Object-Oriented Analysis and Design

車輛租賃管理系統

Car Rental Management System

Homework 6

Name	ID	E-mail
郭振洋	111599003	t111599003@ntut.org.tw
郭駿頤	111598034	t111598034@ntut.org.tw

Department of Computer Science & Information Engineering
National Taipei University of Technology

2023/05/25

目錄

1 Requirement Document

1.1 Change History

Revision	Description	Date
Iteration I		
1	Problem statement.	Mar 1, 2023
2	Problem Statement refinement. System Context Diagram. Summary of System Features. Use Case Diagram. Use Cases. Non-functional Requirements and Constraints. Glossary. Software Environments refinement.	Mar 15, 2023
3	Use Case Diagram Domain class diagram showing only concepts Add associations Add attributes	Mar 29, 2023
4	Use Case Diagram Domain class diagram showing only concepts Add associations Add attributes Logical Architecture Use-Case Realizations with GRASP Patterns Design Class Model	Apr 20,2023
5	Use Case Diagram Domain class diagram showing only concepts Add associations Add attributes Logical Architecture with UML package diagram Use-Case Realizations with GRASP Patterns Design Class Model Draw an implementation class diagram for system Show the difference between implementationclass model and design class model Calculate Line of Code Snapshots of system execution Source Code Listing(note: select and list the most	May 4,2023

	important source code Snapshots of testing result Unit Testing Code Listing	
Iteration I		
6	Use Cases refinement. Classes Identified refinement. Bad Classes refinement. Operation Sequence Diagram refinement. Design Class Model refinement.	May 24,2023

1.2 Problem Statement

提供業者進行欲出租車輛的資訊及管理業務，給需要進行車輛租賃的顧客一個線上預約的平台。

近年來租車風氣逐漸流行，許多車廠都有提供各式的共享車輛給大眾租用，但在使用車輛的過程，大眾並無法得知目前車輛的保養情況，安全性上抱持著疑慮。

顧客可以透過系統向業者進行車輛租賃預約，並且能了解業者現有車輛的保養狀態，使顧客可以更了解欲租用車輛的情況，提升車輛使用的安全；業者也可以透過車輛租賃管理系統，更輕鬆且快速的管理現有車輛的目前的保養以及租賃狀況。

本軟體透過C++撰寫本地端的圖形化介面(Graphical User Interface,GUI)租賃系統，並與Database進行結合，在車廠提供車輛出租的同時，也可以進行用戶的評價與使用分析，供車廠進行車輛配置調整，更符合大眾的口味。

1.3 Summary of System Features

Feature ID	Description
FEA-01	上傳出租車輛資訊
FEA-02	提供車輛保養維修診斷
FEA-03	提供車輛評價統計分析
FEA-04	透過系統進行租車
FEA-05	檢視目前車輛的車況資訊
FEA-06	剔除需要進行保養維修的車輛

1.4 Use Case Diagram

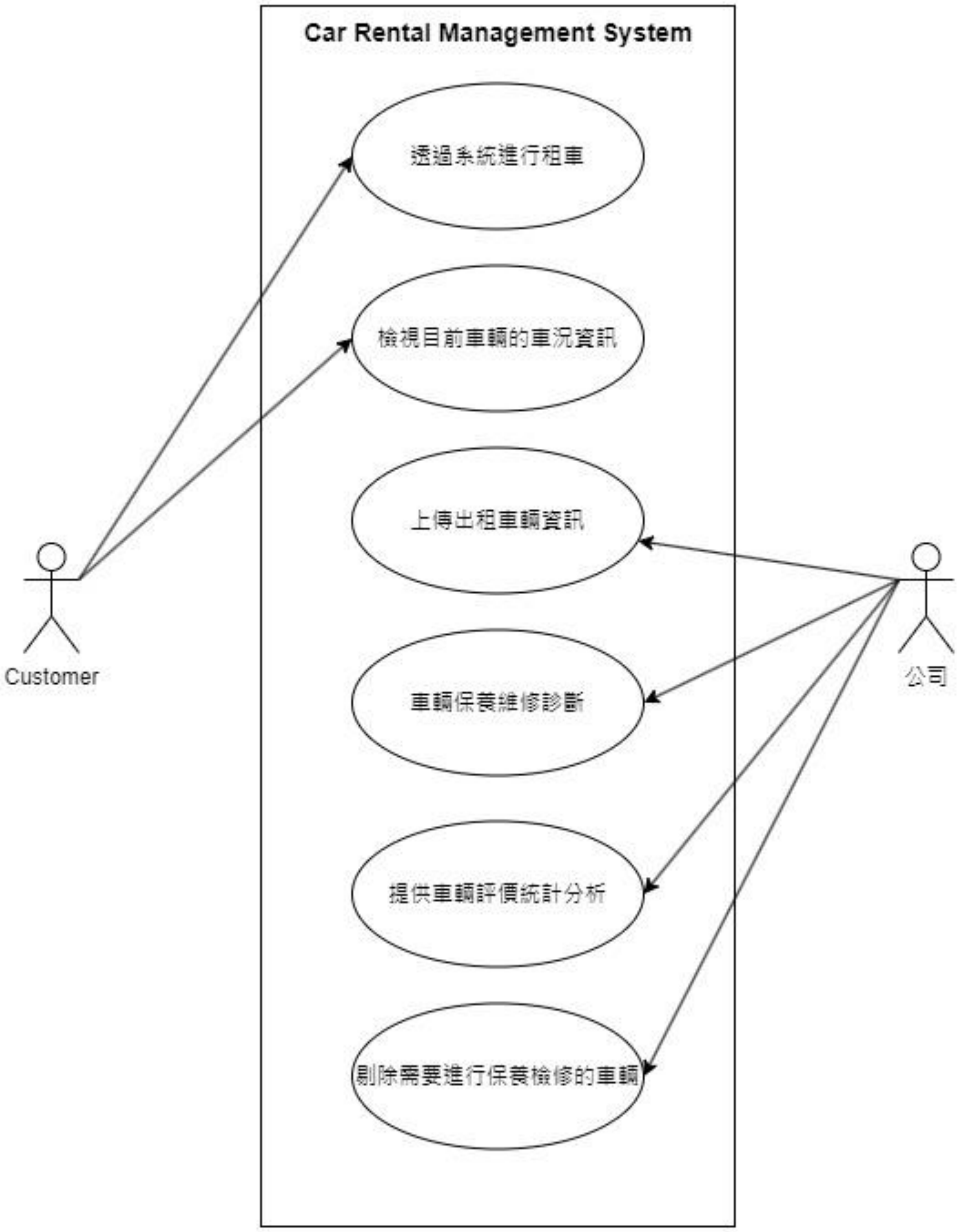


Figure 1. User Case Diagram

1.5 Use Cases

1.5.1 上傳出租車輛資訊

Scope	Rental System
Level	User-goal
Primary Actor	公司
Stakeholders and Interests	1. 公司
Preconditions	1. 公司需要可以登入Rental System後臺權限，且可以進行管理的帳號。 2. Rental System需要打開伺服器，提供服

	務。
Success Guarantee	1. 公司成功輸入欲出租的車輛資訊。 2. Rental System成功紀錄欲出租的車輛資訊。
Main Success Scenario	1. 進入上傳的頁面。 2. 輸入出租車輛的各式訊息。 2a. 出租車輛的廠牌。 2b. 出租車輛的型號。 2c. 出租車輛的可容納人數。 2d. 出租車輛的公里數。 2e. 出租車輛的保養情況。 2f. 出租車輛的圖片。 2g. 出租金額。 3. 確認系統各欄位資訊皆為正確。
Extensions	2a. 員工沒有輸入廠牌： 1. 此欄提示錯誤資料。 2b. 員工輸入不存在(錯誤)的廠牌： 1. 系統查無此廠牌，此欄提示錯誤資料。 2c. 員工沒有輸入型號： 1. 此欄提示錯誤資料。 2d. 員工輸入存在的廠牌，但是不存在的型號： 1. 系統查無此廠牌的型號，此欄提示錯誤資料。 2e. 員工沒有輸入容納人數： 1. 此欄提示錯誤資料。 2f. 員工輸入負數： 1. 此欄提示錯誤資料。 2g. 員工沒有輸入公里數： 1. 此欄提示錯誤資料。 2h. 員工輸入負數： 1. 此欄提示錯誤資料。 2i. 員工沒有上傳出租車的情況： 1. 此欄提示錯誤資料。 2j. 員工沒有上傳出租車的圖片： 1. 此欄提示錯誤資料。 2k. 員工沒有輸入出租金額：

	1. 此欄提示錯誤資料。 2l. 員工輸入負數： 1. 此欄提示錯誤資料。 3a. System偵測到有欄位是錯誤資料: 1. 員工修改錯誤欄位的資料。
Special Requirements	None
Technology and Data Variations List	None
Frequency of Occurrence	Often
Open Issues	

1.5.2 車輛保養維修診斷

Scope	Repair System
Level	User-goal
Primary Actor	公司
Stakeholders and Interests	1. 公司
Preconditions	1. 公司需要可以登入Repair System後臺權限，且可以進行管理的帳號。 2. Repair System需要打開伺服器，提供服務。
Success Guarantee	1. Repair System成功更新車子最新情況。 2. 公司收到車輛維修報告。
Main Success Scenario	1. 公司進入後台點選Repair Car選項，進入對應的頁面。 2. 公司檢查車輛狀況，上傳車輛資訊。 2a. 檢查Car engine 2b. 檢查Car battery。 2c. 檢查Car tire。 3. 回報公司，車子的維修告。
Extensions	1. 顯示需要檢查 2a. 沒有檢查車輛引擎 2b. 沒有檢查車輛電池: 1. 顯示需要檢查 2a. 沒有檢查車輛輪胎:

	1. 顯示需要檢查 3. 存在檢查項目尚未完成檢查 1. 顯示需要檢查
Special Requirements	None
Technology and Data Variations List	None
Frequency of Occurrence	Often
Open Issues	

1.5.3 車輛評價統計分析

Scope	Analyze System
Level	User-goal
Primary Actor	Analyze System
Stakeholders and Interests	1. 公司
Preconditions	1. Analyze System需要打開伺服器，提供服務。
Success Guarantee	1. Analyze System統計3個月所有的車輛評價，產生統計報告。 2. 公司取得統計報告，方便進行商業決策。
Main Success Scenario	1. Analyze System取得Database的三個月內，所有車輛的評價。 2. Analyze System進行統計分析，產生報告。 3. Analyze System回報統計的分析。
Extensions	1a. 3個月紀錄的車子存在不合格的車輛: 1. 過濾掉零件和里程數是不合格的車輛。
Special Requirements	None
Technology and Data Variations List	None
Frequency of Occurrence	Sometime
Open Issues	

1.5.4 透過系統進行租車

Scope	Rental System
Level	User-goal
Primary Actor	Rental System
Stakeholders and Interests	1. Rental System 2. Customer 3. 公司

Preconditions	1. Rental System需要打開伺服器，提供服務。 2. Customer需先註冊並登入Rental System
Success Guarantee	1. Customer成功於系統中預定要使用車輛的時間與車輛。 2. Rental System將Customer的訂單內容回傳給公司。
Main Success Scenario	1. Customer登入租車系統 2. Customer選擇需要使用車輛的日期與時間區間。 3. Rental System 依照 Customer所需要使用車輛的區間，列出可提供出租的車輛資訊以及租金。 4. Customer選擇想要承租的車輛。 5. Customer確認訂單中選擇的車輛、時間、租金後送出訂單。 6. Rental System將訂單內容回傳給公司。
Extensions	1a. 無法登入系統: 1.重新登入或註冊。 2a. 選擇的區間沒有閒置可用的車輛: 1. 選擇其他的使用區間。 6a. 回傳訂單資料失敗: 1. 重新送出訂單。
Special Requirements	None
Technology and Data Variations List	可租賃的車輛、訂單資料
Frequency of Occurrence	Often
Open Issues	

1.5.5 檢視目前車輛的車況資訊

Scope	Rental System
Level	User-goal
Primary Actor	Customer
Stakeholders and Interests	1. Customer
Preconditions	1. Rental System需要打開伺服器，提供服務
Success Guarantee	1. Customer成功於系統中查詢到要使用車輛的車況資訊。
Main Success Scenario	1. Customer登入租車系統。

	2. Customer選擇自己已預訂的訂單。 3. Rental System列出所預定車輛的車況資訊。
Extensions	1a. 無法登入系統: 1.重新登入或註冊。 2a. 並未有已預定的訂單 1. 預約成功後再進行查詢。
Special Requirements	None
Technology and Data Variations List	車輛保養資料
Frequency of Occurrence	Often
Open Issues	

1.5.6 剔除需要進行保養維修的車輛

Scope	Rental System
Level	User-goal
Primary Actor	Rental System
Stakeholders and Interests	1. Rental System 2. 公司
Preconditions	1. Rental System需要打開伺服器，提供服務
Success Guarantee	1. Rental System將已到達保養里程或需要維修的車輛下架。 2. Rental System通知公司對需要進行保養或維修的車輛進行保養維修作業。
Main Success Scenario	1. Rental System檢查所有出租車輛的里程及車輛狀況。 2. Rental System對需要保養或維修的車輛進行下架。 3. Rental System通知公司對需要進行保養或維修的車輛進行保養維修作業。
Extensions	1a. 未有到達保養週期或是維修的車輛: 1. None。
Special Requirements	None
Technology and Data Variations List	需要保養的車輛列表以及事項
Frequency of Occurrence	Sometime
Open Issues	

1.6 Non-functional Requirements and Constraints

NFR ID	Category	Description
NFR-01	Usability	提供使用者介面方便出租與租車。

NFR-02	Performance	從Database中讀取資料不超過3秒鐘。
NFR-03	Portability	支援多個作業系統，達到跨平台的目的。
NFR-04	Reliability	資料即時同步，可獲得最新的車輛資訊。
NFR-05	Modifiability	物件導向設計開發。
NFR-06	Testability	Code Coverage 至少 85%以上。

1.7 Glossary

Item	Definition or Description
GUI	提供用戶圖形化介面(Graphical User Interface)，可以更加簡單且方便的進行租車系統的操作。
統計分析	以客戶評價來分析用戶租用的車輛情況，以及分析用戶喜好的車型進行統計。
UID	替所有的租賃車輛紀錄唯一識別碼Unique identifier(UID)，區分所有車輛。

