UNIVERSITI KEBANGSAAN MALAYSIA

The National University of Malaysia

TTTK1143

REKABENTUK ATURCARA & PENYELESAIAN MASALAH

ASSIGNMENT 1 REPORT

PREPARED BY :

Name	
ViceNish	

Contents

1. Introduction

My system is about UKM Parking Lot. First, you must select new or continue. If you select new, the system will reset every data in it. If you select continue, the system will not reset and you can continue your previous data. Then, it will appear 4 type of choices that is Add, Edit, Remove and List Parking.

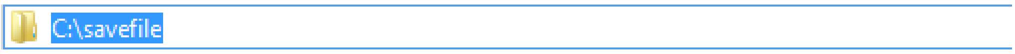
If you select Add, you must register your name, id and password. Then Choose your vehicle (Note : you can only submit one vehicle per time). However if you choose bus or lorry, dialog message will appear show that both vehicle can't parked at UKM. After that, Let say that you are student, you can only choose public parking lot section which is car and motor only. However if you are staff, you can park at staff parking and public parking. If you are Management staff, you can park anywhere in the parking lot. For my system, I have 4 types of parking lot that is public parking (Car and Motor), staff parking and Management parking. Each parking respectively have 5 lot. Those who have more than 1 vehicle may enter again Add button at the Login Menu to register respective vehicle. Every time you success parked will be recorded in a file for respective parking lot folder.

Next, if you select Edit, the system will lead you to 2 choices which is Change Owner Vehicle or Promote/Demote Staff. Change Owner Vehicle may change the owner of the vehicle. You can insert name and id of the new owner of the vehicle. This action will be updated in recorded file. Then, if you select Promote/Demote Staff, the system will change your status job and recorded the updated. However, the system will not change the parking lot for you.

Then, if you select Remove, the system will grant you to remove the vehicle from parking lot depend on your choice. This action may delete the file recorded in the parking lot folder.

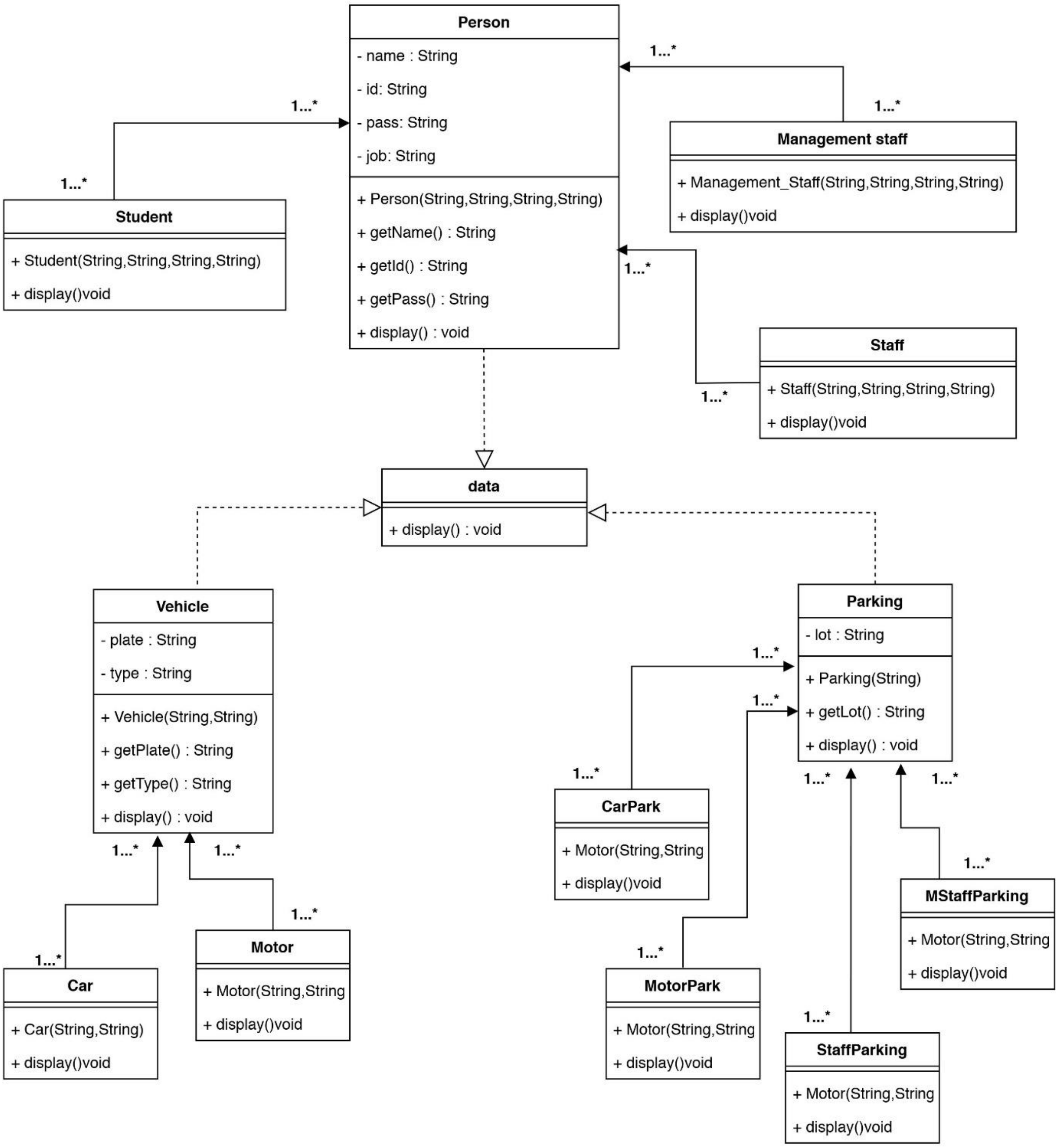
Last but not least, if you select List Parking, you may view your record based on what you input early. You also can view others parking lot information.

Lastly, this program may run if the "savefile" folder put at "C:\" path. I will provided the folders. Thank You.



