



Assignment/Report Cover Sheet

Student Name (lecture group no)	Student Id Number	L. Group	Signature
JOE MICK MALMSTEEN AK DAVID	112843	6	
MUHAMMAD SYAIRUL AKMAL BIN HASSAN	111167	6	
NUR MASYITAH BINTI SUHAIRIN	111650	6	
SYAFIRUL RAZZI BIN MULYADI	112509	6	
VELERRY BRITNEY ANAK ALEXANDER STEPNY	112667	6	
WAN NUR AMIRA BINTI MOHD AZMI	112722	6	

Subject Code: TMF1414	Subject Name : Introduction to Programming		
Assignment Title :	GROUP PROJECT	Lecturer:	MOHAMAD JOHAN BIN AHMAD KHIRI
Due Date : 7th January 2026		Date Submitted: 7th January 2026	

This cover sheet must be completed, signed and firmly attached to the front of the submission. All work must be submitted by the due date. If an extension of work is granted, an assignment extension acknowledgement slip must be signed by the lecturer/tutor and attached to assignment. Please note that is your responsibility to retain copies of your assignment.

Plagiarism and Collusion are methods of cheating that falls under Peraturan Akademik Universiti Malaysia Sarawak para 11: Etika Akademik

Plagiarism

Plagiarism is the presentation of work which has been copied in whole or in part from another person's work, or from any other source such as the internet, published books or periodicals without due acknowledgement given in the text.

Collusion

Collusion is the presentation of work that is the result in whole or in part of unauthorized collaboration with another person or persons.

Where there are reasonable grounds for believing that cheating has occurred, the only action that may be taken when plagiarism or collusion is detected is for the staff member not to mark the item of work and to report or refer the matter to the Dean. This may result in work being disallowed and given a fail grade or if the circumstances warrant, the matter may be referred to a Committee of inquiry for investigation. Such investigation may result in the matter being referred to the University Discipline Committee, **which** has the power to exclude a student.

Upon placing signature above, I certify that I have not plagiarized the work of others or participated in unauthorized collusion when preparing this assignment.

I also certify that I have taken proper care in safeguarding my work and have made all reasonable efforts to ensure that my work not be able to be copied.

MARK :

Contents

1.0 Introduction	1
2.0 Objectives	2
3.0 Flowcharts.....	3
4.0 User Manual	4
4.1 Individual User Manual	4
4.2 Family User Manual.....	5
4.3 History Transaction and Termination	5
5.0 Output Screenshots	6
6.0 Proof of Discussion	11

1.0 Introduction

The Movie Ticketing System is a console-based application designed to simulate a real-world ticket dispensing environment for a Cinema. This system serves as a practical application of fundamental programming concepts. The system is designed to handle ticket purchases for different customer categories, including Family and Individual tickets, with additional options such as VIP seating and Food and Beverage coupons. The program calculates ticket prices automatically and generates a receipt for each transaction.