1.8 Software Environments

該軟體為達到跨平台的目的，所以 C++程式語言進行設計，並結合Mysql資料庫進行車輛與數據的儲存作業。

2 Domain Class Model

2.1 Domain class diagram showing only concepts

2.1.1 Classes Identified

Analyze System	Rental System	Information	Car
Repair System	Parts	Maintainance	Evaluate
Customer Information	Order	Report	Submit
Platform	Car’s milage	Car’s state	

首先在Use Case中列出所有相關的單詞

2.1.2 Bad Classes

Attributes	Vague	Operation	Roles	Implementation Construction
Parts		Evaluate	Analyze System	Information
Customer Information		Order	Rental System	Report
Car’s milage		Submit	Repair System	Car
Car’s state		Maintainance	Platform	

Attributes : 列為屬性
Vague : 定義模糊不清
Operation : 操作步驟中的一部分
Roles : 角色介紹
Implementation Construction : 時做產生的結構

2.1.3 Good Classes

Analyze System	Rental System	Repair System	Report
Car	Parts	Customer	Platform

其餘的Good Classes

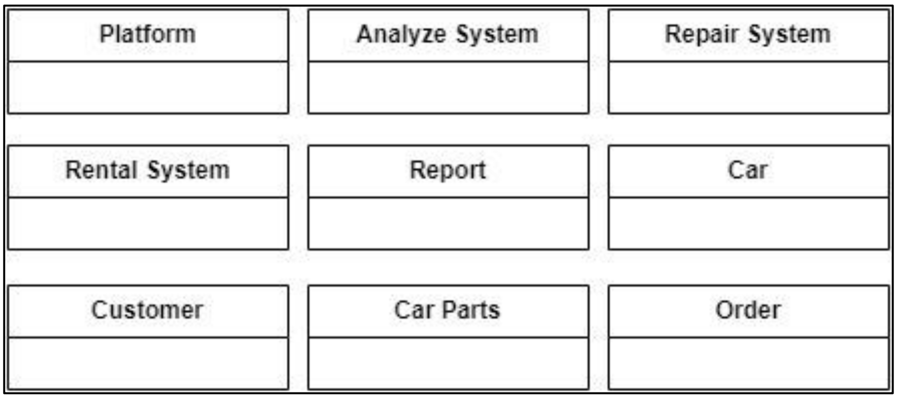


Figure 2-1. Domain class diagram showing only concepts

2.2 Add associations

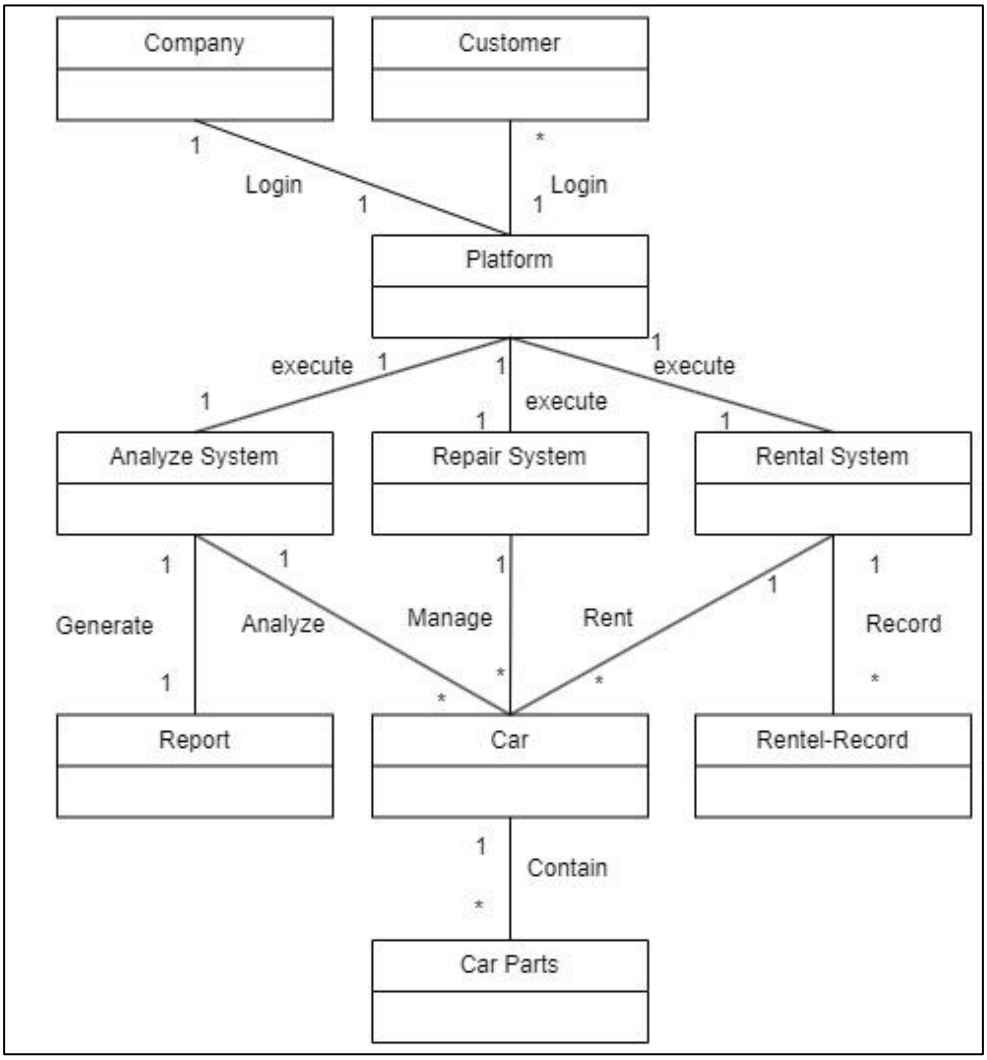


Figure 2-2. Domain Class Diagram with associations

Relationships
Platform 管理 Analyze System、Repair System、Rental System。
Analyze System 分析租車後的回饋訊息，產生 Report用於商業分析。
Repair System 管理車輛的保養維修資訊，供承租人與公司檢閱。
Rental System 管理出租車的訂單資訊。

