

SUBWAY TICKET PURCHASE SYSTEM

Malaysia's Ministry of Transport is planning to develop a simple subway ticket booking system for travelers and residents. The route map is shown in Figure 1 below.

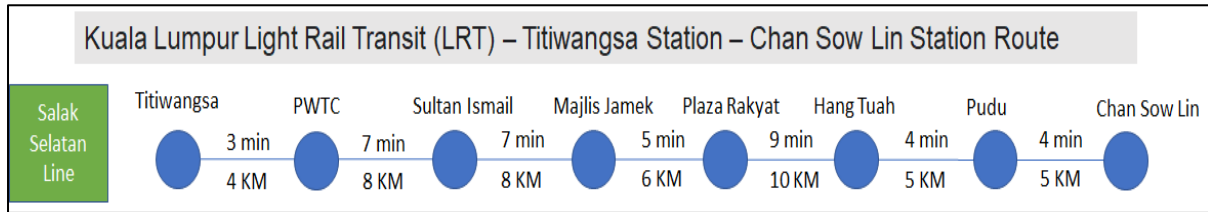


Figure 1 : Subway Route Map Details

Based on their requirements, the system should be able to assist passengers with:

1. Calculate the total travel distance required between the two cities selected by the passenger.
2. Calculate the total travel expenses required between the two cities selected by the passenger.
3. Calculate the total travel time required between the two cities selected by the passenger.
4. Allow passengers to book subway tickets before arriving at the subway station.
5. Allow passengers to get the estimated current arrival time from one city to another in real time during the booking stage.

The LRT station operates daily from 6:00am to 1:00am, with trains from Titiwangsa Station to Chan Sow Lin Station or from Chan Sow Lin Station to Titiwangsa Station in every 30 minutes.

If a passenger wants to travel from Titiwangsa to Sultan Ismail, the fare is calculated as Equation 1 according to Table 1 below:

Equation 1 – Ticket Fare Calculation

$$\text{Total fare} = \text{Ticket price from Titiwangsa to PWTC} + \text{Ticket price from PWTC to Sultan Ismail}$$

Table 1: Fare for two cities

Ticket Price for Salak Selatan Line				
Route between		Ticket Price		Route Between
Titiwangsa – PWTC	➡	RM 0.40	⬅	PWTC – Titiwangsa
PWTC – Sultan Ismail	➡	RM 0.80	⬅	Sultan Ismail – PWTC
Sultan Ismail – Majlis Jamek	➡	RM 0.80	⬅	Majlis Jamek – Sultan Ismail
Majlis Jamek – Plaza Rakyat	➡	RM 0.60	⬅	Plaza Rakyat – Majlis Jamek
Plaza Rakyat – Hang Tuah	➡	RM 1.00	⬅	Hang Tuah – Plaza Rakyat
Hang Tuah – Pudu	➡	RM 0.50	⬅	Pudu – Hang Tuah
Pudu – Chan Sow Lin	➡	RM 0.50	⬅	Chan Sow Lin – Pudu

As a junior software developer in FIRST IT Ltd, you had been assigned to help Subang Jaya City Council to develop this system. On this team, **THREE (3)** other candidates will work with you to complete the project. Your team needs to develop the prototype through C++ programming and implement appropriate data structures to store and manage subway station details and passenger purchasing details.