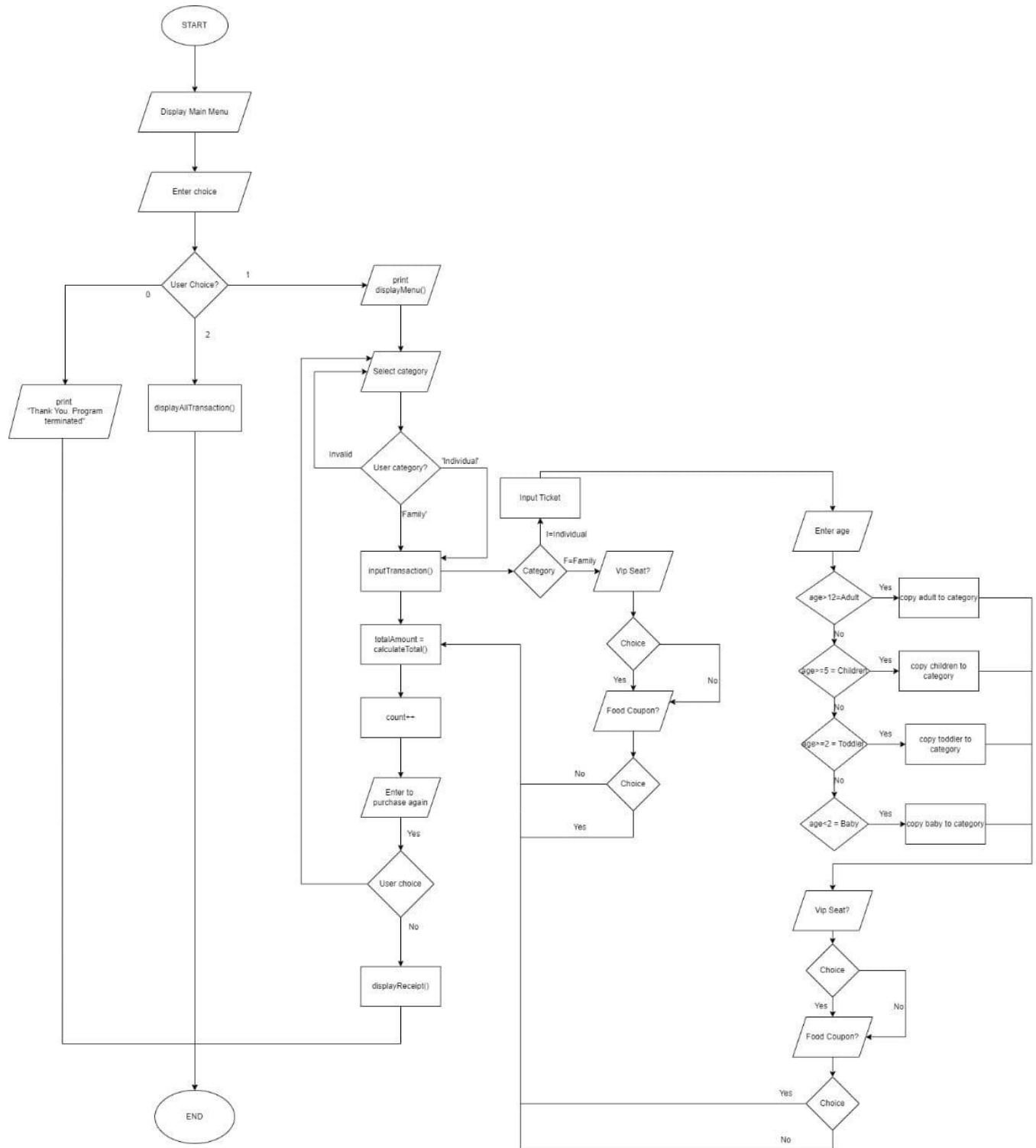
The system is implemented using a command-line interface that provides a streamlined and user-friendly interaction for managing movie ticket purchases. Users are able to select ticket categories, apply optional services such as VIP seating upgrades and Food and Beverage coupons, and receive accurate and well-formatted receipts upon completion of each transaction. Furthermore, the system incorporates file handling to store transaction records permanently, allowing users to retrieve and view transaction history even after the program is closed. By applying structured programming techniques, this project demonstrates the practical use of core C programming concepts while providing a functional solution for managing movie ticket transactions.

2.0 Objectives

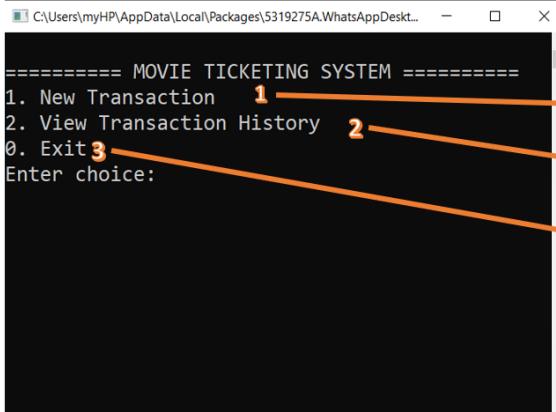
The development of this system aims to demonstrate core competencies in software construction and problem-solving. The specific objectives are to:

- **Develop a Functional Console Program:** Create a bug-free interface that effectively receives user inputs and displays correct ticketing information and receipts.
- **Implement Modular Programming:** Utilize functions to break down the system into manageable components, demonstrating the ability to pass arguments and return values between modules.
- **Utilize Derived Data Types:** Manage complex data effectively by applying structures, arrays, or pointers to organize ticket and customer information.
- **Apply File Operations:** Demonstrate the ability to perform persistent data storage by saving all transaction history to a **.txt or .dat file** for future auditing.