2.3 Add attributes

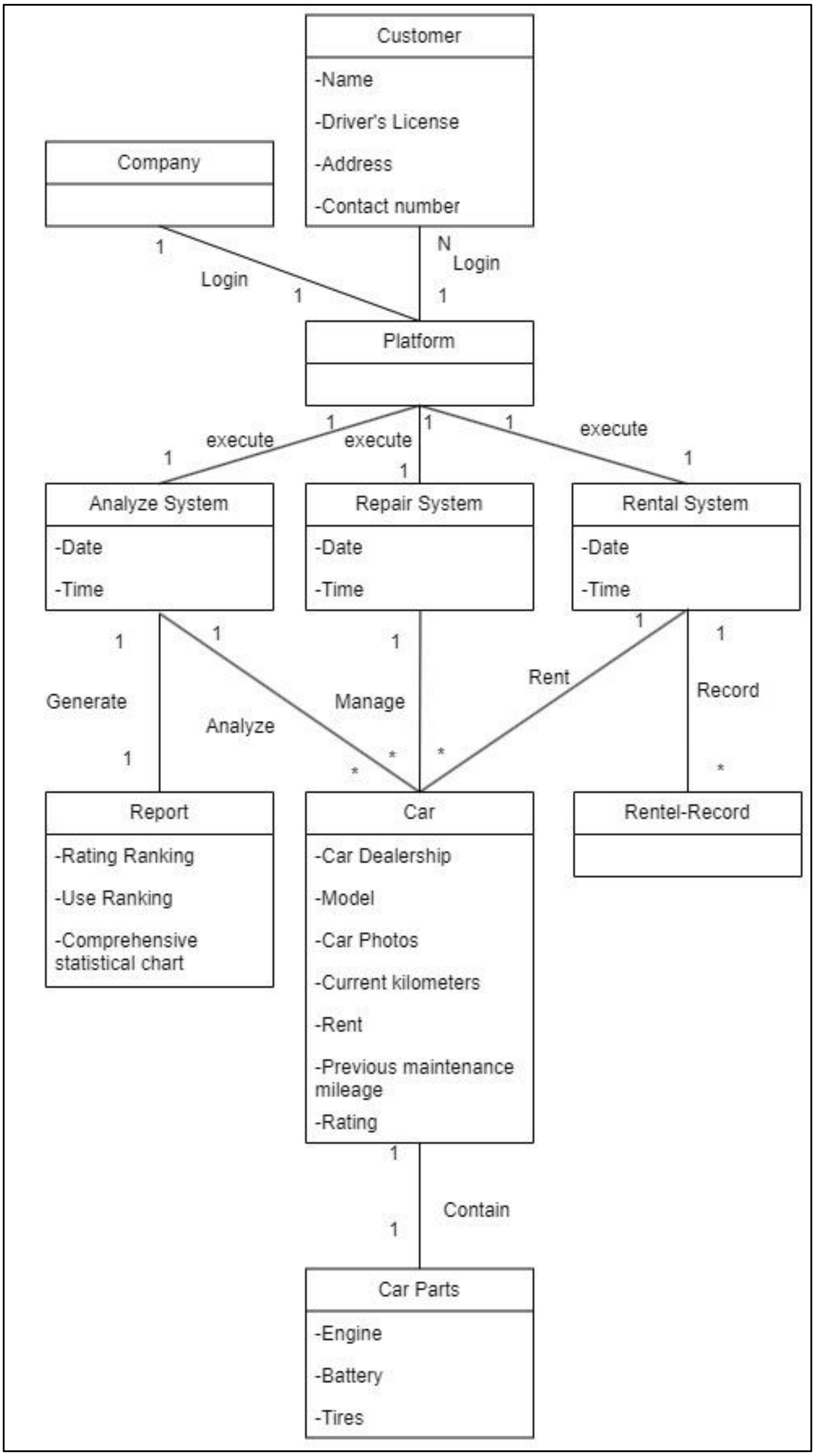


Figure 2. Domain Class Diagram with Attributes

3 Design

3.1 Logical Architecture

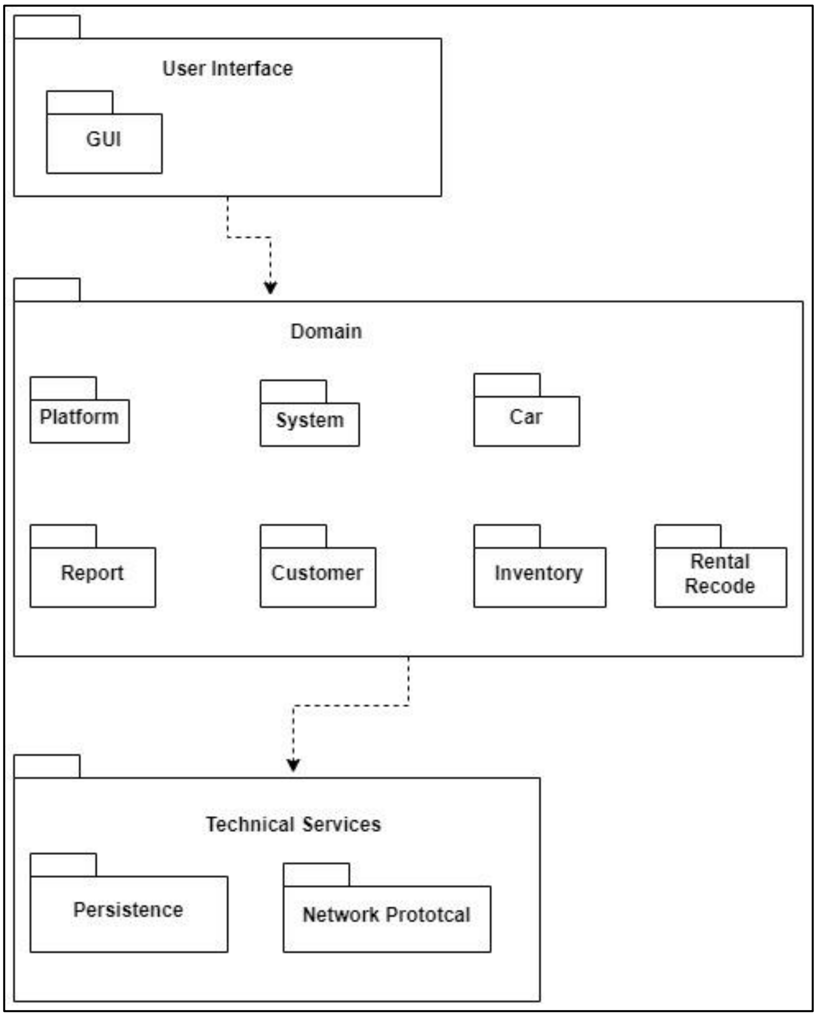


Figure 3-1 Logical Architecture in UML package diagram

3.2 Use-Case Realizations with GRASP Patterns

3.2.1 System Sequence Diagram

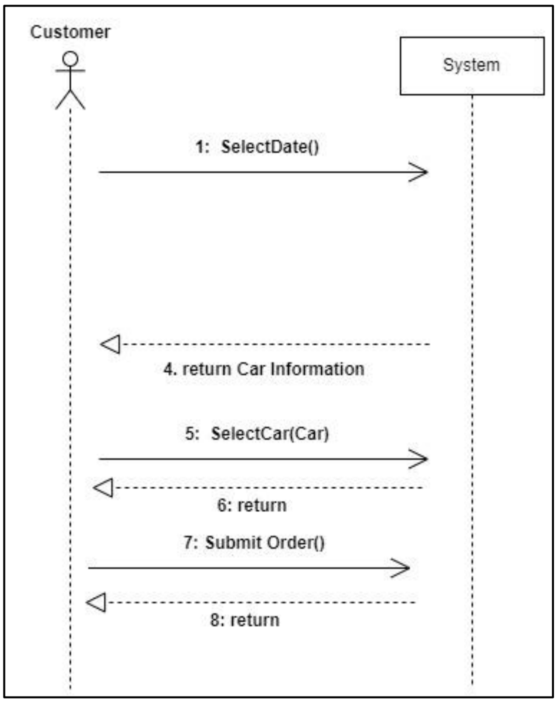


Figure 3-2 System Sequence Diagram of use case:透過系統進行租車

3.2.2 Operation Sequence Diagram

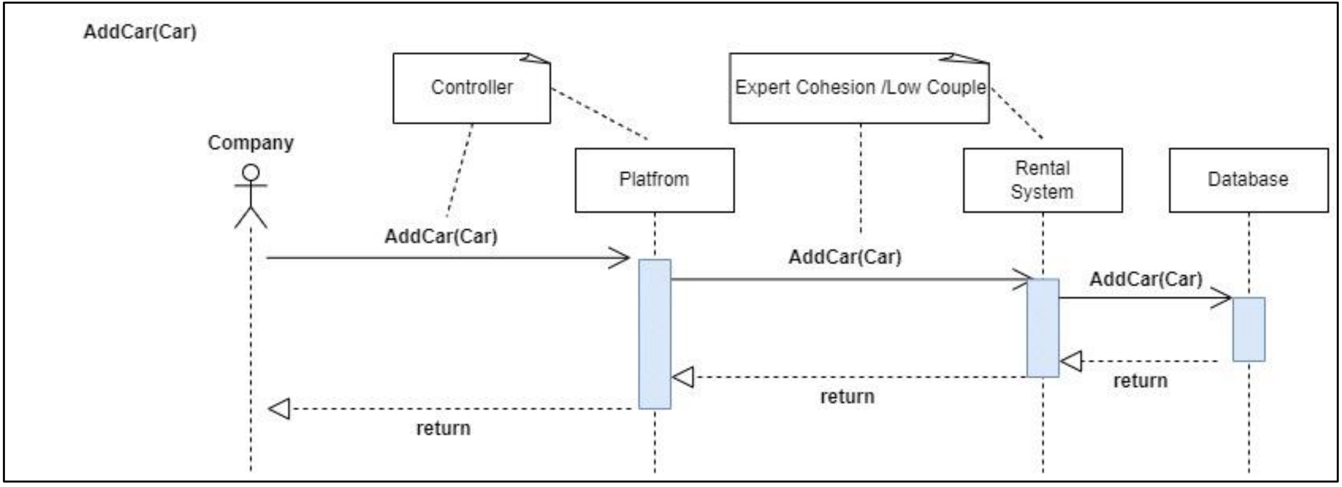


Figure 3-3 Sequence Diagram : Add Car

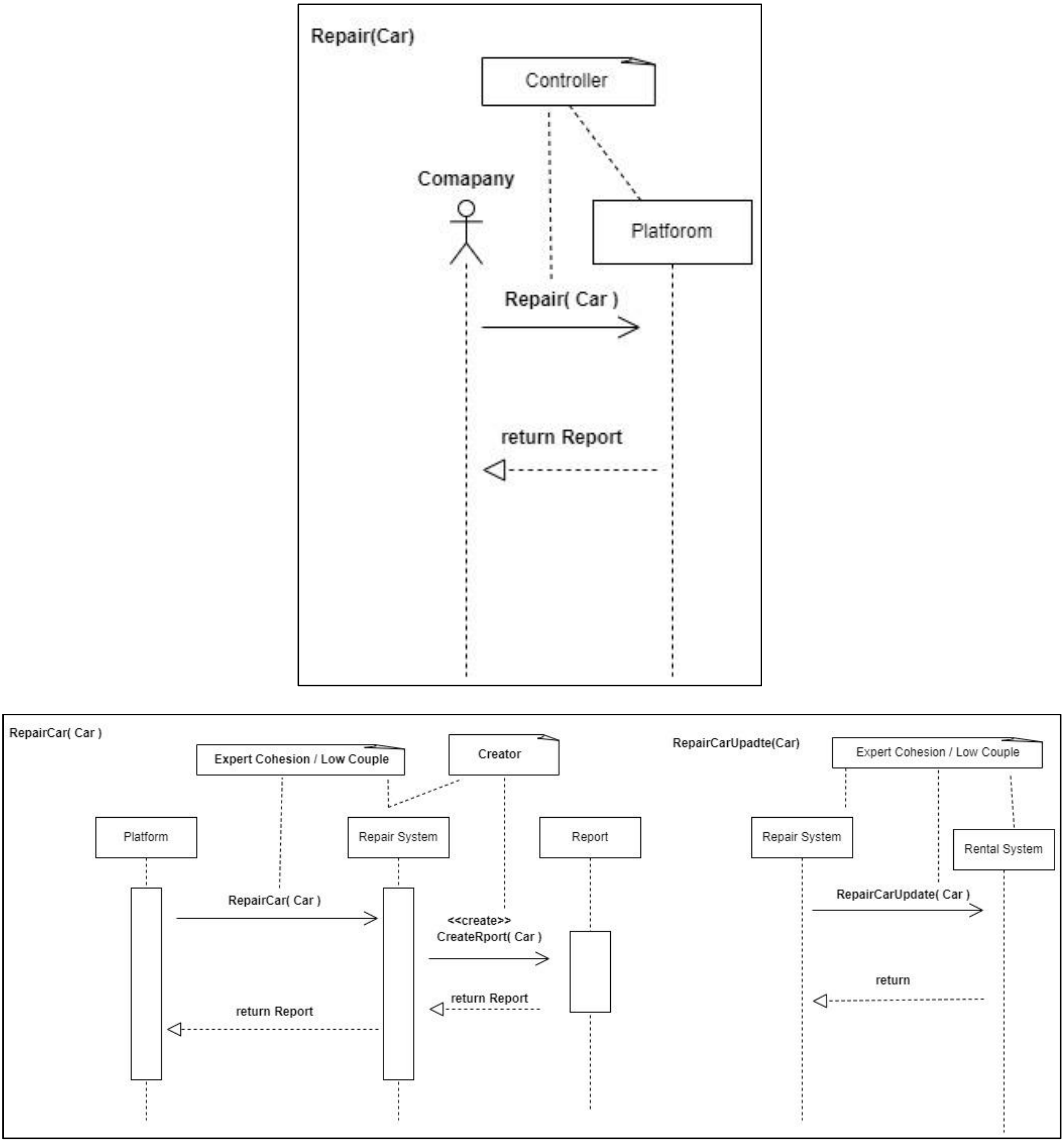


Figure 4-6 Sequence Diagram : Repair Car

- Repair(Car):calss Platform在系統的功能是接收request，並轉發到正確的subsystem，所以是Controller。

- RepairCar(Car):platform和Report沒有關係，但是可以產生Report，所以是Hight Cohesion / Low Couple。

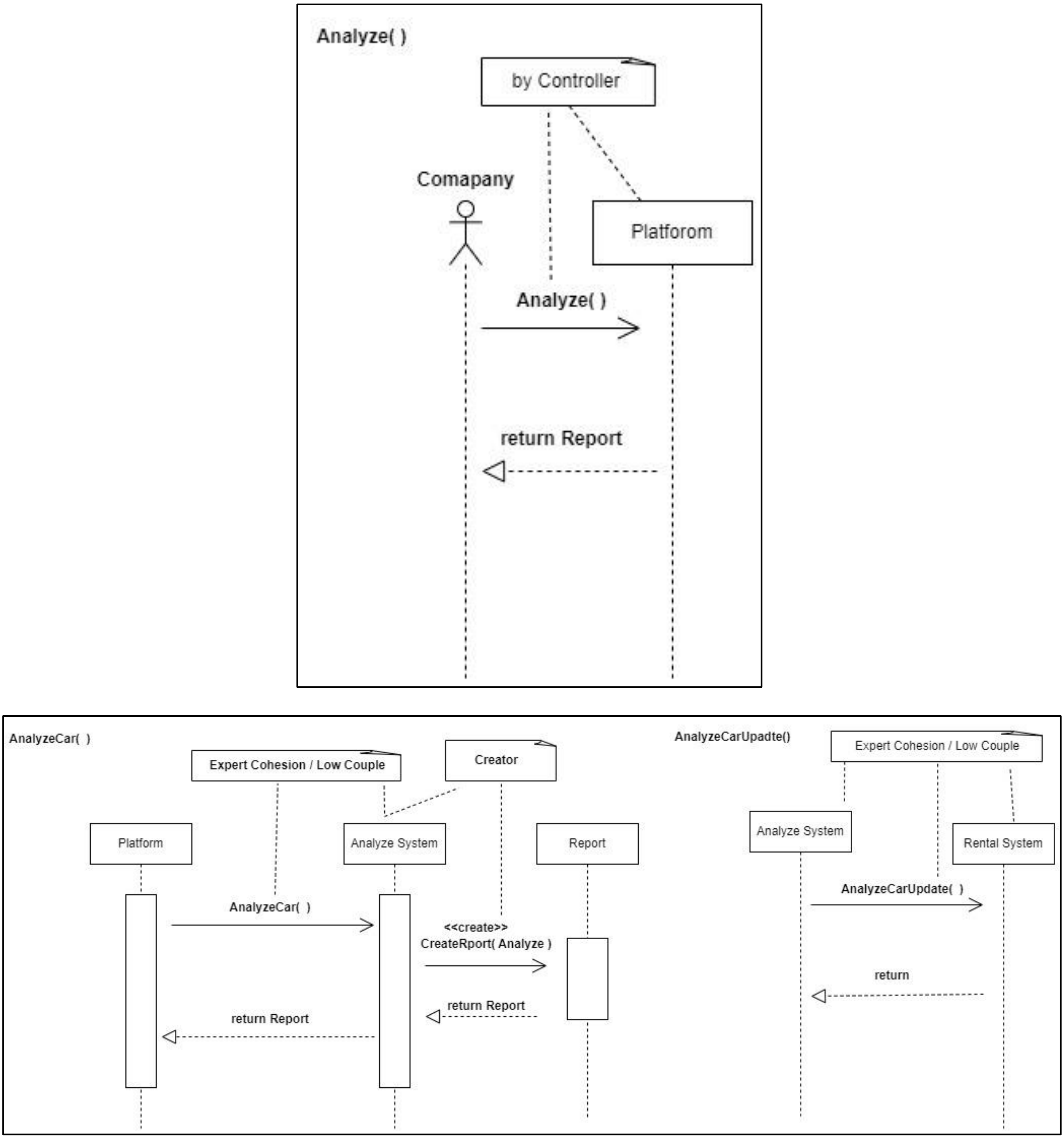


Figure 4-7 Sequence Diagram : Analyze Car

- Analyze():calss Platform在系統的功能是接收request，並轉發到正確的subsystem，所以是Controller。

- AnalyeCar(Car):platform和Report沒有關係，但是可以產生Report，所以是Hight Cohesion / Low Couple。

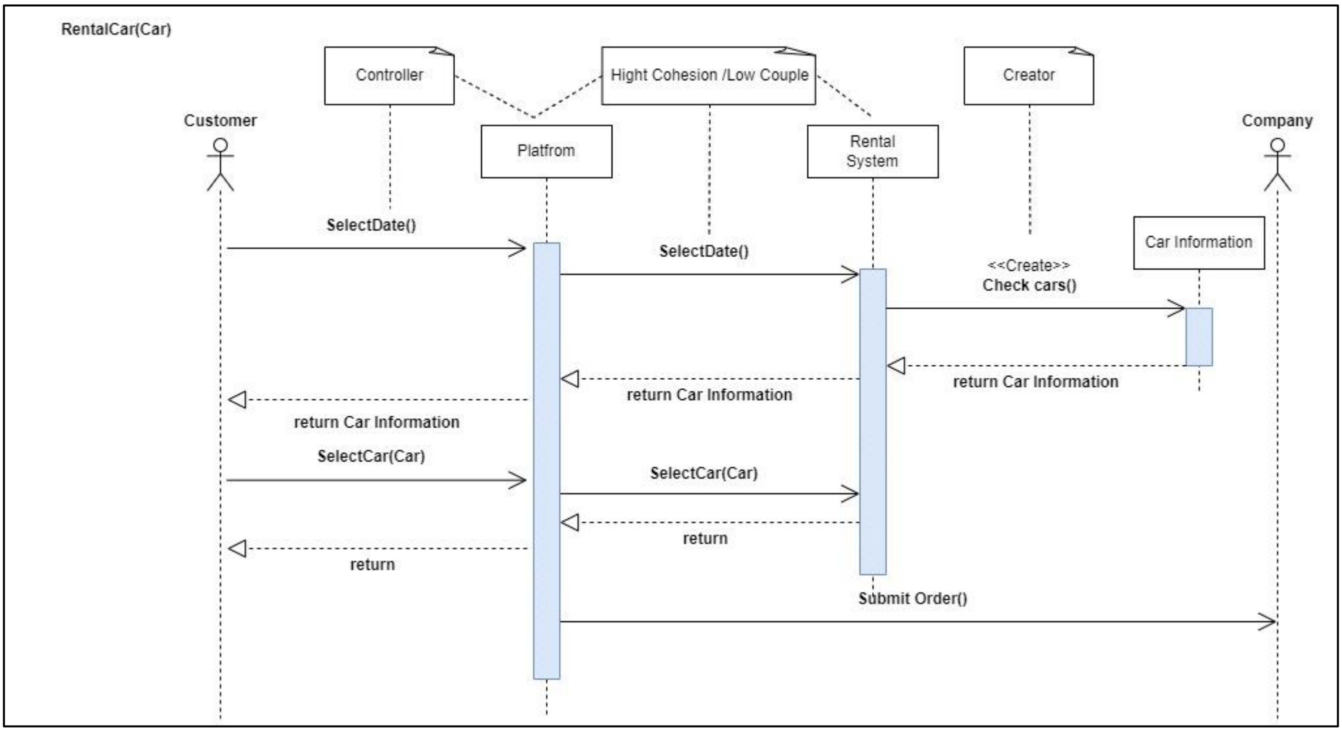
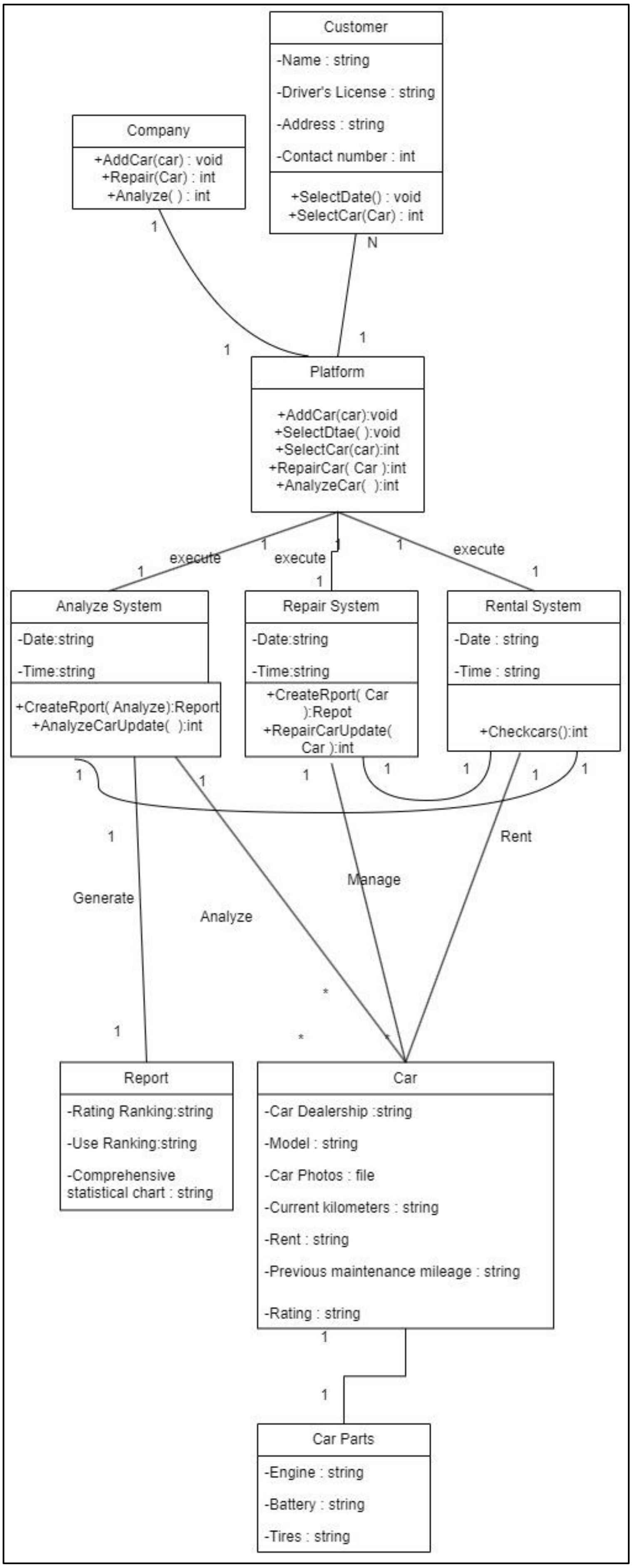


