UNIVERSITI POLY-TECH MALAYSIA
UPTM

Name: NURSYASHA AMIRA BINTI SAIFUL NUR SITI ZAINAB BINTI AZMI NUR IZZATI AFIQAH BINTI KAHIZAR RADHIAH FARZANA BINTI RIDWAN		**Section : 02**
ID Number: **AM2311015156** **AM2311015158** **AM2311015176** **AM2311015163**		
Lecturer: NOORNAJWA BINTI MD AMIN		**Lab group / Tutorial group / Tutor (if applicable):**
Course and Course Code: DATA STRUCTURE SWC3344		**Submission Date:** **21/11/2024**
Assignment No. / Title: PROJECT VEHICLE SERVICE CENTER		**Extension & Late submission:** **Disallowed**
Assignment Type: **GROUP**	**% of Assignment Mark** **30%**	**Returning Date:**

Penalties:

1. 10% of the original mark will be deducted for every one week period after the submission date.
2. No work will be accepted after two weeks of the deadline.
3. If you were unable to submit the coursework on time due to extenuating circumstances you may be eligible for an extension.
4. Extension will not exceed one week.

Declaration: I/we the undersigned confirm that I/we have read and agree to abide by these regulations on plagiarism and cheating. I/we confirm that this piece of work is my/our own. I/we consent to appropriate storage of our work for checking to ensure that there is no plagiarism/ academic cheating.

Signature(s): *SYASHA* *ZAINAB* *IZZATI* *RADHIAH*

This section may be used for feedback or other information

Contents

INTRODUCTION

This project focuses on developing a Java-based Vehicle Service Management System to simulate the operations of a Coachella service centre. With the main goal of enhancing user experience and operational efficiency, the system is made to handle essential features like service requesting, customer and vehicle information, payment processing, and transaction logs.

The system's ability to control three service lanes, each of which is intended to maximize service flow, is one of its noteworthy features. Customers requesting fewer than three services are served by Lanes 1 and 2 alternately by batch of 5 customers, while those requesting more than three services are served by Lane 3. Through the use of a First-In-First-Out (FIFO) queue management technique, the system guarantees equitable and effective service request processing.

Numerous services, such as oil changes, brake repairs, battery replacements, engine tune-ups, and gearbox servicing, are also simulated by the system. Although the project includes a model of these services, the main emphasis is on coding their pricing, tracking, and booking features.

The project demonstrates how a digital solution can effectively organize data and automate procedures to streamline operations in a service center. By creating this system, the project highlights how programming concepts can be used practically to address real-world problems. It acts as an example of how technology can improve user satisfaction and service management in a simulated setting.

GitHub Link

https://github.com/syasha05/SWC3344-COECHELLASERVICECENTER/tree/main

IPO Analysis

INPUT

- Read 100 list of customers from input file named CustomerList.txt document.
- Coechella Service Centre will read how many customers are added.
- Messages Info will display how many customers added and limit are 100 customers.

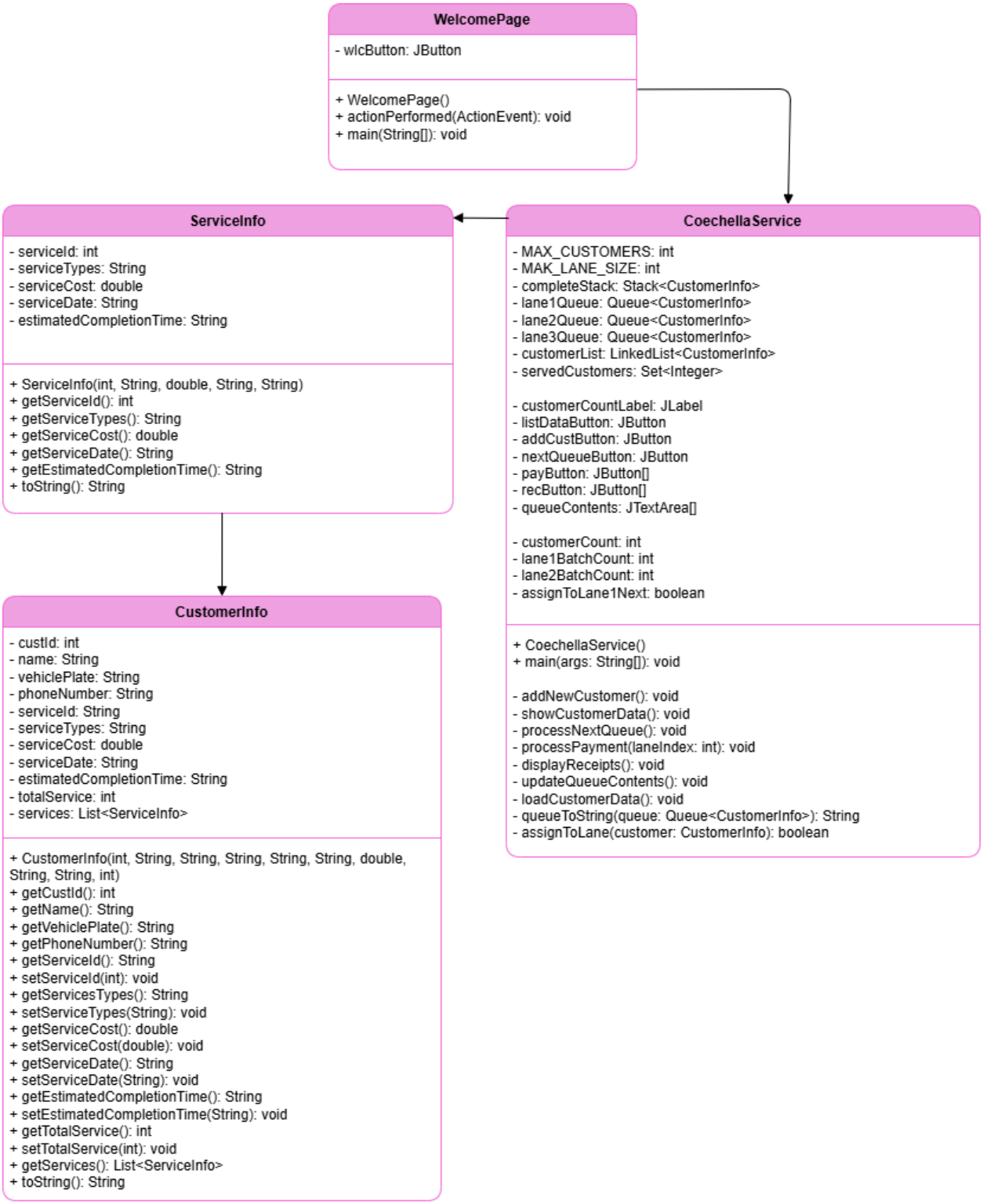
PROCESS

- Coechella Service Centre is required to assign customers fewer than 3 service types to lane 1 and 2, alternately by batch of 5 customers. And customers more than 3 service types to lane 3.
- This programme will read the data from the customerList.txt document and separate it into three lane based on the requested services.
- Lane one and two for customer who requested less than three services.
- Lane three is only for customer who requested more than three services
- The max limit for each counter is 5 customers
- By Click the button “Process Payment” the customers in lane (queue, First-In-First-Out) will be in the receipt for display and remove from the lane.

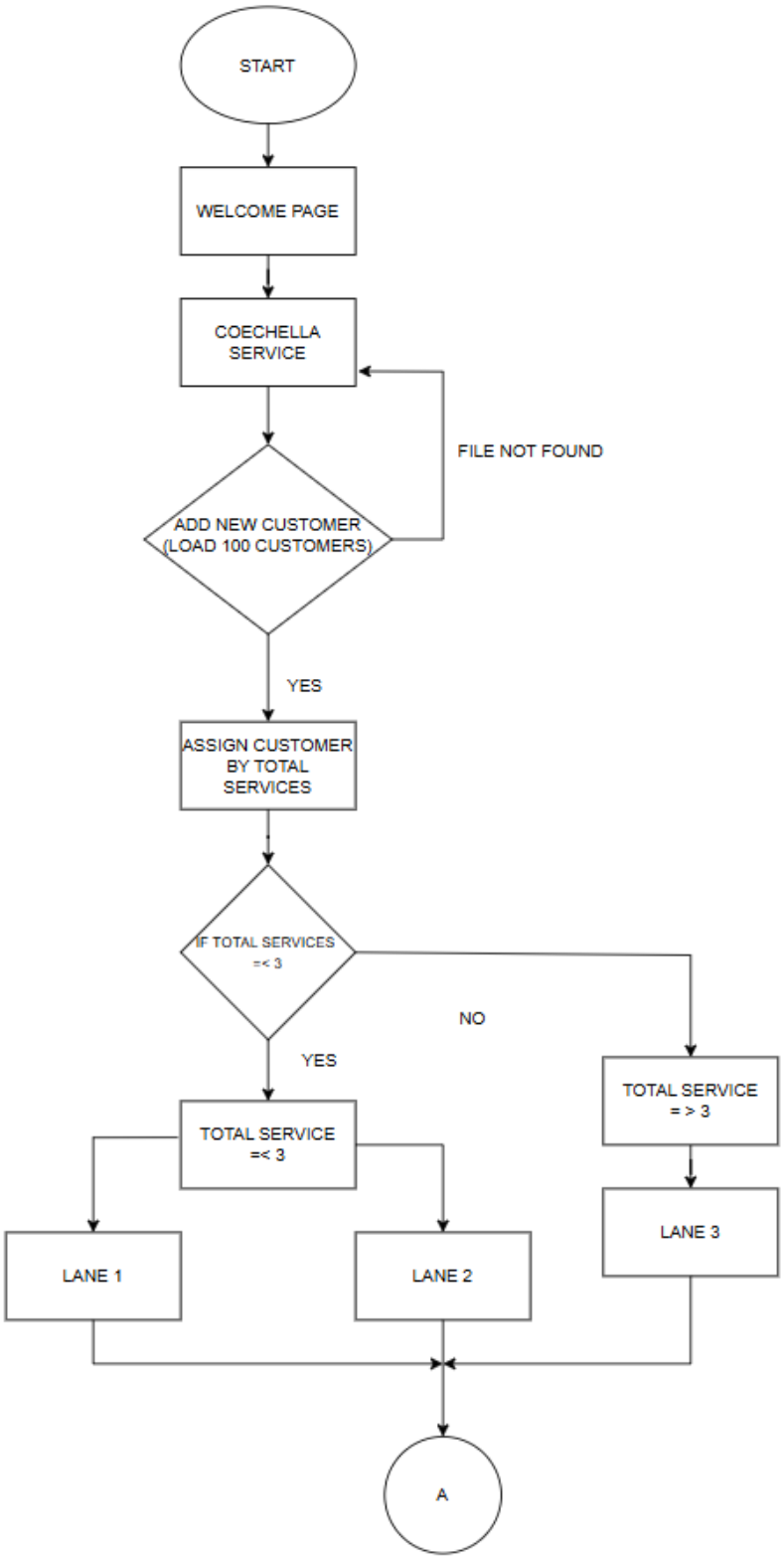
OUTPUT

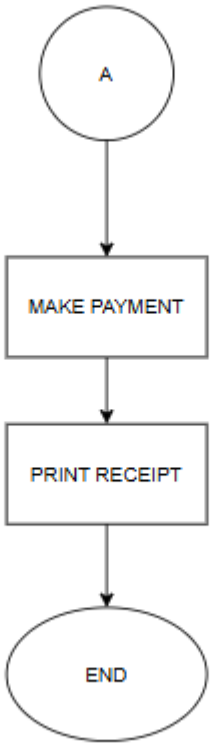
- The customer that already paid (get through the Process Payment) will be on display receipt.
- On the receipt, customers detail and service info such as customer Id, Name, Vehicle Plate Number, Phone Number, Service Types, Service Date and Service Cost are on display.