3.0 Flowcharts



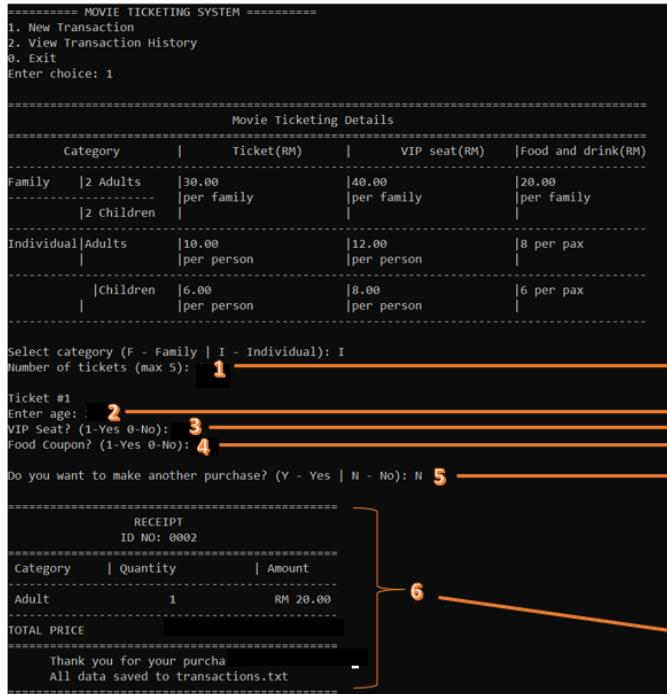
4.0 User Manual



FEATURES

- **Type [1]** - Start a process to book and purchase tickets
- **Type [2]** - Display all past ticket sales and transactions
- **Type [0]** - To ensure all data is saved correctly, always exit using the menu

4.1 Individual User Manual



FEATURES

If you select the **Individual (I)** category,

1. **Number of tickets:** Enter how many tickets you wish to buy (Maximum of 5)
2. **Enter age:** The system uses age to determine if the ticket is for an **Adult** or **Children**
3. **VIP Seat:** Selecting VIP seats for upgrade
4. **Food Coupon:** For adding the food package
5. **Add more purchase:** You will be asked if you want to make another purchase (Y/N).

After completing the prompts, a receipt is generated containing:

- **ID NO:** A unique transaction identifier
- **Category & Quantity:** Breakdown of the tickets purchased.
- **Total Price:** The final sum of all tickets, seat upgrades, and food coupons.
- **Data Confirmation:** A message confirming that "All data saved to transactions.txt".

4.2 Family User Manual

The screenshot shows the movie ticketing system interface. The user has selected the 'Family' category. The system displays movie ticketing details for Family and Individual categories. For Family, the price is 70.00 per adult and 35.00 per child. For Individual, the price is 35.00 per adult and 18.00 per child. The user selects category F. It asks if they want a Family VIP seat (Y/No) and adds a food coupon (Y/No). The user enters N for both. It then asks if they want to make another purchase (Y/Yes | N/No), and the user enters N. The receipt shows ID NO: 0003, Category: Family, Quantity: 1, Amount: 70.00, and TOTAL PRICE: 70.00. A message at the bottom says "Thank you for your purchase! All data saved to transactions.txt".

FEATURES

If you select the Family(F) category:

- Ticket Category:** Enter 1 for Yes or 0 for No. This provides VIP seating for the entire family.
- Food coupon:** Enter 1 for Yes or 0 for No to add a food coupon for the family.
- Session Completion:** The system will ask if you wish to make another purchase. Enter N to finalize the transaction and view the family receipt.

Once the purchase is finalized, the system generates a receipt specifically for the family session:

- ID NO:** A unique number transaction identifier
- Total Price:** The final sum of the base ticket and any selected add-ons is displayed.
- Data Confirmation:** A message confirms "All data saved to transactions.txt" indicating the family record has been permanently logged.

