responsibilities	Collaborators
Can convert speech to text commands	<ul style="list-style-type: none"> • Trip • Account
Can provide audio confirmation for actions	
Can guide users through booking, cancel or rescheduling	
Can validate and execute voice commands like "Book a trip to X"	

3.3.13 Statistic

Class Name: Statistic	
The Statistic class aggregates and analyses trip data to provide valuable insights to the Admin. It supports features such as trip popularity summaries, ticket sales analysis, and trip cancellation overviews, enabling data-driven decisions for improving and optimizing the ART scheduling system.	
Attributes	Collaborators
Knows most popular trip	
Knows least popular trip	
Knows creation date and time	
Knows the statistic ID	
Knows the total account	
Knows the ticket sales	
Knows the trip cancellation	
Responsibilities	Collaborators
Can generate statistical report about ticket sales and trip cancellations	<ul style="list-style-type: none"> • Admin • Trip • Account
Can summarise trip popularity	
Can analyse the purchased trip	
Can analyse trip cancellation	
Can support data visualisation for admin review	

4.0 Design Quality

In this section, the team will make several design heuristics based on our refined UML-type diagram and clarify why each heuristic are being consider in designing the ART ticketing system.

4.1 Design Heuristics

H1 A class should capture one and only one key abstraction

This heuristic is considered as each class has their own responsibilities and it helps to prevent God class and improve maintainability.

H2 Keep related data and behaviour in one place

This heuristic is considered as Trip class encapsulates trip data and booking, reschedule and cancel logic, and how the Sale class includes both payment and receipt generation. It helps to promote cohesion in the design.

H3 Distribute system intelligence horizontally

This heuristic is considered as the responsibilities are spread across functional and data classes like Notification, Top-up and Statistic, instead of squeeze everything in Account or Trip classes. This can avoid bottlenecks and fosters scalability.

H4 Do not create any 'Super class' or God class

This heuristic is considered as in the refined diagram, we discarded User, Booking and Payment classes and reassigned their responsibilities into other classes. This is to encourage single responsibility is assigned for each class but not create a God class and combine the responsibilities under God class.

H5 Inheritance should be used only to model a specialization hierarchy

This heuristic is considered as in the design we actually store different Admin roles and Top-up types as their class attributes. For instance, Admin class has attribute called admin role, which store different role of admin, while same goes to Top-up class where it also has a top-up type to store different method of top-up. This approach prevents any misused inheritance in the OOD.

H6 Minimize the number of classes with which another class collaborates

This heuristic is considered as in the design we keeps collaborations scoped logically. For example, Trip class doesn't directly interact with Point or Feedback even they have some connections in it. This helps to reduce coupling and complexity in designing the diagram.

H7 Minimize the amount of collaboration between a class and its collaborators

This heuristic is considered as each class sends a small number of focused methods like Sale handles Promotion and Point during checkout only. This enhances the performance and clarity on how the ART system works.

H8 Prefer containment over association when possible

This heuristic is considered as Line of Sale is contained in Sale, and Notification is generated within functional flows. This approach simplifies lifecycle management and structure.

5.0 Design Patterns

In this section, the team will discuss the patterns that may be employed for the ART system to address specific architectural challenges to ensure scalability, modularity and maintainability.

5.1 Creational Patterns

The Singleton Pattern ensures that just one instance of a class exists globally, which allows for centralized access to common resources. In the online ART system, the Notification class uses this pattern to manage all notification-related operations, such as sending booking confirmations, trip reminders, and promotion notifications. By limiting instantiation to a single object, the pattern avoids duplicate resource utilization and ensures consistent notice delivery across online and mobile platforms. For example, when a user completes a ticket purchase, the Singleton Notification instance sends alerts via email, SMS, and in-app messaging without requiring numerous notification handlers.

5.2 Behavioural Patterns

The Observer Pattern defines a subscription system in which objects (AKA observers) receive updates from a subject as its state changes. The Trip class serves as the topic, informing registered Account objects of real-time updates like schedule changes, delays, or cancellations. For example, if the departure time of a trip is changed, the Trip class will automatically notify all impacted users, allowing them to quickly amend their arrangements. This architecture separates the Trip management logic from the user notification procedures, increasing modularity and making it easier to add other notification channels such as voice prompts assistant in the future.

5.3 Structural Patterns

The Facade Pattern improves interactions with complex subsystems by offering a unifying interface. The Sale class acts as a front, coordinating payment processing via Top-up, promotion validation via Promotion, and ticket production via Trip during checkout. For example, when a user buys a ticket, the Sale facade validates payment information, applies eligible discounts, generates a QR-coded ticket, and changes reward points through a single `checkoutAmount()` method. This abstraction

protects users and developers from the complexities of particular subsystems, speeding the transaction process and reducing error-prone code practice.

6.0 Bootstrap Process

The bootstrap process reflects a user registering an account and proceeding to purchase the ART online ticket.

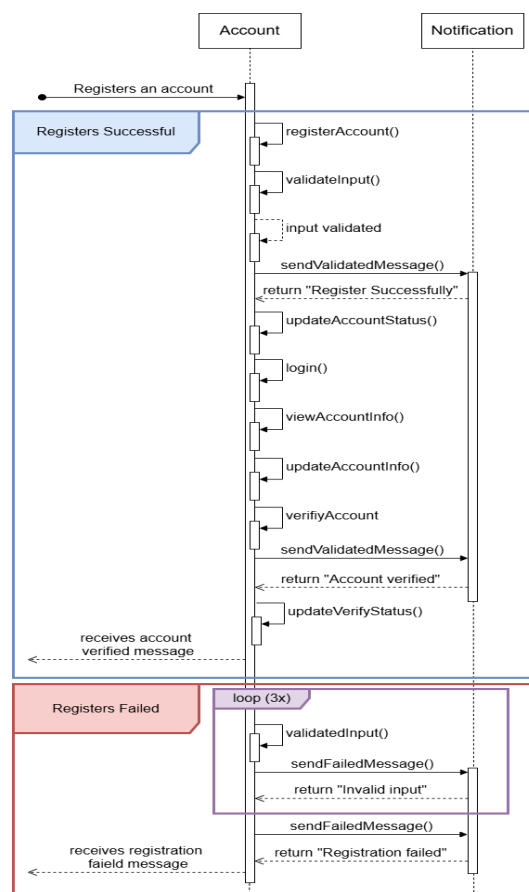
- A new Account instance is created with the user's details during registration.
- The user browses available trips, the system retrieves pre-created Trip instances for display and checks for seat availability.
- If the user applies a discount, the system retrieves a Promotion instance, validates its expiry and availability then calculates the updated price.
- If the user lacks sufficient balance, a Top-up instance is created and linked to the user's account.
- Once the user confirms the trip, a new Sale instance is created to represent the transaction.
- Each ticket or merchandise item within the transaction is recorded as a Line of Sale instance linked to the Sale.
- After the purchase, the system awards points to the user's account, stored in a Point instance.
- A Notification instance is generated to confirm the successful ticket purchase or inform the user of any updates.
- After completing a trip or using the service, the user may submit feedback for resolution by the admin, which creates a new Feedback instance.
- Notification instance is initialized and subscribed to system events like confirmation and top-up success, which stored and accessible for later review.
- The system pre-loads available Trip instance from persistent storage to present real-time availability.
- Upon successful booking purchase, a QR code is generated by the Trip class and embedded in the digital ticket PDF for mobile access or optional printing.
- During registration and login, Account instance are validated against the stored records and the verification status is updated appropriately.