Figure 4-8 Sequence Diagram : Rental Car

3.3 Design Class Model


Customer
-Name : string
-Driver's License : string
-Address : string
-Contact number : int
+SelectDate() : void
+SelectCar(Car) : int
Company
+AddCar(car) : void
+Repair(Car) : int
+Analyze( ) : int
1
N
1
1
Platform
+AddCar(car):void
+SelectDtae( ):void
+SelectCar(car):int
+RepairCar(Car):int
+AnalyzeCar( ):int
execute
execute
execute
1
1
1
1
1
1
Analyze System
-Date:string
-Time:string
+CreateRport(Analyze):Report
+AnalyzeCarUpdate( ):int
Repair System
-Date:string
-Time:string
+CreateRport(Car):Repot
+RepairCarUpdate(Car):int
Rental System
-Date : string
-Time : string
+Checkcars():int
1
1
1
1
1
1
1
1
Generate
Analyze
Manage
Rent
1
*
*
*
Report
-Rating Ranking:string
-Use Ranking:string
-Comprehensive statistical chart : string
Car
-Car Dealership :string
-Model : string
-Car Photos : file
-Current kilometers : string
-Rent : string
-Previous maintenance mileage : string
-Rating : string
1
1
Car Parts
-Engine : string
-Battery : string
-Tires : string

4 Implementation Class Model

4.1 Draw an implementationclass diagram for your system (including associations, attributes and methods)

4.2 Show the difference between implementationclass model and design class model

4.2.1 Table: Comparison with design and implementation class

Class	Method	Design	Imp.
Platfrom	AddCar()	Yes	Yes
	SelectDate()	Yes	No
	SelectCar()	Yes	No
	RepairCar()	Yes	No
	AnalyzeCar()	Yes	No
	check_car_info()	No	Yes
Analyze System	CreateReport()	Yes	No
	AnalyzeCarUpdate()	Yes	No
Repair System	RepairCarUpdate()	Yes	No
Rentel System	Checkcars()	Yes	No

4.2.2 Table: Summary of implementation class/method changed

	Number of added	Number of removed	Number of modified
Class	10	0	8
Method	20	0	13

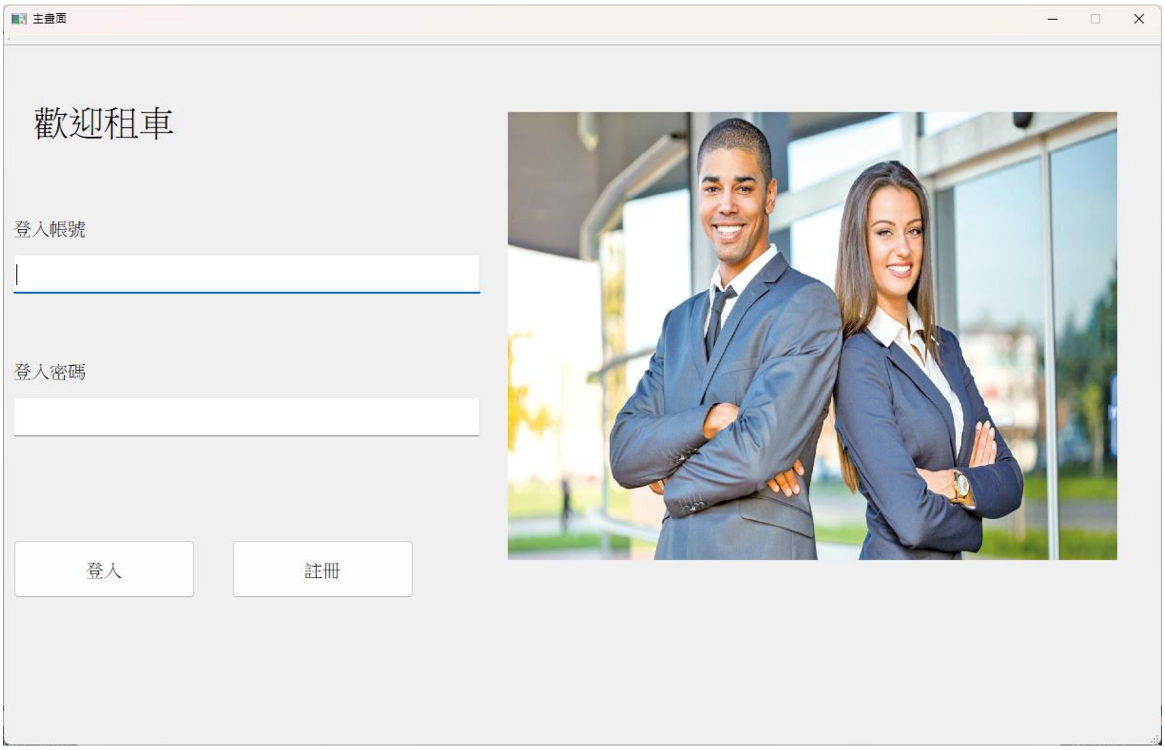
4.3 Calculate Line of Code

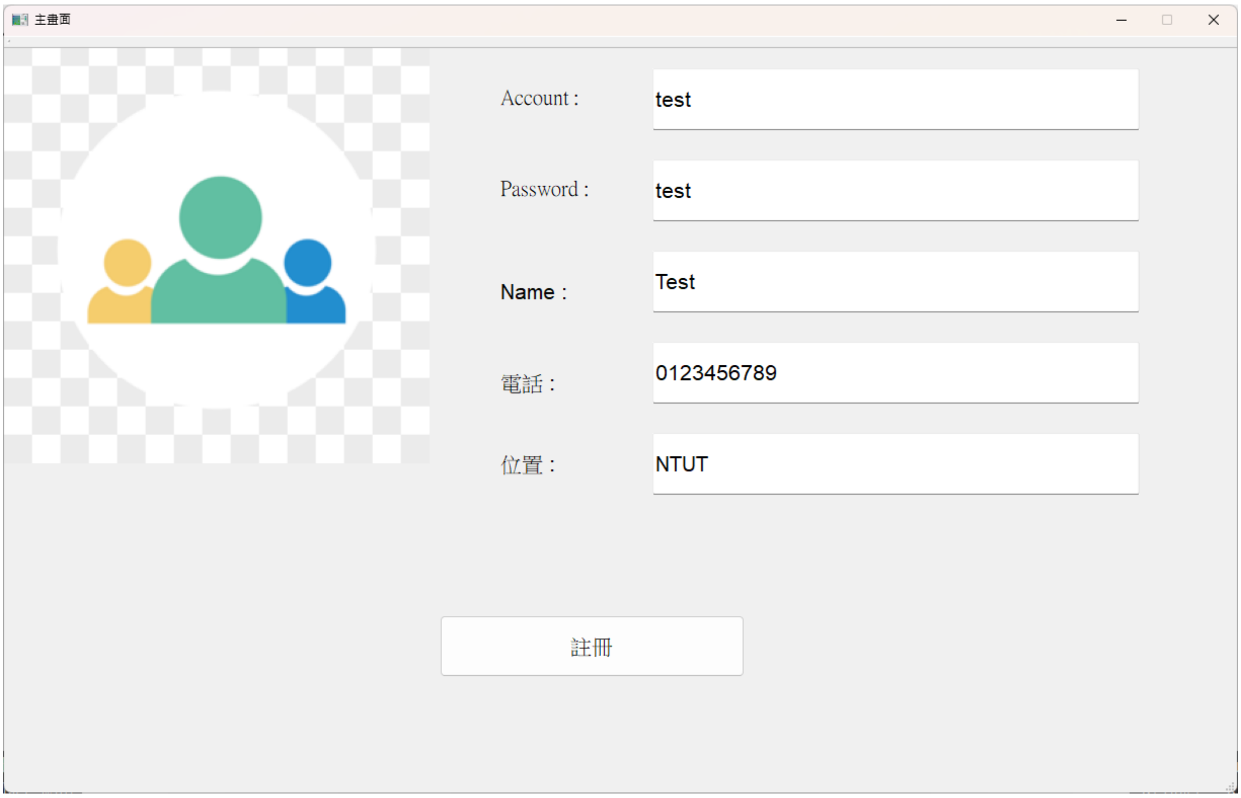
4.3.1 Table: Line of Code of classes

No	Method	Number of methods	Line of Code in Class (without comment)
1	Platfrom	10	221
2	Analyze System	2	6
3	Repair System	3	3
4	Rentel System	2	3