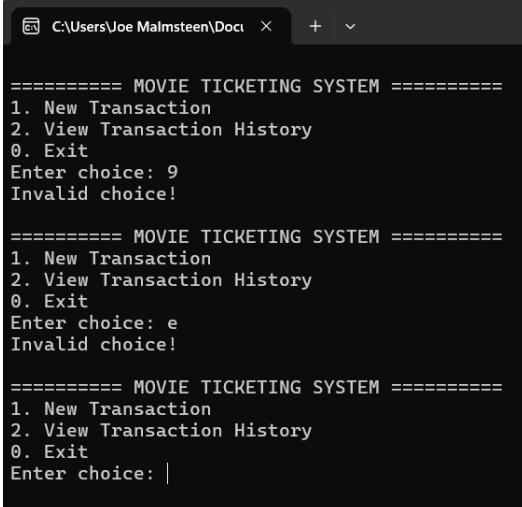
4.3 History Transaction and Termination

The screenshot shows the movie ticketing system interface. The user has selected option 2 to view transaction history. It displays a table of transaction history with columns: Receipt No., Category, Qty, VIP Seat, Food/Drink, and Amount(RM). It shows two entries: one for Receipt No. 0001 (Individual - Adult, Qty 1, VIP Seat 1, Food/Drink 1, Amount 20.00) and one for Receipt No. 0002 (Individual - Adult, Qty 1, VIP Seat 1, Food/Drink 1, Amount 20.00). Both have a 'Total:' row at the bottom. After viewing history, the user exits the application by selecting option 0, which terminates the program.

FEATURES

- Type [2] to view transaction history:** Access records of all previous ticket sales saved in the system
- System generates a detailed receipt containing:**
 - The Receipt No. assigned to the sale.
 - The Category of tickets sold (Individual - Adult/Children or Family Package).
 - The status of VIP Seats and Food/Drink upgrades (indicated by '1' for yes or '-' for no).
 - The Total Amount (RM) for individual entries and the overall receipt total
- Type [0] to Exit:** Close the application and terminate the program.

5.0 Output Screenshots

Output	Explanation
<p>1) Selecting the Menu with wrong inputs.</p>  <pre> ===== MOVIE TICKETING SYSTEM ===== 1. New Transaction 2. View Transaction History 0. Exit Enter choice: 9 Invalid choice! ===== MOVIE TICKETING SYSTEM ===== 1. New Transaction 2. View Transaction History 0. Exit Enter choice: e Invalid choice! ===== MOVIE TICKETING SYSTEM ===== 1. New Transaction 2. View Transaction History 0. Exit Enter choice: </pre>	<ul style="list-style-type: none"> This shows when the user enters the wrong input. First, the user entered the number '9', which is not an option on the menu and received an 'Invalid choice!' message. Then, the user asked to re-enter again, but the user entered the letter 'e' by mistake again which is also not an option on the menu.
<p>2) Selecting the Ticket Category with wrong inputs.</p>	<ul style="list-style-type: none"> When the user enter the letter 'y' by mistake, the program immediately caught the error and displayed an

```

C:\Users\Joe Malmsteen\Docx + v

===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: 1

=====
Movie Ticketing Data
=====

Category | Ticket(RM) |
Family   | 2 Adults    | 30.00      | 40.
          |              | per family | per
          | 2 Children   |            | per
          |              |            | per
Individual| Adults     | 10.00      | 12.
          |              | per person | per
          | Children    | 6.00       | 8.
          |              | per person | per

Select category (F - Family | I - Individual): y
Invalid category! Try again.

Select category (F - Family | I - Individual): |

```

error message and give the user another chance to enter the correct category.