Design UML Class Diagram



Flow Chart


START
WELCOME PAGE
COECHELLA SERVICE
FILE NOT FOUND
ADD NEW CUSTOMER (LOAD 100 CUSTOMERS)
YES
ASSIGN CUSTOMER BY TOTAL SERVICES
IF TOTAL SERVICES =< 3
NO
YES
TOTAL SERVICE =< 3
TOTAL SERVICE = > 3
LANE 1
LANE 2
LANE 3
A


A
MAKE PAYMENT
PRINT RECEIPT
END

Sample Data

customerList.TXT (100customers)

```
1024,Siti,WEY0010,017-3321107,1,04,Battery Service,500.00,2024-01-04,1hours
1069,Johan,JXZ1357,011-9011443,2,49,Fluid Checks And Replacements|Gearbox Service|,1380.00,2024-02-24,4hours 45minutes
1105,Meerqeen,VW99,017-8123456,5,85,Tire Rotation and Alignment|Suspension Service|Brake Service|Battery Service|Air Conditioning(A/C) Service|,2020.00,2024-05-05,8hours
1055,Patricia,PWR7890,019-0087123,2,35,Suspension Service|Air Conditioning(A/C) Service|,1100.00,2024-02-10,5hours
1099,Laila,LBA2348,019-9911200,4,79,Brake Service|Engine Tune-Up|Battery Service|Air Conditioning(A/C) Service|,1500.00,2024-04-14,6hours
1042,Rayyan,KDA5682,011-9901144,1,22,Air Conditioning(A/C) Service,250.00,2024-01-22,2hours
1063,Icaa,VMX2713,011-1204583,2,43,Fluid Checks and Replacement|Inspection and Emissions Testing|,280.00,2024-02-18,1hour 15minutes
1031,Mikail,WWJ4439,017-2237789,1,11,Fluid Checks and Replacement,180.00,2024-01-11,45minutes
1097,Ayda,CFH4382,019-76852345,4,77,Brake Service|Engine Tune-Up|Battery Service|Air Conditioning(A/C) Service|,1500.00,2024-04-12,6hours
1043,Anis,JTR4371,017-2157457,1,23,Oil Change,120.00,2024-01-23,45minutes
1096,Syaheera,CEV708,013-0952345,4,76,Oil Change|Battery Service|Engine Tune-Up|Gearbox Service|,2270.00,2024-04-11,7 hours 45minutes
1104,Fakhrul,LRT1505,019-8765332,5,84,Brake Service|Oil Change|Tire Rotation and Alignment|Air Conditioning(A/C) Service|Inspection and Emissions Testing|,890.00,2024-05-04,5hours 15minutes
1062,Miera,WTG5482,010-3362034,2,42,Oil Change|Brake Service|,420.00,2024-02-17,1hour 45minutes
1077,Shamim,WAD2901,010-4675298,3,57,Fluid Checks and Replacements|Suspension Service|Tire Rotation and Alignment|,1150.00,2024-03-07,4hours 45minutes
1032,Syaazni,JJK1288,011-2366142,1,12,Tire Rotation And Alignment,120.00,2024-01-12,1hours
1094,Nur,WAE3456,013-7654321,4,74,Fluid Checks and Replacements|Battery Service|Air Conditioning (A/C) Service|Gearbox Service|,2150.00,2024-04-09,8hours 30minutes
1110,fimie,M3,010-9988776,5,90,Battery Service|Suspension Service|Air Conditioning(A/C) Service|Tire Rotation and Alignment|Engine Tune-Up|,2170.00,2024-05-10,9hours
1111,Zara,WAF3412,012-9876543,5,91,Oil Change|Brake Service|Fluid Checks and Replacements|Engine Tune-Up|Inspection and Emissions Testing|,1450.00,2024-05-11,5hours 30minutes
1091,Alya,WAG5520,011-3364789,4,71,Oil Change|Brake Service|Battery Service|Engine Tune-Up|,1370.00,2024-04-06,5hours 45minutes
1083,Syed,SKD3314,014-0093780,3,63,Suspension Service|Inspection And Emissions Testing|Engine Tune-Up|,1400.00,2024-03-13,5hours 30minutes
1021,Ridwan,R1579,012-4521776,1,01,Brake Service,300.00,2024-01-01,1hours
1061,Kamal,PFN909,017-4562312,2,41,Gearbox Service|Brake Service|,1500.00,2024-02-16,5hours
1088,Nina,JWY6321,019-3033345,4,68,Suspension Service|Oil Change|Engine Tune-Up|Gearbox Service|,2620.00,2024-03-03,9hours 45minutes
1022,Syasha,SHA072,014-3201099,1,02,Oil Change,120.00,2024-01-02,45minutes
1076,Aminah,WAB3029,011-5566789,3,56,Oil Change|Brake Service|Battery Service|,920.00,2024-03-06,2hours 45minutes
1056,Tasha,PGH2683,011-4122843,2,36,Suspension Service|Gearbox Service|,2050.00,2024-02-11,7hours
1117,Khairul,KDN4538,011-04624525,5,97,Battery Service|Oil Change|Gearbox Service|Brake Service|Engine Tune-Up|,2570.00,2024-05-17,8hours 45minutes
1060,Wardina,WPD6671,011-9012708,2,40,Suspension Service|Gearbox Service|,2050.00,2024-02-15,7hours
1093,Imran,WAD4567,010-1234567,4,73,Brake Service|Engine Tune-Up|Tire Rotation and Alignment|Suspension Service|,2050.00,8hours,2024-04-08,15minutes
1108,Shukri,WW999,012-4567890,5,88,Tire Rotation and Alignment|Inspection and Emissions Testing|Gearbox Service|Engine Tune-Up|Brake Service|,2170.00,2024-05-08,8 hours 30 minutes
1072,Daniel,WYT4152,018-9238899,3,52,Air Conditioning(A/C) Service|Engine Tune-Up|Inspection and Emissions Testing|,800.00,2024-03-02,4hours 30minutes
1058,Johan,JXZ1357,011-9011443,2,38,Fluid Checks And Replacements|Gearbox Service|,1380.00,2024-02-13,4hours 45minutes
1066,Patrick,PPT1212,019-0102211,2,46,Suspension Service|Air Conditioning(A/C) Service|,1100.00,2024-02-21,5hours
1095,Kayla,KYG1122,012-0223411,4,75,Inspections and Emissions Testing|Engine Tune-Up|Gearbox Service|Battery Service|,2250.00,2024-04-10,5hours 30minutes
1092,Najib,WAB6733,012-2457890,4,72,Fluid Checks and Replacements|Suspension Service|Air Conditioning (A/C) Service|Inspection and Emissions Testing|,1880.00,2024-04-07,7hours 15minutes
1095,Kayla,KYG1122,012-0223411,4,75,Inspections and Emissions Testing|Engine Tune-Up|Gearbox Service|Battery Service|,2250.00,2024-04-10,5hours 30minutes
1078,Reza,WAE6633,012-5589341,3,58,Engine Tune-Up|Inspection and Emissions Testing|Air Conditioning Service|,1000.00,2024-03-08,4hours 30minutes
1090,Nadya,KDT6321,018-8883329,4,70,Engine Tune-Up|Air Conditioning(A/C) Service|Brake Service|Fluid Checks and replacements|,1180.00,2024-04-05,5hours 45minutes
1114,Hafiz,WAE6781,010-8765432,5,94,Oil Change|Battery Service|Suspension Service|Air Conditioning(A/C) Service|Gearbox Service|,2860.00,2024-05-14,11hours 15minutes
1115,Waiz,WQS1130,019-011211,5,95,Battery Service|Suspension Service|Gearbox Service|Oil Change|Engine Tune-Up|,3120.00,2024-05-15,10hours
1079,Hasni,WAF7890,013-6782394,3,59,Brake Service|Fluid Checks and Replacements|Gearbox Service|,1600.00,2024-03-09,6hours 15minutes
1094,Siti,WAE3456,013-7654321,4,74,Fluid Checks and Replacements|Battery Service|Air Conditioning (A/C) Service|Gearbox Service|,2150.00,2024-04-09,8hours 30minutes
1085,Syeerah,SGF1132,013-9033641,3,65,Brake Service|Gearbox Service|Engine Tune-Up|,1270.00,2024-03-15,7hours
1107,Nassar,WV33,018-3456789,5,87,Suspension Service|Tire Rotation and Alignment|Brake Service|Oil Change|Air Conditioning(A/C) Service|,1640.00,2024-05-07,7hours 45minutes
1045,Farhan,JTH2353,012-7307342,1,25,Brake Service,300.00,2024-01-25,1hours
```

```
1027,Rabi,WRQ556,011-8765432,1,07,Engine Tune-Up,450.00,2024-01-07,2hours
1091,Alya,WAG5520,011-3364789,4,71,Oil Change|Brake Service|Battery Service|Engine Tune-Up|,1370.00,2024-04-06,5hours 45minutes
1037,Haikal,KBA5475,013-0908812,1,17,Brake Service,300.00,2024-01-17,1hours
1038,Geivano,VKA5678,013-0997413,1,18,Inspection And Emissions Testing,100.00,2024-01-18,30minutes
1116,Amir,ARK5472,019-12962432,5,96,Air Conditioning(A/C) Service|OilChange|Tire Rotation And Alignment|Fluid Checks and replacements|Battery Service|,1170.00,2024-05-16,5hours 30minutes
1101,Nadlyn,JHT2555,017-3355667,5,81,Fluid Checks and replacements|Brake Service|Air Conditioning(A/C) Service|Inspection and Emissions Testing|Tire Rotation and Alignment|,950.00,2024-05-01,5 hours 15 minutes
1062,Miera,WTG5482,010-3362034,2,42,Oil Change|Brake Service|,420.00,2024-02-17,1hour 45minutes
1098,Thalia,TDC2311,014-0023611,4,78,Fluid Checks And Replacements|Suspension Service|Gearbox Service|Oil Change|,2350.00,2024-04-13,9hours
1023,Izzati,WUU0585,011-2309988,1,03,Air Conditioning(A/C) Service,250.00,2024-01-03,2hours
1083,Syed,SKD3314,014-0093780,3,63,Suspension Service|Inspection And Emissions Testing|Engine Tune-Up|,1400.00,2024-03-13,5hours 30minutes
1025,Zainab,WKU5647,019-5483290,1,05,Engine Tune-Up,450.00,2024-01-05,2hours
1030,Khai,JKL11,019-7766554,1,10,Oil Change,120.00,2024-01-10,45minutes
1036,Syahmin,BMW1245,011-2034556,1,16,Oil Change,120.00,2024-01-16,45minutes
1039,Jay,JTR9012,014-2201244,1,19,Tire Rotation And Alignment,120.00,2024-01-19,1hours
1100,Veena,VDE3691,011-0231422,4,80,Suspension Service|Gearbox Service|Oil Change|Tire Rotation And Alignment|,2290.00,2024-04-15,8hours 45minutes
1060,Wardina,WPD6671,011-9012708,2,40,Suspension Service|Gearbox Service|,2050.00,2024-02-15,7hours
1064,Ihsan,WGA8352,012-1470396,2,44,Brake Service|Battery Service|,800.00,2024-02-19,2hours
1040,Badrol,BMG3455,011-9901144,1,20,Air Conditioning(A/C) Service,250.00,2024-01-20,2hours
1041,Jannah,AVE7562,014-2201244,1,21,Tire Rotation And Alignment,120.00,2024-01-21,1hours
1026,Aina,WND5741,013-7654321,1,06,Inspection and Emissions Testing,100.00,2024-01-06,30minutes
1044,Adam,RM3935,012-6428132,1,24,Inspection and Emissions Testing,100.00,2024-01-24,30minutes
1046,Radhiah,WSF4659,018-4099212,2,26,Fluid Checks and Replacement|Inspection and Emissions Testing|,280.00,2024-02-01,1hours 15minutes
1047,Farzana,WKI4125,013-3204567,2,27,Suspension Service|Tire Rotation and Alignment|,970.00,2024-02-02,4hours
1071,Amin,WKI6352,013-9822200,3,51,Oil Change|Inspection and Emissions Testing|Engine Tune-Up|,670.00,2024-03-01,3hours 15minutes
1048,Syira,WIU8899,016-9871234,2,28,Gearbox Service|Brake Service|,1500.00,2024-02-03,5hours
1063,Icaa,VMX2713,011-1204583,2,43,Fluid Checks and Replacement|Inspection and Emissions Testing|,280.00,2024-02-18,1hour 15minutes
1108,Shukri,WW999,012-4567890,5,88,Tire Rotation and Alignment|Inspection and Emissions Testing|Gearbox Service|Engine Tune-Up|Brake Service|,2170.00,2024-05-08,8 hours 30 minutes
1069,Johan,JXZ1357,011-9011443,2,49,Fluid Checks And Replacements|Gearbox Service|,1380.00,2024-02-24,4hours 45minutes
1070,Nissa,NSX2244,017-0013422,2,50,Inspection And Emissions Testing|Suspension Service|,950.00,2024-02-25,3hours 30minutes
1052,Syafiqah,WAD8743,010-4523678,2,32,Oil Change|Brake Service|,420.00,2024-02-07,1hour 45minutes
1064,Ihsan,WGA8352,012-1470396,2,44,Brake Service|Battery Service|,800.00,2024-02-19,2hours
1049,Amira,WKC2805,012-9883321,2,29,Inspection and Emissions Testing|Battery Service|,600.00,2024-02-04,1hour 30minutes
1051,Kamelia,WAF9917,010-3365613,2,31,Air Conditioning(A/C) Service|Gearbox Service|,1450.00,2024-02-06,2hours
1072,Daniel,WYT4152,018-9238899,3,52,Air Conditioning(A/C) Service|Engine Tune-Up|Inspection and Emissions Testing|,800.00,2024-03-02,4hours 30minutes
1053,Hakim,WAX9932,012-9876543,33,2,Fluid Checks and Replacements|Engine Tune-Up|,630.00,2024-02-08,2hours 45minutes
1112,Anwar,WAG2358,013-4567890,5,92,Battery Service|Suspension Service|Tire Rotation and Alignment|Air Conditioning(A/C) Service|Brake Service|,2024-05-12,2460.00,9hours
1113,Nadia,WAD8234,011-6543210,5,93,Fluid Checks and Replacements|Brake Service|Engine Tune-Up|Inspection and Emissions Testing|Air Conditioning (A/C) Service|,2200.00,2024-05-13,8hours 15minutes
1054,Farid,WAS5548,013-1122334,2,34,Air Conditioning Service|Suspension Service|,1100.00,2024-02-09,5hours
1055,Patricia,PWR7890,019-0087123,2,35,Suspension Service|Air Conditioning(A/C) Service|,1100.00,2024-02-10,5hours