7.0 Verification

In this section, we will go through the verification process for each use case scenario of the ART system based on the proposed UML-type diagram. Each scenario will be illustrated using UML sequence diagram to demonstrate the design solution along with the justification.

7.1 Registers and Verifies Account

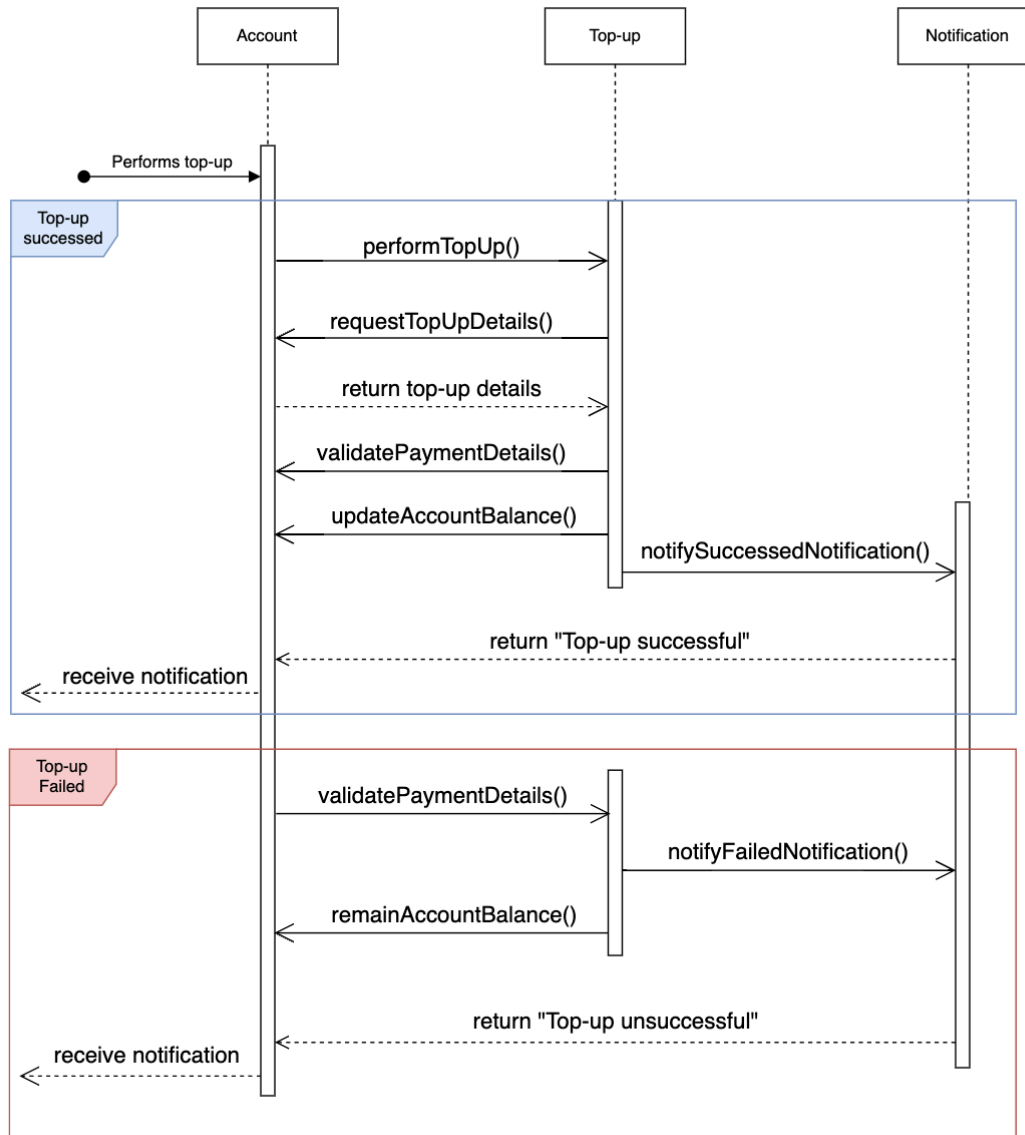
This solution shows an account performs registration and verification processes in online ART ticketing system.



Account starts register by filling the inputs like name, phone number and password. The account input will be validated and if all are validated, it will send a notification to indicate account register successfully and update the account status to “active”. If the inputs entered invalid for three times, it will send failed notification to show account register failed. Once the account logs the application, they can view the account info and update the identity information in the system. Once the account identity is verified, the account will receive message “Account verified” and the account verify status will change to “verified”.