C:\savefile

2. Class Relationship (UML Diagram)



3. List Of Classes

Class Name	Description
pperson	This class will implements data class. Attribute static : name, id, pass, job. Method : +pperson(String,String,String,String) role is constructor. +getName() to get and return the name insert. +getId() to get and return the id insert. +getPass() to get and return the password insert. +getJob() to get and return the job depend on the user. +display() will display the output.
pstudent	This class will inherit person class. Method : +pstudent(String,String,String,String) will act as constructor. +display() will display the output.
pstaff	This class will inherit pperson class. Method : +pstaff(String,String,String,String) will act as constructor. +display() will display the output.
pmstaff	This class will inherit pperson class. Method : +pmstaff(String,String,String,String) will act as constructor. +display() will display the output.
pvehicle	This class will implements data class. Attribute static: plate, type. Method : +pvehicle(String,String) will act as constructor.. +display() will display the output.
pcar	This class will inherit pvehicle class. Method : +pcar(String,String) will act as constructor. +display() will display the output.
pmotor	This class will inherit pvehicle class. Method : +pmotor(String,String) will act as constructor. +display() will display the output.
Pparking	This class will implements data class. Attribute static: lot. Method : +pparking(String) will act as constructor. +display() will display the output.
Pcarpark	This class will inherit pparking class. Method : +pcarpark(String) will act as constructor. +display() will display the output.

pmotorpark	This class will inherit pparking class. Method : +pmotorpark(String) will act as constructor. +display() will display the output.
pstaffpark	This class will inherit pparking class. Method : +pstaffpark(String) will act as constructor. +display() will display the output.
pmstaffpark	This class will inherit pparking class. Method : +pmstaffpark(String) will act as constructor. +display() will display the output.
Data	This is interface class. Method : +display() will make other class have this method.
Person	This class will be link to LoginMenu class. This class's constructor will be my GUI. Method : +Person() act as my GUI. User must choose whether they are students, staff or management staff .
Student	This class's constructor will be my GUI. Other related classes will inherit below attribute for recod filing process. Method : +Student() act as my GUI. User must input their name, id and password.
Staff	This class's constructor will be my GUI Other related classes will inherit below attribute for recod filing process. Method : +Staff() act as my GUI. User must input their name, id and password.
Management_Staff	This class's constructor will be my GUI. Other related classes will inherit below attribute for recod filing process. Method : +Management_Staff() act as my GUI. User must input their name, id and password.
VehicleS (student)	This class will inherit data from Student class. This class's constructor will be my GUI. Method : +VehicleS(String,String,String,String) act as my GUI. User must choose to select their vehicle which are car, motorcycle, bus or lorry. If the user choose bus and lorry, message dialog will appear said that vehicle can't park at UKM. In this case, I will not make Bus and Lorry classes.

VehicleSt (staff)	This class will inherit data from Staff class. This class's constructor will be my GUI. Method : +VehicleSt(String,String,String,String) act as my GUI. User must choose to select their vehicle which are car, motorcycle, bus or lorry. If the user choose bus and lorry, message dialog will appear said that vehicle can't park at UKM. In this case, I will not make Bus and Lorry classes.
VehicleM (mstaff)	This class will inherit data from Management_Staff class. This class's constructor will be my GUI. Method : +VehicleM(String,String,String,String) act as my GUI. User must choose to select their vehicle which are car, motorcycle, bus or lorry. If the user choose bus and lorry, message dialog will appear said that vehicle can't park at UKM. In this case, I will not make Bus and Lorry classes.
CarS (student)	This class will inherit data from VehicleS class. This class's constructor will be my GUI. Method : +CarS(String,String,String,String) act as my GUI. User must input their PlateNo. of car.
CarSt (staff)	This class will inherit data from VehicleSt class. This class's constructor will be my GUI. Method : +CarSt(String,String,String,String) act as my GUI. User must input their PlateNo. of car.
CarM (mstaff)	This class will inherit data from VehicleM class. This class's constructor will be my GUI. Method : +CarS(String,String,String,String) act as my GUI. User must input their PlateNo. of car.
MotorS (student)	This class will inherit data from VehicleS class. This class's constructor will be my GUI. Method : +MotorS(String,String,String,String) act as my GUI. User must input their PlateNo. Of motor.
MotorSt (staff)	This class will inherit data from VehicleSt class. This class's constructor will be my GUI. Method : +MotorSt(String,String,String,String) act as my GUI. User must input their PlateNo. Of motor.
MotorM (mstaff)	This class will inherit data from VehicleM class. This class's constructor will be my GUI. Method : +MotorM(String,String,String,String) act as my GUI. User must input their PlateNo. Of motor.

ParkingSc (student car)	This class will inherit data from CarS class. This class's constructor will be my GUI. Method : +ParkingSc(String,String,String,String,String) act as my GUI. User must select which place to park. Since there are some problem occur, user can only choose to park at FTSM only.
ParkingStc (staff car)	This class will inherit data from CarSt class. This class's constructor will be my GUI. Method : +ParkingStc(String,String,String,String,String) act as my GUI. User must select which place to park. Since there are some problem occur, user can only choose to park at FTSM only.
ParkingMc (mstaff car)	This class will inherit data from CarM class. This class's constructor will be my GUI. Method : +ParkingMc(String,String,String,String,String) act as my GUI. User must select which place to park. Since there are some problem occur, user can only choose to park at FTSM only.
ParkingSm (student motor)	This class will inherit data from MotorS class. This class's constructor will be my GUI. Method : +ParkingSm(String,String,String,String,String) act as my GUI. User must select which place to park. Since there are some problem occur, user can only choose to park at FTSM only.
ParkingStm (staff motor)	This class will inherit data from MotorSt class. This class's constructor will be my GUI. Method : +ParkingStm(String,String,String,String,String) act as my GUI. User must select which place to park. Since there are some problem occur, user can only choose to park at FTSM only.
ParkingMm (mstaff motor)	This class will inherit data from MotorM class. This class's constructor will be my GUI. Method : +ParkingMm(String,String,String,String,String) act as my GUI. User must select which place to park. Since there are some problem occur, user can only choose to park at FTSM only.
ChooseSc (staff car)	This class will inherit data from ParkingStc class. This class's constructor will be my GUI. Method : +ChooseSc(String,String,String,String,String,String,String) act as my GUI. User (Staff) may choose where he/she want to park whether public park (car lot) or premium park (staff lot).
ChooseSm (staff motor)	This class will inherit data from ParkingStm class. This class's constructor will be my GUI. Method : +ChooseSm(String,String,String,String,String,String,String) act as my GUI. User (Staff) may choose where he/she want to park whether public park (car lot) or premium park (staff lot).

ChooseMc (mstaff car)	This class will inherit data from ParkingMc class. This class's constructor will be my GUI. Method : +ChooseMc(String,String,String,String,String,String,String) act as my GUI. User (MStaff) may choose where he/she want to park whether public park (car lot) , premium park (staff lot) or premium m park (mstaff lot).
ChooseMm (mstaff motor)	This class will inherit data from ParkingMm class. This class's constructor will be my GUI. Method : +ChooseMm(String,String,String,String,String,String,String) act as my GUI. User (MStaff) may choose where he/she want to park whether public park (car lot) , premium park (staff lot) or premium m park (mstaff lot).
CarPark	This class will inherit data from ParkingSc, ChooseSc and ChooseMc class. This class's constructor will be my GUI. Method : +CarPark(String,String,String,String,String,String,String) act as my GUI. User may select any lot parking that he/she like to parked. This process may record all information from inheritance above in a textfile.
MotorPark	This class will inherit data from ParkingSm, ChooseSm and ChooseMm class. This class's constructor will be my GUI. Method : +MotorPark(String,String,String,String,String,String,String) act as my GUI. User may select any lot parking that he/she like to parked. This process may record all information from inheritance above in a textfile.
StaffParkingc (car)	This class will inherit data from ChooseSc class. (car) This class's constructor will be my GUI. Method : +StaffParkingc(String,String,String,String,String,String,String) act as my GUI. User may select any lot parking in Staff lot that he/she like to parked. This process may record all information from inheritance above in a textfile.
StaffParkingm (motor)	This class will inherit data from ChooseSm class. (motor) This class's constructor will be my GUI. Method : +StaffParkingm(String,String,String,String,String,String,String) act as my GUI. User may select any lot parking in Staff lot that he/she like to parked. This process may record all information from inheritance above in a textfile.