System features must include, but are not limited to, the following options:

Passengers

- 1.1 Choose the travel route from Titiwangsa to Chan Sow Lin or from Chan Sow Lin to Titiwangsa.
- 1.2 Display the complete list of subway stations according to the selection of option 1.1.
- 1.3 Search the subway station details.
- 1.4 View the details between two selected cities, e.g.
 - Display the total travel distance required between cities.
 - Display the total travel expenses required between cities.
 - Display the total travel time required between cities.
- 1.5 Purchase subway ticket
 - Display Ticket Details.
 - Add Customer Information
 - Display the estimated current arrival time from departure city to arrival city.
- 1.6 View purchase transaction history
- 1.7 Delete purchase transaction

Admin

- 1.8 Add new subway station information
- 1.9 Edit current subway stations' information (*Only the fare, travel time between cities, station name can be modified*)
- 1.10 View purchase – View all purchase transactions
- 1.11 Sort purchase – Sort the purchase transactions based on passenger name in ascending order
- 1.12 Search specific customer ticket purchase information
- 1.13 Edit specific customer ticket purchase information (*Cannot edit customer information, only the travel information on the ticket*)
- 1.14 Delete specific customer ticket purchase information

Additional information for this assignment:**a) The list of subway station should contain the following information but not limited to:**

Subway ID, Current Subway Station Name, Previous Station Name, Next Station Name, Travel Distance Between Previous Station, Travel Fare Between Previous Station, Travel Time Between Previous Station, Travel Distance Between Next Station, Travel Cost Between Next Station, Travel Time Between Next Station, Nearby Sightseeing Spots, etc.

b) The list of ticket purchase should contain the following information but not limited to:

Transaction ID, Ticket ID, Name of the Source Station, Name of the Target Station, Total Ticket Amount, Transaction Date and Time, Departure Time, Customer ID, Customer Name, Customer Identity Card / Passport Details, etc.

c) You need to include some sorting and searching algorithms in the system. Justify your choices and explain in detail how they can help you to make your system more efficient.**d) You must hardcode the subway station details in the program. This assignment does not require any SQL or NO-SQL database or text-based document to store subway stations' details.****e) Include all validations required for the system and use good programming practices (e.g., indentation, meaningful identifier names, comments, etc.).**

Assignment Requirements

A group can **only have a maximum of 4 students.**

There are two submissions of the required in this assignment:

1. Proposal Stage (Week 8)

- You need to submit your proposal on the **LMS in week 8.**
- The proposal should include the proposed data structure together with related algorithms.
- The algorithms should be presented in **flowchart or pseudocode.**
- This proposal will contribute **20%** for the final assignment marks (under Design criteria).
- Refer to the Appendix 1 for the PowerPoint Slides contents.

Approximation of Total Pages for the PowerPoint slides: 12 - 40 (max).

2. Final Report and Presentation Stage (follow the hand-in date)

You are required to submit a **softcopy** of assignment report and source code to the Moodle system. The report should contain:

- Detailed explanation of the data structures and classes created, with proper justification on your decisions (include source code defining classes, data members, and method headers only).
- Brief explanation about the algorithms used to implement the functionalities stated above (include code snippets of important parts of implementation).
- Source code of the main function, with screenshots showing program's input and output interactions.

Approximation of Total Pages for the documentation: 30 - 60 (max).

Approximation of Words for the documentation: 2000 words (min)

You must **present your assignment solution and answers** to the lecturer during a Q&A session that will be conducted after the hand-in date.

If you use some code which has been taken or adapted from another source (book, magazine, internet, forum, etc.) then this must be **cited and referenced** using **APA Referencing Style within your source code**, and this must be mentioned explicitly in the **report**. Failure to reference code properly will be treated as plagiarism. **Automated tools for checking code similarities** among submissions will be used, and all detected cases will be treated as cheating. Assessment marks are divided as follows:

Total Percentage	80%		20%	
	Design (proposal)	Implementation	Documentation	Presentation
Total raw marks in each section	25	75	50	50

What You Need to Hand In during the final submission?

1. You are required to hand in the group project report on or before the due date mentioned on the cover sheet of the assignment.
2. A softcopy of the report (in Word Document / PDF format), in addition to the C++ files of the programs. The organization of files and folders must adhere to the following instructions precisely:
 - The report should be named using format
“<GroupNo>_<student ID-leader>_<student ID-member1>_<student ID-member2>_<student ID-member3>.docx”.