5 Programming

5.1 Snapshots of system execution





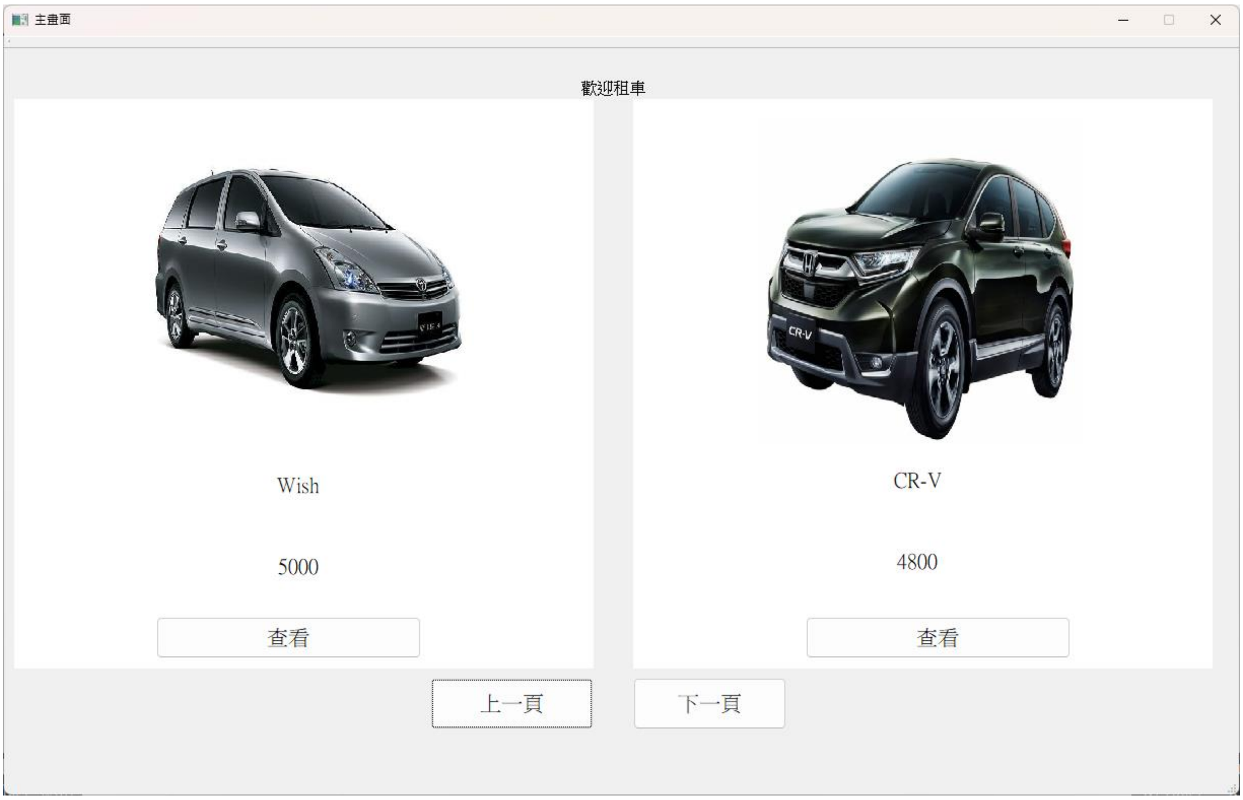
Account : test

Password : test

Name : Test

電話 : 0123456789

位置 : NTUT

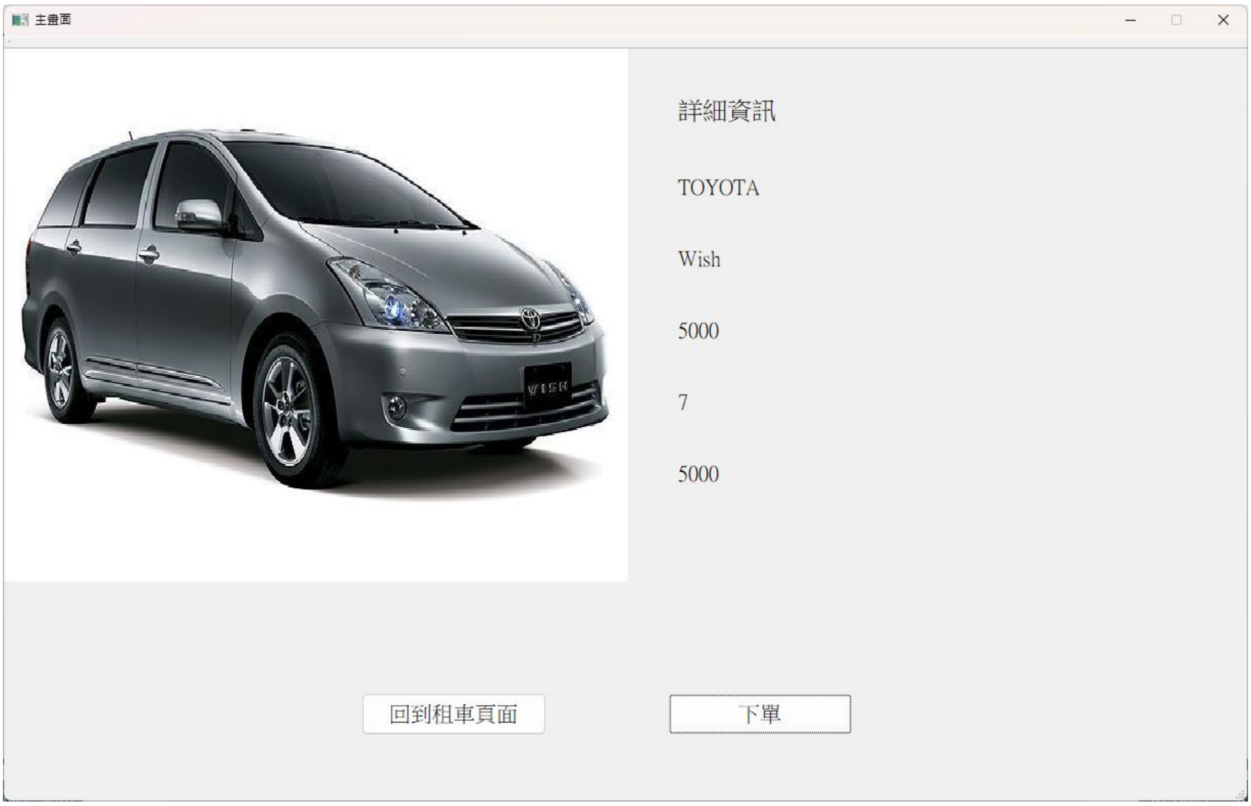


Wish

5000

CR-V

4800



詳細資訊

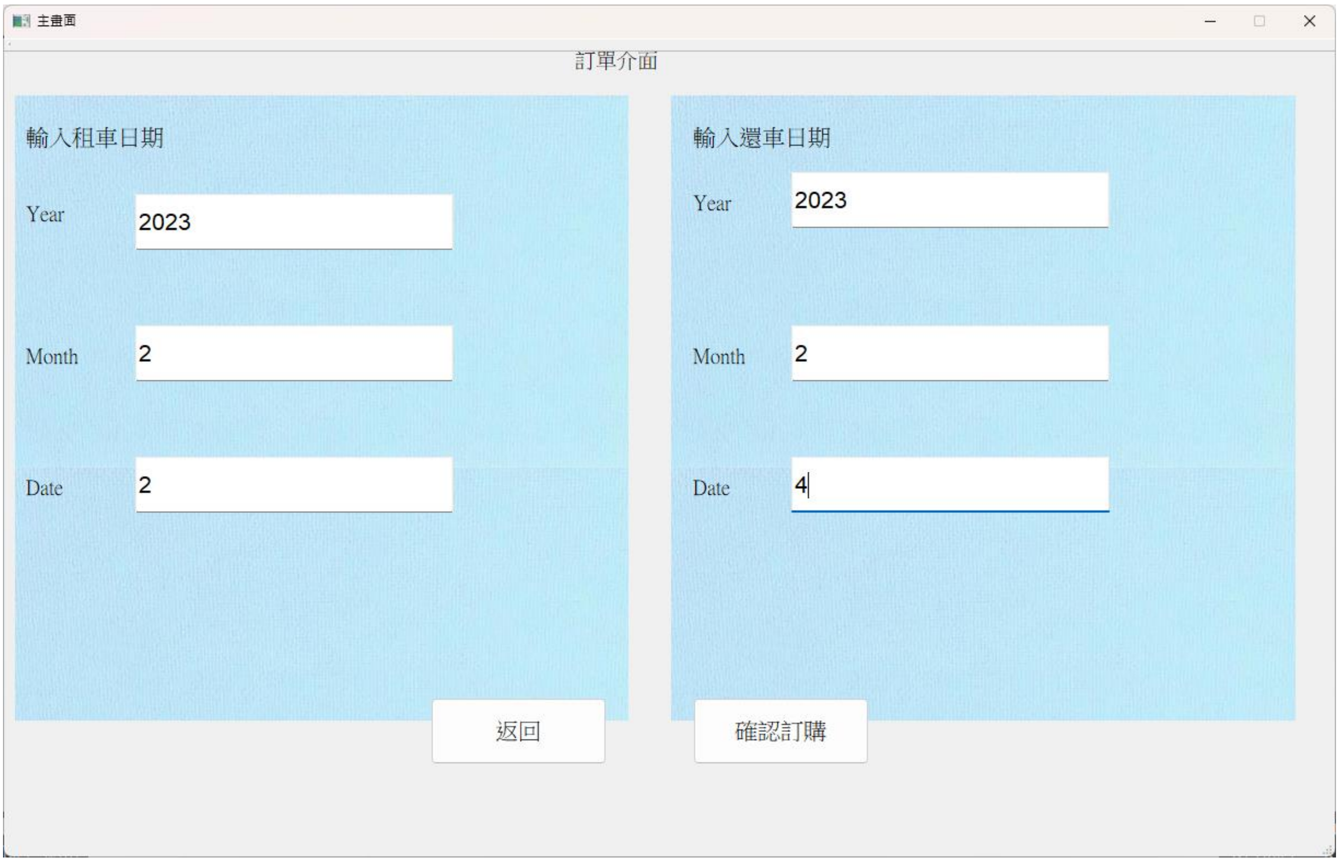
TOYOTA

Wish

5000

7

5000



訂單介面

輸入租車日期

Year 2023

Month 2

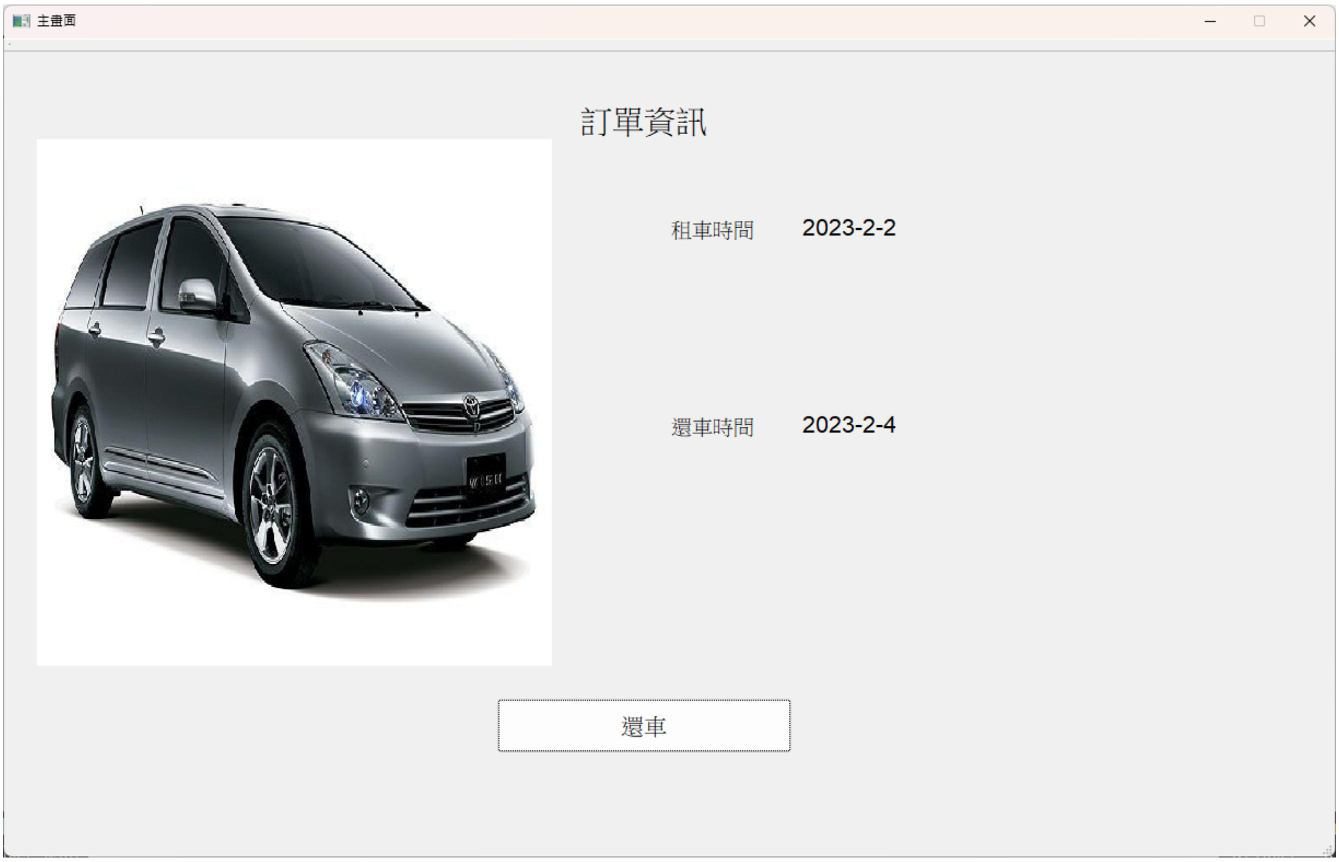
Date 2

輸入還車日期

Year 2023

Month 2

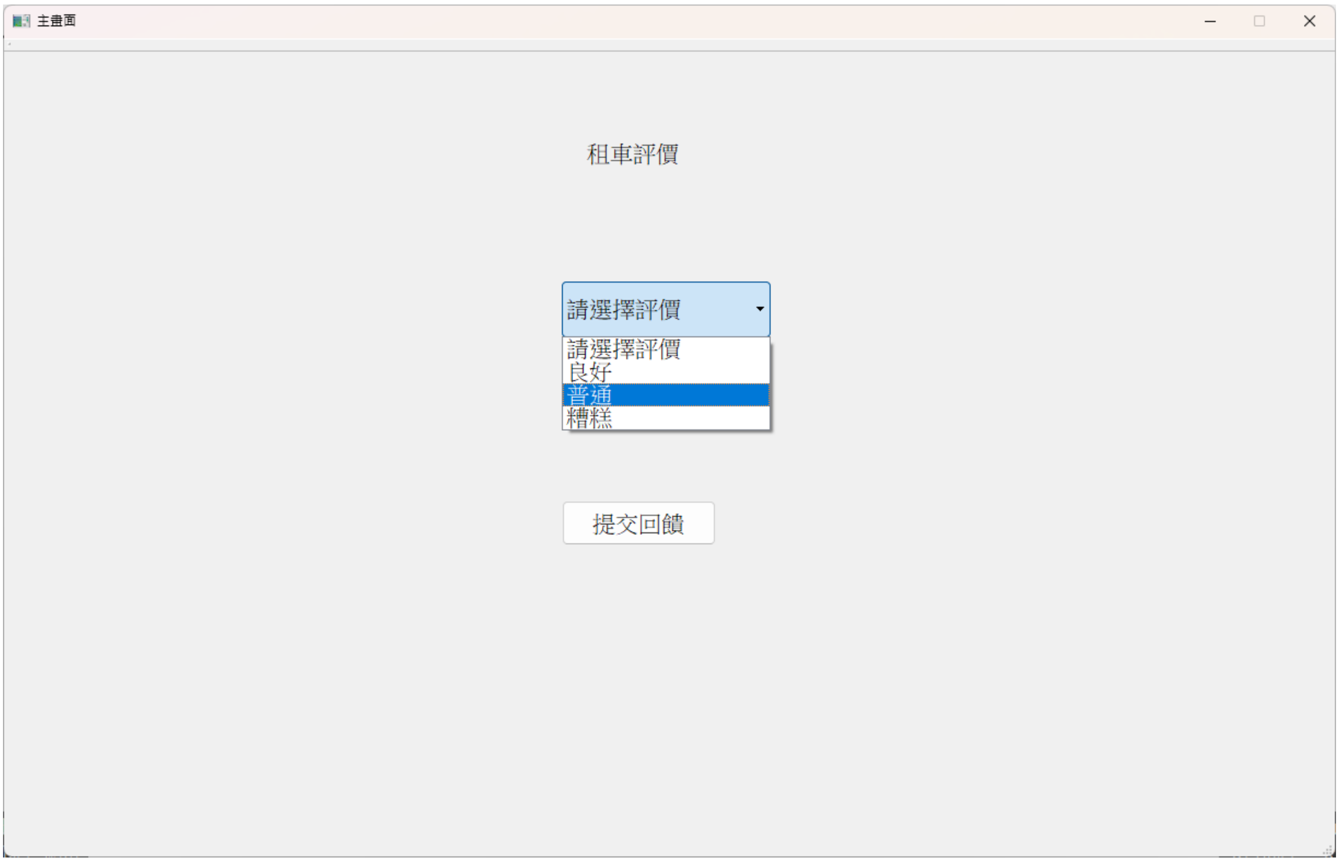
Date 4



訂單資訊

租車時間 2023-2-2

還車時間 2023-2-4



5.2 Source Code Listing

5.2.1 Main.cpp

```cpp
#include "test1.h"
#include "ui_test1.h"
#include <QDebug>
#include <QString>
#include <QFileDialog>


Test1::Test1(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::Test1)
{
    ui->setupUi(this);
    this->setFixedSize(1100, 700);   //fix screen size
    this->setWindowTitle("上傳車輛資訊");

}

void Test1::on_BrandLine_textEdited(const QString &arg1)
{
    BrandInput = arg1;
}
```

```
void Test1::on_KmLine_textEdited(const QString &arg1)
{
    KmInput = arg1;
}

void Test1::on_MaxSeatLine_textEdited(const QString &arg1)
{
    MaxSeatLine = arg1;
}

void Test1::on_ModelLine_textEdited(const QString &arg1)
{
    ModelLine = arg1;
}

void Test1::on_RentalPriceLine_textEdited(const QString &arg1)
{
    RentalPriceLine = arg1;
}

void Test1::on_comboBox_currentIndexChanged(const QString &arg1)
{
    CarStatus = arg1;
}

void Test1::on_submitPicture_clicked()
{
       QFileDialog fileDialog;
       fileDialog.setWindowTitle("開啟檔案");
       fileDialog.setDirectory("C:\\");
       fileDialog.setNameFilter("圖片 (*.png *.jpg *.bmp)");
       fileDialog.setFileMode(QFileDialog::ExistingFile);
       filePath = fileDialog.getOpenFileName();
       qDebug() << "所選擇的檔案為：" << filePath;

       QLabel *pictureLabel = ui->pictureLabel;
       pictureLabel->setText(filePath);
}

int Test1::save_file(QString filePath){
    int r = 0;
    if (!filePath.isEmpty()) {
        QFile file(filePath);
        if (file.open(QIODevice::ReadOnly)) {
            newFilePath =
"C:/Users/CSS_Tim/Desktop/OOAD/CarRentalProject/image/"+file.fileName().split('/').
last();
            QFile newFile(newFilePath);
            if (newFile.open(QIODevice::WriteOnly | QIODevice::Truncate |
QIODevice::Unbuffered)) {
                QDataStream out(&newFile);
                out.setByteOrder(QDataStream::LittleEndian);
                out.writeRawData(file.readAll().constData(), file.size());
                newFile.close();
                qDebug() << "儲存檔案成功：" << newFilePath;
            }
            file.close();
        }
        r = 1;
    }
    return r;
}

void Test1::Rental_car_info(){
    car.set_car(BrandInput, KmInput, ModelLine, CarStatus, MaxSeatLine,
RentalPriceLine, newFilePath);
    // db.saveData(car.get_BrandInput(), car.KmInput)
}

int Test1::regex_label(QString str){
    int r = 0;
    int len = str.length();
    for(int i=0; i<len; i++){
        if(i==0 && str[i]=='0'){
            //qDebug()<<"err 92";
            return r;
        }
        if(str[i]=='1'||str[i]=='2'||str[i]=='3'||str[i]=='4'||str[i]=='5'||str[i]=
='6'||str[i]=='7'||str[i]=='8'||str[i]=='9'){
            //qDebug()<<"err 96";
            continue;
        }else{
            return r;
        }
    }
    r = 1;
    return r;
}

int Test1::check_label(){
```

```
int err =0;
// BrandInput 檢查
if(BrandInput == ""){
    ui->BrandError->setText("不可以空白");
    err++;
}else{
    ui->BrandError->setText("");
}

// KInput 檢查
if(KmInput == ""){
    ui->KmError->setText("不可以空白");
    err++;
}else if (regex_label(KmInput)==0) {
    ui->KmError->setText("有錯誤");
    err++;
}else{
    ui->KmError->setText("");
}

// MaxSeatLine 檢查
if(MaxSeatLine == ""){
    ui->MaxSeatsError->setText("不可以空白");
    err++;
}else if(regex_label(MaxSeatLine)==0){
    ui->MaxSeatsError->setText("有錯誤");
    err++;
}else{
    ui->MaxSeatsError->setText("");
}

// ModelLine 檢查
if(ModelLine == ""){
    ui->ModelError->setText("不可以空白");
    err++;
}else{
    ui->ModelError->setText("");
}

// RentalPriceLine 檢查
if(RentalPriceLine == ""){
    ui->PriceError->setText("不可以空白");
    err++;
}else if(regex_label(RentalPriceLine) == 0){
    ui->PriceError->setText("有錯誤");
    err++;
    }else{
        ui->PriceError->setText("");
    }

    // CarStatus 檢查
    if(CarStatus == ""){
        ui->StatusError->setText("不可以空白");
        err++;
    }else{
        ui->StatusError->setText("");
    }

    // filePath 檢查
    if(filePath == ""){
        ui->ImageError->setText("不可以空白");
        err++;
    }else{
        ui->ImageError->setText("");
    }

    if (err>0){
        return 0;
    }else{
        return 1;
    }
};


void Test1::check_car_info(){
    qDebug()<<"Car Check !!";
    qDebug()<<"Brand Line "<<car.BrandInput;
    qDebug()<<"Km Line "<<car.KmInput;
    qDebug()<<"MaxSeatLine Line "<<car.MaxSeatLine;
    qDebug()<<"ModelLine Line "<<car.ModelLine;
    qDebug()<<"Car Status "<<car.CarStatus;
    qDebug()<<"RentalPriceLine Line "<<car.RentalPriceLine;
}



void Test1::on_SubmitButton_clicked()
{
    // qDebug()<<__cplusplus;
    int  r1 = check_label();
    Database db;
    qDebug()<<"Brand Line "<<BrandInput;
    qDebug()<<"Km Line "<<KmInput;
```

```
        qDebug()<<"MaxSeatLine Line "<<MaxSeatLine;
        qDebug()<<"ModelLine Line "<<ModelLine;
        qDebug()<<"Car Status "<<CarStatus;
        qDebug()<<"RentalPriceLine Line "<<RentalPriceLine;
        QLabel *message = ui->message;
        int r2 = save_file(filePath);
        qDebug()<<"r1 = "<<r1<<" r2 = "<<r2;
        Rental_car_info();
        qDebug()<<"Brand Line "<<car.BrandInput;
        int r3 =
    db.addCar(car.BrandInput,car.ModelLine,car.KmInput,car.MaxSeatLine,car.RentalPriceLine,car.CarStatus,car.filePath);
        if (r1==1 && r2==1 && r3==1){ // add && r3==1

           message->setText("上傳成功");
        }else{
            message->setText("上傳失敗");
        }

    }

    Test1::~Test1()
    {
        delete ui;
    }
```

5.2.2 Car.cpp

```
#include "car.h"

#include <QDebug>


Car::Car(){

    qDebug() << "Car create";

}


QString Car::get_BrandInput(){

    return this->BrandInput;


}


void Car::set_car(QString BrandInput, QString KmInput, QString ModelLine, QString CarStatus,
        QString MaxSeatLine, QString RentalPriceLine, QString filePath){
```

```