7.2 Performs Top-up

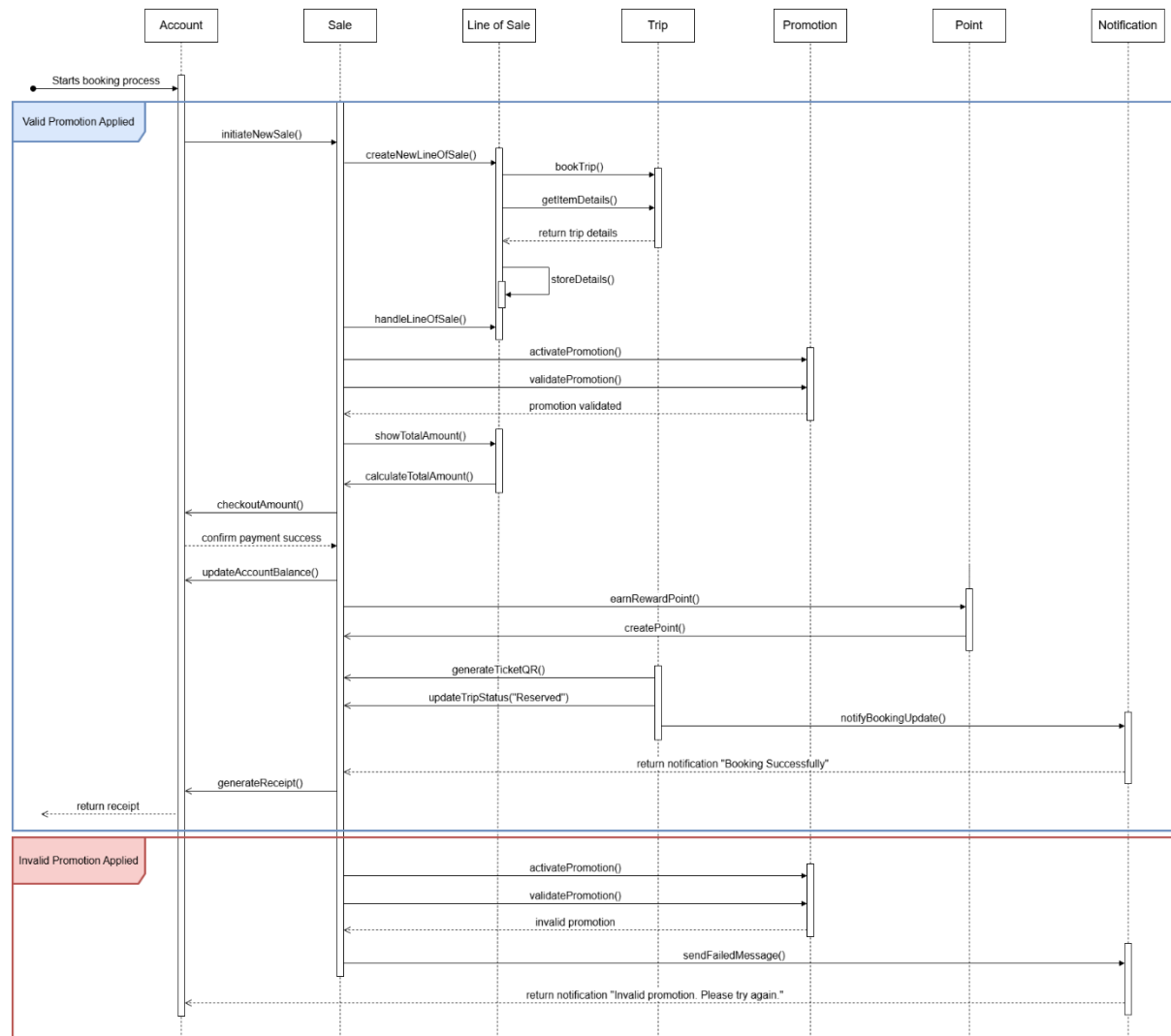
This solution shows the system performs top-up process to the account and send notification to the account.



Account initiates the top-up process and provides top-up details like top-up amount and top-up methods. The system will then validate the payment, if the payment is success the account balance will be update. Otherwise, the payment is unsuccess the account balance will be remain. If there is top-up succussed or failed, the system will send a notification to the account.

7.3 Books a Trip with Promotion

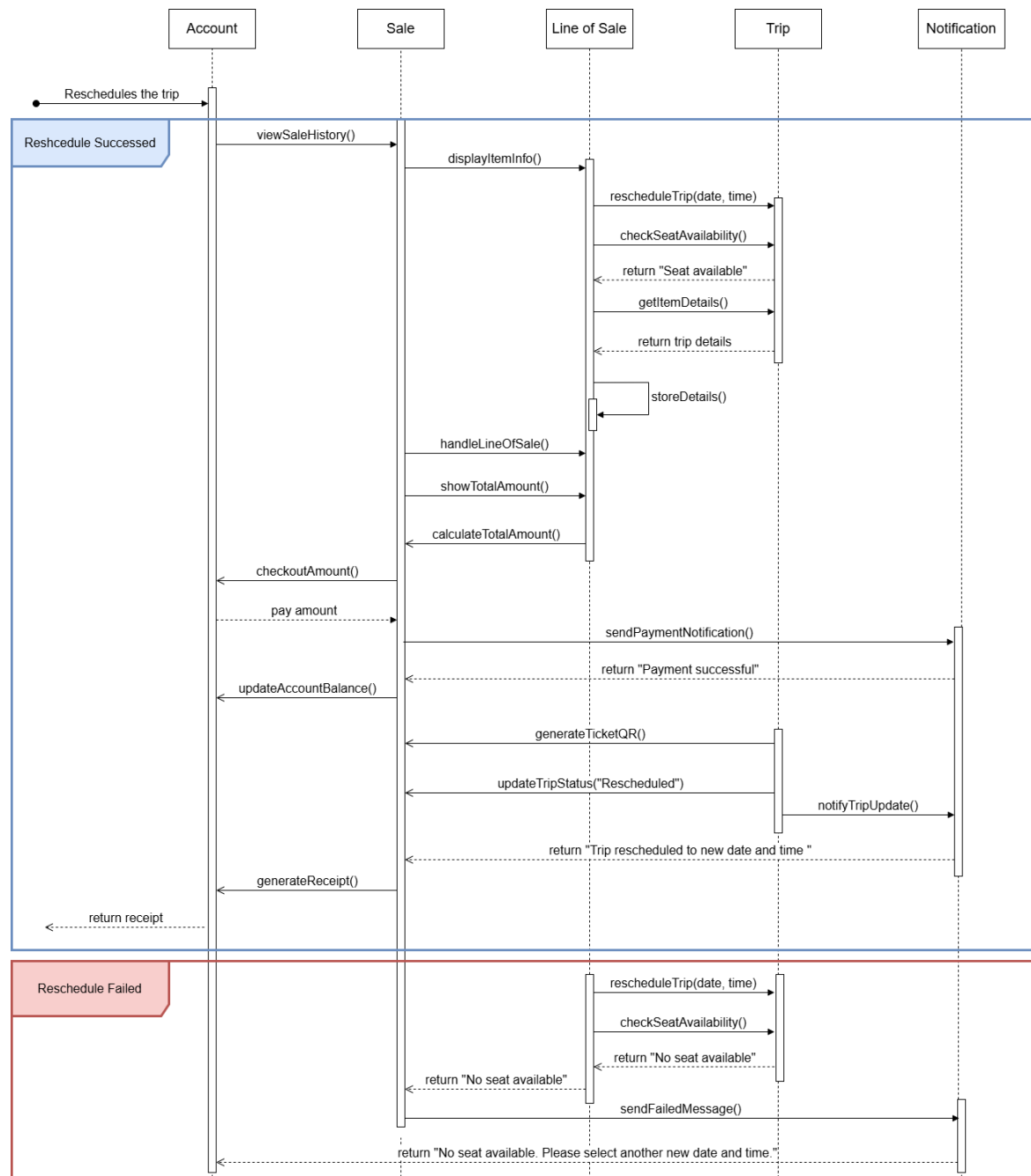
This solution shows that the interaction between the account and the ART system works when the account books a trip with a promotion.