For example “G1_TP012345_TP012344_TP012123_TP012126.docx”

 - All the source codes (.cpp and .h) should be zipped into one file and named following the above format. Make sure to **DELETE** all non-source-code files, including executables (*.exe).
3. You should **present an executable solution** during Q&A session to demonstrate program execution, the working of the data structure, your understanding of the code, and ability to modify / fix it.

Marking Criteria:

The program submitted will be evaluated according to the following performance criteria:

Distinction (90% and above)

- Program compiles and executes perfectly
- At least 90% of the required functionalities are correctly implemented
- Efficient data structures and/or algorithms are used in the implementation
- Clear coding style and structure, and code is properly commented
- Functionalities are fully tested/validated in program execution

Credit (70% – 89%)

- Program compiles and executes
- Between 70% and 90% of the required functionalities are correctly implemented
- Implementation uses a data structure or algorithm that is not most efficient
- Clear coding style, and code is properly commented
- Functionalities are not fully tested/validated in program execution

Pass (50% - 69%)

- Program compiles perfectly and executes
- Between 50% and 70% of the required functionalities are correctly implemented
- Implementation uses inefficient data structures or algorithms
- Unclear coding style, or code is not properly commented
- Functionalities are not fully tested/validated in program execution, or produce errors in some cases

Marginal Fail (30% - 49%)

- Program does not compile or run, but coding logic is almost correct
- Between 30% and 50% of the required functionalities are correctly implemented
- Implementation uses inefficient data structures or algorithms
- Unclear coding style, and no comments provided
- Functionalities are not tested/validated in program execution

Fail (below 30%)

- Program is not given
- Program does not compile or run
- Less than 30% of the required functionalities are implemented
- Implementation uses very inefficient data structures or algorithms
- No proper code structure and no comments provided

APPENDIX 1: SHORT PROPOSAL OF YOUR GROUP PROJECT:

Note: Approximation of Total Pages in PowerPoint Slides: 12 - 40 (max).

Before your group start to develop the system, your group is required to write a short proposal to identify what are the data members will be included your structure, what are workflows involved in the system, and which of the algorithms will be implemented in your future system.

Below are the minimum elements that you need to include in your slides:

1. Introduce your group's members.
2. Identify what are the important data structures or classes (optional) will be included in your system.
 - Visualization of data structures
 - A detailed description of the data structures and classes created, with appropriate justification for your decision (including only source code defining classes, data members, and method headers).
3. Briefly explain all the workflows planning for your future system.
 - Use flowchart / pseudocode to describe each of the functionalities' workflows.
 - Provide a brief explanation about the algorithms that you plan to use in your function
 - Provide a proper justification on your decisions.
4. Provide a workload distribution table to determine the tasks of the members.

You are required to submit a softcopy of proposal to the LMS before week 8. If your group fail to submit the proposal report, it will cause your group to get the 0 mark from your design section (20%).

Assessment marks in design section are divided as follows:

Total Percentage	Design Section (20%)	
	Data Structure (design)	System Workflow and Algorithm
Total raw marks in each section	10 Marks	15 Marks

APPENDIX 2: OUTLINE OF YOUR FINAL REPORT:

The final report outline as below:

- Cover Page
 - Module Code and Name
 - Intake code
 - Proposal Title
 - Include your group member list.
- Introduction
 - Brief introduction about the system
 - Source code of each developed data structure / class
 - Detailed explanation of the data structures and classes created
 - Provide justification on your decisions
- Implementation:
 - Source code of each algorithm.
 - Brief explanation about the algorithms
 - Provide justification on your decisions
- Result
 - System Input / Output Screenshot
 - Brief explanation about screenshots
- Conclusion, Future Works, and Reflection.
 - A summary about your system (including limitations)
 - Brief description of your future works based on your system limitations
 - Briefly describe your experience/feedback on the assignment
- References
- Appendix: Workload Matrix Table with signature