    this->BrandInput = BrandInput;

    this->KmInput = KmInput;

    this->ModelLine = ModelLine;

    this->CarStatus = CarStatus;

    this->MaxSeatLine = MaxSeatLine;

    this->RentalPriceLine = RentalPriceLine;

    this->filePath = filePath;

}
```

5.2.3 Database.cpp

```
#include "database.h"
#include <QDebug>
#include <QT>

bool Database::dbdConnection() {
        QSqlDatabase db = QSqlDatabase::addDatabase("QODBC");
        db.setHostName("140.124.183.46");
        db.setPort(3306);
        db.setDatabaseName("carrentalsystem");
        db.setUserName("ooad");
        db.setPassword("Csslab438");
        if (!db.open()) {
            return false;
        }
        db.close();
        return true;
}
int Database::addCar(QString Brand,QString Model,QString kilometers,QString
MaxSeat,QString RentalPrice,QString CarStatus,QString filePath){
        int err = 1;
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("INSERT INTO car VALUES
        (0,:brand,:model,:kilometers,:MaxSeat,:RentalPrice,:carStatus,:filePath,0,3,
        3,3);");
        q.bindValue(":brand", Brand);
        q.bindValue(":model", Model);
        q.bindValue(":kilometers", kilometers);
        q.bindValue(":MaxSeat", MaxSeat);
        q.bindValue(":RentalPrice", RentalPrice);
        if(CarStatus=="良好"){
            q.bindValue(":carStatus",3);
        }else if(CarStatus=="尚可"){
            q.bindValue(":carStatus",2);
        }else{
            q.bindValue(":carStatus",1);
        }
        q.bindValue(":filePath", filePath);
        if(!q.exec()){
            err=0;
        }
        db.close();
        return err;

}
bool Database::signUp(QString Account,QString Password,QString Name,QString
Phone,QString Address){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        bool st=true;
        QSqlQuery q(db);
        q.prepare("INSERT INTO user VALUES
        (0,:account,:password,0,:name,:phone,:address,0);");
        q.bindValue(":account", Account);
        q.bindValue(":password", Password);
        q.bindValue(":name", Name);
        q.bindValue(":phone", Phone);
        q.bindValue(":address", Address);
        if(!q.exec()){
            st=false;
            qDebug()<<"失敗";
        }
        db.close();
        return st;
}
bool Database::checkAccount(QString Account){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("SELECT id FROM carrentalsystem.user where account=:account;");
        q.bindValue(":account", Account);
        int r=0;
        if(q.exec()){
            while(q.next()){
                r++;
            }
        }
        if(r==0){
            return true;
        }else{
            return false;
        }
        db.close();
}
bool Database::checkPassword(QString Account,QString Password){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("SELECT password FROM carrentalsystem.user where
        account=:account;");
        q.bindValue(":account", Account);
        int r=0;
        if(q.exec()){
            while(q.next()){
                if(Password==q.value("password").toString()){
                    return true;
                }else{
                    return false;
                }
                r++;
            }
        }
        if(r==0){
            return false;
        }
}
int Database::getAccountType(QString Account){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("SELECT type FROM carrentalsystem.user where account=:account;");
        q.bindValue(":account", Account);
        if(q.exec()){
            while(q.next()){
                return q.value("type").toInt();
            }
        }
        db.close();
}
QStringList Database::getCarBrandInfo(){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        QStringList brand;
        q.prepare("SELECT brand,model,kilomenter,maxSeat,rentalPrice FROM
        carrentalsystem.car where rentalStatus=0 order by id;");
        if(q.exec()){
            while(q.next()){
                brand << q.value("brand").toString();
            }
        }
        db.close();
        return brand;

}
QString Database::getCustomerID(QString Account){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("SELECT id FROM carrentalsystem.user where account=:account;");
        q.bindValue(":account", Account);
        if(q.exec()){
            while(q.next()){
                return q.value("id").toString();
            }
        }
        db.close();
}
bool Database::getCustomerRentalCar(QString customertID){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("SELECT rentalCar FROM carrentalsystem.user where id=:ID;");
        q.bindValue(":ID", customertID);
        if(q.exec()){
            while(q.next()){
                if("1"==q.value("rentalCar").toString()){
                    return true;
                }else{
                    return false;
                }

            }
        }
        db.close();
}
void Database::updateRentalCar(QString customertID){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        qDebug()<<customertID.toInt();
        q.prepare("UPDATE carrentalsystem.user SET rentalCar=1 where id=:ID;");
        q.bindValue(":ID", customertID.toInt());
        q.exec();
        db.close();
}
void Database::escheatRentalCar(QString customertID){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("UPDATE carrentalsystem.user SET rentalCar=0 where id=:ID;");
        q.bindValue(":ID", customertID.toInt());
        q.exec();
        db.close();
}
QString Database::addOrder(QString customertID,QString carID,QString
rentalDate,QString expireDate){
        QSqlDatabase db = QSqlDatabase::database();
        db.open();
        QSqlQuery q(db);
        q.prepare("INSERT INTO executionorder VALUES
        (0,:customertID,:carID,:rentalDate,:expireDate);");
        q.bindValue(":customertID", customertID);
        q.bindValue(":carID", carID);
        q.bindValue(":rentalDate", rentalDate);
        q.bindValue(":expireDate", expireDate);
        q.exec();
        q.prepare("SELECT orderID FROM carrentalsystem.executionorder where
        customerID=:customertID AND carID=:carID;");
        q.bindValue(":customertID", customertID);
        q.bindValue(":carID", carID);
        if(q.exec()){
            while(q.next()){
                return q.value("orderID").toString();
            }
        }
        db.close();
}
```

5.2.4 Database.cpp

```
#include "mainwindow.h"
#include "rental_car_system.h"
#include "database.h"
#include <QPixmap>
#include <QDebug>
```

```
#include <QString>
#include <QLabel>
#include <QPushButton>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    this->setWindowTitle("主畫面");
    this->setFixedSize(1217, 747);

    upload_car_infor_w = new Test1(this);
    upload_car_infor_w->hide();

    rental_page = new Rental_Car(ui);
    rental_page_index = 0;
    repair_report_pg = 0;
    repair_report_index = 0;
    repair_report_remain_item = 0;
    check_report_1 = false;
    check_report_2 = false;
    check_report_3 = false;

    // Login
    ui->stackedWidget->setCurrentIndex(0); // Login Page
    int width = ui->Login_cover->width();
    int height = ui->Login_cover->height();
    QPixmap image("C:/Users/CSS_Tim/Desktop/OOAD/CarRentalProject/image/cover.jpg");
    image = image.scaled(width, height);
    ui->Login_cover->setPixmap(image);
    ui->Login_cover->show();
}

MainWindow::~MainWindow()
{
    delete ui;
    delete upload_car_infor_w;
}

void MainWindow::on_Account_lineEdit_textEdited(const QString &arg1)
{
    Account = arg1;
}

void MainWindow::on_Password_lineEdit_textEdited(const QString &arg1)
```

```
{
    Password = arg1;
}

void MainWindow::on_Login_Button_clicked()
{
    Database db;
    //if(Account=="customer" && Password=="password"){
    if(db.checkPassword(Account,Password)){
        int type=db.getAccountType(Account);
        if(type==0){
            customer = new Person();
            customer->user_id = db.getCustomerID(Account);
            customer->is_rental_car = db.getCustomerRentalCar(customer->user_id);
            if(customer->is_rental_car){
                ui->stackedWidget->setCurrentIndex(5);
                //getInfo
            }else{
                //all_car_array = ;
                ui->stackedWidget->setCurrentIndex(2);
                car_array = rental_page->show_Rental_page(rental_page_index);
            }
        }else if(type==1){
            ui->stackedWidget->setCurrentIndex(7);
        }
    }else{
        ui->Login_Message->setText("登入失敗 !!");
    }
}

void MainWindow::on_Upload_Car_Button_clicked()
{
    upload_car_infor_w->show();
}

void MainWindow::on_picture1_Button_clicked()
{
    ui->stackedWidget->setCurrentIndex(3);
    rental_car = car_array[0];
    QString carID = rental_car.car_id;
    QString img_qpath = rental_car.filePath;
    QString brand = rental_car.BrandInput;
    QString kilomenter = rental_car.KmInput;
    QString model = rental_car.ModelLine;
    QString rentalPrice1 = rental_car.RentalPriceLine;
    QString maxSeat = rental_car.MaxSeatLine;
    rental_page->show_Product_page(img_qpath);
    ui->brand->setText(brand);
    ui->model->setText(model);
    ui->km->setText(kilomenter);
    ui->Reantal_Price->setText(rentalPricel);
    ui->Max_Seat->setText(maxSeat);
}

void MainWindow::on_picture2_Button_clicked()//----
{
    ui->stackedWidget->setCurrentIndex(3);
    rental_car = car_array[1];
    QString img_qpath = rental_car.filePath;
    rental_page->show_Product_page(img_qpath);
}

void MainWindow::on_NextPageButton_clicked()//----
{
    if(rental_page_index == 4){

    }else{
        rental_page_index = rental_page_index +2;
        car_array = rental_page->show_Rental_page(rental_page_index);
    }
}

void MainWindow::on_PreviousPageButton_clicked()
{
    if(rental_page_index == 0){

    }else{
        rental_page_index = rental_page_index -2;
        car_array = rental_page->show_Rental_page(rental_page_index);
    }
}

void MainWindow::on_Return_RentalCar_Button_clicked()
{
    ui->stackedWidget->setCurrentIndex(2);
}

void MainWindow::on_Order_Button_clicked()
{
    rental_order = new Order();
    ui->stackedWidget->setCurrentIndex(4);
}
```

```
void MainWindow::on_Return_Button_clicked()
{
    ui->Rental_Date_Year->setText("");
    ui->Rental_Date_Month->setText("");
    ui->Rental_Date_Date->setText("");
    ui->Expire_Date_Year->setText("");
    ui->Expire_Date_Month->setText("");
    ui->Expire_Date_Date->setText("");
    ui->stackedWidget->setCurrentIndex(3);
}

bool MainWindow::check_Date_label_format(QString year, QString month, QString
date){
    QRegularExpression regex("[^0-9]");
    if(year=="" || month=="" || date==""){
        return false;
    }else if(year.at(0)=='0' || year.contains(regex) || month.at(0)=='0' ||
month.contains(regex) || month.toInt() >12 || date.at(0)=='0' ||
date.contains(regex) || date.toInt() > 31) {
        return false;
    }else{
        return true;
    }
}

void MainWindow::on_decide_order_Button_clicked()
{
    rental_order->rental_car = &rental_car;

    rental_order->rental_date[0] = ui->Rental_Date_Year->text().toInt();
    rental_order->rental_date[1] = ui->Rental_Date_Month->text().toInt();
    rental_order->rental_date[2] = ui->Rental_Date_Date->text().toInt();

    rental_order->expire_date[0] = ui->Expire_Date_Year->text().toInt();
    rental_order->expire_date[1] = ui->Expire_Date_Month->text().toInt();
    rental_order->expire_date[2] = ui->Expire_Date_Date->text().toInt();

    qDebug()<<"img : "<<rental_order->rental_car->filePath;
    qDebug()<<"Rental year= "<<rental_order->rental_date[0]<<" month=
"<<rental_order->rental_date[1]<<" Date= "<<rental_order->rental_date[2];
    qDebug()<<"Expire year= "<<rental_order->expire_date[0]<<" month=
"<<rental_order->expire_date[1]<<" Date= "<<rental_order->expire_date[2];
    /*save order*/
    rental_order->car_id = rental_car.car_id;
    rental_order->user_id = customer->user_id;
    rental_order->order_id = "";
    Database db;
    QString rentalDate=QString::number(rental_order->rental_date[0])+"-
"+QString::number(rental_order->rental_date[1])+"-"+QString::number(rental_order-
>rental_date[2]);
    QString expireDate=QString::number(rental_order->expire_date[0])+"-
"+QString::number(rental_order->expire_date[1])+"-"+QString::number(rental_order-
>expire_date[2]);
    db.updateRentalCar(customer->user_id);
    db.addOrder(rental_order->user_id,rental_order->car_id,rentalDate,expireDate);
    customer->is_rental_car = true;
    /*
    這裡要儲存客戶的 order
    rental_order->car_id
    rental_order->user_id
    rental_order->order_id (DataBase 自動產生)
    */

    /*save customer*/
    customer->is_rental_car = true;
    /*
    這裡要存 customer 的資料
    customer->is_rental_car
    */
    ui->stackedWidget->setCurrentIndex(5);
    int width = ui->order_product_picture->width();
    int height = ui->product_picture->height();

    QPixmap image(rental_order->rental_car->filePath);
    image = image.scaled(width, height);
    ui->order_product_picture->setPixmap(image);
    ui->order_product_picture->show();

    QString rental_date = QString::number(rental_order->rental_date[0])+"-
"+QString::number(rental_order->rental_date[1])+"-"+QString::number(rental_order-
>rental_date[2]);
    ui->label_25->setText(rental_date);
    QString expire_date = QString::number(rental_order->expire_date[0])+"-
"+QString::number(rental_order->expire_date[1])+"-"+QString::number(rental_order-
>expire_date[2]);
    ui->label_27->setText(expire_date);
}

void MainWindow::on_Return_Car_Button_clicked()
{
    customer->is_rental_car = false;
    /*
    這裡要存 customer 的資料
    customer->is_rental_car
    */
    rental_page_index = 0;
    car_array = rental_page->show_Rental_page(rental_page_index);
    /* create report */
    rental_order->order_report = new Report();
    ui->stackedWidget->setCurrentIndex(6);
}

void MainWindow::on_SigIn_Button_clicked()
{
    ui->stackedWidget->setCurrentIndex(1);
    sigin_customer = new Person();
}

void MainWindow::on_SigIN_Account_lineEdit_textEdited(const QString &arg1)
{
    sigin_customer->Account = arg1;
}


void MainWindow::on_SignIN_Password_lineEdit_2_textEdited(const QString &arg1)
{
    sigin_customer->Password = arg1;
}

void MainWindow::on_SignIN_Name_lineEdit_3_textEdited(const QString &arg1)
{
    sigin_customer->Name = arg1;
}

void MainWindow::on_SignIN_phone_lineEdit_4_textEdited(const QString &arg1)
{
    sigin_customer->phone = arg1;
}

void MainWindow::on_SigIN_address_lineEdit_5_textEdited(const QString &arg1)
{
    sigin_customer->address = arg1;
}

void MainWindow::on_DecideSigin_Button_clicked()
{
    Database db;
    // save object sign_customer
    /*
    存 sign_customer
    sigin_customer->Name
    sigin_customer->phone
    sigin_customer->Account
    sigin_customer->Password
    sigin_customer->address
    sigin_customer->user_id (DataBase 自動產生)
    sigin_customer->is_rental_car (預設為 false)
    */
    if(db.checkAccount(sigin_customer->Account)){
        db.signUp(sigin_customer->Account,sigin_customer->Password,sigin_customer-
>Name,sigin_customer->phone,sigin_customer->address);
        qDebug()<<"Success";
        ui->stackedWidget->setCurrentIndex(0);
    }else{
        qDebug()<<"Fail Account 重複!!";
        ui->error_message->setText("重複的 Account! 請重新輸入!");
    }

}

void MainWindow::on_comboBox_currentIndexChanged(const QString &arg1)
{
    //qDebug()<<"test "<<arg1;
    rental_order->order_report->Evaluation = arg1;
}

void MainWindow::on_customer_report_button_clicked()
{
    /* save report */
    /*
     存 report 資料
     rental_order->order_report->Evaluation;
     */
    ui->stackedWidget->setCurrentIndex(2);
}

/* company page */

void MainWindow::on_company_Rapair_Report_Button_clicked()
{
    ui->stackedWidget->setCurrentIndex(8);
    calculate_repair_report_pg();
    show_repair_report();
```

```
}

void MainWindow::calculate_repair_report_pg(){
    /*test order*/
    int total_order = 13;
    order_array = new Order[total_order];
    /*
    這裡要改成從 DataBase 調出資料，存到 order_array
    */

    QString BrandInput = "Toyota";
    for(int i=0; i<total_order; i++){
        order_array[i].car_id = "Test_car"+QString::number(i);
        order_array[i].user_id = "Test_user"+QString::number(i);
        order_array[i].order_id = "Test_oder"+QString::number(i);
        order_array[i].rental_date[0] = i;
        order_array[i].rental_date[1] = 10+i;
        order_array[i].rental_date[2] = 20+i;
        order_array[i].expire_date[0] = i;
        order_array[i].expire_date[1] = 100+i;
        order_array[i].expire_date[2] = 200+i;
        order_array[i].order_report = new Report();
        order_array[i].order_report->car_id = order_array[i].car_id;
        order_array[i].order_report->user_id = order_array[i].user_id;
        order_array[i].order_report->order_id = order_array[i].order_id;
        order_array[i].rental_car = new Car();
        order_array[i].rental_car->ModelLine = "Model "+QString::number(i);
    }


    if(total_order%6==0){
        repair_report_pg = total_order / 6;
        repair_report_remain_item = 0;
    }else{
        if(total_order/6==0){
            repair_report_pg = 1;
        }else{
            repair_report_pg = (total_order / 6)+1;
        }
        repair_report_remain_item = total_order%6;
    }
    qDebug()<<"repair_report_pg = "<<repair_report_pg;
    qDebug()<<"remain_item = "<<repair_report_remain_item;
}

void MainWindow::show_repair_report(){
    /*show content on frontEnd*/
    front_label[0] = ui->repaire_report_label_1;
    front_button[0] = ui->repaire_report_button_1;

    front_label[1] = ui->repaire_report_label_2;
    front_button[1] = ui->repaire_report_button_2;

    front_label[2] = ui->repaire_report_label_3;
    front_button[2] = ui->repaire_report_button_3;

    front_label[3] = ui->repaire_report_label_4;
    front_button[3] = ui->repaire_report_button_4;

    front_label[4] = ui->repaire_report_label_5;
    front_button[4] = ui->repaire_report_button_5;

    front_label[5] = ui->repaire_report_label_6;
    front_button[5] = ui->repaire_report_button_6;

    if(repair_report_index==repair_report_pg-1 && repair_report_remain_item != 0){
        int temp_index = 0;
        qDebug()<<"290 repair_report_index = "<<repair_report_index;
        for(int j=repair_report_index*6;
j<(repair_report_index*6+repair_report_remain_item); j++){
            qDebug()<<"j = "<<j;
            front_label[temp_index]->setText( order_array[j].rental_car-
>ModelLine );
            front_button[temp_index]->setVisible(true);
            temp_index = temp_index+1;
        }
        for(int j=repair_report_remain_item; j<6; j++){
            front_label[j]->setText( "尚無資料" );
            front_button[j]->setVisible(false);
        }
    }else{
        qDebug()<<"299 repair_report_index = "<<repair_report_index;
        front_label[0]->setText( order_array[repair_report_index*6].rental_car-
>ModelLine );
        front_button[0]->setVisible(true);

        front_label[1]->setText( order_array[repair_report_index*6+1].rental_car-
>ModelLine );
        front_button[1]->setVisible(true);

        front_label[2]->setText( order_array[repair_report_index*6+2].rental_car-
>ModelLine );
```

```
        front_button[2]->setVisible(true);

        front_label[3]->setText( order_array[repair_report_index*6+3].rental_car-
>ModelLine );
        front_button[3]->setVisible(true);

        front_label[4]->setText( order_array[repair_report_index*6+4].rental_car-
>ModelLine );
        front_button[4]->setVisible(true);

        front_label[5]->setText( order_array[repair_report_index*6+5].rental_car-
>ModelLine );
        front_button[5]->setVisible(true);
    }
}

void MainWindow::clear_repair_report_pg(){
    front_label[0]->setText(" ");
    front_button[0]->setVisible(false);

    front_label[1]->setText(" ");
    front_button[1]->setVisible(false);

    front_label[2]->setText(" ");
    front_button[2]->setVisible(false);

    front_label[3]->setText(" ");
    front_button[3]->setVisible(false);

    front_label[4]->setText(" ");
    front_button[4]->setVisible(false);

    front_label[5]->setText(" ");
    front_button[5]->setVisible(false);
}

void MainWindow::on_repaire_report_last_pg_button_clicked()
{
    if(repair_report_index >0){
        repair_report_index = repair_report_index -1;
        clear_repair_report_pg();
        show_repair_report();
    }
}

void MainWindow::on_repaire_report_next_pg_button_clicked()
```

```
{
    if(repair_report_index < repair_report_pg-1){
        repair_report_index = repair_report_index +1;
        clear_repair_report_pg();
        show_repair_report();
    }
}

void MainWindow::show_repair_report_detail(Order *order){
    process_order = order;
    ui->repair_report_detail_user_id->setText("User_id = "+order->user_id);
    ui->repair_report_detail_car_id->setText("Car_id = "+order->car_id);
    ui->repair_report_detail_order_id->setText("Order_id = "+order->order_id);
    ui->repair_report_detail_ExpireDate->setText("Expire_Date =
"+QString::number(order->expire_date[0])+"-"+QString::number(order-
>expire_date[1])+"-"+QString::number(order->expire_date[2]));
    ui->repair_report_detail_RentalDate->setText("Rental_Date =
"+QString::number(order->rental_date[0])+"-"+QString::number(order-
>rental_date[1])+"-"+QString::number(order->rental_date[2]));
}

void MainWindow::on_repaire_report_button_1_clicked()
{
    ui->stackedWidget->setCurrentIndex(9);
    show_repair_report_detail(&order_array[0]);
}

void MainWindow::on_repaire_report_button_2_clicked()
{
    ui->stackedWidget->setCurrentIndex(9);
    show_repair_report_detail(&order_array[1]);
}

void MainWindow::on_repaire_report_button_3_clicked()
{
    ui->stackedWidget->setCurrentIndex(9);
    show_repair_report_detail(&order_array[2]);
}

void MainWindow::on_repaire_report_button_4_clicked()
{
    ui->stackedWidget->setCurrentIndex(9);
    show_repair_report_detail(&order_array[3]);
}

void MainWindow::on_repaire_report_button_5_clicked()
```

```
{
    ui->stackedWidget->setCurrentIndex(9);
    show_repair_report_detail(&order_array[4]);
}

void MainWindow::on_repaire_report_button_6_clicked()
{
    ui->stackedWidget->setCurrentIndex(9);
    show_repair_report_detail(&order_array[5]);
}

void MainWindow::on_comboBox_2_currentIndexChanged(const QString &arg1)
{
    if(arg1=="選擇" || arg1==""){
        ui->battery_err_msg_label->setText("error 請選擇 !");
        check_report_1 = false;
    }else if (arg1=="良好") {
        process_order->order_report->battery = true;
        ui->battery_err_msg_label->setText(" ");
        check_report_1 = true;
    }else {
        process_order->order_report->battery = false;
        ui->battery_err_msg_label->setText(" ");
        check_report_1 = true;
    }
}

void MainWindow::on_comboBox_3_currentIndexChanged(const QString &arg1)
{
    if(arg1=="選擇" || arg1==""){
        ui->engine_err_msg_label->setText("error 請選擇 !");
        check_report_2 = false;
    }else if (arg1=="良好") {
        process_order->order_report->engine = true;
        ui->engine_err_msg_label->setText(" ");
        check_report_2 = true;
        qDebug()<<"model "<<process_order->car_id<<" engine status true";
    }else {
        process_order->order_report->engine = false;
        ui->engine_err_msg_label->setText(" ");
        check_report_2 = true;
        qDebug()<<"model "<<process_order->car_id<<" engine status false";
    }
}

void MainWindow::on_comboBox_4_currentIndexChanged(const QString &arg1)
```

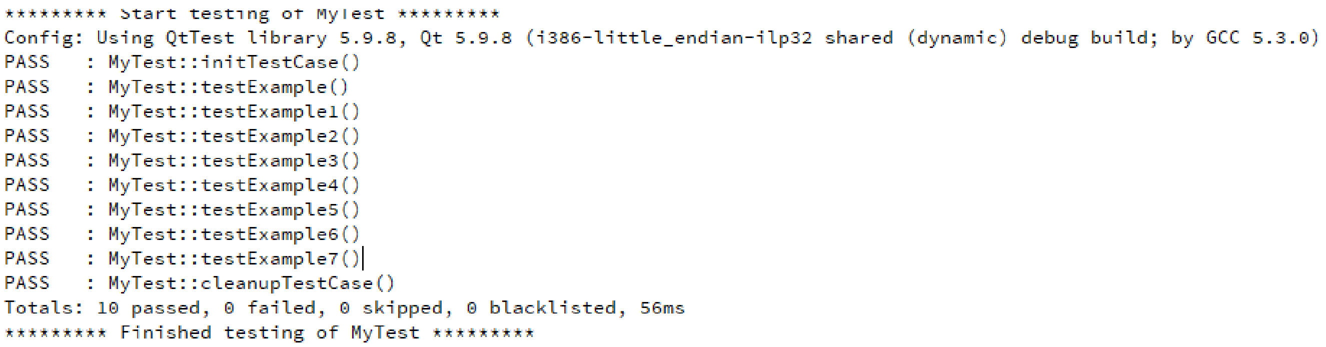
```
{
    if(arg1=="選擇" || arg1==""){
        ui->tire_err_msg_label->setText("error 請選擇 !");
        check_report_3 = false;
    }else if (arg1=="良好") {
        process_order->order_report->tire = true;
        check_report_3 = true;
        ui->tire_err_msg_label->setText(" ");
    }else {
        process_order->order_report->tire = false;
        check_report_3 = true;
        ui->tire_err_msg_label->setText(" ");
    }
}

void MainWindow::clear_repair_report_detail_pg()
{
    ui->Return_Report_err_msg_label->setText(" ");
    ui->comboBox_2->setCurrentIndex(0);
    ui->comboBox_3->setCurrentIndex(0);
    ui->comboBox_4->setCurrentIndex(0);
    ui->battery_err_msg_label->setText(" ");
    ui->engine_err_msg_label->setText(" ");
    ui->tire_err_msg_label->setText(" ");
    check_report_1 = false;
    check_report_2 = false;
    check_report_3 = false;
    repair_report_index = 0;
}

void MainWindow::on_Return_Report_Button_clicked()
{
    if(!check_report_1 || !check_report_2 || !check_report_3){
        ui->Return_Report_err_msg_label->setText("尚有資料未選擇 !");
    }else{
        /*
        這裡要把 repaire report 寫入 dataBase
        process_order->order_report->battery
        process_order->order_report->engine
        process_order->order_report->tire
        */
        clear_repair_report_detail_pg();
        ui->stackedWidget->setCurrentIndex(8);
        clear_repair_report_pg();
        show_repair_report();
    }
}
```

5.2.5 Rental_car_system.cpp

```
#include "rental_car_system.h"
#include <QDebug>
#include "database.h"
#include <QPixmap>
#include "mainwindow.h"
#include <iostream>
#include <string>

Rental_Car::Rental_Car(Ui::MainWindow *ui){
        mainwindow_ui = ui;
}

Car* Rental_Car::show_Rental_page(int rental_page_index)
{
        Database db;
        QStringList brandList= db.getCarBrandInfo();
        int width = mainwindow_ui->picture1->height();
        int height = mainwindow_ui->picture1->height();
        std::string path = "C:/Users/CSS_Tim/Desktop/OOAD/CarRentalProject/image/";
        std::string temp[] = {"car.jpg", "car2.jpg", "car3.jpg", "car4.jpg",
        "car5.jpg", "car6.jpg"};
        std::string brand[] = {"TOYOTA","HONDA","NISSAN","NISSAN","NISSAN","NISSAN"};
        std::string model[] = {"Wish","CR-V","TIIDA","SENTRA","ALTIMA","X-TRAIL"};
        std::string kilomenter[] = {"5000","4820","5513","3654","5420","4251"};
        std::string maxSeat[] = {"7","5","5","5","5","5"};
        std::string rentalPrice[] = {"5000","4800","5500","3500","5500","4200"};
        std::string img1_path = path + temp[rental_page_index];
        std::string img2_path = path + temp[rental_page_index+1];
        QString brandStr1 = QString::fromStdString(brand[rental_page_index]);
        QString kilomenterStr1 =
        QString::fromStdString(kilomenter[rental_page_index]);
        QString maxSeatStr1 = QString::fromStdString(maxSeat[rental_page_index]);
        QString modelStr1 = QString::fromStdString(model[rental_page_index]);
        QString rentalPricelStr1 =
        QString::fromStdString(rentalPrice[rental_page_index]);
        QString brandStr2 = QString::fromStdString(brand[rental_page_index+1]);
        QString kilomenterStr2 =
        QString::fromStdString(kilomenter[rental_page_index+1]);
        QString maxSeatStr2 = QString::fromStdString(maxSeat[rental_page_index+1]);
        QString modelStr2 = QString::fromStdString(model[rental_page_index+1]);
        QString rentalPricelStr2 =
        QString::fromStdString(rentalPrice[rental_page_index+1]);
        QString img1_qpath = QString::fromStdString(img1_path);
        QString img2_qpath = QString::fromStdString(img2_path);
        qDebug()<<"brandStr1";
        qDebug()<<brandStr1;
        mainwindow_ui->label_7->setText(modelStr1);
        mainwindow_ui->label_8->setText(rentalPricelStr1);
        mainwindow_ui->label_9->setText(modelStr2);
        mainwindow_ui->label_10->setText(rentalPricelStr2);
        Car *car_array = new Car[2];
        int carid1 = rental_page_index+1;
        int carid2 = rental_page_index+2;
        car_array[0].car_id = QString::number(carid1);
        car_array[0].BrandInput = brandStr1;
        car_array[0].KmInput = kilomenterStr1;
        car_array[0].ModelLine = modelStr1;
        car_array[0].RentalPriceLine = rentalPricelStr1;
        car_array[0].MaxSeatLine = maxSeatStr1;
        car_array[0].filePath = img1_qpath;
        car_array[1].car_id = QString::number(carid2);
        car_array[1].BrandInput = brandStr2;
        car_array[1].KmInput = kilomenterStr2;
        car_array[1].ModelLine = modelStr2;
        car_array[1].RentalPriceLine = rentalPricelStr2;
        car_array[1].MaxSeatLine = maxSeatStr2;
        car_array[1].filePath = img2_qpath;

        QPixmap image(img1_qpath);
        image = image.scaled(width, height);
        mainwindow_ui->picture1->setPixmap(image);
        mainwindow_ui->picture1->show();

        QPixmap image2(img2_qpath);
        image2 = image2.scaled(width, height);
        mainwindow_ui->picture2->setPixmap(image2);
        mainwindow_ui->picture2->show();

        return car_array;
}

void Rental_Car::show_Product_page(QString img_qpath)
{
        int width = mainwindow_ui->product_picture->width();
        int height = mainwindow_ui->product_picture->height();

        QPixmap image(img_qpath);
        image = image.scaled(width, height);
        mainwindow_ui->product_picture->setPixmap(image);
        mainwindow_ui->product_picture->show();
}
```

6 Unit Testing

6.1 Snapshots of testing result



```
********* Start testing of MyTest *********
Config: Using QtTest library 5.9.8, Qt 5.9.8 (i386-little_endian-ilp32 shared (dynamic) debug build; by GCC 5.3.0)
PASS   : MyTest::initTestCase()
PASS   : MyTest::testExample()
PASS   : MyTest::testExample1()
PASS   : MyTest::testExample2()
PASS   : MyTest::testExample3()
PASS   : MyTest::testExample4()
PASS   : MyTest::testExample5()
PASS   : MyTest::testExample6()
PASS   : MyTest::testExample7()
PASS   : MyTest::cleanupTestCase()
Totals: 10 passed, 0 failed, 0 skipped, 0 blacklisted, 56ms
********* Finished testing of MyTest *********
```

6.2 Unit Testing Code Listing

```
#include <QtTest>
#include <QCoreApplication>
#include "mainwindow.h"
#include "database.h"
#include <QDebug>

class Function_Test{
public:
    Function_Test(){

    }
    bool check_Date_label_format(QString year, QString month, QString date){
        QRegularExpression regex("[^0-9]");
        if(year=="" || month=="" || date==""){
            return false;
        }else if(year.at(0)=='0' || year.contains(regex) || month.at(0)=='0' ||
month.contains(regex) || month.toInt() >12 || date.at(0)=='0' ||
date.contains(regex) || date.toInt() > 31) {
            return false;
        }else{
            return true;
        }
    }
};

class MyTest : public QObject
{
    Q_OBJECT
```

```
private slots:

    void testExample()
    {
        QString year = "2000";
        QString month = "3";
        QString date = "7";
        Function_Test *ft = new Function_Test();

        bool result = ft->check_Date_label_format(year, month, date);
        int r = 0;
        if(result){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 1);
    }
    void testExample1()
    {
        QString year = "2000";
        QString month = "23";
        QString date = "7";
        Function_Test *ft = new Function_Test();

        bool result = ft->check_Date_label_format(year, month, date);
        int r = 0;
        if(result){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 0);
    }
    void testExample2()
    {
        QString account = "test";
        Database db;
        bool result = db.checkAccount(account);
        int r=-1;
        if(result){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 0);
    }
    void testExample3()
    {
        QString account = "dsfsdifisu";
        Database db;
        bool result = db.checkAccount(account);
        int r=-1;
        if(result){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 1);
    }
    void testExample4()
    {
        Database db;
        QString account = "test";
        QString password = "test";

        bool result = db.checkPassword(account,password);//檢查密碼
        int r=-1;
        if(result){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 1);
    }
    void testExample5()
    {
        Database db;
        QString account = "test";
        QString password = "testpassword";

        bool result = db.checkPassword(account,password);//檢查密碼
        int r=-1;
        if(result){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 0);
    }
    void testExample6()
```

```
    {
        Database db;
        QString account = "test";
        QString password = "test";

        int result = db.getAccountType(account);
        int r = -1;
        if(result==1){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 0);
    }
    void testExample7()
    {
        Database db;
        QString account = "company";

        int result = db.getAccountType(account);
        int r = -1;
        if(result==1){
            r = 1;
        }else{
            r = 0;
        }
        QCOMPARE(r, 1);
    }
};

int main(int argc, char *argv[])
{
    QCoreApplication app(argc, argv);
    Database db;
    db.dbdConnection();
    MyTest test;
    QTest::qExec(&test);

    return 0;
}


#include "test_first.moc"
```

7 Measurement

112/03/01	20:30 ~ 22:30	2 H
112/03/14	16:00 ~ 18:00	2 H
112/03/15	13:00 ~ 18:00	5 H
112/03/15	19:00 ~ 21:30	2.5 H
112/03/29	21:00 ~ 23:30	2.5 H
112/04/19	22:00 ~ 00:00	2 H
112/04/20	00:00 ~ 03:00	3 H
112/04/30	20:00 ~ 22:00	2 H
112/05/01	15:00 ~ 17:00	2 H
112/05/03	19:00 ~ 20:00	1 H
112/05/03	21:00 ~ 00:00	3 H
112/05/04	00:00 ~ 04:30	4.5 H
112/05/04	05:00 ~ 08:00	3 H
112/05/10	18:00 ~ 22:00	4 H
112/05/16	18:00 ~ 22:00	4 H
112/05/17	18:00 ~ 22:00	4 H
112/05/23	13:00 ~ 15:00	2 H
112/05/24	19:00 ~ 22:00	3 H
112/05/31	19:00 ~ 22:00	3 H
112/06/03	19:00 ~ 22:00	3 H
112/06/07	19:00 ~ 22:00	3 H
112/06/14	20:00 ~ 24:00	3 H
112/06/15	00:00 ~ 09:00	9 H