Account starts booking a trip by filling in the trip details, and the details will then store in a line of sale. A sale instance is created to store the line of sale and if the account applies a promotion voucher, the system will validate the voucher. If the voucher is validated, it offers a discount and updates the total amount payment; otherwise, it shows an invalid notification back to the account. Once the total amount calculated, the account need to checkout the amount and pay the exact amount needed for the trip. After performing a successful transaction, it will update the account balance and generate a receipt to account for record purpose. At the same time, a ticket QR that contains the trip information will be generated and shows a booking successfully notification to the account.

7.4 Reschedules a Trip

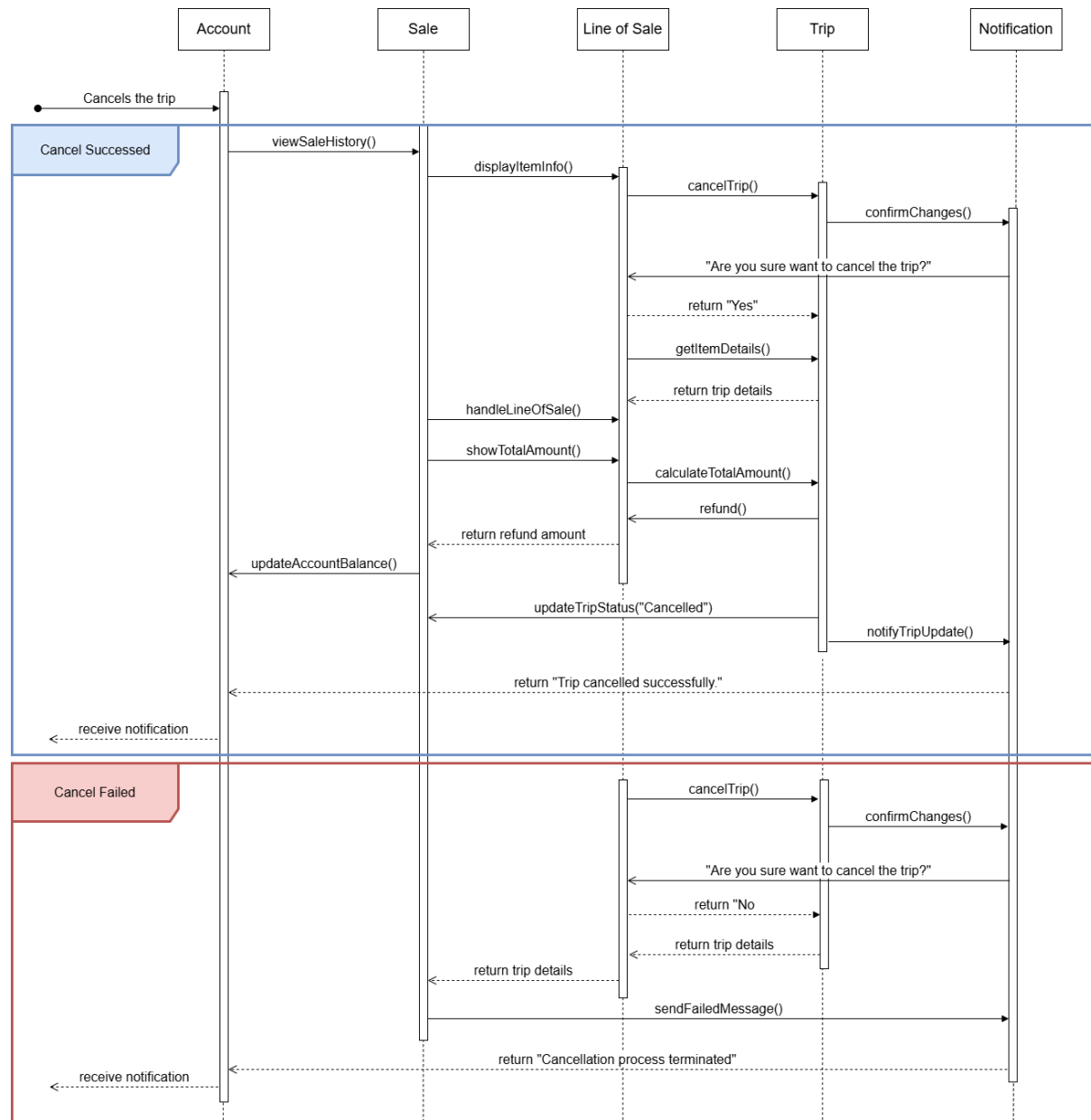
This solution shows the account performs reschedule process in the ART system.



Account views the sale history, and selects the trip that need to reschedule. The sale will then display line of sale with the item information. Account performs reschedule process by selecting another new data and time for the trip, and the system will check the seat availability. If there is a seat available, it continues the reschedule process, otherwise the system shows reschedule failed notification. The new trip will be charges extra fees and account need to checkout the amount and pay for it. The trip status will change to “Rescheduled” and generate a new ticket QR for account.