1104,Fakhrul,LRT1505,019-8765332,5,84,Brake Service|Oil Change|Tire Rotation and Alignment|Air Conditioning(A/C) Service|Inspection and Emissions Testing|,890.00,2024-05-04,5hours 15minutes
1056,Tasha,PGH2683,011-4122843,2,36,Suspension Service|Gearbox Service|,2050.00,2024-02-11,7hours
1061,Kamal,PFN909,017-4562312,2,41,Gearbox Service|Brake Service|,1500.00,2024-02-16,5hours
1057,TanCheng,TAK2486,012-9077343,2,37,Brake Service|Battery Service|,800.00,2024-02-12,2hours
1080,Kamila,KNM9011,019-0023411,3,60,Suspension Service|Brake Service|Tire Rotation And Alignment|,1270.00,2024-03-10,6hours
1059,Nissa,NSX2244,017-0013422,2,39,Inspection and Emissions Testing|Suspension Service|,950.00,2024-02-14,3hours 30minutes
1065,Redza,PKR7093,012-5677090,2,45,Air Conditioning(A/C) Service|Gearbox Service|,1450.00,2024-02-20,2hours
1067,Tasha,PGH2683,011-4128453,2,47,Suspension Service|Gearbox Service|,2050.00,2024-02-22,7hours
1102,Fikri,LPS1254,011-6677899,5,82,Suspension Service|Gearbox Service|Engine Tune-up|Tire Rotation and Alignment|Battery Service|,2720.00,2024-05-02,11hours
1068,TanCheng,TAK2486,012-9077343,2,48,Brake Service|Battery Service|,800.00,2024-02-23,2hours
1070,Nissa,NSX2244,017-0013422,2,50,Inspection And Emissions Testing|Suspension Service|,950.00,2024-02-25,3hours 30minutes
1093,Imran,WAD4567,010-1234567,4,73,Brake Service|Engine Tune-Up|Tire Rotation and Alignment|Suspension Service|,2050.00,8hours,2024-04-08,15minutes
1071,Amin,WKI6352,013-9822200,3,51,Oil Change|Inspection and Emissions Testing|Engine Tune-Up|,670.00,2024-03-01,3hours 15minutes
1087,Hasby,JYF5412,018-9982233,4,67,Inspections and Emissions Testing|Engine Tune-Up|Gearbox Service|Battery Service|,2250.00,2024-04-02,5hours 30minutes
1074,Aqil,WLP174,017-7783445,3,54,Air Conditioning(A/C) Service|Engine Tune-Up|Inspection and Emissions Testing|,800.00,2024-03-04,4hours 30minutes
1075,Hassan,WAD6238,012-6523349,3,55,Brake Service|Gearbox Service|Engine Tune-Up|,1270.00,2024-03-05,7hours
1076,Aminah,WAB3029,011-5566789,3,56,Oil Change|Brake Service|Battery Service|,920.00,2024-03-06,2hours 45minutes
1081,Waniy,KTR3621,019-9455935,3,61,Gearbox Service|Tire Rotation And Alignment|Engine Tune-Up|,1770.00,2024-03-11,7hours
1082,Wafiy,PEQ7063,014-12347432,3,62,Brake Service|Inspection And Emissions Testing|Oil Change|,520.00,2024-03-12,2hours 15minutes
1087,Hasby,JYF5412,018-9982233,4,67,Inspections and Emissions Testing|Engine Tune-Up|Gearbox Service|Battery Service|,2250.00,2024-04-02,5hours 30minutes
```

```
1090,Nadya,KDT6321,018-8883329,4,70,Engine Tune-Up|Air Conditioning(A/C) Service|Brake Service|Fluid Checks and replacements|,1180.00,2024-04-05,5hours 45minutes
1092,Najib,WAB6733,012-2457890,4,72,Fluid Checks and Replacements|Suspension Service|Air Conditioning (A/C) Service|Inspection and Emissions Testing|,1880.00,2024-04-07,7hours 15minutes
1096,Syaheera,CEV708,013-0952345,4,76,Oil Change|Battery Service|Engine Tune-Up|Gearbox Service|,2270.00,2024-04-11,7 hours 45minutes
1077,Shamim,WAD2901,010-4675298,3,57,Fluid Checks and Replacements|Suspension Service|Tire Rotation and Alignment|,1150.00,2024-03-07,4hours 45minutes
1098,Thalia,TDC2311,014-0023611,4,78,Fluid Checks And Replacements|Suspension Service|Gearbox Service|Oil Change|,2350.00,2024-04-13,9hours
1099,Laila,LBA2348,019-9911200,4,79,Brake Service|Engine Tune-Up|Battery Service|Air Conditioning(A/C) Service|,1500.00,2024-04-14,6hours
1101,Nadlyn,JHT2555,017-3355667,5,81,Fluid Checks and replacements|Brake Service|Air Conditioning(A/C) Service|Inspection and Emissions Testing|Tire Rotation and Alignment|,950.00,2024-05-01,5 hours 15 minutes
1103,Amar,LPQ6999,012-6767543,5,83,Gearbox Service|Oil Change|Engine Tune-Up|Brake Service|Tire Rotation and Alignment|,2190.00,2024-05-03,8hours 45minutes
1106,Nadhir,KO72,011-5678912,5,86,Fluid Checks and replacements|Oil Change|Air Conditioning(A/C) Service|Tire Rotation and Alignment|Gearbox Service|,1870.00,2024-05-06,8hours 30minutes
1109,Syafiq,J1,016-9876543,5,89,Fluid Checks and replacements|Air Conditioning(A/C) Service|Tire Rotation and Alignment|Battery Service|Engine Tune-Up|,1500.00,2024-05-09,6hours 45minutes
1115,Waiz,WQS1130,019-011211,5,95,Battery Service|Suspension Service|Gearbox Service|Oil Change|Engine Tune-Up|,3120.00,2024-05-15,10hours
1082,Wafiy,PEQ7063,014-12347432,3,62,Brake Service|Inspection And Emissions Testing|Oil Change|,520.00,2024-03-12,2hours 15minutes
1117,Khairul,KDN4538,011-04624525,5,97,Battery Service|Oil Change|Gearbox Service|Brake Service|Engine Tune-Up|,2570.00,2024-05-17,8hours 45minutes
1105,Meerqeen,VW99,017-8123456,5,85,Tire Rotation and Alignment|Suspension Service|Brake Service|Battery Service|Air Conditioning(A/C) Service|,2020.00,2024-05-05,8hours
1118,Amran,AXJ9900,019-5109811,5,98,Gearbox Service|Suspension Service|Tire Rotation And Alignment|Engine Tune-Up|Battery Service|,3120.00,2024-05-18,11hours
1119,Redza,RZG5561,011-0101332,5,99,Fluid Checks and replacements|Oil Change|Gearbox Service|Air Conditioning(A/C) Service|Engine Tune-Up|,2200.00,2024-05-19,9hours 30minutes
1120,Purnama,PLK8080,012-0990112,5,100,Oil Change|Brake Service|Battery Service|Air Conditioning(A/C) Service|Gearbox Service|,2370.00,2024-05-20,8hours 45minutes
1057,TanCheng,TAK2486,012-9077343,2,37,Brake Service|Battery Service|,800.00,2024-02-12,2hours
1112,Anwar,WAG2358,013-4567890,5,92,Battery Service|Suspension Service|Tire Rotation and Alignment|Air Conditioning(A/C) Service|Brake Service|,2024-05-12,2460.00,9hours
1113,Nadia,WADB234,011-6543210,5,93,Fluid Checks and Replacements|Brake Service|Engine Tune-Up|Inspection and Emissions Testing|Air Conditioning (A/C) Service|,2200.00,2024-05-13,8hours 15minutes
1058,Johan,JXZ1357,011-9011443,2,38,Fluid Checks And Replacements|Gearbox Service|,1380.00,2024-02-13,4hours 45minutes
1059,Nissa,NSX2244,017-0013422,2,39,Inspection and Emissions Testing|Suspension Service|,950.00,2024-02-14,3hours 30minutes
1065,Redza,PKR7093,012-56770909,2,45,Air Conditioning(A/C) Service|Gearbox Service|,1450.00,2024-02-20,2hours
1073,Farhan,WQR8965,014-5532245,3,53,Battery Service|Engine Tune-Up|Air Conditioning(A/C) Service|,1000.00,2024-03-03,5hours
1074,Aqil,WLP174,017-7783445,3,54,Air Conditioning(A/C) Service|Engine Tune-Up|Inspection and Emissions Testing|,800.00,2024-03-04,4hours 30minutes
1066,Patrick,PPT1212,019-0102211,2,46,Suspension Service|Air Conditioning(A/C) Service|,1100.00,2024-02-21,5hours
1084,Mara,MLN7610,012-00349018,3,64,Engine Tune-Up|Oil Change|Fluid Checks And Replacements|,750.00,2024-03-14,3hours
1085,Syeerah,SGF1132,013-9033641,3,65,Brake Service|Gearbox Service|Engine Tune-Up|,1270.00,2024-03-15,7hours
1089,Zul,KWD5421,012-5543321,4,69,Oil Change|Fluid Checks and replacements|Air Conditioning(A/C) Service|Tire Rotation and Alignment|,670.00,2024-04-04,4hours 30minutes
1097,Ayda,CFH4382,019-76852345,4,77,Brake Service|Engine Tune-Up|Battery Service|Air Conditioning(A/C) Service|,1500.00,2024-04-12,6hours
1103,Amar,LPQ6999,012-6767543,5,83,Gearbox Service|Oil Change|Engine Tune-Up|Brake Service|Tire Rotation and Alignment|,2190.00,2024-05-03,8hours 45minutes
1028,Odish,WON777,016-0187633,1,08,Suspension Service,850.00,2024-01-08,3hours
1029,Lisa,LLY711,018-4433221,1,09,Air Conditioning(A/C) Service,250.00,2024-01-09,2hours
1050,Sobella,WSS6351,019-7743320,2,30,Inspection and Emissions Testing|Engine Tune-Up|,550.00,2024-02-05,2hours 30minutes
1106,Nadhir,KO72,011-5678912,5,86,Fluid Checks and replacements|Oil Change|Air Conditioning(A/C) Service|Tire Rotation and Alignment|Gearbox Service|,1870.00,2024-05-06,8hours 30minutes
1084,Mara,MLN7610,012-00349018,3,64,Engine Tune-Up|Oil Change|Fluid Checks And Replacements|,750.00,2024-03-14,3hours
1035,Roy,WYY1234,011-2203456,1,15,Fluid Checks And Replacements,180.00,2024-01-15,45minutes
1067,Tasha,PGH2683,011-4128453,2,47,Suspension Service|Gearbox Service|,2050.00,2024-02-22,7hours
1107,Nassar,WV33,018-3456789,5,87,Suspension Service|Tire Rotation and Alignment|Brake Service|Oil Change|Air Conditioning(A/C) Service|,1640.00,2024-05-07,7hours 45minutes
1068,TanCheng,TAK2486,012-9077343,2,48,Brake Service|Battery Service|,800.00,2024-02-23,2hours
1109,Syafiq,J1,016-9876543,5,89,Fluid Checks and replacements|Air Conditioning(A/C) Service|Tire Rotation and Alignment|Battery Service|Engine Tune-Up|,1500.00,2024-05-09,6hours 45minutes
1086,Adam,JOR6953,011-2199332,4,66,Air Conditioning(A/C) Service|Gearbox Service|Battery Service|Brake Service|,2250.00,2024-04-01,8hours
1081,Waniy,KTR3621,019-9455935,3,61,Gearbox Service|Tire Rotation And Alignment|Engine Tune-Up|,1770.00,2024-03-11,7hours
1111,Zara,WAF3412,012-9876543,5,91,Oil Change|Brake Service|Fluid Checks and Replacements|Engine Tune-Up|Inspection and Emissions Testing|,1450.00,2024-05-11,5hours 30minutes
1089,Zul,KWD5421,012-5543321,4,69,Oil Change|Fluid Checks and replacements|Air Conditioning(A/C) Service|Tire Rotation and Alignment|,670.00,2024-04-04,4hours 30minutes
1034,Farah,KWF3389,013-2788422,1,14,Brake Service,300.00,2024-01-14,1hours
1086,Adam,JOR6953,011-2199332,4,66,Air Conditioning(A/C) Service|Gearbox Service|Battery Service|Brake Service|,2250.00,2024-04-01,8hours
1114,Hafiz,WAE6781,010-8765432,5,94,Oil Change|Battery Service|Suspension Service|Air Conditioning(A/C) Service|Gearbox Service|,2860.00,2024-05-14,11hours 15minutes
1073,Farhan,WQR8965,014-5532245,3,53,Battery Service|Engine Tune-Up|Air Conditioning(A/C) Service|,1000.00,2024-03-03,5hours
1110,fimie,M3,010-9988776,5,90,Battery Service|Suspension Service|Air Conditioning(A/C) Service|Tire Rotation and Alignment|Engine Tune-Up|,2170.00,2024-05-10,9hours
1080,Kamila,KNM9011,019-0023411,3,60,Suspension Service|Brake Service|Tire Rotation And Alignment|,1270.00,2024-03-10,6hours
1033,Hanif,HHZ7749,011-5528796,1,13,Battery Service,500.00,2024-01-13,1hours
1078,Reza,WAE6633,012-5589341,3,58,Engine Tune-Up|Inspection and Emissions Testing|Air Conditioning Service|,1000.00,2024-03-08,4hours 30minutes
1102,Fikri,LPS1254,011-6677899,5,82,Suspension Service|Gearbox Service|Engine Tune-up|Tire Rotation and Alignment|Battery Service|,2720.00,2024-05-02,11hours
1088,Nina,JWY6321,019-3033345,4,68,Suspension Service|Oil Change|Engine Tune-Up|Gearbox Service|,2620.00,2024-03-03,9hours 45minutes
1116,Amir,ARK5472,019-12962432,5,96,Air Conditioning(A/C) Service|OilChange|Tire Rotation And Alignment|Fluid Checks and replacements|Battery Service|,1170.00,2024-05-16,5hours 30minutes
1075,Hassan,WAD6238,012-6523349,3,55,Brake Service|Gearbox Service|Engine Tune-Up|,1270.00,2024-03-05,7hours
1100,Veena,VDE3691,011-0231422,4,80,Suspension Service|Gearbox Service|Oil Change|Tire Rotation And Alignment|,2290.00,2024-04-15,8hours 45minutes
1079,Hasni,WAF7890,013-6782394,3,59,Brake Service|Fluid Checks and Replacements|Gearbox Service|,1600.00,2024-03-09,6hours 15minutes
```