- 3) The user selecting the VIP seat and food coupon with wrong inputs.

```

C:\Users\Joe Malmsteen\Docx + v

===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: 1

=====
Movie Ticketing Data
=====

Category | Ticket(RM) |
Family   | 2 Adults    | 30.00      | 40.
          |              | per family | per
          | 2 Children   |            | per
          |              |            | per
Individual| Adults     | 10.00      | 12.
          |              | per person | per
          | Children    | 6.00       | 8.
          |              | per person | per

Select category (F - Family | I - Individual): f
Purchase for Family VIP seat? (1- YES | 0- NO): 2
Invalid Input. Please enter again!
Purchase for Family VIP seat? (1- YES | 0- NO): |

```

- However, when asked about the VIP Seat upgrade, the user enters ‘2’, which triggers an error message because there is only ‘1’ or ‘0’ option.

```

C:\Users\Joe Malmsteen\Docu + ▾

===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: 1

===== Movie Ticketing Detail =====
Category | Ticket(RM) |
Family   |2 Adults    |30.00      |40.00
          |            |per family |per f
          |2 Children   |           |per f
Individual|Adults     |10.00      |12.00
          |            |per person |per p
          |Children    |6.00       |8.00
          |            |per person |per p

Select category (F - Family | I - Individual): f
Purchase for Family VIP seat? (1- YES | 0- NO): 1
Add Food Coupon for the family? (1 - YES | 0 - NO): 7
Invalid Input. Please enter again!
Add Food Coupon for the family? (1 - YES | 0 - NO): |

```

- Same goes when the user enters the same mistake by enters number ‘7’ which is out of option and an error message displayed and asks the user to select again until the value is true.

- 4) The user selecting the wrong inputs when trying to make another purchase.

```

C:\Users\Joe Malmsteen\Docu + ▾

1. New Transaction
2. View Transaction History
0. Exit
Enter choice: 1

===== Movie Ticketing Details =====
Category | Ticket(RM) | VIP
Family   |2 Adults    |30.00      |40.00
          |            |per family |per family
          |2 Children   |           |per family
Individual|Adults     |10.00      |12.00
          |            |per person |per person
          |Children    |6.00       |8.00
          |            |per person |per person

Select category (F - Family | I - Individual): f
Purchase for Family VIP seat? (1- YES | 0- NO): 1
Add Food Coupon for the family? (1 - YES | 0 - NO): 0
Do you want to make another purchase? (Y - Yes | N - No): j
Invalid input! Please enter Y or N.

Do you want to make another purchase? (Y - Yes | N - No): |

```

- The user is asked if they want to make another purchase and the user enters letter ‘j’ as thos does not match the required ‘Y’ and ‘N’ options.
- Then an error message is displayed and re-enter the user for a correct response.

- 5) The user enter the correct input (“1”).

```

C:\Users\User\Downloads\tic> + ^

===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: 1

=====
Movie Ticketing Details
=====
Category | Ticket(RM) | VIP seat(RM) | Food and drink(RM)
Family   | 30.00      | 40.00      | 20.00
          | per family | per family | per family
          | 2 Children |           |           |
Individual| 10.00      | 12.00      | 8 per pax
          | per person  | per person | per pax
          | Children   | 6.00       | 8.00      | 6 per pax
          | per person  | per person | per pax

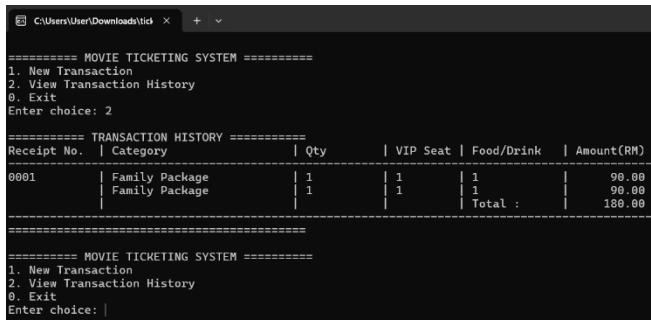
Select category (F - Family | I - Individual): F
Purchase for Family VIP seat? (1 - YES | 0 - NO): 1
Add Food Coupon for the Family? (1 - YES | 0 - NO): 1
Do you want to make another purchase? (Y - Yes | N - No): y
Select category (F - Family | I - Individual): f
Purchase for Family VIP seat? (1 - YES | 0 - NO): 1
Add Food Coupon for the Family? (1 - YES | 0 - NO): 1
Do you want to make another purchase? (Y - Yes | N - No): n
===== RECEIPT
ID NO: 0004
=====
Category | Quantity | Amount
Family   | 1         | RM 90.00
Family   | 1         | RM 90.00
TOTAL PRICE |           | RM 180.00
=====
Thank you for your purchase!
All data saved to transactions.txt
===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: |

```

- Users enter the right input, and the program runs successfully. The system will display the receipt for users to see, and all information will be saved inside “transaction.txt” file.
- The system will also ask user to make a choice again.