7.5 Cancels a Trip and Requests Refund

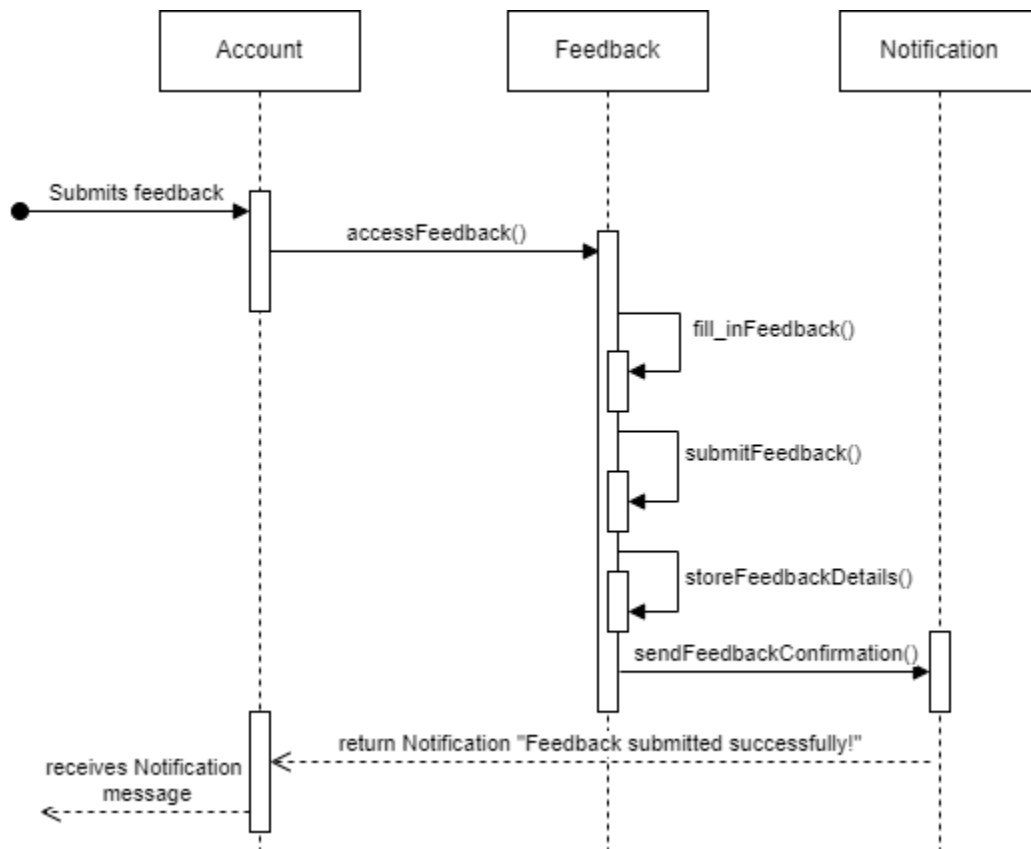
This solution shows the account performs trip cancellation and the system refunds back to the account.



Account views the sale history, and the sale will display a list of line of sale that contain the trip information. Account cancels the trip and the system will send notification to confirm the changes made by account. If yes, the system will proceed to cancellation process and refund respective amount paid to the account and update the account balance. The trip status will change to “Cancelled” and the ticket QR will no longer valid. If account discards changes, all the cancellation process will terminate and account will receive notification “Cancellation process terminated”.

7.6 Submits Feedback

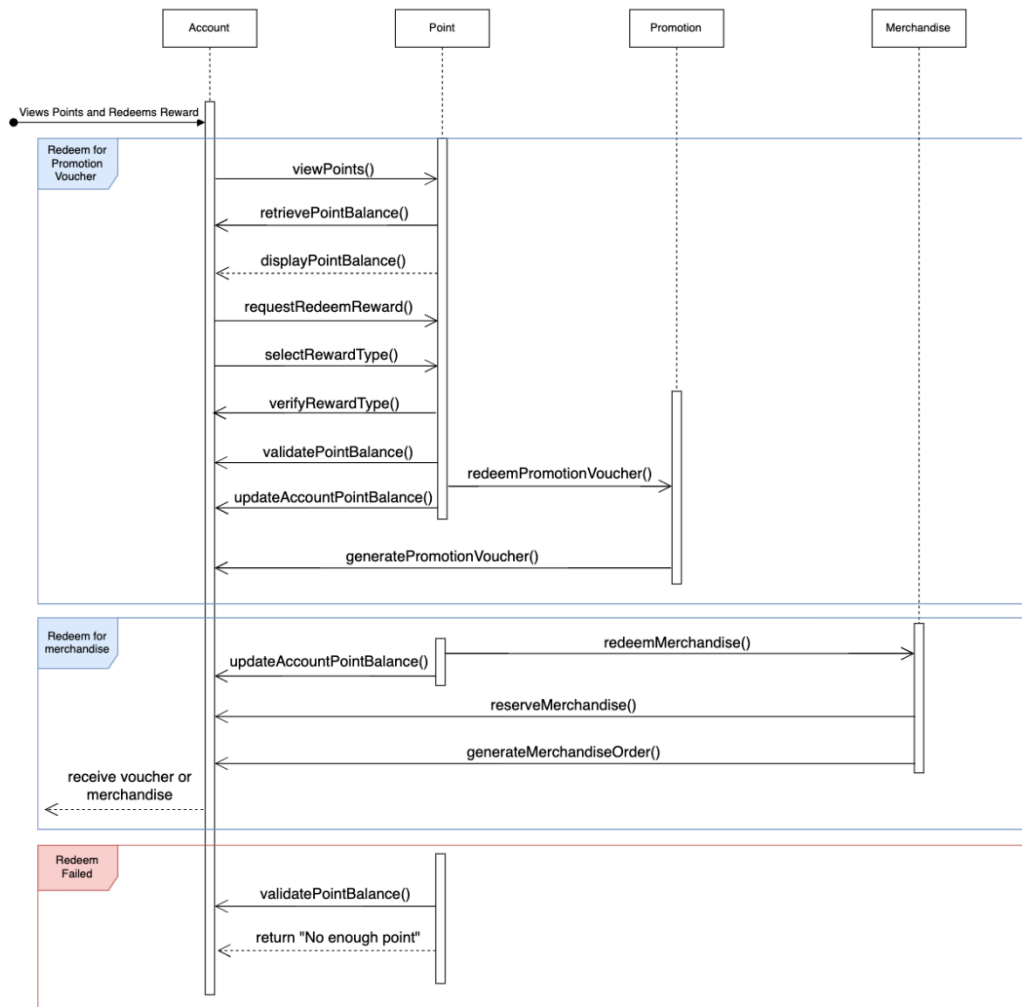
This situation shows that the user submitted the feedback.



Account accesses the feedback form, fills in the details based on their ART trip experience, and submits the feedback. The submitted feedback is stored in the system. Upon successful submission, a confirmation notification is sent to the Notification class, which then notifies the Account that their feedback has been submitted successfully.