MStaffParkingc (car)	This class will inherit data from ChooseMc class. (car) This class's constructor will be my GUI. Method : +MStaffParkingc(String,String,String,String,String,String,String) act as my GUI. User may select any lot parking in MStaff lot that he/she like to parked. This process may record all information from inheritance above in a textfile.
MStaffParkingm (motor)	This class will inherit data from ChooseMc class. (motor) This class's constructor will be my GUI. Method : +MStaffParkingm(String,String,String,String,String,String,String) act as my GUI. User may select any lot parking in MStaff lot that he/she like to parked. This process may record all information from inheritance above in a textfile.
Edit	This class will be link to LoginMenu class. This class's constructor will be my GUI. Method : +Edit() act as my GUI. Here user will choose whether he/she want to change vehicle owner or promote/demote staff.
ChangeVehicle	This class's constructor will be my GUI. Method : +ChangeVehicle() act as my GUI. Here user will have to insert name and id to change owner of the vehicle.
ECAR (edit car lot)	This class will inherit name and id form ChangeVehicle class. This class's constructor will be my GUI. Method : +ECAR(String,String) act as my GUI. User may select his/her car in carparking lot. This action may updated the information into the existed textfile.
EMOTOR (edit motor lot)	This class will inherit name and id form ChangeVehicle class. This class's constructor will be my GUI. Method : +EMOTOR(String,String) act as my GUI. User may select his/her car in motorparking lot. This action may updated the information into the existed textfile.
ESTAFF (edit staff lot)	This class will inherit name and id form ChangeVehicle class. This class's constructor will be my GUI. Method : +ESTAFF (String,String) act as my GUI. User may select his/her car in staffparking lot. This action may updated the information into the existed textfile.
EMSTAFF (edit mstaff lot)	This class will inherit name and id form ChangeVehicle class. This class's constructor will be my GUI. Method : +EMSTAFF(String,String) act as my GUI. User may select his/her car in mstaffparking lot. This action may updated the information into the existed textfile.

PDS (Promote/Demote Staff)	This class’s constructor will be my GUI. Method : +PDS() act as my GUI. User may choose to promote staff or demote staff.
Promote	This class’s constructor will be my GUI. Method : +Promote() act as my GUI. User may choose parking lot which is carpark,motorpark,staffpark and mstaffpark.
ECP (edit car promote)	This class’s constructor will be my GUI. Method : +ECP() act as my GUI. User may choose lot parking in carpark. If there are any staff, that staff will be promoted to mstaff.
EMP (edit motor promote)	This class’s constructor will be my GUI. Method : +EMP() act as my GUI. User may choose lot parking in motorpark. If there are any staff, that staff will be promoted to mstaff.
ESP (edit staff promote)	This class’s constructor will be my GUI. Method : +ESP() act as my GUI. User may choose lot parking in staffrpark. If there are any staff, that staff will be promoted to mstaff.
EMSP (edit mstaff promote)	This class’s constructor will be my GUI. Method : +EMSP() act as my GUI. User may choose lot parking in mstaffpark. If there are any staff, that staff will be promoted to mstaff.
Demote	This class’s constructor will be my GUI. Method : +Demote() act as my GUI. User may choose parking lot which is carpark,motorpark,staffpark and mstaffpark.
ECD (edit car demote)	This class’s constructor will be my GUI. Method : +ECD() act as my GUI. User may choose lot parking in carpark. If there are any mstaff, that mstaff will be demoted to staff.
EMD (edit motor demote)	This class’s constructor will be my GUI. Method : +EMD() act as my GUI. User may choose lot parking in carpark. If there are any mstaff, that mstaff will be demoted to staff.
ESD (edit staff demote)	This class’s constructor will be my GUI. Method : +ESD() act as my GUI. User may choose lot parking in carpark. If there are any mstaff, that mstaff will be demoted to staff.
EMSD (edit mstaff demote)	This class’s constructor will be my GUI. Method : +EMSD() act as my GUI. User may choose lot parking in carpark. If there are any mstaff, that mstaff will be demoted to staff.
Remove	This class will be link to LoginMenu class. This class’s constructor will be my GUI. Method : +Remove() act as my GUI. User will choose public park, staff park or mstaff park that vehicle to be parked out.

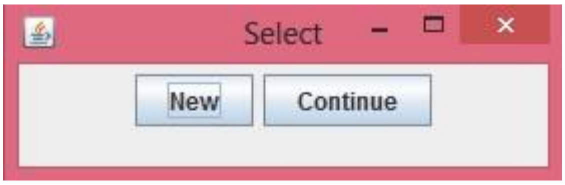
RPUBLIC (remove public)	This class's constructor will be my GUI. Method : +RPUBLIC() act as my GUI. User will choose car or motor park that vehicle to be parked out
RPPARK (remove premium park)	This class's constructor will be my GUI. (staff) Method : +RPPARK() act as my GUI. User will select which parking lot that his/her vehicle want to parked out.
RMPPARK (remove m premium park)	This class's constructor will be my GUI. (mstaff) Method : +RMPPARK() act as my GUI. User will select which parking lot that his/her vehicle want to parked out.
RCAR (remove car)	This class's constructor will be my GUI. (public) Method : +RCAR() act as my GUI. User will select which parking lot that his/her vehicle want to parked out.
RMOTOR (remove motor)	This class's constructor will be my GUI. (public) Method : +RMOTOR() act as my GUI. User will select which parking lot that his/her vehicle want to parked out.
ListParking	This class will be link to LoginMenu class. This class's constructor will be my GUI. Method : +ListParking() act as my GUI. User must choose which type of parking they want to view.
LCP (list car park)	This class's constructor will be my GUI. Method : +LCP() act as my GUI. User able to view information based on respective carpark lot.
CLot_1 (car lot)	This class's constructor will be my GUI. Method : +CLot_1() act as my GUI. User able to view information in this lot.
CLot_2 (car lot)	This class's constructor will be my GUI. Method : +CLot_2() act as my GUI. User able to view information in this lot.
CLot_3 (car lot)	This class's constructor will be my GUI. Method : +CLot_3() act as my GUI. User able to view information in this lot.
CLot_4 (car lot)	This class's constructor will be my GUI. Method : +CLot_4() act as my GUI. User able to view information in this lot.
CLot_5 (car lot)	This class's constructor will be my GUI. Method : +CLot_5() act as my GUI. User able to view information in this lot.
LMP (list motor park)	This class's constructor will be my GUI. Method : +LMP() act as my GUI. User able to view information based on respective motorpark lot.