Receipt No.	Category	Qty	VIP Seat	Food/Drink	Amount(RM)
0001	Family Package	1	1	1	90.00
	Family Package	1	1	1	90.00
				Total :	180.00

- 6) The user enter the correct input (“2”).



```
===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: 2

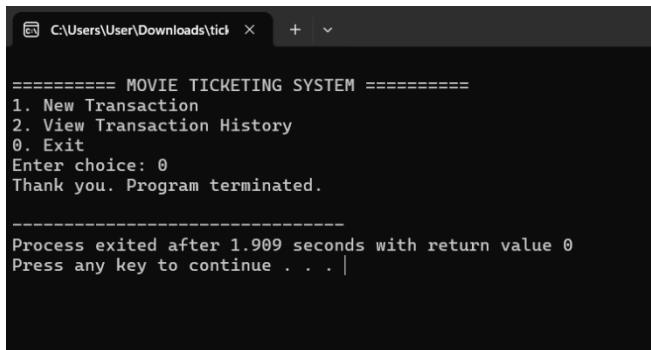
===== TRANSACTION HISTORY =====
Receipt No. | Category | Qty | VIP Seat | Food/Drink | Amount(RM)
0001 | Family Package | 1 | 1 | 1 | 98.00
          | Family Package | 1 | 1 | 1 | 98.00
                                         Total : 196.00
===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: |
```

- After the user entered the corrected input the user will see all transaction history.

- The transaction history will be include receipt number, category, quantity, vip seat or not ('1'=yes, '0'=no), Food/Drink or not ('1'=yes, '0'=no) and the total price of the ticket.

- The system will also ask user to make a choice again.

- 7) The user enter the correct input (“0”).



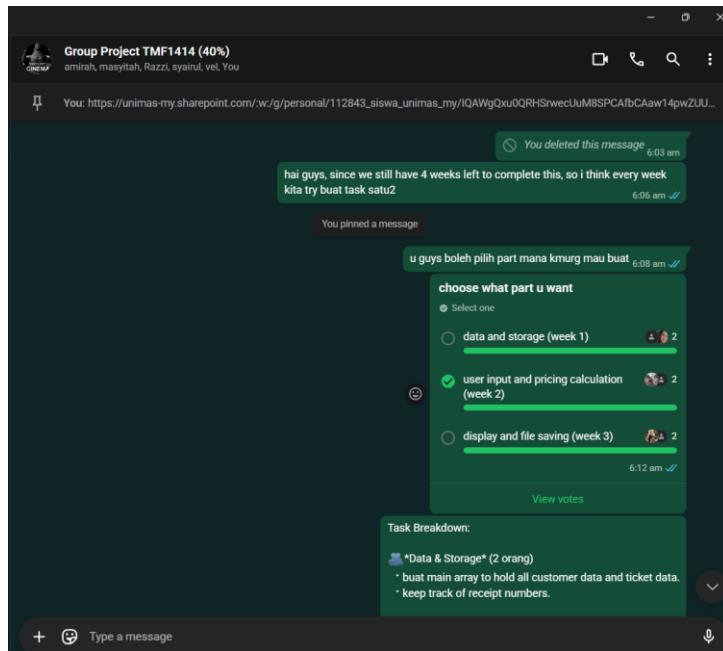
```
===== MOVIE TICKETING SYSTEM =====
1. New Transaction
2. View Transaction History
0. Exit
Enter choice: 0
Thank you. Program terminated.

Process exited after 1.909 seconds with return value 0
Press any key to continue . . . |
```

- The system will exit and display “Thank you. Program terminated”

6.0 Proof of Discussion

- Task distribution:



- Task progress:

