IY4113 Milestone 1

Assessment Details	Please Complete All Details
Group	B
Module Title	Applied Software Engineering using Object Orientated Programming
Assessment Type	Java fundamentals part 1
Module Tutor Name	Jonathan Shore
Student ID Number	P501367
Date of Submission	25th of Jan 2026
Word Count	834

- [x] *I confirm that this assignment is my own work. Where I have referred to academic sources, I have provided in-text citations and included the sources in the final reference list.*

- [ ] *Where I have used AI, I have cited and referenced appropriately.

Purpose of the Program

Purpose of the Program

The purpose of the program is to create a console-based application for public transportation that provides detailed information. It is designed to help a user track their public transport journeys for a single day by recording trip details, calculating accurate fares based on specific rules, and monitoring expenses against daily caps.

Core Program functionality

- Journey Recording: Captures details for each trip, including the date, peak or off-peak, starting zone, destination zone, and passenger type.
- Fare Calculation: Different fares based on the user's age.

- Discount and Daily Cap: Applies discounts and implements daily price caps for each passenger type.
- Filtering: Allows users to view a summary of journeys by passenger type, time band, zone, or date.
- Remove Journeys: Allows the user to remove specific journeys by their unique ID or reset all data for the day with user confirmation.
- Summary: Outputs summaries, including the total number of journeys, total expenditure, average cost, and the most expensive journey.

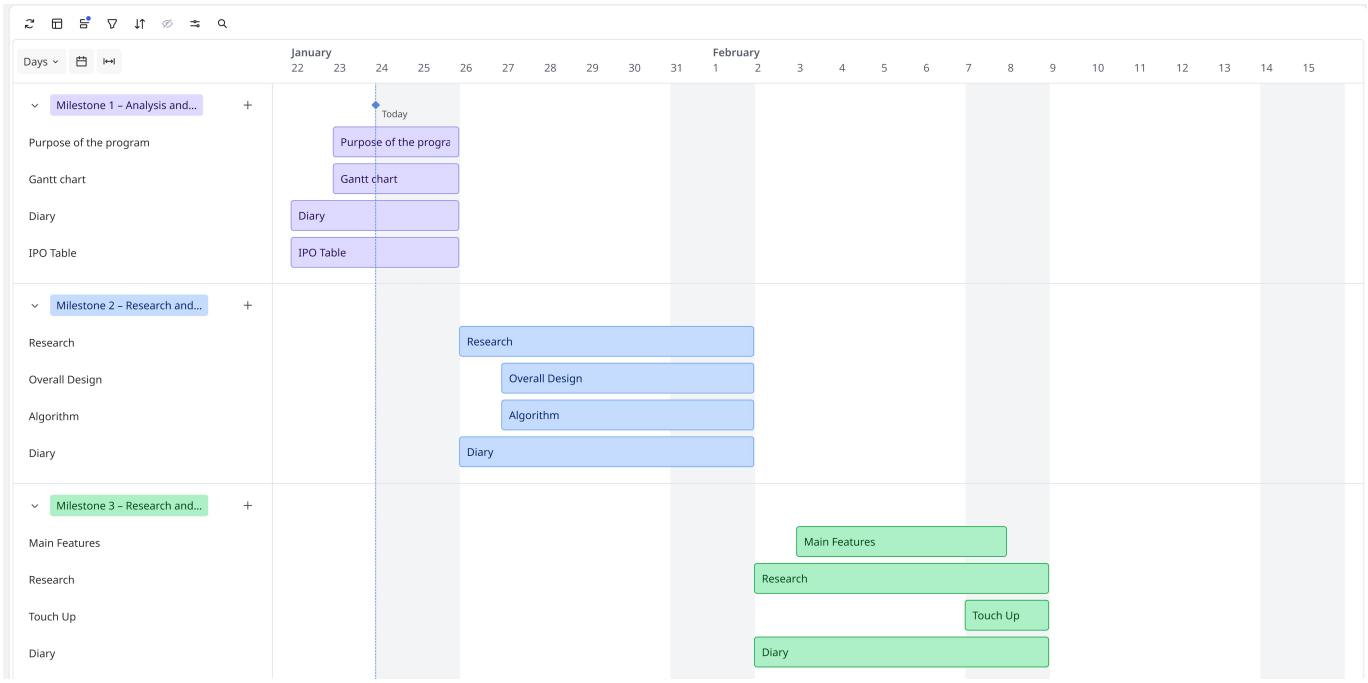
System Constraints (Limitations / Rules)

- Storage: All journey data and totals are stored in memory during the session, and no files or databases are used.
- Data Storing Limitation: The application is limited to a single day's session.
- Zone Boundaries: Valid zones are restricted to the range of 1–5.
- Distance Calculating: The zones crossed must be calculated as "abs(toZone – fromZone) + 1".
- Validating: The system must reject invalid inputs such as zones outside the 1–5 range, invalid passenger types, non-integer zone entries, or blank inputs.
- Daily Cap: If a new journey's fare exceeds the daily cap, the fare must be reduced to the daily cap.

Input Process Output Table

Feature/Tasks	INPUT	PROCESS	OUTPUT
Start program	None (program launch)		Main menu displayed
Menu Selection	User choice (1-8) to navigate the system (Add, List, Filter, etc.).	Check for non-numeric zones, blank inputs, and out-of-bounds values (Zones 1-5).	**Main Menu:** Loop of options until the user chooses to exit.
Journey Details	Date, Time Band, From Zone, To Zone, and Passenger Type.	Calculate zones crossed using "abs(tozone – fromzone) + 1".	Alerts for successful additions, removals, or resets.
Session Data	Active passenger type and optional name.	* Retrieve base fare from the dataset based on zones and time band.	Displays IDs, routes, passenger types, and calculated fares.
Filter Criteria	Search by type, band, zone, or date.	Apply reduction (e.g., Student 25%) to the base fare.	Daily summary such as total count, total cost, average cost, and ID of the most expensive trip.

Gantt Chart



Diary Entries

22/01/2026 - Diary Entry 1 – IPO table and Analysing the program

Today's main goal was to understand the program logic and work out the IPO table. Firstly, I carefully read the assessment instructions, identifying the processes necessary for the program and its main functions. After that, I analysed other IPO tables to understand how to do it myself. Then, I noted down the possible inputs and outputs for the IPO table.

The main problem I encountered was understanding the process between input and output. At first, I could not figure out how the input would be transformed into the output, or what specific steps were involved in the process. This made it difficult to complete the IPO table accurately. However, after further research and reviewing the table examples, I finally understood the logic behind it. By decomposing each step and connecting them, I was able to identify what happens between receiving the input and producing the output. This made it much easier for me to complete the IPO table with confidence.

24/01/2026 - Diary Entry 2 – Purpose of the Program and Gantt Chart

For today, I finalised writing the purpose of the program and the Gantt chart. I briefly read through the system requirements again to pinpoint the functional requirements, which helps me to explain the functions more accurately. After that, I took a look at few Gantt chart templates.

After pinpointing the functional requirements, I started writing around the points. Whenever I encounter uncertainties, I revisit the system requirements to clarify them. The Gantt chart was easier than I thought, since I was familiar with it from another module.

Finally, I polished and finalized the other tasks and made it ready to submit.