MLot_1 (motor lot)	This class's constructor will be my GUI. Method : +MLot_1() act as my GUI. User able to view information in this lot.
MLot_2 (motor lot)	This class's constructor will be my GUI. Method : +MLot_2() act as my GUI. User able to view information in this lot.
MLot_3 (motor lot)	This class's constructor will be my GUI. Method : +MLot_3() act as my GUI. User able to view information in this lot.
MLot_4 (motor lot)	This class's constructor will be my GUI. Method : +MLot_4() act as my GUI. User able to view information in this lot.
MLot_5 (motor lot)	This class's constructor will be my GUI. Method : +MLot_5() act as my GUI. User able to view information in this lot.
LSP (list staff park)	This class's constructor will be my GUI. Method : +LSP() act as my GUI. User able to view information based on respective staffpark lot.
SLot_1 (staff lot)	This class's constructor will be my GUI. Method : +SLot_1() act as my GUI. User able to view information in this lot.
SLot_2 (staff lot)	This class's constructor will be my GUI. Method : +SLot_2() act as my GUI. User able to view information in this lot.
SLot_3 (staff lot)	This class's constructor will be my GUI. Method : +SLot_3() act as my GUI. User able to view information in this lot.
SLot_4 (staff lot)	This class's constructor will be my GUI. Method : +SLot_4() act as my GUI. User able to view information in this lot.
SLot_5 (staff lot)	This class's constructor will be my GUI. Method : +SLot_5() act as my GUI. User able to view information in this lot.
LMSP (list mstaff park)	This class's constructor will be my GUI. Method : +LMSP() act as my GUI. User able to view information based on respective mstaffpark lot.
MSLot_1 (mstaff lot)	This class's constructor will be my GUI. Method : +MSLot_1() act as my GUI. User able to view information in this lot.
MSLot_2 (mstaff lot)	This class's constructor will be my GUI. Method : +MSLot_2() act as my GUI. User able to view information in this lot.
MSLot_3 (mstaff lot)	This class's constructor will be my GUI. Method : +MSLot_3() act as my GUI. User able to view information in this lot.

MSLot_4 (mstaff lot)	This class’s constructor will be my GUI. Method : +MSLot_4() act as my GUI. User able to view information in this lot.
MSLot_5 (mstaff lot)	This class’s constructor will be my GUI. Method : +MSLot_5() act as my GUI. User able to view information in this lot.
LoginMenu	This class will link to Person, Edit, Remove and ListParking classes. This class’s constructor will be my GUI. Method : +LoginMenu() act as my GUI. This is like Main menu interface for user. User may choose what he/she want to do with the choices given.
NEWC	This is the first GUI that user need to be through. This class will lead user to LoginMenu class. This class’s constructor will be my GUI. Method : +NEWC() act as my GUI. User may choose to create new or continue previous data/work. If user choose new, the data will be reseted. If user choose continue, the data will keep stored.
MainApps	This class is my main class that execute only NEWC class only.

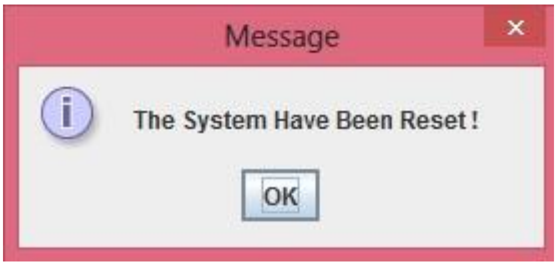
4. GUI Design

- **Select Menu**



-If user choose “continue” button, it just take you to next GUI which is Login Menu. For “New” button, the system will be reset and message dialog will appear as shown below.

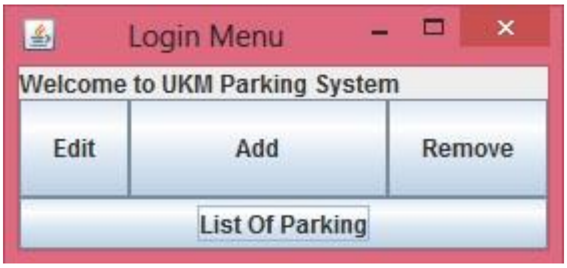
-Used FlowLayout


Message

The System Have Been Reset !

-Then, it will also lead you to the Login Menu. **NEWC** class involved in this GUI.

- **Login Menu (Main Menu)**


Login Menu

Welcome to UKM Parking System

-“Add” button will take you to register vehicle and park at parking lot.

-“Edit” button will make you to change owner of the vehicle or promote/demote staff.

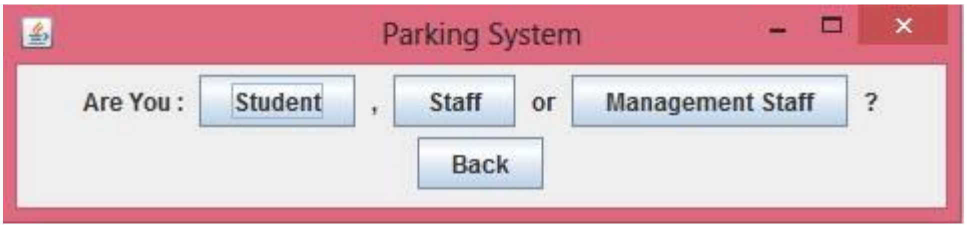
-“Remove” button may make you to parked out from the parking.

-“List Of Parking” button may display the information recorded in file.

-LoginMenu class involved in this GUI.

-Used BorderLayout

- **Parking System**



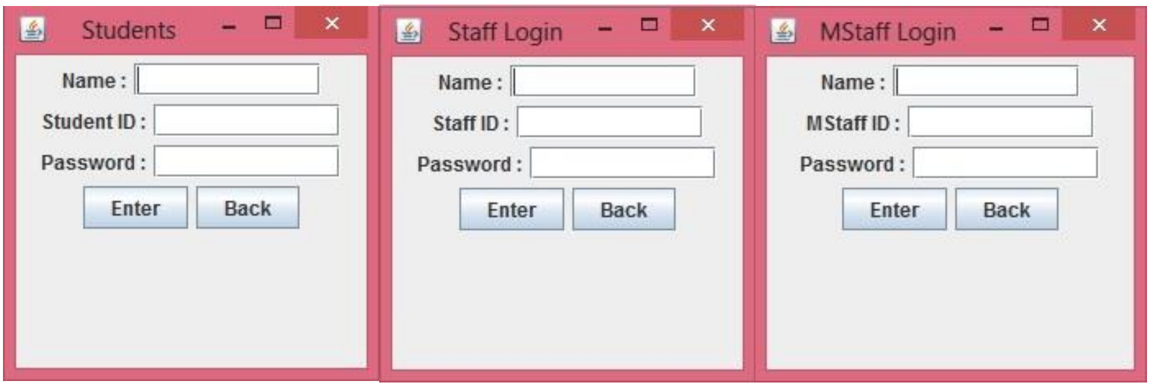
-This GUI will make the user choose whether he/she are student, staff or management staff.

-“Back” button will take you back to main menu.

-**Person** class involved in this GUI.

-Used Flow Layout

- **Student, Staff and Management Staff input**

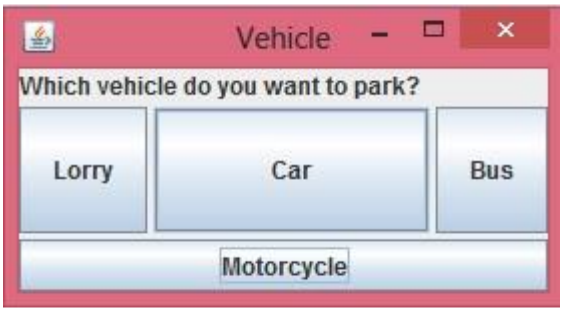


-User may insert Name, ID, and Password in their respective roles.

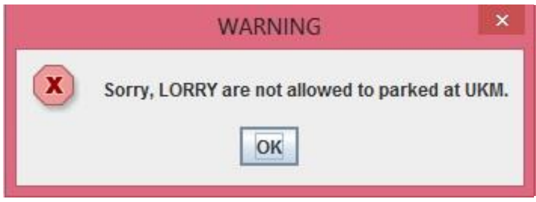
-**Student**, **Staff** and **Management_Staff** classes involved in GUI respectively.

-Used Flow Layout

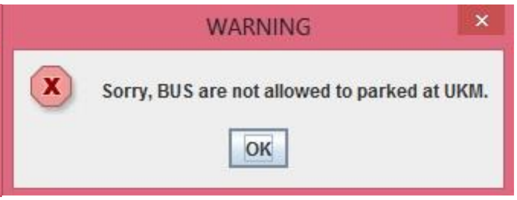
- **Vehicle**



-User may choose vehicle that his/her want to register. Note that if user choose bus and lorry button, message dialog will appear as shown below.



Sorry, LORRY are not allowed to parked at UKM.



Sorry, BUS are not allowed to parked at UKM.

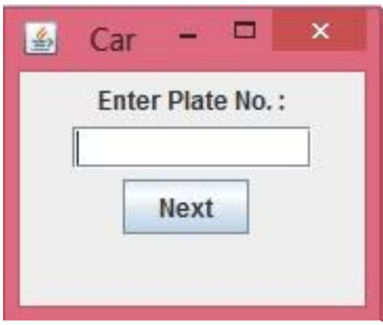
-This shows that user can’t park bus and lorry inside UKM.

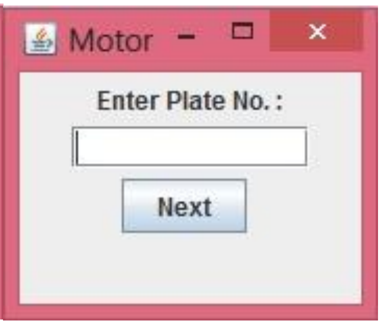
-This GUI involve **VehicleS, VehicleSt,** and **VehicleM** classes.

-If you choose car or motor button, the system will take you to the next GUI.

-Used Flow Layout.

- **Car and Motor (PlateNo.)**



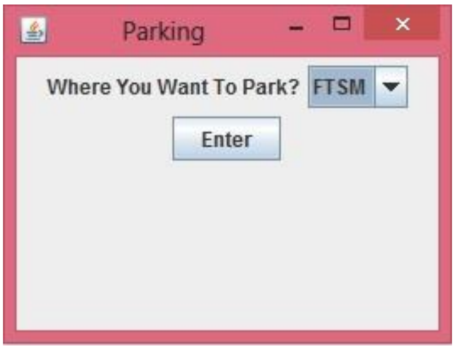


-User will enter their type of vehicle’s plate number.

-This GUI involve **CarS, CarSt, CarM, MotorS, MotorSt** and **MotorM** classes.

-Used Flow Layout.

- **Select Faculty to park**



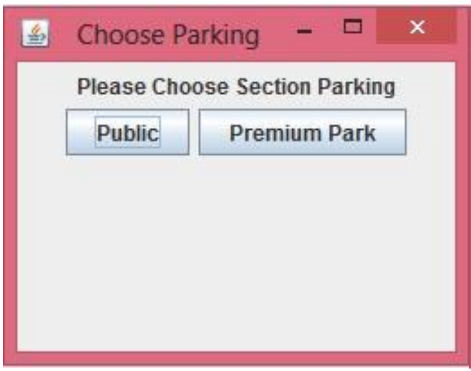
Where You Want To Park? FTSM

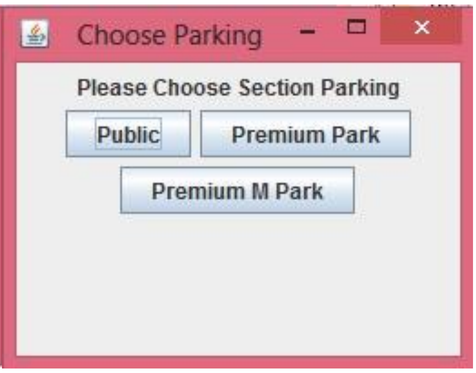
-In original idea, user may choose to park any faculty. But since there a lot of problem occur, user can park only at FTSM.

-**ParkingSc, ParkingStc, ParkingMc, ParkingSm, ParkingStm** and **ParkingMm** classes are involved in this GUI.

-Used Flow Layout.

- **Choose**





- For Student user, This GUI will not shown because student can only park at public parking.

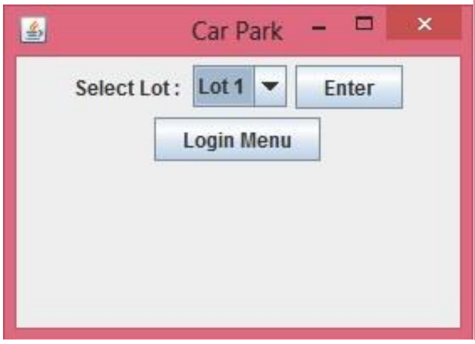
-For Staff, (left GUI) he/she can choose to park whether public or premium park.

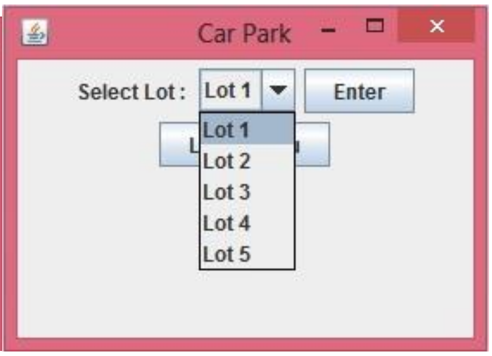
-But for the management staff, he/she can park anywhere including his/her park (premium m park).

-Used Flow Layout

-This GUI involve **ChooseSc, ChooseSm, ChooseMc** and **ChooseMm** classes.

- **Lot Parking**

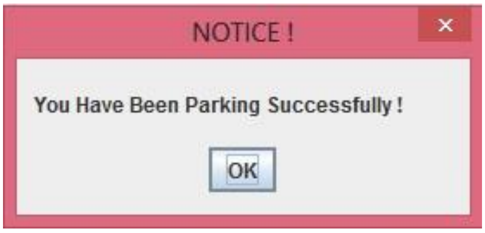




-This is the GUI for public parking (carpark). User may select any parking lot that their like and press enter button to be parked.

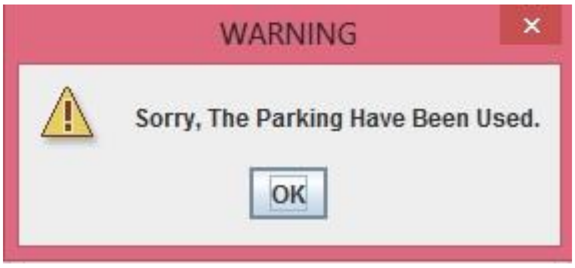
-The Login Menu button will lead you to the main menu.

-If user parking successfully, dialog message will appear as shown below and the information will be recorded in a folder through textfile.



You Have Been Parking Successfully !

-If not, then dialog message as below will be appeared then the system will lead you to main menu



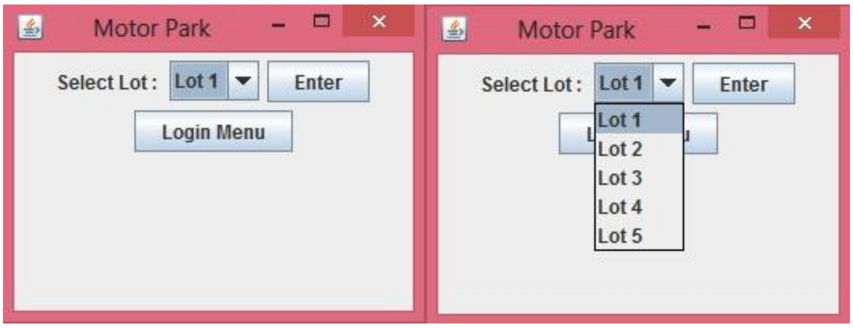
Sorry, The Parking Have Been Used.

-This message shows that you can’t parked if the parking have been used.

- Used Flow Layout.

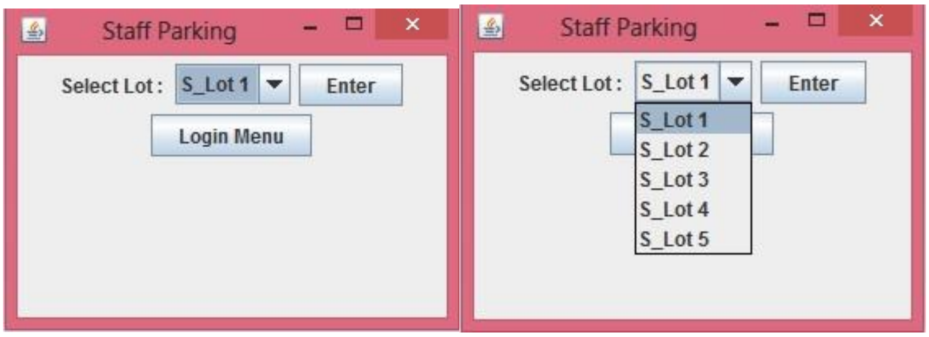
-The rest parking like Motorpark, Staffpark and MStaffpark are about the same process but recorded at different folder.

-Class involved in this GUI is **CarPark.**



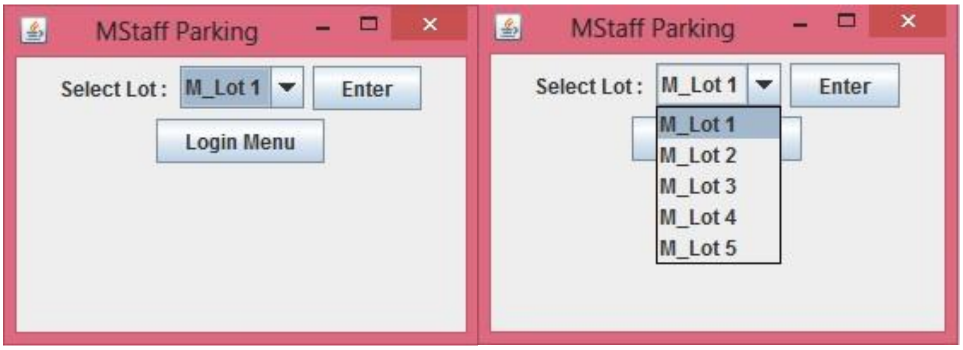
- Used Flow Layout.

- Class involved in this GUI is MotorPark.



- Used Flow Layout.

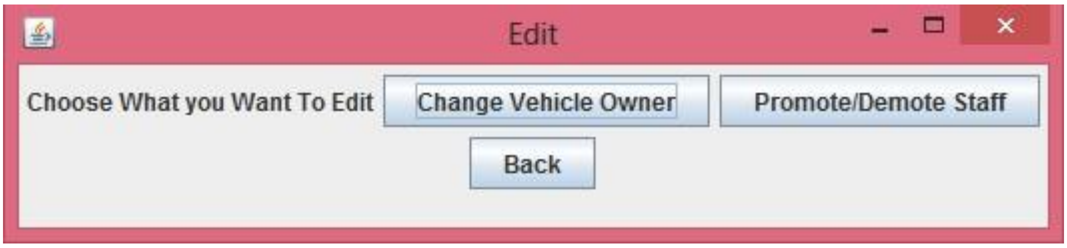
- Classes involved in this GUI is StaffParkingc and StaffParkingm.



- Used Flow Layout.

- Classes involved in this GUI is MStaffParkingc and MStaffParkingm.

- **Edit**



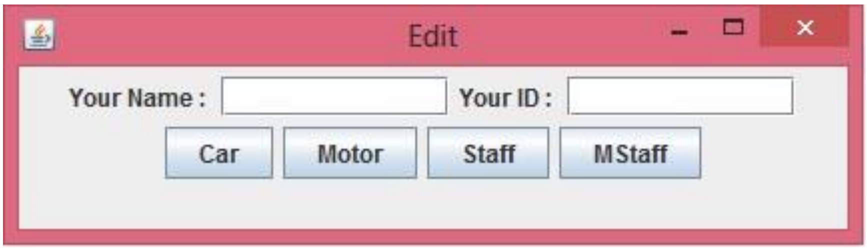
-User can change the owner of the vehicle or prmote/demote staff.

-Button back will lead you back to the main menu.

-This GUI involve **Edit** class.

- Used Flow Layout.

- **Change Vehicle Owner**

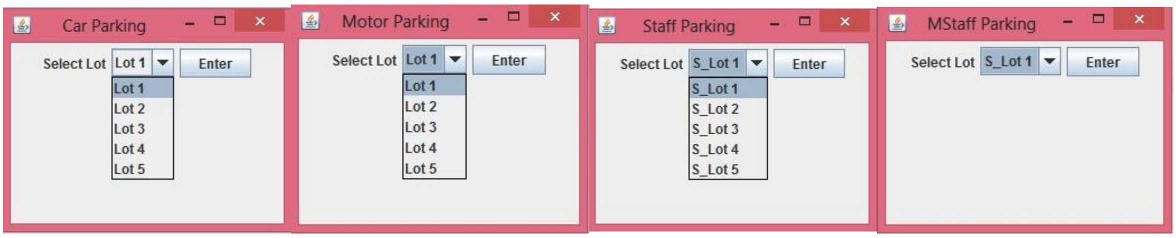


-User may insert his/het name and id and select which parking lot that have vehicle to change yours.

- Used Flow Layout.

-**ChangeVehicle** class involved in this GUI.

- **Select Car lot, Motor lot, Staff lot or MStaff lot**

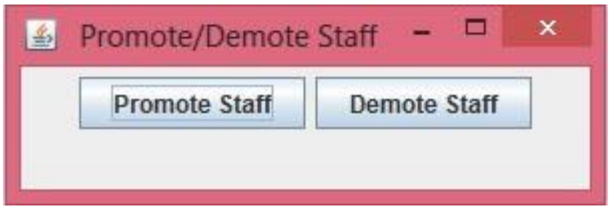


-The GUI above make user to select which lot that he/she want to change name and id of the vehicle.

- Used Flow Layout.

-**ECAR, EMOTOR, ESTAFF, EMSTAFF** classes are involved in this GUI.

- **Promote/Demote Staff**

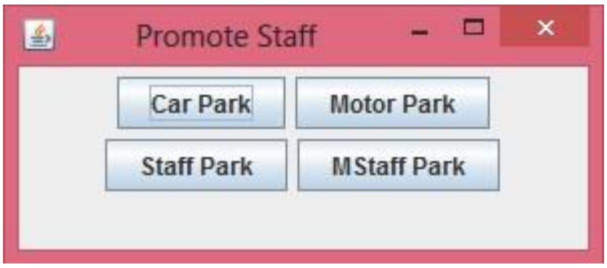


-User may choose whether promote or demote staff.

-**PDS** class involved in this GUI.

- Used Flow Layout.

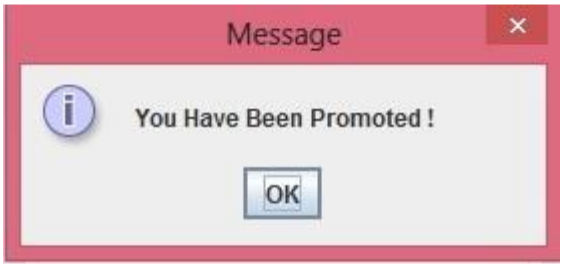
- **Promote Staff**



-User select which parking lot that have staff and promote him/her

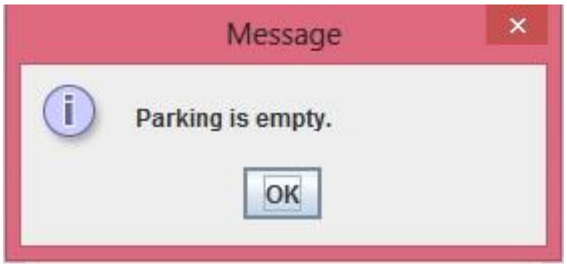
- Used Flow Layout.

-**Promote** class involved in this GUI



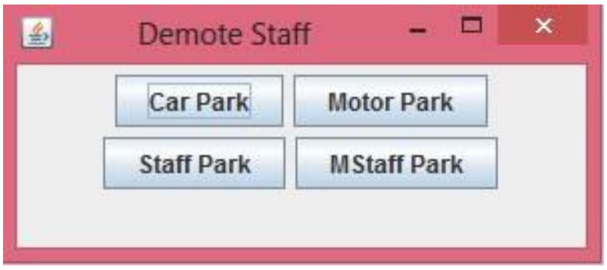
You Have Been Promoted !

-If user success promote the staff, dialog message above will appear. If not, it will appear as below.



Parking is empty.

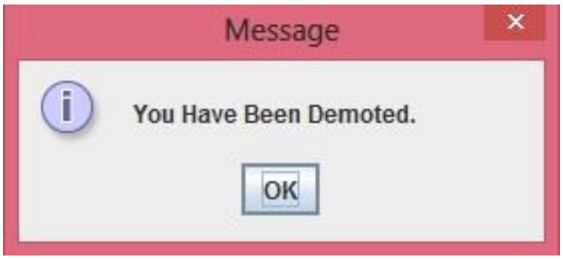
- **Demote**



-User select which parking lot that have staff and demote him/her

- Used Flow Layout.

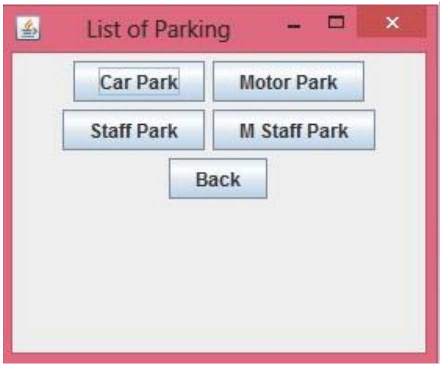
-**Demote** class involved in this GUI

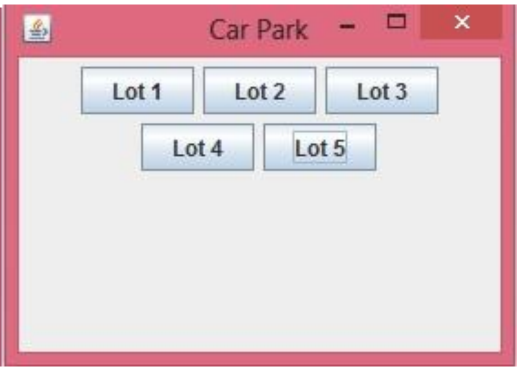


You Have Been Demoted.

-This dialog message will appear if you success and will show dialog message empty parking if not.

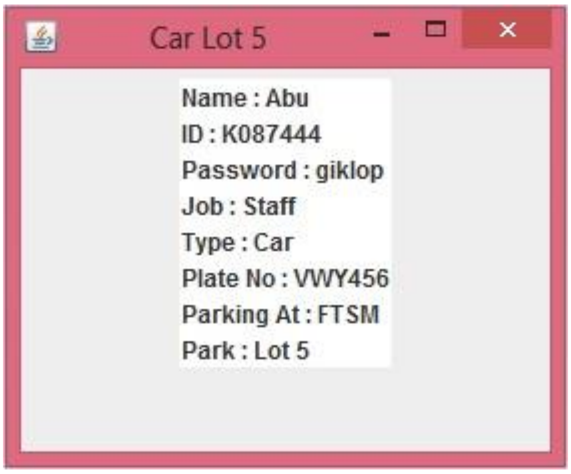
- **List information of parking**





-User can view the information data depend on type of parking and lot

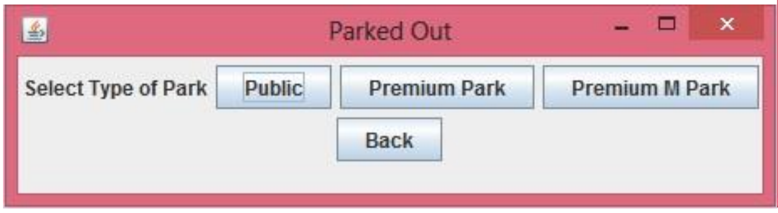
- Used Flow Layout.