7.7 Views Points and Redeems Reward

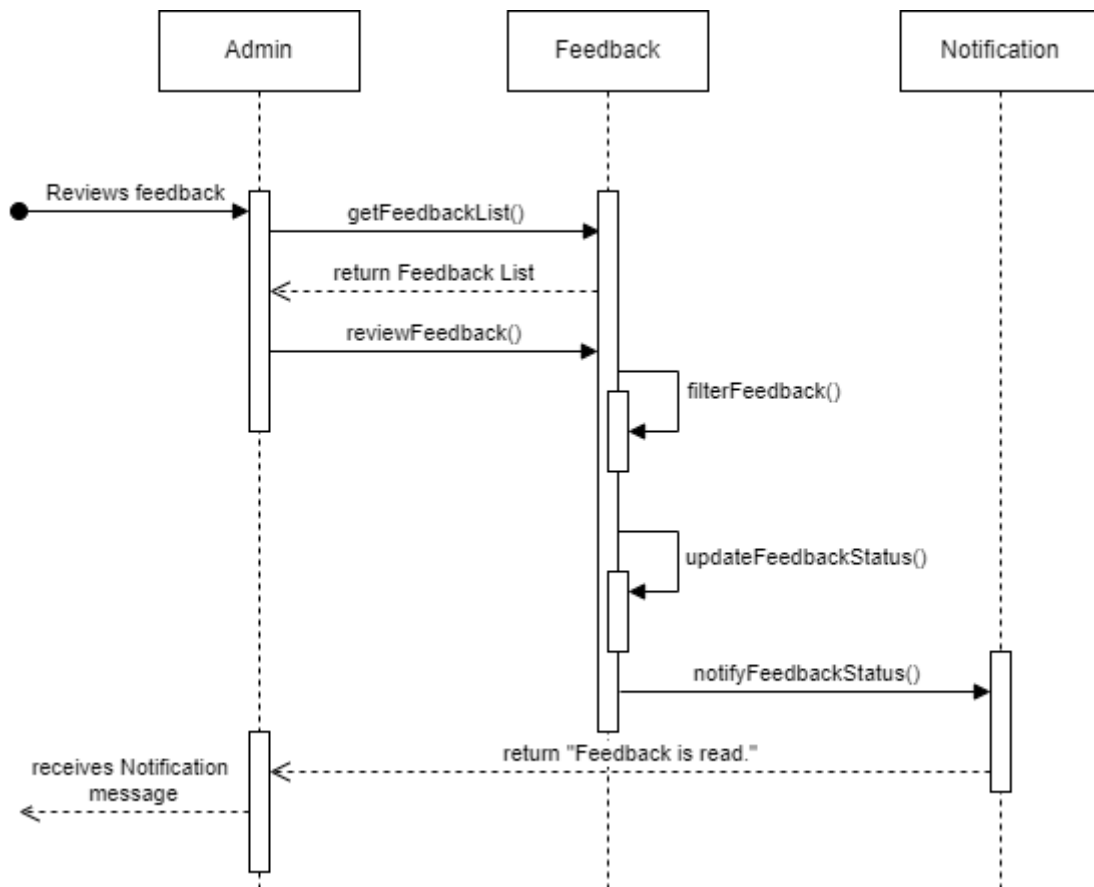
This situation shows the account views points and select a reward type for redemption in the ART system.



Account views the point balance then select a reward between promotion voucher and merchandise. After account selects a reward type, the system will validate the account point balance to check the account is enough point to redeem reward or not. If the account point balance is not enough, the system will not perform redemption and remain the point balance.

7.8 Reviews Feedback

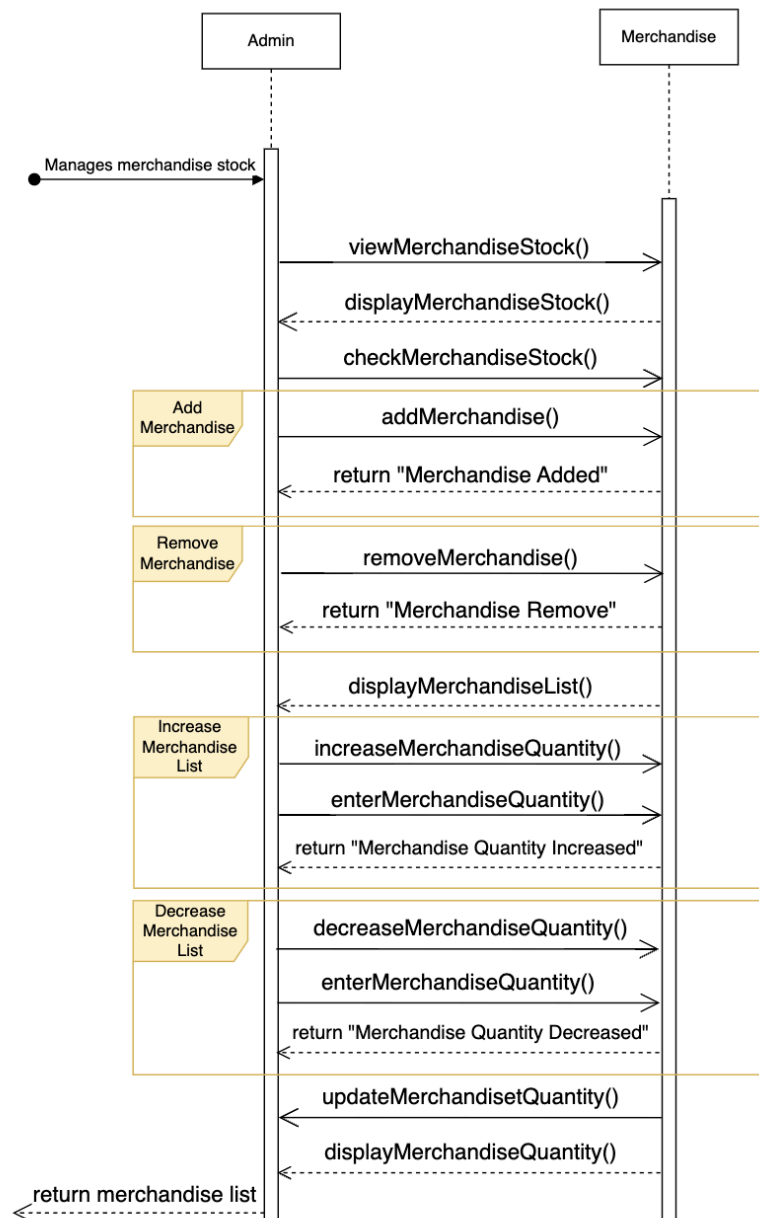
This situation shows that the admin will review the feedback.



Admin retrieves the feedback list from the Feedback class. The Admin reviews each feedback entry and filters those that require action. Once the Admin reads the feedback, the feedback status is updated. The system then uses the Notification class to notify the Admin that the feedback has been marked as read.

7.9 Manages Merchandise Stock

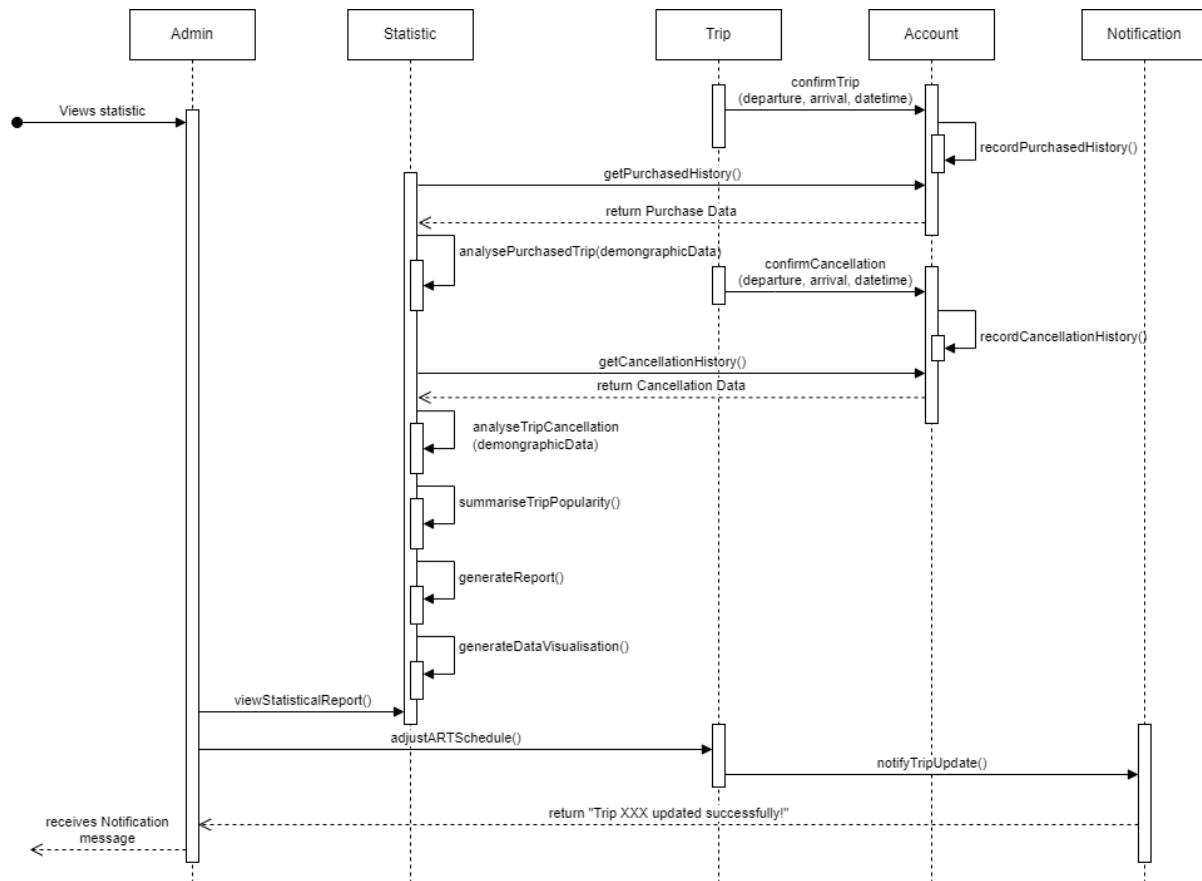
This solution shows the admin manages the merchandise stock within the ART system.



Admin request to view the merchandise stock levels so admin can manage the merchandise stock. The system displays the merchandise stock for admin to view. Admin then check the merchandise stock to decide either want to add new merchandise or remove out of stock merchandise, increase or decrease merchandise quantity. If admin make any change of the merchandise stock, the system will then update the merchandise list and the merchandise quantity.

7.10 Views Statistic

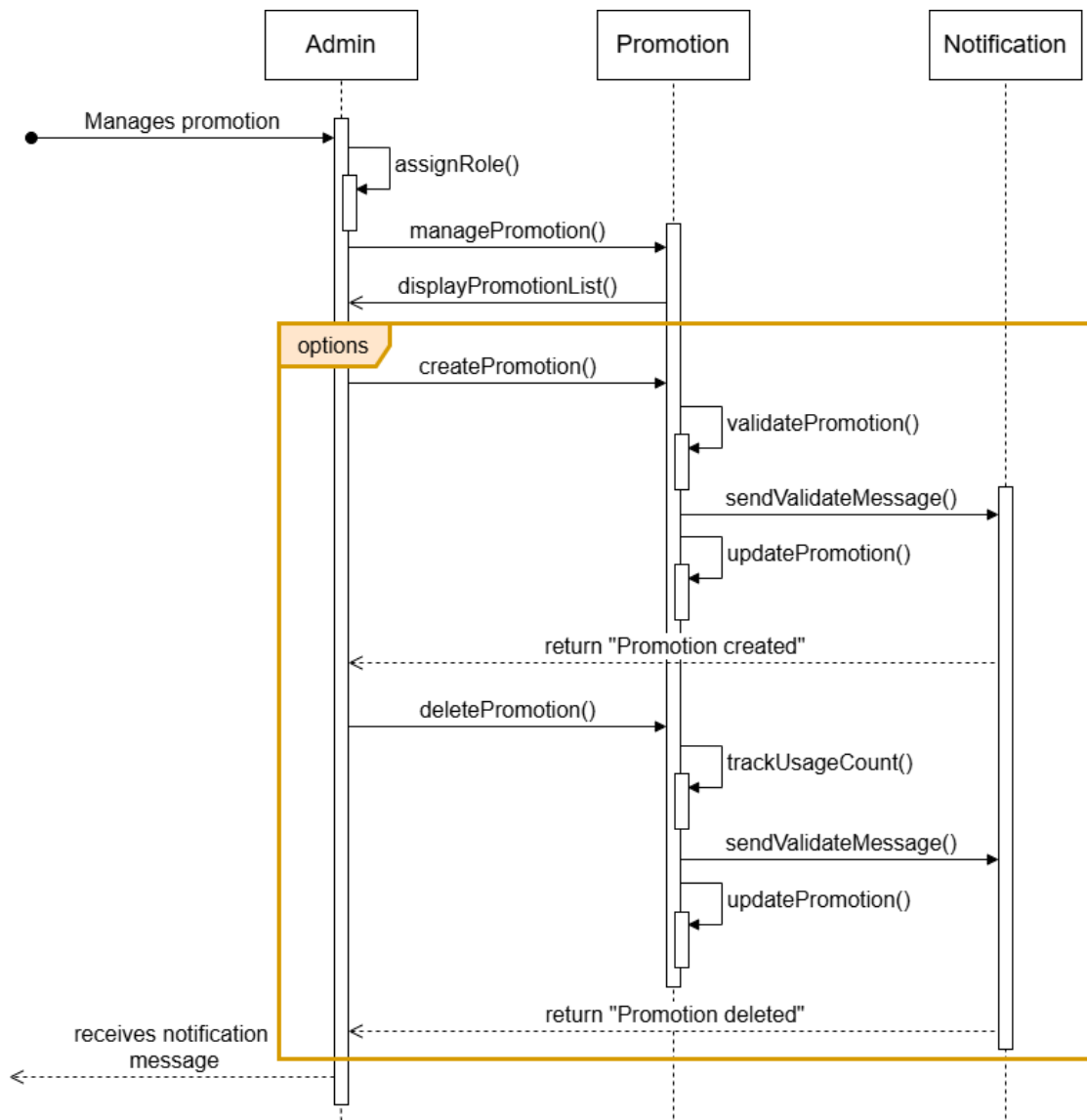
The admin uses demographic statistics from user ticket purchases and trip cancellations to optimize and adjust trip schedules.



Admin views statistical reports generated by the Statistic class. This class collects data from the Account class to analyse ticket purchases and trip cancellations for specific ART routes. When an Account confirms a trip purchase or cancellation through the Trip class, the corresponding history is recorded in the Account class. The Statistic class retrieves this data to analyse and summarise trip popularity, and then generates a comprehensive report, including data visualisations to help the Admin easily compare trends. The Admin views the report to make informed decisions, such as increasing ART schedules during peak hours or reducing schedules for destinations with low user demand. Once a trip is adjusted, the Notification class sends a message to confirm that the Admin has successfully updated trip XXX.

7.11 Manages Promotions

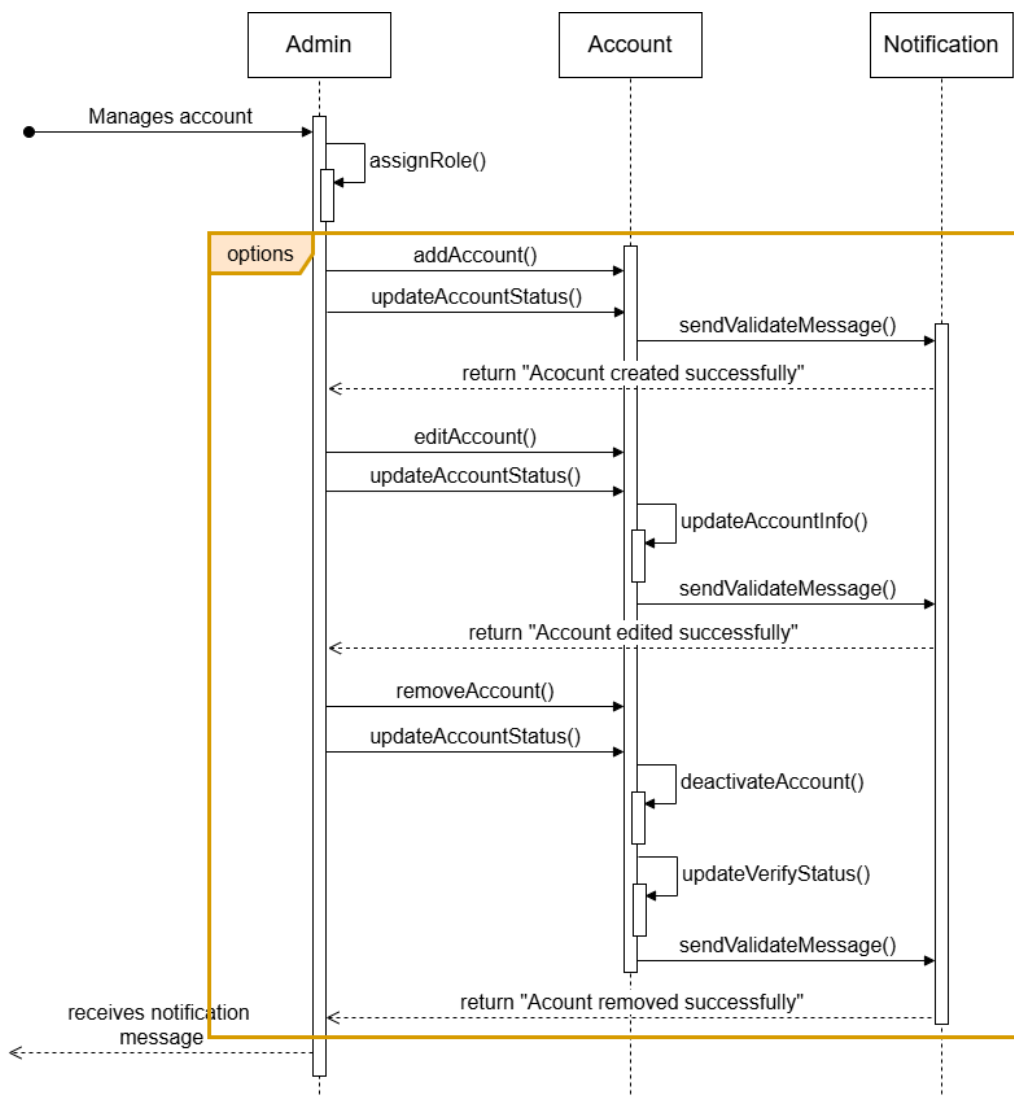
This solution shows admin performs CRUD method to manage the promotions.



Admin will first assign the role itself and starts to manage the promotion by get the promotion list from the Promotion class, then it will display the promotion list to the admin. Admin can choose either create new promotion or delete promotion from the list. If admin choose to create a new promotion, it will validate the new promotion start date and expire date. Once validated admin will receive notification message "Promotion created", and the promotion list will update the new promotion into the list. On the other hand, admin can delete the available promotion from the list, then the Promotion class will track the usage count to check the quantity of the selected promotion. If the promotion quantity is higher than the threshold set, it will send validate message "Promotion delete" to admin and update the promotion list.

7.12 Updates Account Status

This solution shows admin updates the account status based on the activeness of the account in the ART system by using CRUD method.



Admin will first assign a role to itself and performs CRUD to update an account status. Admin can choose either add, edit or remove an account. Let's say admin performs adding account to the system, it will set the credentials for that account and update the status to "Active" and admin will receive message to indicate account is created successfully. Admin can also edit an account by updating the account info or update the account status to either "Active" or "Inactive" and receiving message after editing process finished. In addition, admin can remove an account and update the account status to "Inactive". That account itself will deactivate and update the verify status to "Unverified", then send a message to admin to indicate the account is removed successfully.

8.0 Appendix

Assignment 1 link:

<https://drive.google.com/file/d/1KNgBXK87f7KneJpA9iGfjyzxyO4bNUd5/view?usp=sharing>