Java Code

welcomePage Class

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;


public class WelcomePage extends JFrame implements ActionListener {

    private JButton wlcButton;


    public WelcomePage() {

        // Set up the frame

        setTitle("Coechella Entrance");

        setSize(800, 710);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setLocationRelativeTo(null);


        // Create two labels for the title with multiline effect

        JLabel titleLabel1 = new JLabel("COCHELLA SERVICE CENTER", SwingConstants.CENTER);

        titleLabel1.setFont(new Font("Arial", Font.BOLD, 26));

        JLabel titleLabel2 = new JLabel("Founded by Syasha, Zainab, Izzati and Radhiah", SwingConstants.CENTER);

        titleLabel2.setFont(new Font("Arial", Font.PLAIN, 20));
```

```
// Create a panel for the title and add the title labels to it
JPanel titlePanel = new JPanel(new GridLayout(2, 1));
titlePanel.add(titleLabel1);
titlePanel.add(titleLabel2);

// Create an ImageIcon logo
ImageIcon logo = new ImageIcon("chl1.jpg");
JLabel logoLabel = new JLabel(logo, SwingConstants.CENTER);

// Create a tagline label
JLabel taglineLabel = new JLabel("WE PROVIDE, WE ASSIST, WE HAPPY!", SwingConstants.CENTER);
taglineLabel.setFont(new Font("Arial", Font.ITALIC, 18));

// Create the welcome button
wlcButton = new JButton("WELCOME");
wlcButton.addActionListener(this);

// Create the main panel with BorderLayout
JPanel mainPanel = new JPanel(new BorderLayout());
mainPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
mainPanel.add(titlePanel, BorderLayout.NORTH);
```

```
    mainPanel.add(logoLabel, BorderLayout.CENTER);

    // Create a panel for the tagline and button, and add them to it
    JPanel footerPanel = new JPanel(new GridLayout(2, 1, 0, 10));
    footerPanel.add(taglineLabel);
    footerPanel.add(wlcButton);

    // Add the footer panel to the main panel's south region
    mainPanel.add(footerPanel, BorderLayout.SOUTH);



    // Add the main panel to the frame
    add(mainPanel);
    setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == wlcButton) {
        dispose();
        new CoechellaService();
    }
}
```

```
    public static void main(String[] args) {
        new WelcomePage();
    }
}
```

CoechellaService Class

```
// Import required libraries for the program
import javax.swing.*; // For graphical user interface components
import java.awt.*; // For GUI layouts and colors
import java.io.*; // For reading and writing files
import java.util.LinkedList; // For LinkedList data structure
import java.util.Queue; // For Queue interface
import java.util.Stack; // For Stack data structure
import java.util.Set; // For Set interface
import java.util.stream.Collectors; // For stream operations
import java.util.HashSet; // For HashSet implementation

public class CoechellaService extends JFrame {
    // Constants for system limits
    private static final int MAX_CUSTOMERS = 100; // Maximum number of customers allowed
    private static final int MAX_LANE_SIZE = 5; // Maximum size for each service lane

    // Data structures for managing customers
```

```
    private Stack<CustomerInfo> completeStack = new Stack<>(); // Stack for completed payments

    private Queue<CustomerInfo> lane1Queue = new LinkedList<>(); // Queue for Lane 1

    private Queue<CustomerInfo> lane2Queue = new LinkedList<>(); // Queue for Lane 2

    private Queue<CustomerInfo> lane3Queue = new LinkedList<>(); // Queue for Lane 3

    private LinkedList<CustomerInfo> customerList = new LinkedList<>(); // List of customers

    private Set<Integer> servedCustomers = new HashSet<>(); // Set to track served customers by their ID


    // UI Components

    private JLabel customerCountLabel; // Label to display the number of customers

    private JButton listDataButton, addCustButton, nextQueueButton; // Buttons for various actions

    private JButton[] payButton, recButton; // Buttons for payment and receipt actions per lane

    private JTextArea[] queueContents; // Text areas to display the contents of each lane


    // Variables for lane assignment batching

    private int customerCount = 0; // Counter for total customers

    private int lane1BatchCount = 0, lane2BatchCount = 0; // Batch counters for lanes

    private boolean assignToLane1Next = true; // Flag to alternate lane assignment


    // Constructor for the CoachellaService GUI

    public CoechellaService() {

        // JFrame properties
```

```
        setTitle("Coachella Service Centre"); // Window title
        setSize(1050, 730); // Window dimensions
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Exit application on close
        setLocationRelativeTo(null); // Center window on screen

        // Main panel for the application layout
        JPanel mainPanel = new JPanel(new BorderLayout());
        mainPanel.setBackground(new Color(255, 182, 193)); // Light pink background
        add(mainPanel); // Add panel to the frame

        // Top panel with customer count label and buttons
        JPanel topPanel = new JPanel(new BorderLayout());
        topPanel.setBackground(new Color(255, 182, 193)); // Same background color as main panel
        customerCountLabel = new JLabel("Number of Customers: 0", SwingConstants.CENTER); // Initial customer count
        customerCountLabel.setFont(new Font("Arial", Font.BOLD, 21)); // Set font style and size
        customerCountLabel.setForeground(Color.WHITE); // White text color
        topPanel.add(customerCountLabel, BorderLayout.NORTH); // Add label to the top of the top panel

        // Button panel to hold action buttons
        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 10)); // Flow layout with spacing
```

```
        buttonPanel.setBackground(new Color(255, 182, 193)); // Match background color

        // Create and style buttons
        addCustButton = new JButton("Add New Customer");
        listDataButton = new JButton("List of Data");
        nextQueueButton = new JButton("Next Queue");

        Color buttonColor = new Color(255, 105, 180); // Hot pink for buttons
        JButton[] buttons = {addCustButton, listDataButton, nextQueueButton};
        for (JButton button : buttons) {
            button.setBackground(buttonColor); // Set button background color
            button.setForeground(Color.WHITE); // Set button text color
            buttonPanel.add(button); // Add button to the button panel
        }

        topPanel.add(buttonPanel, BorderLayout.CENTER); // Add button panel to the center of top panel
        mainPanel.add(topPanel, BorderLayout.NORTH); // Add top panel to the north of main panel

        // Center panel for displaying lanes and their contents
        JPanel centerPanel = new JPanel(new GridLayout(1, 3, 10, 10)); // Grid layout for three lanes
        centerPanel.setBackground(new Color(240, 240, 240)); // Neutral background
        queueContents = new JTextArea[3]; // Array for text areas to display queue contents
```

```
payButton = new JButton[3]; // Array for payment buttons

recButton = new JButton[3]; // Array for receipt buttons


// Create panels for each lane

for (int i = 0; i < 3; i++) {

    JPanel counterPanel = new JPanel(new BorderLayout(10, 10)); // Border layout for lane panels

    counterPanel.setBackground(new Color(255, 182, 193)); // Match main panel background

    counterPanel.setBorder(BorderFactory.createLineBorder(new Color(255, 105, 180), 3)); // Add border


    JLabel counterLabel = new JLabel("Counter " + (i + 1), JLabel.CENTER); // Counter label

    counterLabel.setFont(new Font("Arial", Font.BOLD, 16)); // Font styling

    counterLabel.setOpaque(true); // Make label background visible

    counterLabel.setBackground(buttonColor); // Hot pink background

    counterLabel.setForeground(Color.WHITE); // White text color

    counterPanel.add(counterLabel, BorderLayout.NORTH); // Add label to top of counter panel


    queueContents[i] = new JTextArea(); // Text area for lane contents

    queueContents[i].setEditable(false); // Make text area non-editable

    queueContents[i].setFont(new Font("Monospaced", Font.PLAIN, 16)); // Monospaced font for alignment

    JScrollPane scrollPane = new JScrollPane(queueContents[i]); // Scroll pane for text area
```

```
        counterPanel.add(scrollPane, BorderLayout.CENTER); // Add scroll pane to center of counter panel


        // Panel for lane buttons
        JPanel buttonPanelCounter = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 5)); // Flow layout for buttons
        buttonPanelCounter.setBackground(new Color(255, 182, 193)); // Match background color


        payButton[i] = new JButton("Process Payment"); // Button for processing payments
        recButton[i] = new JButton("Print Receipt"); // Button for printing receipts


        JButton[] laneButtons = {payButton[i], recButton[i]}; // Array of lane buttons
        for (JButton btn : laneButtons) {
            btn.setBackground(buttonColor); // Set button background color
            btn.setForeground(Color.WHITE); // Set button text color
            buttonPanelCounter.add(btn); // Add button to lane's button panel
        }


        // Add action listeners for buttons
        int laneIndex = i; // Track lane index for button actions
        payButton[i].addActionListener(e -> processPayment(laneIndex)); // Process payment action
        recButton[i].addActionListener(e -> displayReceipts()); // Display receipt action
```

```
        counterPanel.add(buttonPanelCounter, BorderLayout.SOUTH); // Add button panel to bottom of counter panel
        centerPanel.add(counterPanel); // Add counter panel to center panel
    }

    mainPanel.add(centerPanel, BorderLayout.CENTER); // Add center panel to main panel

    // Add action listeners for top-level buttons
    addCustButton.addActionListener(e -> addNewCustomer()); // Add new customer action
    listDataButton.addActionListener(e -> showCustomerData()); // Show customer data action
    nextQueueButton.addActionListener(e -> processNextQueue()); // Process next queue action

    setVisible(true); // Make the frame visible
  }

  private void addNewCustomer() {
    // Check if customer limit is already reached
    if (customerCount >= MAX_CUSTOMERS) {
        JOptionPane.showMessageDialog(this, "Customer limit reached! No new customers can be added.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
```

```
    }

    File file = new File("CustomerList.txt");
    if (!file.exists()) {
        JOptionPane.showMessageDialog(this, "CustomerList.txt file not found!", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    int customersAdded = 0;

    try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
        String line;

        // Read customers from file until the list is full or file ends
        while ((line = reader.readLine()) != null) {
            if (customerCount >= MAX_CUSTOMERS) {
                break;
            }

            // Split and validate the data from the line
            String[] data = line.split(",");
            if (data.length < 10) {
```

```
                JOptionPane.showMessageDialog(this, "Invalid data format in CustomerList.txt! Skipping entry.", "Warning", JOptionPane.WARNING_MESSAGE);
                continue;
            }

            try {
                // Parse customer details and add to the list
                CustomerInfo customer = new CustomerInfo(
                    Integer.parseInt(data[0].trim()),  // Customer ID
                    data[1].trim(),                    // Name
                    data[2].trim(),                    // Vehicle Plate Number
                    data[3].trim(),                    // Phone Number
                    Integer.parseInt(data[4].trim()), // Total Services
                    Integer.parseInt(data[5].trim()), // Service ID
                    data[6].trim(),                    // Service Types
                    Double.parseDouble(data[7].trim()), // Service Cost
                    data[8].trim(),                    // Service Date
                    data[9].trim()                     // Estimated Completion Time
                );

                customerList.add(customer);
                customersAdded++;
                customerCount++;
```

```
            } catch (NumberFormatException ex) {
                // Simply skip the invalid entry and continue
                continue;
            }
        }

        // Update the customer count label
        customerCountLabel.setText("Number of Customers: " + customerCount);
        // Show the total number of customers added
        JOptionPane.showMessageDialog(this, customersAdded + " customers successfully added to the waiting list.", "Info", JOptionPane.INFORMATION_MESSAGE);

    } catch (IOException e) {
        JOptionPane.showMessageDialog(this, "Error reading CustomerList.txt!", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

    private void showCustomerData() {
    JFrame dataFrame = new JFrame("Customer and Service Info");
    dataFrame.setSize(480, 640);
    dataFrame.setLocationRelativeTo(this);
    dataFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
        JTextArea textArea = new JTextArea();
        textArea.setEditable(false);
        textArea.setFont(new Font("Monospaced", Font.PLAIN, 16));

        // ScrollPane for the JTextArea
        JScrollPane scrollPane = new JScrollPane(textArea);
        dataFrame.add(scrollPane);

        File file = new File("CustomerList.txt");
        if (!file.exists()) {
            JOptionPane.showMessageDialog(this, "CustomerList.txt file not found!", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        // Read the file and display data
        try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
            String line;
            StringBuilder dataBuilder = new StringBuilder();

            while ((line = reader.readLine()) != null) {
                String[] data = line.split(",");
```

```
// Ensure there's enough data (at least 8 fields)
if (data.length >= 10) {
    dataBuilder.append("*****************************************\n")
    .append("Customer ID: ").append(data[0].trim()).append("\n")
    .append("Name: ").append(data[1].trim()).append("\n")
    .append("Vehicle Plate Number: ").append(data[2].trim()).append("\n")
    .append("Phone Number: ").append(data[3].trim()).append("\n")
    .append("Total Services: ").append(data[4].trim()).append("\n")
    .append("Service ID: ").append(data[5].trim()).append("\n")
    .append("Service Cost: ").append(data[7].trim()).append("\n")
    .append("Service Date: ").append(data[8].trim()).append("\n")
    .append("Estimate Completion Time: ").append(data[9].trim()).append("\n");

    // Handle multiple service types (splitting by "|")
    String[] serviceTypes = data[6].trim().split("\\|");
    dataBuilder.append("Service Type(s): \n");
    for (String serviceType : serviceTypes) {
        dataBuilder.append("  - ").append(serviceType.trim()).append("\n");
    }

    dataBuilder.append("*****************************************\n\n");
```

```
            }
        }

        textArea.setText(dataBuilder.toString());
    } catch (IOException e) {
        JOptionPane.showMessageDialog(this, "Error reading file!", "Error", JOptionPane.ERROR_MESSAGE);
    }

    dataFrame.setVisible(true);
}

private void processNextQueue() {
    if (customerList.isEmpty()) {
        JOptionPane.showMessageDialog(this, "No more customers in the waiting list!", "Info", JOptionPane.INFORMATION_MESSAGE);
        return;
    }

    int assignedToLane1 = 0;
    int assignedToLane2 = 0;
    int assignedToLane3 = 0;
```

```
boolean alternateLane1 = true;  // This will help alternate between Lane 1 and Lane 2 for customers with fewer than 3 services


// Loop through the customer list and assign to lanes
while (!customerList.isEmpty()) {
    // Get the next customer
    CustomerInfo nextCustomer = customerList.poll(); // Get the next customer
    if (nextCustomer == null) continue;


    boolean assigned = false;
    if (nextCustomer.getTotalservice() <= 3) {
        // Assign customers with fewer than 3 services alternately to Lane 1 or Lane 2
        if (lane1Queue.size() < MAX_LANE_SIZE && assignedToLane1 < 5) {
            lane1Queue.add(nextCustomer);
            assignedToLane1++;
            assigned = true;
        } else if (lane2Queue.size() < MAX_LANE_SIZE && assignedToLane2 < 5) {
            lane2Queue.add(nextCustomer);
            assignedToLane2++;
            assigned = true;
        } else {
            // If Lane 1 and Lane 2 are full, put the customer back in the queue
            customerList.addLast(nextCustomer);
```

```
            break;  // Stop processing further for now, we will try again next time
        }
    } else {
        // Customers with more than 3 services go to Lane 3
        if (lane3Queue.size() < MAX_LANE_SIZE && assignedToLane3 < 5) {
            lane3Queue.add(nextCustomer);
            assignedToLane3++;
            assigned = true;
        } else {
            // If Lane 3 is full, put the customer back in the queue
            customerList.addLast(nextCustomer);
            break;  // Stop processing for now
        }
    }

    // If the customer couldn't be assigned to a lane, put them back in the customer list
    if (!assigned) {
        customerList.addLast(nextCustomer); // Re-add the customer to the waiting list
        break;  // Stop processing for now, we will try again next time
    }
}
```

```
    // Update the customer count after processing
    customerCount -= (assignedToLane1 + assignedToLane2 + assignedToLane3);
    customerCountLabel.setText("Number of Customers: " + customerCount);

    // Display a summary of assignments
    JOptionPane.showMessageDialog(
        this,
        "Batch processed:\n"
        + "Lane 1: " + assignedToLane1 + " customers\n"
        + "Lane 2: " + assignedToLane2 + " customers\n"
        + "Lane 3: " + assignedToLane3 + " customers",
        "Batch Assignment",
        JOptionPane.INFORMATION_MESSAGE
    );

    // Refresh the UI to display updated queues
    updateQueueContents();
}

private void processPayment(int laneIndex) {
    Queue<CustomerInfo> targetQueue = switch (laneIndex) {
            case 0 -> lane1Queue;
```

```
            case 1 -> lane2Queue;
            case 2 -> lane3Queue;
            default -> null;
        };

    if (targetQueue == null || targetQueue.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Queue is empty or invalid!", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    CustomerInfo servedCustomer = targetQueue.poll(); // Serve the customer
    completeStack.push(servedCustomer); // Add to completed stack
    servedCustomers.add(servedCustomer.hashCode()); // Track served customer
    updateQueueContents();
    JOptionPane.showMessageDialog(this, "Payment processed for \n" + servedCustomer.toString(), "Payment", JOptionPane.INFORMATION_MESSAGE);
  }

  private boolean assignToLane(CustomerInfo customer) {
    // If the customer has 3 or more services, assign them directly to Lane 3
    if (customer.getTotalservice() >= 3) {
        if (lane3Queue.size() < MAX_LANE_SIZE) {
```

```
            lane3Queue.add(customer);
            return true;
        }
        // If Lane 3 is full, return false
        return false;
    }

    // If the customer has less than 3 services, assign them to either Lane 1 or Lane 2
    if (lane1Queue.size() < MAX_LANE_SIZE) {
        lane1Queue.add(customer);
        return true;
    } else if (lane2Queue.size() < MAX_LANE_SIZE) {
        lane2Queue.add(customer);
        return true;
    }

    // If both Lane 1 and Lane 2 are full, return false (no space available)
    return false;
}

private void loadCustomerData() {
    // Check if customerList is empty
```

```
        if (customerList.isEmpty()) {
            JOptionPane.showMessageDialog(this, "No data added. Please add new customers first.", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }
        File file = new File("CustomerList.txt");
        if (!file.exists()) {
            JOptionPane.showMessageDialog(this, "File not found!", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] data = line.split(",");
                if (data.length >= 10) {  // Ensure there are at least 4 elements for custId, custName, vehiclePlate, and phoneNumber
                    CustomerInfo customer = new CustomerInfo(
                            Integer.parseInt(data[0].trim()),  // Customer ID
                            data[1].trim(),                    // Name
                            data[2].trim(),                    // Vehicle Plate Number
                            data[3].trim(),                    // Phone Number
```

```
                    Integer.parseInt(data[4].trim()), // Total Services
                    Integer.parseInt(data[5].trim()), // Service ID
                    data[6].trim(),                   // Service Types
                    Double.parseDouble(data[7].trim()), // Service Cost
                    data[8].trim(),                   // Service Date
                    data[9].trim()                    // Estimated Completion Time
                );
                customerList.add(customer);
                assignToLane(customer);
            }
        }
        updateQueueContents();
    } catch (IOException | NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Error reading file!", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
// Define the updateQueueContents method
private void updateQueueContents() {

queueContents[0].setText(lane1Queue.stream().map(CustomerInfo::toString).collect(Collectors.j
oining("\n\n")));
```

```
queueContents[1].setText(lane2Queue.stream().map(CustomerInfo::toString).collect(Collectors.joining("\n\n")));

queueContents[2].setText(lane3Queue.stream().map(CustomerInfo::toString).collect(Collectors.joining("\n\n")));
    }

    private void displayReceipts() {
        if (completeStack.isEmpty()) {
            JOptionPane.showMessageDialog(this, "No completed payments to show.", "Info", JOptionPane.INFORMATION_MESSAGE);
            return;
        }

        // Build receipt information
        StringBuilder receiptBuilder = new StringBuilder("**** Completed Payments ****\n\n");
        for (CustomerInfo customer : completeStack) {
            receiptBuilder.append("Service Date: ").append(customer.getServiceDate()).append("\n")
            .append("Customer ID: ").append(customer.getCustId()).append("\n")
            .append("Name: ").append(customer.getName()).append("\n")
            .append("Vehicle Plate Number: ").append(customer.getVehiclePlate()).append("\n")
            .append("Phone Number: ").append(customer.getPhoneNumber()).append("\n")
```

```
        .append("Service ID: ").append(customer.getServiceId()).append("\n");


        // Handle multiple service types (splitting by "|")
        String[] serviceTypes = customer.getServiceTypes() != null &&
!customer.getServiceTypes().isEmpty()
            ? customer.getServiceTypes().trim().split("\\|") // Split by "|"
            : new String[0]; // Default empty array if service types are missing


        // Display service types
        if (serviceTypes.length > 0) {
            receiptBuilder.append("Service Type(s): \n");
            for (String serviceType : serviceTypes) {
                receiptBuilder.append("  - ").append(serviceType.trim()).append("\n");
            }
        } else {
            receiptBuilder.append("Service Type(s): Not specified\n");
        }


        receiptBuilder.append("Service Cost: RM
").append(customer.getServiceCost()).append("\n")
        .append("-----------------------------------------------\n");
    }
```

```
    // Display receipt in a new JFrame or JTextArea
    JTextArea receiptArea = new JTextArea(receiptBuilder.toString());
    receiptArea.setEditable(false);
    receiptArea.setFont(new Font("Monospaced", Font.PLAIN, 16));
    JScrollPane scrollPane = new JScrollPane(receiptArea);

    JFrame receiptFrame = new JFrame("Receipts");
    receiptFrame.setSize(500, 640);
    receiptFrame.add(scrollPane);
    receiptFrame.setLocationRelativeTo(this);
    receiptFrame.setVisible(true);
}

private String queueToString(Queue<CustomerInfo> queue) {
    StringBuilder sb = new StringBuilder();
    for (CustomerInfo customer : queue) {
        sb.append(customer.toString()).append("\n\n");  // Adding double newline for separation
    }
    return sb.toString();
}

public static void main(String[] args) {
```

```
        SwingUtilities.invokeLater(CoechellaService::new);
    }
}
```

CustomerInfo Class

```
import java.util.ArrayList;
import java.util.List;


// This class represents customer information
public class CustomerInfo {
    // Unique customer identifier
    private int custId;
    // Name of the customer
    private String name;
    // Vehicle plate number of the customer
    private String vehiclePlate;
    // Phone number of the customer
    private String phoneNumber;

    // Track the count of services for the customer
    private int serviceId;

    private String serviceTypes;
```

```
    private double serviceCost;

    private String serviceDate;

    private String estimatedCompletionTime;

    private int totalService;
    // Track the assigned lane for the customer
    // List of services made by the customer
    private List<ServiceInfo> services;

    // Constructor to initialize customer information
    public CustomerInfo(int custId, String name, String vehiclePlate, String phoneNumber, int totalService) {
        this.custId = custId;
        this.name = name;
        this.vehiclePlate = vehiclePlate;
        this.phoneNumber = phoneNumber;
        this.totalService = totalService;
        this.services = new ArrayList<>();
    }
    // Add overloaded constructor to handle all fields
```

```
    public CustomerInfo(int custId, String name, String vehiclePlate, String phoneNumber, int totalService, int serviceId,
    String serviceTypes, double serviceCost, String serviceDate, String estimatedCompletionTime) {
        this.custId = custId;
        this.name = name;
        this.vehiclePlate = vehiclePlate;
        this.phoneNumber = phoneNumber;
        this.totalService = totalService;
        this.serviceId = serviceId;
        this.serviceTypes = serviceTypes;
        this.serviceCost = serviceCost;
        this.serviceDate = serviceDate;
        this.estimatedCompletionTime = estimatedCompletionTime;
    }

    // Getter methods
    public int getCustId() {
        return custId;
    }

    public String getName() {
        return name;
```

```
}

public String getVehiclePlate() {
    return vehiclePlate;
}

public String getPhoneNumber() {
    return phoneNumber;
}

public int getServiceId() {
    return serviceId;
}

public void setServiceId(int serviceId) {
    this.serviceId = serviceId;
}

public String getServiceTypes() {
    return serviceTypes;
}
```

```
public void setServiceTypes(String serviceTypes) {
    this.serviceTypes = serviceTypes;
}

public double getServiceCost() {
    return serviceCost;
}

public void setServiceCost(double serviceCost) {
    this.serviceCost = serviceCost;
}

public String getServiceDate() {
    return serviceDate;
}

public void setServiceDate(String serviceDate) {
    this.serviceDate = serviceDate;
}

public String getEstimatedCompletionTime() {
    return estimatedCompletionTime;
```

```
}

public void setEstimatedCompletionTime(String estimatedCompletionTime) {
    this.estimatedCompletionTime = estimatedCompletionTime;
}

public int getTotalservice() {
    return totalService;
}

public void addService(ServiceInfo service) {
    services.add(service);
}

public void setTotalService(int totalService)
{
    this.totalService = totalService;
}

// Services getter
public List<ServiceInfo> getServices() {
    return services;
```

```
    }



    // Overriding toString() to format the customer info
    @Override
    public String toString() {
        return custId + " | " + name + "\n" +
        "Vehicle Plate Number: " + vehiclePlate + "\n" +
        "Phone Number: " + phoneNumber + "\n" + "Total Service:" + totalService;
    }
}
```

ServiceInfo Class

