

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 – Software Engineering

Lab 1 Deliverables

Lab Group	SCEX
Team	SmartCommute
Members	CHAN ZI HAO (U2222242B)
	AMANDA RAE JOSEPHINE (U2420764F)
	AW YONG WING KIAN, ALVIN (U2223300F)
	IVAN CHENG LI HAO (U2221078L)
	JACE SEOW WEN HUI (U2222469F)

Table of Contents

Documentation of functional and non-functional requirements.....	3
Functional Requirements.....	3
Non-Functional Requirements.....	7
Data Dictionary.....	8
Initial Use Case Model, consisting of Use Case diagram and Use Case descriptions..	9
A. Use Case Diagram.....	9
B. Use Case Description.....	10
UI Mockups.....	19

Documentation of functional and non-functional requirements

Functional Requirements

1. The system shall retrieve the user's location automatically or manually.
 - 1.1. The system shall automatically detect the user's location.
 - 1.1.1. The system shall retrieve the user's current location using the device's GPS when the application is launched.
 - 1.1.2. The system shall request the user's permission to access location services upon first use.
 - 1.1.3. If location services are disabled, the system shall prompt the user to enable them through device settings.
 - 1.2. The system shall update the user's location dynamically.
 - 1.2.1. The system shall provide a manual refresh button for users to update their location on demand.
2. The system shall provide real-time taxi availability based on the user's location.
 - 2.1. The system shall retrieve and update taxi availability data.
 - 2.1.1. The system shall fetch real-time taxi availability data from the LTA Taxi Availability API.
 - 2.1.2. If taxi data retrieval fails, the system shall display an error message.
 - 2.2. The system shall display available taxis.
 - 2.2.1. The system shall identify all available taxis within a 3-kilometer radius from the user's location.
 - 2.2.2. The system shall display a map interface that visually represents the user's location and the locations of available taxis.
 - 2.2.3. The system shall provide a booking button to book the available taxi.
 - 2.2.3.1. When the user clicks on the booking button, the system shall display an option to call a taxi service provider.
3. The system shall provide real-time bus stop and bus arrival information.
 - 3.1. The system shall retrieve and display nearby bus stops.
 - 3.1.1. The system shall retrieve a list of bus stops within a 3-kilometer radius using the LTA Bus Arrival API.
 - 3.1.2. The system shall display nearby bus stops in a list format.

3.1.2.1. The system shall display the nearest bus stop at the top of the list format.

3.1.3. The system shall allow users to tap on a bus stop to view real-time bus arrival details.

3.2. Upon selection of a bus stop, the system shall display real-time bus information details.

3.2.1. The system shall display real-time bus arrival timings.

3.2.1.1. The system shall display bus arrival times in a countdown format.

3.2.2. The system shall display bus service numbers.

3.2.3. The system shall display bus occupancy levels.

3.2.3.1. The occupancy level shall be displayed using low, moderate, or high.

4. The system shall provide real-time MRT station information.

4.1. The system shall identify MRT stations within a 5-kilometer radius using the LTA Station Crowd API.

4.2. The system shall display nearby MRT stations in a list format.

4.2.1. The system shall display the nearest MRT station at the top of the list format.

4.3. The system shall allow users to select an MRT station to view real-time crowd density data.

4.3.1. Upon selection of an MRT station, the system shall display real-time crowd density data using the LTA Station Crowd API.

4.3.1.1. The crowd density level shall be displayed using low, moderate, or high.

5. The system shall provide a weather button.

5.1. Upon clicking the weather button, the system shall retrieve real-time weather conditions using an external weather API.

5.2. Upon clicking the weather button, the system shall display the current weather conditions based in Singapore.

6. The system shall provide an interactive tutorial guide for users.

6.1. The tutorial guide shall display the functional overview of the application.

6.2. The tutorial guide shall display instructions on how to view and book nearby taxis.

- 6.3. The tutorial guide shall display instructions on how to view bus stop and bus arrival information.
- 6.4. The tutorial guide shall display instructions on how to view nearby MRT stations and crowd density levels.
- 7. The system shall provide a search bar that enables the user to retrieve transportation-related information.
 - 7.1. When the user enters "taxi hotline" in the search bar, the system shall display the taxi hotline number.
 - 7.2. The system shall allow the user to search for MRT station information.
 - 7.2.1. When the user enters "MRT station" in the search bar, the system shall display Singapore's MRT map.
 - 7.2.2. When the user clicks on an MRT station on the map, the system shall display the crowd density level at the selected station.
 - 7.3. The system shall allow the user to search for bus stop information.
 - 7.3.1. When the user enters "bus stop" in the search bar, the system shall display a list of bus stops.
 - 7.3.2. When the user selects a bus stop from the list, the system shall display real-time bus arrival times for that stop.
 - 7.4. When the user enters "MRT operating hours" in the search bar, the system shall display the operating hours of the MRT system.
 - 7.5. When the user enters "bus operating hours" in the search bar, the system shall display the operating hours of public buses.
- 8. The system shall allow the user to check transportation fares.
 - 8.1. The system shall prompt the user to select the mode of transport before displaying any fare information.
 - 8.2. The system shall allow the user to check MRT fares
 - 8.2.1. The system shall allow the user to select their commuter category before checking MRT fares.
 - 8.1.2. The system shall display the MRT fare table based on the selected commuter category.
 - 8.3. The system shall allow the user to check bus fares.
 - 8.3.1. The system shall allow the user to select their commuter category before checking bus fares.
 - 8.3.2. The system shall display the bus fare table based on the selected commuter category.

8.4. The system shall allow the user to check taxi fares.

8.4.1. The system shall display the flag-down fare for taxis.

8.4.2. The system shall display peak period surcharges for taxis.

8.4.3. The system shall display location-based surcharges for taxis.

8.4.4. The system shall display booking fees for taxis.

Non-Functional Requirements

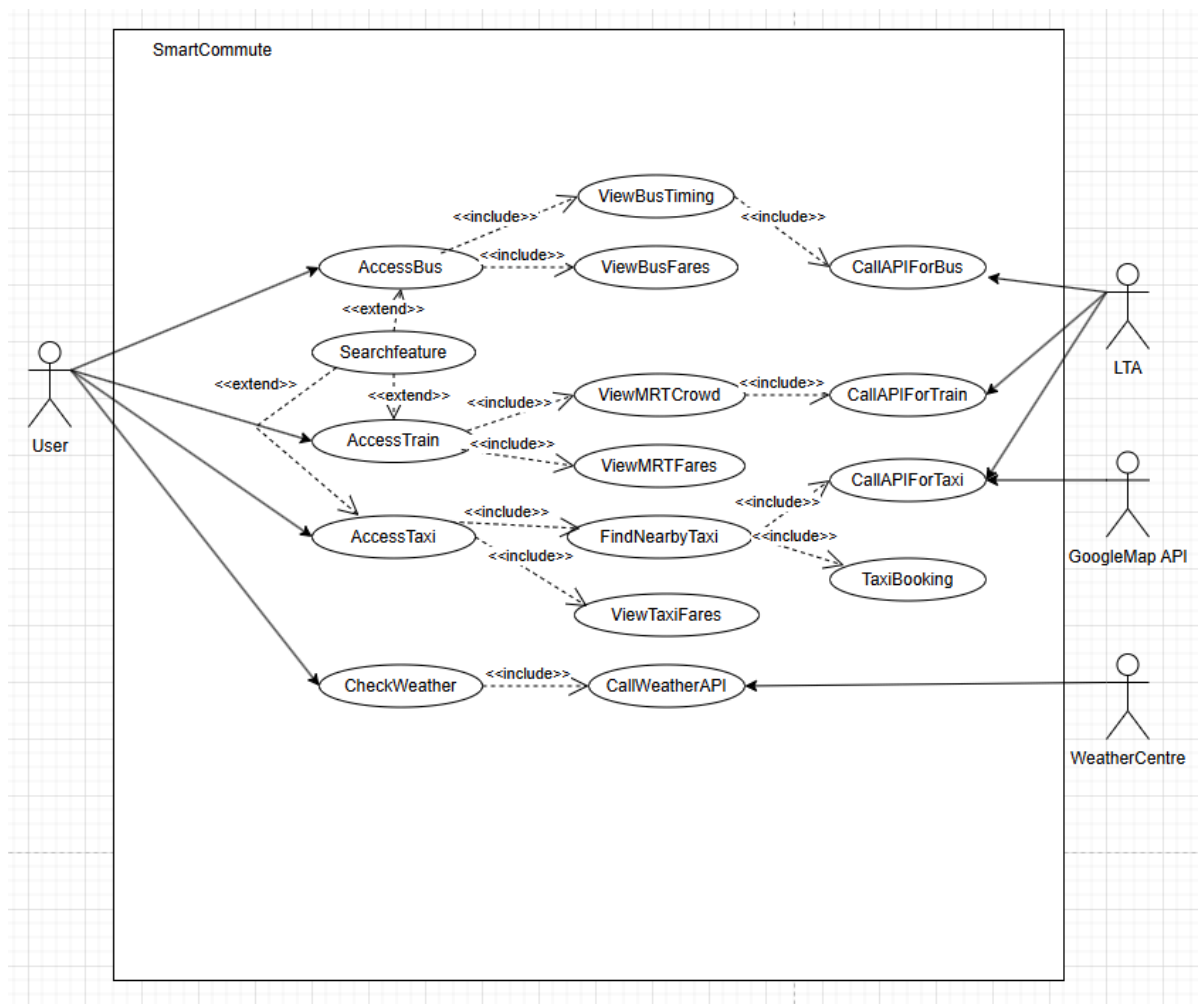
Performance	<ul style="list-style-type: none">- The system must detect the user's location and update it within 15 seconds of application launch or user request.- The system should retrieve and update taxi availability data within 10 seconds from the LTA Taxi Availability API.- The system must refresh and update the user's real-time location every 10 seconds to provide real-time updates.- The system must refresh and update bus arrival times every 10 seconds to provide real-time updates.- The system must refresh MRT congestion levels every 20 seconds to provide real-time updates.
Availability	<ul style="list-style-type: none">- The system must be available 99% of the time, excluding planned maintenance.- Location-based services must function without interruption during system operation.
Reliability	<ul style="list-style-type: none">- The system should be able to handle errors related to data retrieval, and display appropriate error messages to the user while allowing retry options.- The system should maintain consistency across all displayed data and ensure the data presented is synchronised with real-time updates.
Usability	<ul style="list-style-type: none">- The user interface (UI) must be intuitive and easy to use. Icons, buttons, and controls must be clearly labeled.- 85% of new users must be able to view taxi availability, find nearby bus stops, and/or find nearby MRT stations within 2 minutes of starting to use the system.
Customisability	<ul style="list-style-type: none">- The system should allow users to enable night mode through a settings page.

Data Dictionary

Term	Definition
Crowd Density Level	The real-time or estimated level of passenger congestion within MRT stations, train cabins, or platforms, categorized by density levels: Low,High,Moderate,N/A.
Bus Occupancy Level	The real-time or estimated level of passenger occupancy within Bus categorized by density levels: Low,High,Moderate,N/A.
Commuter Category	The type of commuter based on the fares they are paying. There are four types of Commuters: Elderly, Student, Adult, Workfare
MRT Fares	The cost of travel on Singapore's public MRT System. Fare is based on distance from Boarding Station to Alighting Station. Type of Fares depend on payment method: Ez-link, cash
Bus Fares	The cost of travel on Singapore's public bus system based on distance traveled and payment mode (e.g., EZ-Link, cash).
Nearby	A function or service that provides location-based information about transport options. For Bus: 3KM, MRT: 5KM, Taxi: 3KM
Application	A software program or digital platform designed to perform specific tasks or services done on Mobile.
Taxi Fares	The cost of travel in a taxi, calculated based on factors such as distance, time, surcharges, and dynamic pricing models
LTA	(Land Transport Authority) – The government agency responsible for planning, regulating, and developing Singapore's land transport infrastructure and policies.
User	An individual who interacts with a digital system, service, or application, typically requiring authentication or identification.

Initial Use Case Model, consisting of Use Case diagram and Use Case descriptions

A. Use Case Diagram



B. Use Case Description

Use Case ID:	#1		
Use Case Name:	AccessBus		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows access to bus information
Preconditions:	.
Postconditions:	
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User selects and views different bus information
Alternative Flows:	
Exceptions:	
Includes:	ViewBusTiming and ViewBusFares
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#1.1		
Use Case Name:	ViewBusTiming		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows user to view all bus arriving time and bus occupancy in the selected bus stop
Preconditions:	1. User device has working WiFi or Cellular connection 2. Connection established between system and LTA(Bus). 3. LTA(Bus) server online
Postconditions:	1. Display a list of bus stops with their respective bus arriving time & occupancy level.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User selects the bus stop he/she wants. 2. Check for the bus arriving time and occupancy level towards the selected bus stop.
Alternative Flows:	
Exceptions:	EX1: Failed to fetch API 1. An error message will pop up.
Includes:	CallAPIForBus
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#1.1.1		
Use Case Name:	CallAPIForBus		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	LTA
Description:	This use case manages connections between the system and LTA(Bus).
Preconditions:	1. User device has working WiFi or Cellular connection 2. Connection established between system and LTA(Bus). 3. LTA(Bus) server is online
Postconditions:	
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Forward API request to LTA(Bus). 2. Both server received request and send acknowledgement and bus arriving time and occupancy level data back
Alternative Flows:	
Exceptions:	EX1: Unable to reach server 1. No acknowledgement send back to system which leads to exception in #1.1
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#1.2		
Use Case Name:	ViewBusFares		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows user to view all required bus fares (Age Category, Distance, Cash/Card, Bus Service Types and etc)
Preconditions:	
Postconditions:	1. Display bus fares table with different prices, distances, age groups and bus service types.
Priority:	Moderate
Frequency of Use:	Moderate
Flow of Events:	1. User will need to select the bus fare types (Student, Senior Citizen, Adult or Workforce Transport Concession Scheme)

	2. After selection, it will display a table with column attributes like “Distance” and “Bus Service Type” with Price generated below cells.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#2		
Use Case Name:	Searchfeature		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	
Description:	This use case allows users to search for common info such as “BusStopNames”, “MRT Maps”, “Taxi Hotlines”, “MRT Operating Hours”
Preconditions:	
Postconditions:	1. Display the necessary information based on the different selection.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1. User will scroll or search the common info they want. 2. User will require to select. 3. After selecting, it displays the necessary information.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#3		
Use Case Name:	AccessTrain		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows access to train information
Preconditions:	
Postconditions:	
Priority:	High
Frequency of Use:	High

Flow of Events:	1. User selects and views different train information
Alternative Flows:	
Exceptions:	
Includes:	ViewMRTFares and ViewMRTCrowd
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#3.1		
Use Case Name:	ViewMRTCrowd		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows user to check different MRT Station Crowd indicated by 3 colors (Green, Orange and Red)
Preconditions:	1. User device has working WiFi or Cellular connection 2. Connection established between system and LTA(Train). 3. LTA(Train) server online
Postconditions:	1. Display MRT Stations Crowd level
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The system will list all MRT stations with nearest station at the top. 2. User can press each station to see the crowd level indicated (Green, Orange and Red)
Alternative Flows:	
Exceptions:	EX1: Failed to fetch API: 1. An error message will pop up.
Includes:	CallAPIForTrain
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#3.1.1		
Use Case Name:	CallAPIForTrain		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	LTA
Description:	This use case manages connections between the system and LTA(Train).

Preconditions:	1. User device has working WiFi or Cellular connection 2. Connection established between system and LTA(Train). 3. LTA(Train) server online
Postconditions:	
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Forward API request to LTA(Train). 2. LTA(Train) receives request and sends acknowledgement and train crowd data back
Alternative Flows:	
Exceptions:	EX1: Unable to reach server 1. 1. No acknowledgement send back to system which leads to exception in #3.1
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#3.2		
Use Case Name:	ViewMRTFares		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows users to view all expected train fares based on Age Group, Distance, Time and etc.
Preconditions:	
Postconditions:	1. Display train fares table with different prices, distances, age groups and timing
Priority:	Moderate
Frequency of Use:	Moderate
Flow of Events:	1. User will need to select the train fare types (Student, Senior Citizen, Adult or Workforce Transport Concession Scheme) 2. After selection, it will display a table with column attributes like "Distance" and "Timing" with Price generated below cells.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#4		
Use Case Name:	AccessTaxi		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows access to taxi information
Preconditions:	
Postconditions:	
Priority:	Moderate
Frequency of Use:	Low
Flow of Events:	1. User selects and views different taxi information
Alternative Flows:	
Exceptions:	
Includes:	ViewTaxiFares and FindNearbyTaxi
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#4.1		
Use Case Name:	FindNearbyTaxi		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case allows user to see all the nearby taxi around him/her on a map.
Preconditions:	2. User device has working WiFi or Cellular connection 3. Connection established between system and LTA(Taxi) and GoogleMap API. 4. LTA(Taxi) and GoogleMap server online
Postconditions:	1. Display a map of your location coordinates with nearby taxi showing.
Priority:	Moderate
Frequency of Use:	Low
Flow of Events:	
Alternative Flows:	
Exceptions:	EX1: Failed to fetch API 1. An error message will pop up.
Includes:	CallAPIForTaxi and TaxiBooking
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#4.1.1		
Use Case Name:	CallAPIForTaxi		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	LTA
Description:	This use case manages connection between the system, LTA(Taxi) and GoogleMap API
Preconditions:	1. User device has working WiFi or Cellular connection 2. Connection established between system and LTA(Taxi) and GoogleMap API. 3. LTA(Taxi) server online
Postconditions:	
Priority:	Moderate
Frequency of Use:	Low
Flow of Events:	1. Forward API request to LTA(Taxi) and GoogleMap API. 2. Both servers receive request and send acknowledgement, TaxiCoordinates and map back.
Alternative Flows:	
Exceptions:	EX1: Unable to reach server 1. 1. No acknowledgement send back to system which leads to exception in #4.1
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#4.1.2		
Use Case Name:	TaxiBooking		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	LTA
Description:	This use case allows users to call taxi service provider
Preconditions:	1. User device has working WiFi or Cellular connection
Postconditions:	
Priority:	Moderate
Frequency of Use:	Low
Flow of Events:	1.User selects booking button 2.After selecting, it will display an option to call the taxi service provider.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#4.2		
Use Case Name:	ViewTaxiFares		
Created By:	Ivan Cheng Li Hao	Last Updated By:	

Date Created:	6/2/2025	Date Last Updated:	
---------------	----------	--------------------	--

Actor:	User
Description:	This use case allows user to view all expected taxi fares such as “Flag-down Fare”, “Peak Periods”, “Location Surcharges” and “Booking Fees”.
Preconditions:	
Postconditions:	1. Display Taxi Fare Table with its respective prices
Priority:	Moderate
Frequency of Use:	Low
Flow of Events:	1.User will need to select the type of fare they looking to find from the dropdown. 2. After selecting, fare table will be displayed.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#5		
Use Case Name:	CheckWeather		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	User
Description:	This use case creates a weather panel for display.
Preconditions:	1. User device has working WiFi or Cellular connection 2. Connection established between system and WeatherCentre 3. WeatherCentre server online
Postconditions:	1. Display Weather panel with today’s temperatures and weather condition.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	
Alternative Flows:	
Exceptions:	EX1: Failed to fetch API 1. An error message will pop up.
Includes:	CallWeatherAPI
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	#5.1		
Use Case Name:	CallWeatherAPI		
Created By:	Ivan Cheng Li Hao	Last Updated By:	
Date Created:	6/2/2025	Date Last Updated:	

Actor:	WeatherCentre
Description:	This use case manages connection between the system and WeatherCentre.
Preconditions:	1. User device has working WiFi or Cellular connection 2. Connection established between system and WeatherCentre. 3. WeatherCentre server online
Postconditions:	
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1.Forward Request to WeatherCentre using API. 2.WeatherCentre sends acknowledgement and weather data back to system.
Alternative Flows:	
Exceptions:	EX1: Unable to reach server 1. 1. No acknowledgement send back to system which leads to exception in #5
Includes:	

UI Mockups