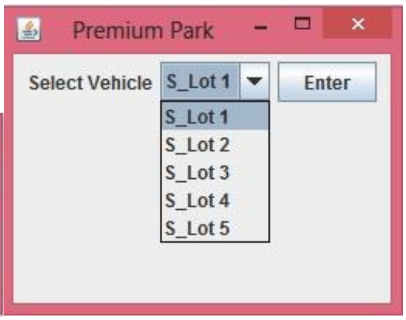


Name : Abu
ID : K087444
Password : giklop
Job : Staff
Type : Car
Plate No : VWY456
Parking At : FTSM
Park : Lot 5

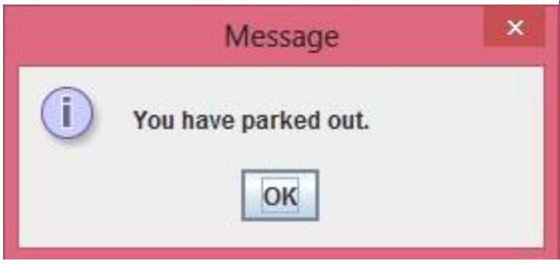
-Example view information in car lot 5.

- **Parked out**

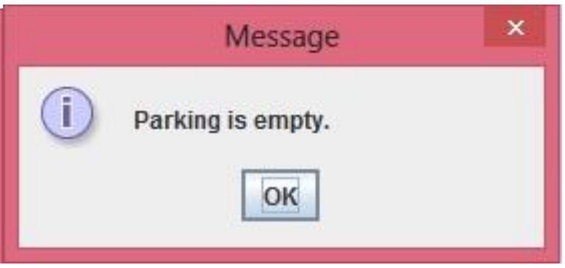




-This GUI make user to parked out from parking depend on the type of parking and parking lot that have been parked.



You have parked out.

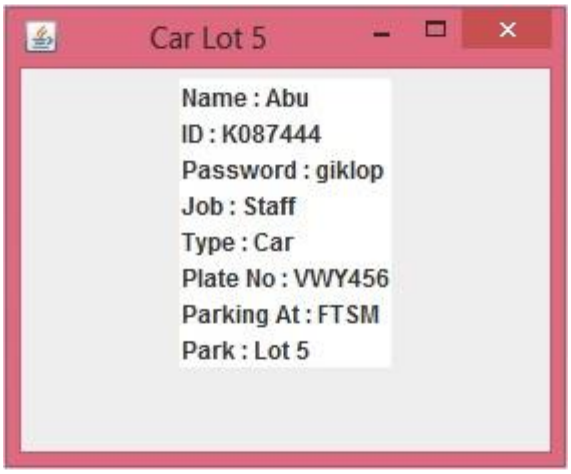


Parking is empty.

-If that lot have vehicle, dialog message (left GUI) will appear, else the right GUI will.

5. File input and output format

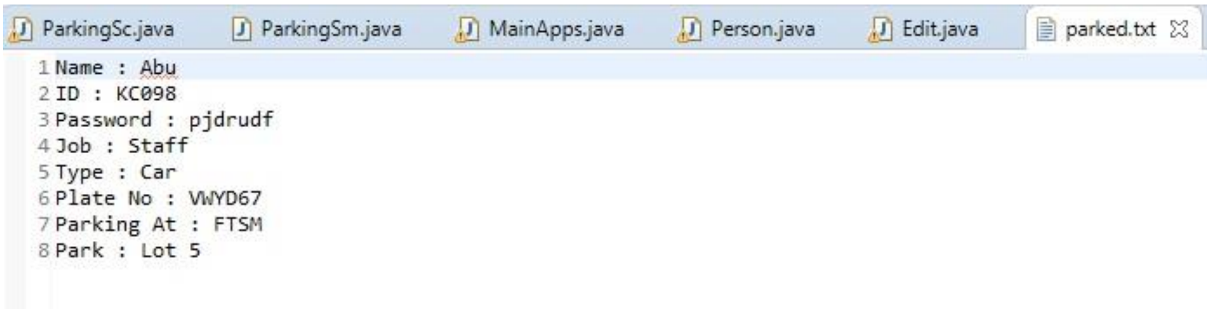
- I use text file to recorded and store the file depend on the parking lot that user choose.



```
Name : Abu
ID : K087444
Password : giklop
Job : Staff
Type : Car
Plate No : VWY456
Parking At : FTSM
Park : Lot 5
```

-This is the example of the output in listparking.

-Every insert of the user will be in String variable.



```
Name : Abu
ID : KC098
Password : pjdrudf
Job : Staff
Type : Car
Plate No : VWYD67
Parking At : FTSM
Park : Lot 5
```

-Example of information in view file.

6. How To Use The Software

Guideline to run the system.

Follow this step to register :

1. First, make sure the path and folder given same as mention in introduction.
2. Select new button if you are new user.
3. Select add button to register. Then choose your job
4. Enter your name, id, and password.
5. Select vehicle. If you choose bus and lorry, a message box will appear that you can’t park at UKM.
6. If you are staff or management staff, you may select which type of parking you want which is public, premium park or premium m park.
7. Note that if you are student, it will directly take you to the choose lot parking depend on your vehicle.
8. After you enter with your favourite parking lot, the information will be recorded and save in a file.
9. It will back to main menu GUI.

Follow this step to view list parking information :

1. Select list of parking button.
2. Choose any type of parking and parking lot you want to view.
3. The system will show you the details of the information inside the file.

Follow this step to change vehicle owner :

1. Select Edit button.
2. Click change vehicle owner.
3. Enter your name and id.
4. Select where are you parked the vehicle previously.
5. Then, press enter when you done.
6. This process will update the information inside the file.

Follow this step to promote/demote staff :

1. Select Edit button.
2. Click Promote/Demote Staff button.
3. Select where are you parked the vehicle previously.
4. Then, press enter when you done.
5. This process will update the information inside the file.

Follow this step to parked out :

1. Select Remove button.
2. Select where are you parked the vehicle previously.
3. Then, press enter when you done.
4. This process will delete the file inside the folder.

7. Solution Constraint and Weakneses

-This program only can insert one owner at one time. So if the user have many vehicles to register, the I recommend the user to register again with same name, id and password but different plate no.

-Make sure the path of the file given saved as I mention in introduction so the system will saved your information.

-When successfully registered, if you open the file in file explorer, the arrangement of the data isn't appropriate to view. However, if you open it in eclipse IDE, the data display appropriately.

-The change vehicle owner have a little bit problem. After you change the owner, if you view it in eclipse, it will not display the data appropriately.