```
public class ServiceInfo {
    // Unique identifier for the ticket
    private int serviceId;
    // Name of the service type
    private String serviceTypes;
    // Cost for the service made
    private double serviceCost;
    // Date when the service was completed
    private String serviceDate;
    // Time estimation of service completed
    private String estimatedCompletionTime;
```

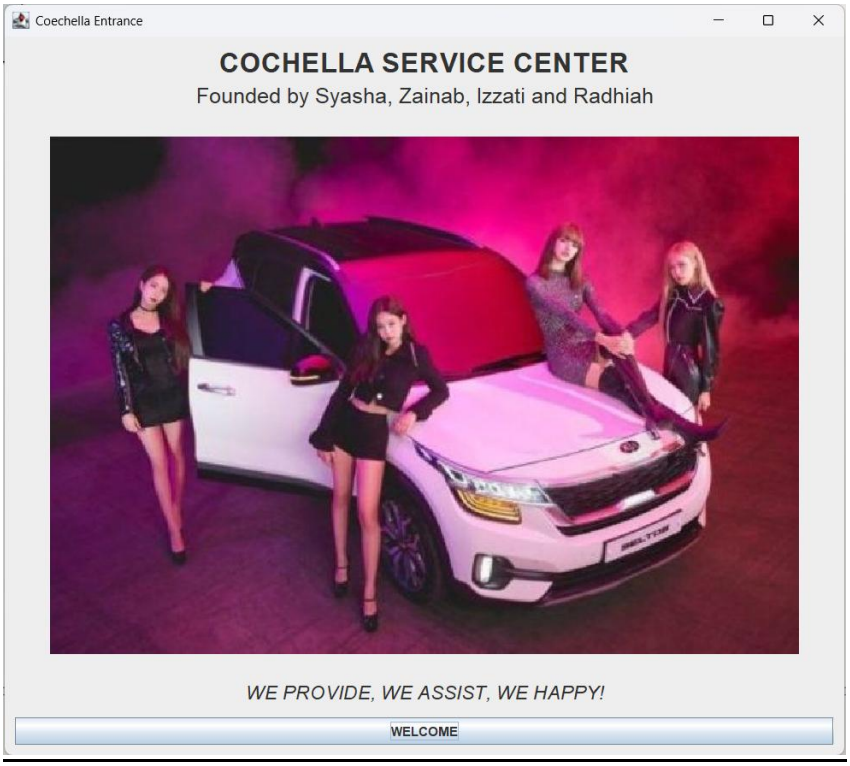
```
    // Constructor to initialize all fields in service information
    public ServiceInfo(int serviceId, String serviceTypes, double serviceCost, String serviceDate, String estimatedCompletionTime) {
        this.serviceId = serviceId;
        this.serviceTypes = serviceTypes;
        this.serviceCost = serviceCost;
        this.serviceDate = serviceDate;
        this.estimatedCompletionTime = estimatedCompletionTime;
    }

    // Return the unique Id of the service
    public int getServiceId() {
        return serviceId;
    }

    // Return the type of the service
    public String getServiceTypes() {
        return serviceTypes;
    }

    // Return the total cost of service
    public double getServiceCost() {
```

```
    return serviceCost;
}

// Return the date the service was completed
public String getServiceDate() {
    return serviceDate;
}

// Return the estimated completion time
public String getEstimatedCompletionTime() {
    return estimatedCompletionTime;
}

// Override the toString method to return a custom string representation
@Override
public String toString() {
    return "Service ID: " + serviceId + "\n" +
           "Service Type: " + serviceTypes + "\n" +
           "Service Cost: RM " + serviceCost + "\n" +
           "Service Date: " + serviceDate + "\n" +
           "Estimated Completion Time: " + estimatedCompletionTime;
}
```

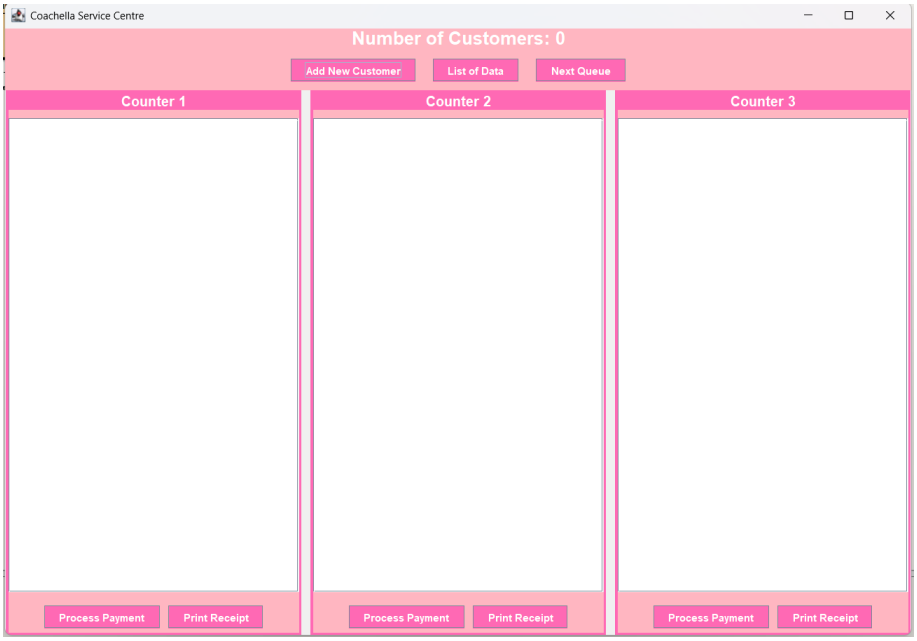
```
}
```

Sample Input/Output

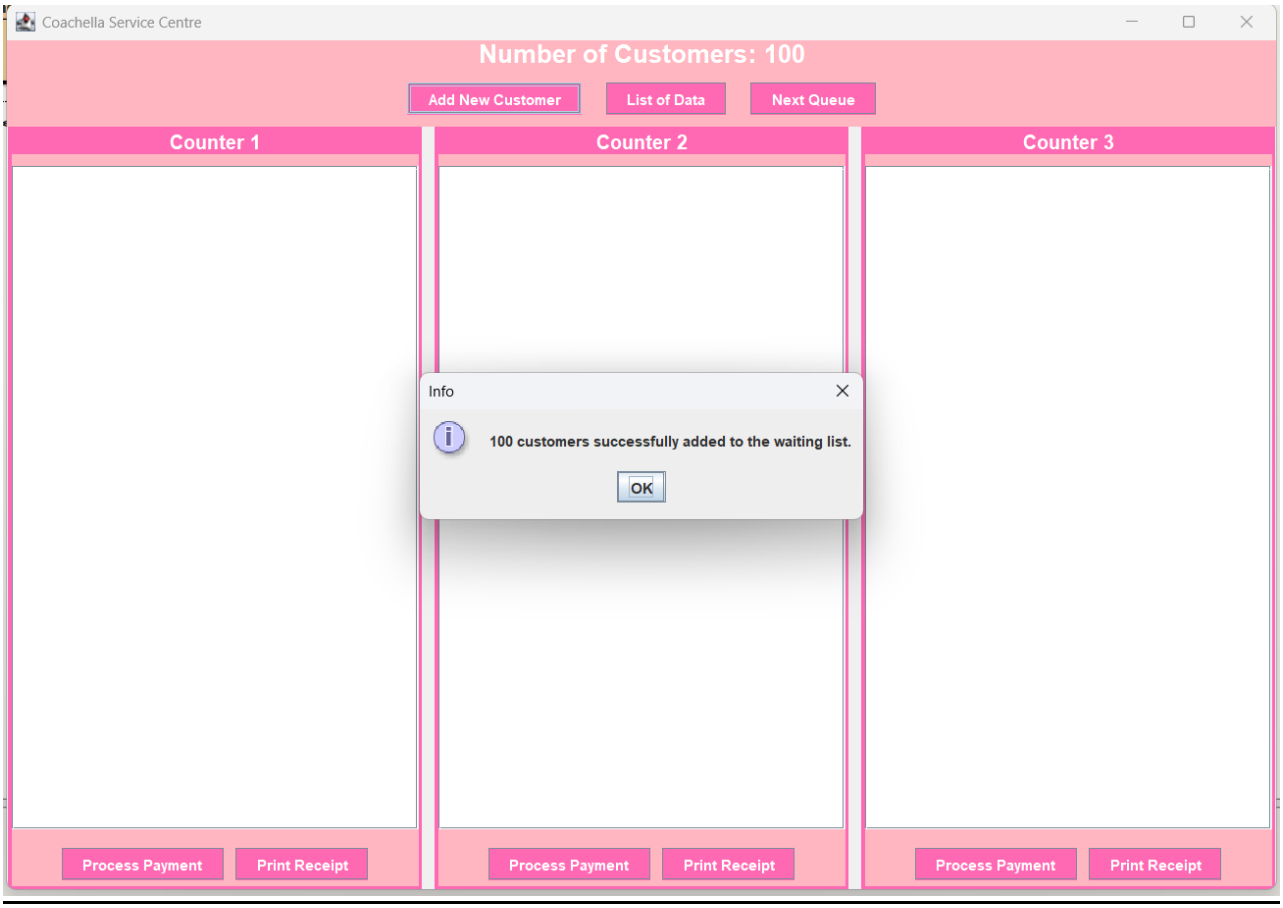
- The user will be greeted by Coechella Entrance (Welcome Page)



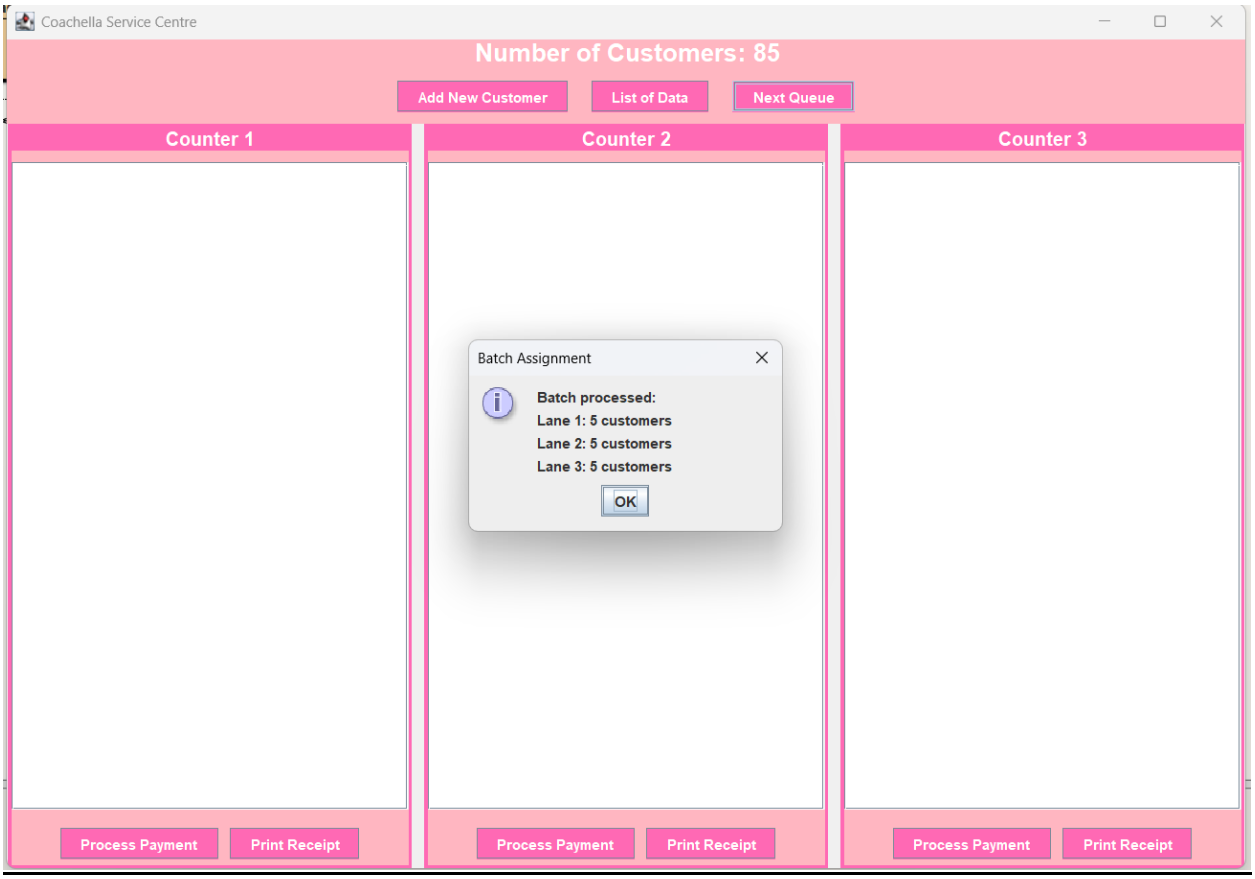
- The Number of Customers is 0 because there are no data added.
- All the button cannot function due to no data.



- By Click the button ‘ Add New Customer ‘ ,first, the system will read the textFile and then info messages pop out and inform how many customer successfully added.
- Automatically, it will update the ‘Number of Customers’ to what its counted, 100 customers.

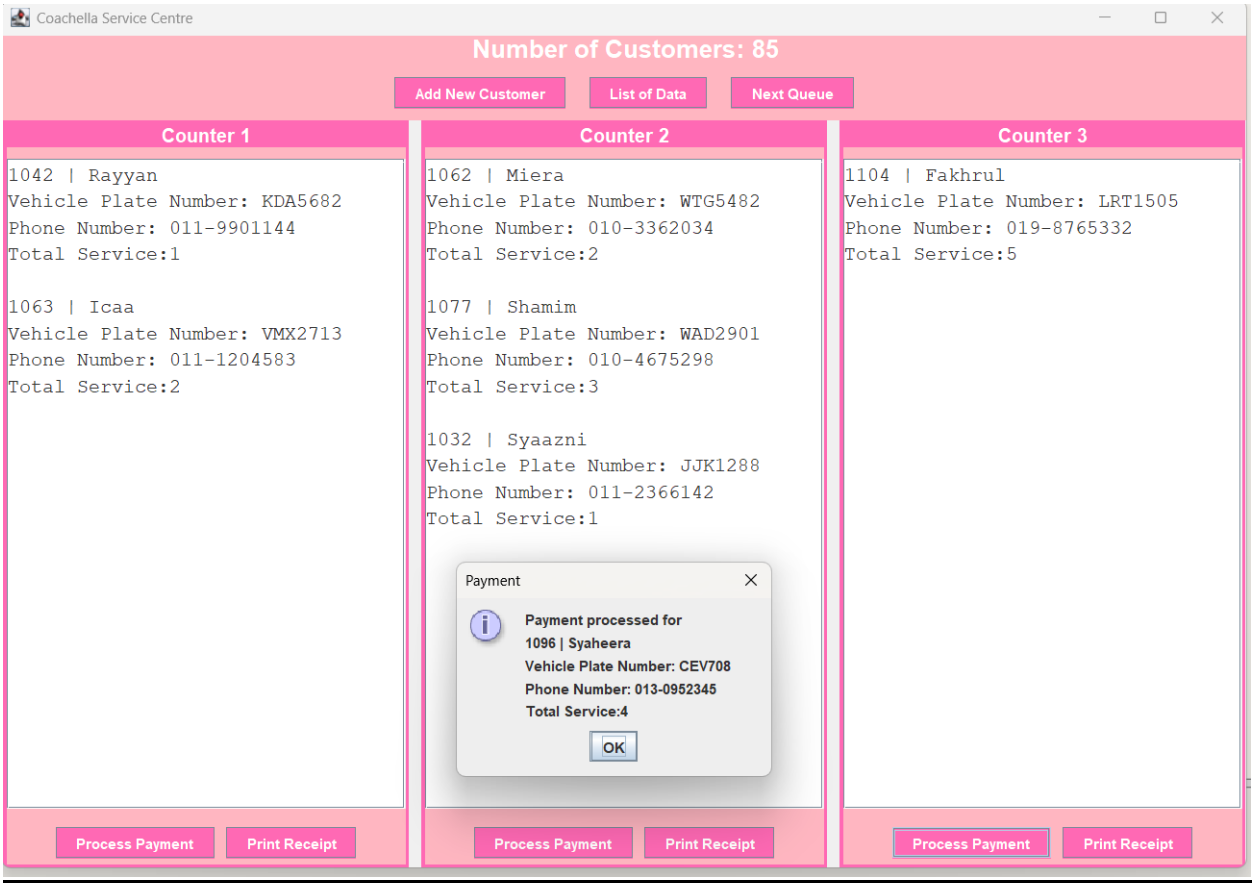


- To proceed, button ‘ Next Queue ‘ need to be click to add the customers to the lane. Each Customer have their services type and we adding services total by each customer to assign them to their lane.
- To confirm, we create Batch Assignment Messages to acknowledge user how many customers will be added to their lane.
- Since the assigning (use queue, First-In-First-Out), theres a time where the customers will be skip in the waiting list cause their lane are full, and need to proceed by other customer(other lane is empty), But all the customers will be assign until there are no customer.

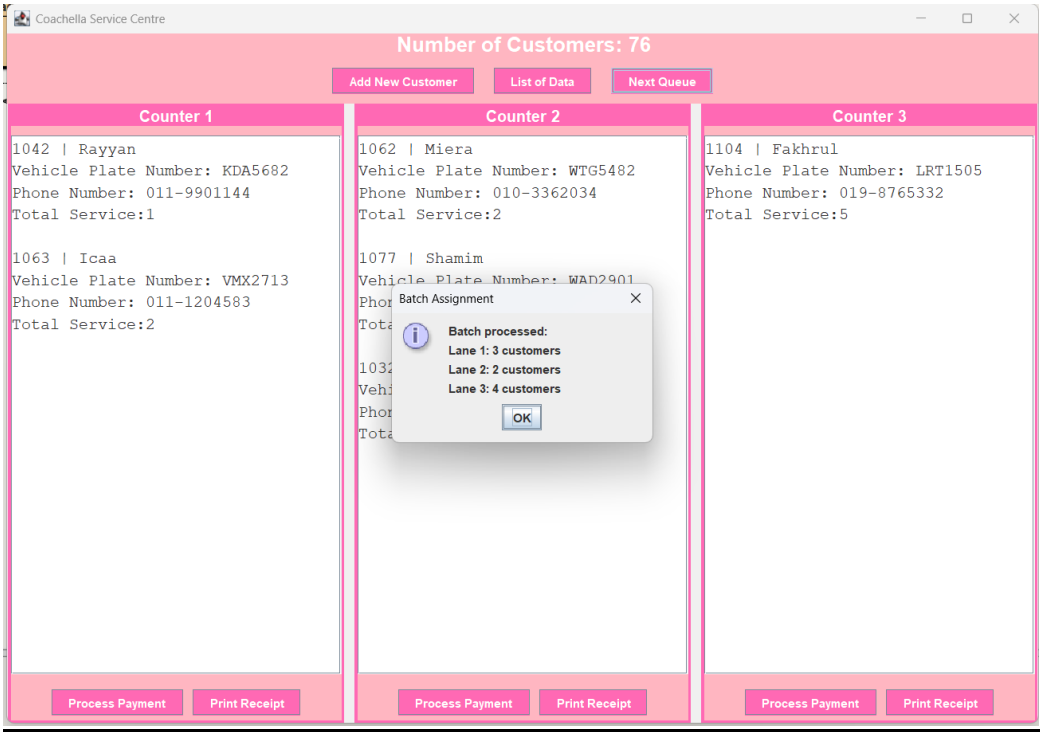


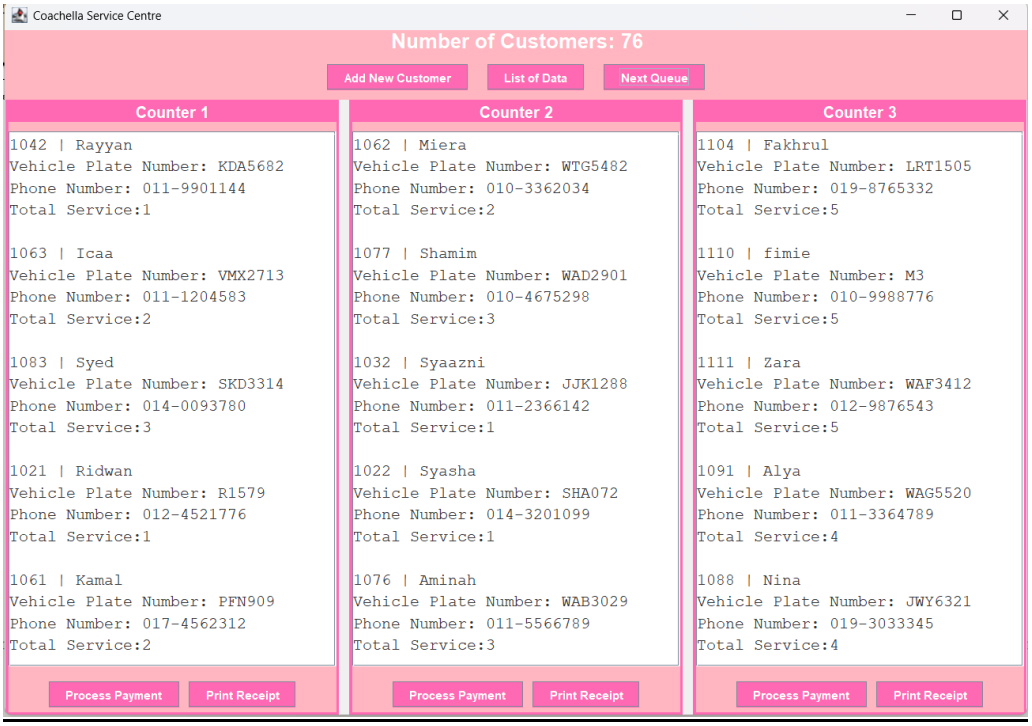
- In each counter, for each customer there with their customer Id, name, vehicle plate number, phone number and total services. Known as their Customer Info (except total service).
- The number of customer will always be update cause after the customers were assign to their lane, the number of customer will be updated.

- Button ‘ Process Payment ‘ will remove customer in the queue and transfer the date to the receipt that will be display in receipt display.
- The messages payment (contains all the customer information) will be pop up for confirmation.

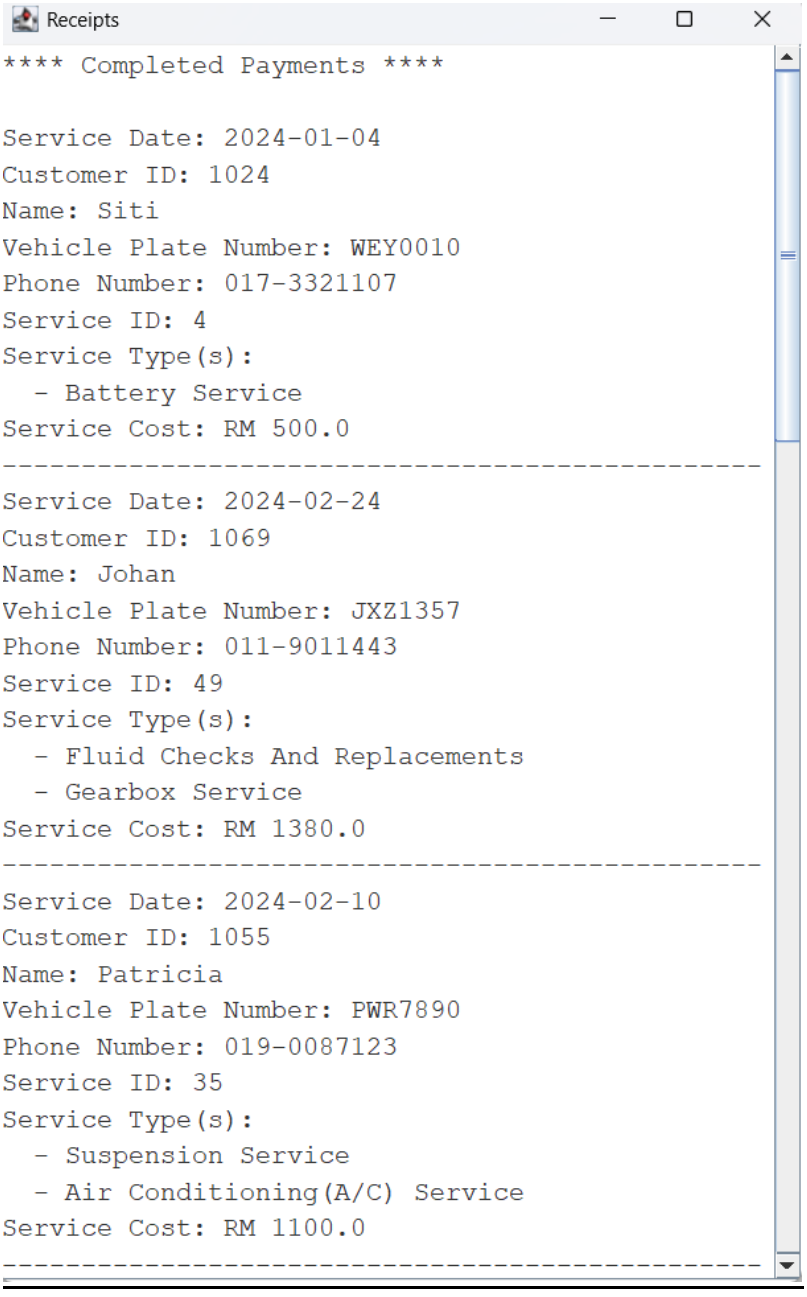


- To continue for all the customers to make payment, Button ‘ Next Queue ‘ will be repeated until there are no customers longer.
- For the output below, the new customers are added to their lane after the confirmation.

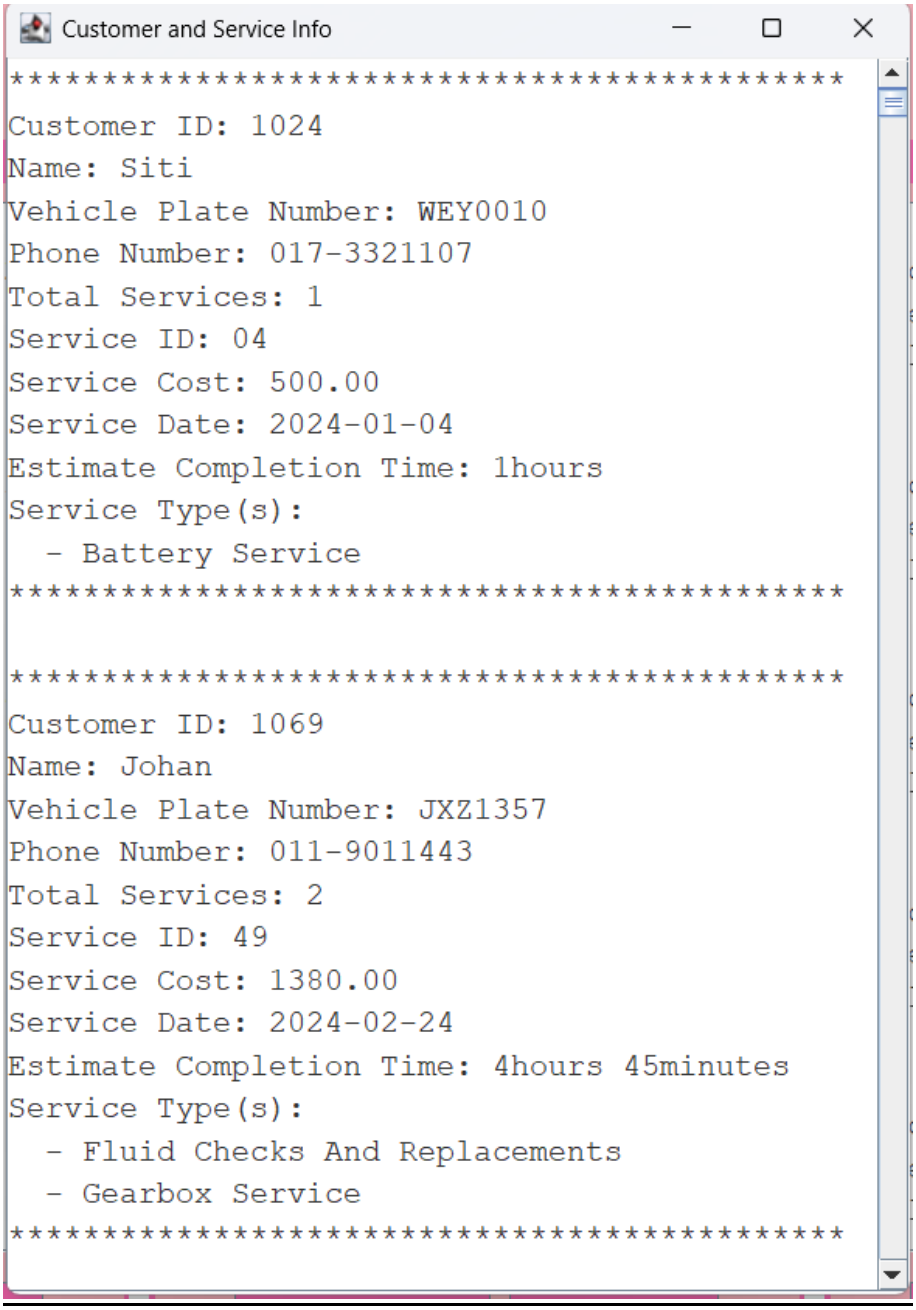




- Button ‘ Print Receipt ‘ will display the list of customer that already make payments(after ‘Process Payment’)
- The data includes service date, Customer Id, Name, Vehicle Plate Number, Phone Number, Service Id, Service Type and Service Cost.



- Button ‘ List of Data’ is to list all the Customer Information and Service Information based on the text. File .



Lesson Learned

NURSYASHA AMIRA BINTI SAIFUL (AM2311015156)

It has been both difficult and rewarding for me to work on this project , Coechella about vehicle service centre. My journey as a project leader and contributor was full of opportunities to learn more than just how to code. My thoughts on the lessons I took away and how this project influenced my abilities and viewpoint are presented here.

One of my primary contributions was to the Coachella Services class, which houses the system's essential features. Because it manages the primary tasks that support the project's goals, this serves as the project's framework. As the team leader, I made sure that everyone had defined roles and due dates by allocating tasks to team members. I actively solicited feedback from my team and led brainstorming sessions where we exchanged ideas both inside and outside the project's criteria. To make sure that our implementation met the project requirements, research and trial and error were essential components of my strategy. In particular, the first week was challenging. Initially, I had trouble understanding the project specifications and what was expected of us. It was difficult to decide which team member's suggestions should be implemented because it was necessary to balance everyone's ideas while adhering to the goals. It was challenging to properly structure the project, from the lane assignment logic to the visual design of the system.

I overcame these challenges by asking our lecturer and friends for clarification until I understood the requirements completely. We were open with each other and settled disputes respectfully by discussing and deciding on the best course of action for the project. Staying focused and having a clear mind was essential, as it taught me to think critically and creatively. I now have a better technical understanding of Java, especially in terms of organizing code and controlling the sequence of operations—from choosing which should be done first to how everything works together in the end. Additionally, I learned how to successfully apply the initially perplexing First-In, First-Out (FIFO) logic for lane management.

In terms of non-technical skills, this project greatly improved my ability to work in a team, solve problems, and manage my time. To finish the project, we studied together after class for a long time—sometimes from morning until evening. Although there was a lot of trial and error during brainstorming sessions, they aided in our development. We made sure that everyone's voice was heard and used effective communication to resolve any disagreements that did arise.

I realized that good leadership involves making decisions based on team consensus rather than individual preferences. Any decision I made, whether it concerned the design or the code structure, was carefully considered and approved by the team in order to ensure alignment with the project's goals. One of my favourite instances of teamwork was when my teammates were always checking in on me and offering assistance when I felt overwhelmed or stuck. This cooperation and concern demonstrated to me the value of a strong team. This project forced me to step outside of my comfort zone. It forced me to exercise critical thinking, stay up late for days, and focus on every little detail because I knew that a single error could have a huge effect on the system as a whole. Because I had to consider and evaluate everything before making any changes, I once joked that I needed a pigeon's 360-degree vision.

In addition, I became more aware of my own work habits. For example, I found that I was more productive when I concentrated on a single task at a time rather than multitasking. Coding and other assignments, such as calculus, were too much to handle. Compared to working alone, I found that group discussions and discussions with lecturers were far more effective for expanding my knowledge and clearing up any doubts. This project taught me the value of completing tasks on time and selecting challenges that will push me to learn and grow. While simple tasks may appear more convenient, they don't provide the same sense of fulfilment or personal development as more challenging ones. I want to be more efficient with my time, be grateful for what I have, and never stop learning new things.

I also realized that I needed to broaden my knowledge by conducting research and making connections with people who have more expertise than me. This will allow me to stay ahead and improve over time. The most memorable part of the project was when everything came together and the system worked as planned. That sense of accomplishment justified all of the hard work, sleepless nights, and struggles.

Looking back on the overall experience, I can confidently say that the project met my expectations. This wasn't just the result of my own work; rather, it was a result of our teamwork, sharing the load and encouraging one another. This project demonstrated to me that hard work does indeed pay off.

NUR SITI ZAINAB BINTI AZMI (AM2311015158)

Working on the Coachella Auto Garage project with Syasha, Izzati, and Radhiah was an incredibly valuable experience. My main role was developing the `CoachellaService` class to manage service bookings and queues. Syasha and I collaborated closely on the service logic, while Izzati and Radhiah focused on customer and service information, as well as the welcome page. A key lesson was the importance of clear class responsibilities `CustomerInfo` and `ServiceInfo` stored essential data, while LinkedLists, Queues, and Stacks helped us manage customer records and service lanes.

One of the biggest challenges was designing the queue system. Customers with fewer services were assigned to Service Lanes 1 and 2, while those with more than three services went to Lane 3. We also had to ensure the system could efficiently handle large volumes of customer data, like reading over 100 customer records from a file. Implementing functions like adding/removing(payment and display receipt) customers and calculating total service costs also required careful attention to detail.

Additionally, collaboration and task delegation were crucial to completing the project on time. Syasha and I focused on implementing the core service logic and queue management, while Izzati concentrated on creating efficient data structures for customer and service information. Radhiah was responsible for the user interface, ensuring that the system presented data in an accessible and user-friendly manner. This level of coordination helped us stay on track and deliver a cohesive project.

In summary, this project not only enhanced our knowledge of object-oriented programming concepts and data structures, but also taught us about the complexities of real-world application development. We had to think critically about managing customer data, handling dynamic queues, and processing transactions in a scalable way. Working with Syasha, Izzati, and Radhiah further developed our teamwork and communication skills, which were just as crucial as our technical abilities. Overall, this experience was an invaluable learning opportunity that has equipped us to tackle similar challenges in the future.

NUR IZZATI AFIQAH BINTI KAHIZAR (AM2311015176)

Overall, after completing this project, I realized that working in a group has many benefits that we can get, one of which is helping. This is because, if we do not practice the concept of helping to help, then this project cannot be completed completely. With that, I realized that helping is very important in group work. In addition, doing work in a group also requires the concept of understanding each other. This is because, to complete this project, we need a lot of time and a lot of giving opinions to each other and we also need to always contact and stay back at the university to complete this project. So, by using the concept of mutual understanding, we managed to accept all opinions with an open heart.

I can conclude the good because, with good and cooperative team members, I managed to finish part of the coding given by our group leader Syasha. This is because, when I finished my coding, a group of us found that there were some parts of the coding that were wrong and I couldn't finish them. So with that, a group of us got together and discussed what went wrong and we tried to redo it until the coding was completed.

Not only that, I also learned a lot about coding. Among them is the part of the coding that I did which is "CustomerInfo". Here, I understand how to systematically organize and control service data. This knowledge is very important because it can be used in many real-world situations to ensure that data is managed efficiently. The "CustomerInfo" class is created only to systematically manage customer information, including services received. By taking advantage of features such as "ServiceInfo", this class also enables customer and service data management to be done in an orderly manner by using methods such as "addService" to add service information while "toString" is used to display data in a formatted way. As an example the class shows how Java can be used to build flexible and manageable systems for customer and service management applications.

With the cooperation of group members in doing this project, our group has successfully completed a project named "VehicleService". Overall, completing this project was an unforgettable memory because almost every day a group of us returned late in the evening to complete this task. In this project, it taught me to be more organized when building programs. In addition, in completing this project as well, we need to be more careful and implement the project in stages to ensure that the project runs smoothly.

RADHIAH FARZANA BINTI RIDWAN (AM2311015163)

In developing the WelcomePage class for the Coechella Service System, I learned several valuable lessons. This process also taught me the importance of collaboration and teamwork in software development.

The first lesson is the importance of setting up a clear, visually appealing interface. I focused on creating an engaging welcome screen for users by utilizing a combination of text labels, images and buttons. The design incorporated two title labels with a GridLayout for better organization and a logo image for branding. This taught me how using simple yet effective layouts can enhance the aesthetic appeal of a page, making it welcoming and easy to navigate.

Another key learning was in event handling with the ActionListener. By attaching the actionPerformed() method, I was able to make the button functional and trigger the transition to another part of the application. Disposing of the current window and initializing the CoechellaService frame taught me the value of managing window transitions within a Java GUI application.

Additionally, I learned the benefit using layout managers like BorderLayout and Grid Layput to organize components. These layout managers made it easier to align and arrange components properly, ensuring that the interface looked good across different screen sizes. This was important for creating a user-friendly experience without worrying about hardcoding the positions of each element.

On the technical side, another important lesson was in creating detaile UML diagrams. Creating these diagrams helped me visualize the structure and relationship between different components of the system. The CustomerInfo class, for instance, encapsulates customer-related data such as id, name, service cost and service date.

Overall, working on the WelcomPage and designing the UML diagrams emphasized the significance of thoughtful design, clear separation of responsibilities, and the effective use of event listeners and layout managers. Additionally, working as a team taught me the value of collaborative problem-solving and sharing knowledge and skilss to achieve a common goal. These lessons will help me in both UI development and in maintaining an organized codebase for future enhancements of the Coechella Service system.

References (APA format)

takeuforward - Best Coding Tutorials for Free. (n.d.). takeUforward - ~ Strive for Excellence. https://takeuforward.org/data-structure/implement-queue-using-stack/

GeeksforGeeks. (2024, August 5). *Implement Stack using Queues*. GeeksforGeeks. https://www.geeksforgeeks.org/implement-stack-using-queue/

Java Program to Implement the queue data structure. (n.d.). https://www.programiz.com/java-programming/examples/queue-implementation

Best implementation of Java Queue? (n.d.). Stack Overflow. https://stackoverflow.com/questions/11149707/best-implementation-of-java-queue'

Java stacks and queues. (2024, January 1). https://javacodehouse.com/blog/java-stack-queue/

Бранниган, 3. (2024, March 29). *Data structures in Java - stack and queue*. JavaRush. https://javarush.com/en/groups/posts/en.2321.data-structures-in-java---stack-and-queue

Staff, H. (2023, October 9). How to Implement Stacks and Queues in Java. *dhia*. https://hackajob.com/talent/blog/how-to-implement-stacks-and-queues-in-java

GeeksforGeeks. (2024b, September 10). *12 Tips to Optimize Java Code Performance*. GeeksforGeeks. https://www.geeksforgeeks.org/12-tips-to-optimize-java-code-performance/

String vs String builder. which is faster? If stringbuilder then why use string. (n.d.). Stack Overflow. https://stackoverflow.com/questions/61183595/string-vs-string-builder-which-is-faster-if-stringbuilder-then-why-use-string/61183781

Derek Banas. (2013, March 2). *Stacks and queues* [Video]. YouTube. https://www.youtube.com/watch?v=JvGZh_BdF-8

Smith, J. A. (2020). *Queue management in service systems: Optimizing customer flow with linked lists and queues*. *Journal of Software Engineering & Systems Architecture*, 15(3), 45-58. https://www.codeunderscored.com