

Software Engineering Project

Report

Smart parking

Report #2: SYSTEM DESIGN

Reference: <https://github.com/software-groupp/smart-parking>

Group member and Individual Contributions Breakdown:

2. Thomas: Use Case4 :Camera sensor,Use Case5 :Timer,Class Diagram4& 5, Identifying Subsystems,Mapping Subsystems to

Hardware, User Interface Design and Implementation ;

Evan:Use Case1 :Payment,Use Case2 :Parking ,Use Case3:Going out, Class Diagram1, 2& 3, Architectural Styles;

Alex:Use Case6 :No Availability,Use Case7 :Register, Class Diagram6& 7,Persistent Data Storage,Network Protocol ;

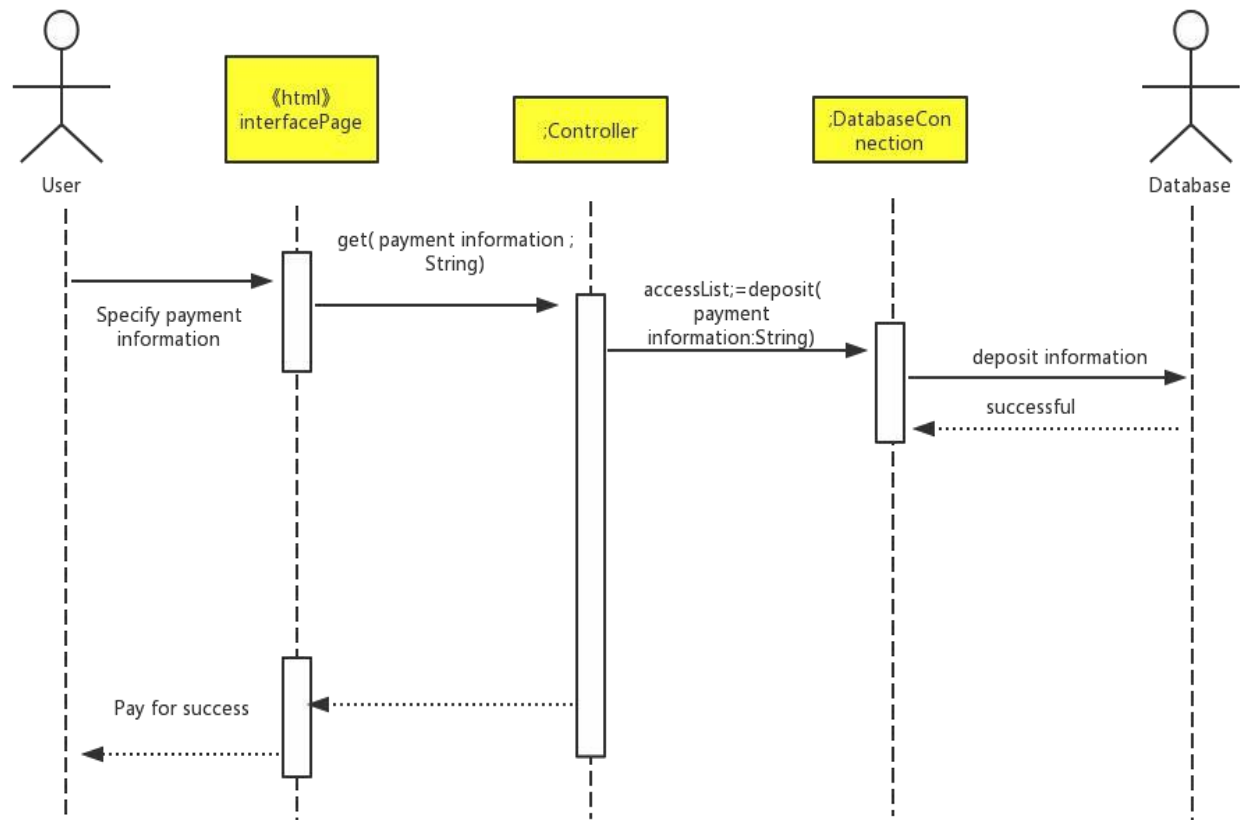
Alan:Use Case8 :Sensors,Use Case9 :ad-hoc,, Class Diagram 8& 9 ,Global Control Flow,Hardware Requirements.

Contents

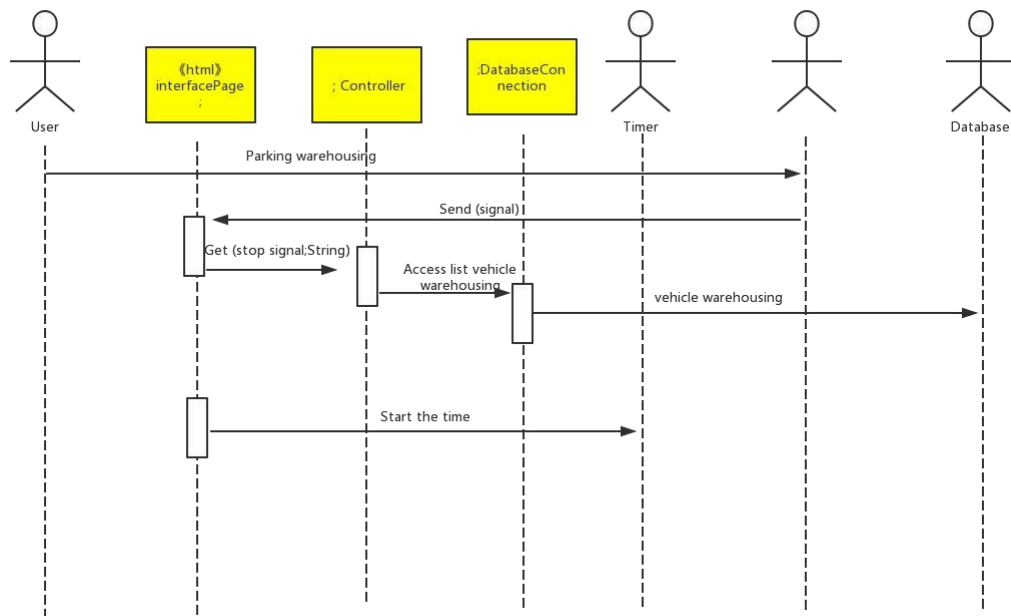
A.Interaction Diagrams.....	3
1. Use Case1 :Payment.....	3
2.Use Case2 :Parking.....	4
3.Use Case3:Going out.....	4
4.Use Case4 :Camera sensor.....	5
5.Use case5:Timer.....	6
6.Use Case6 :No Availability.....	6
7.Use Case7 :Register.....	7
8.Use Case8 :Sensors.....	8
9.Use Case9 :ad-hoc.....	9
B.Project Management.....	10
2. Class Diagram and Interface Specification.....	11
a. Class Diagram.....	11
3. System Architecture and System Design.....	15
a. Architectural Styles	15
B. Identifying Subsystems	16
C. Mapping Subsystems to Hardware	16
D. Persistent Data Storage	16
E. Network Protocol	17
F.Global Control Flow.....	17
G.Hardware Requirements.....	17
4. User Interface Design and Implementation.....	17

A.Interaction Diagrams

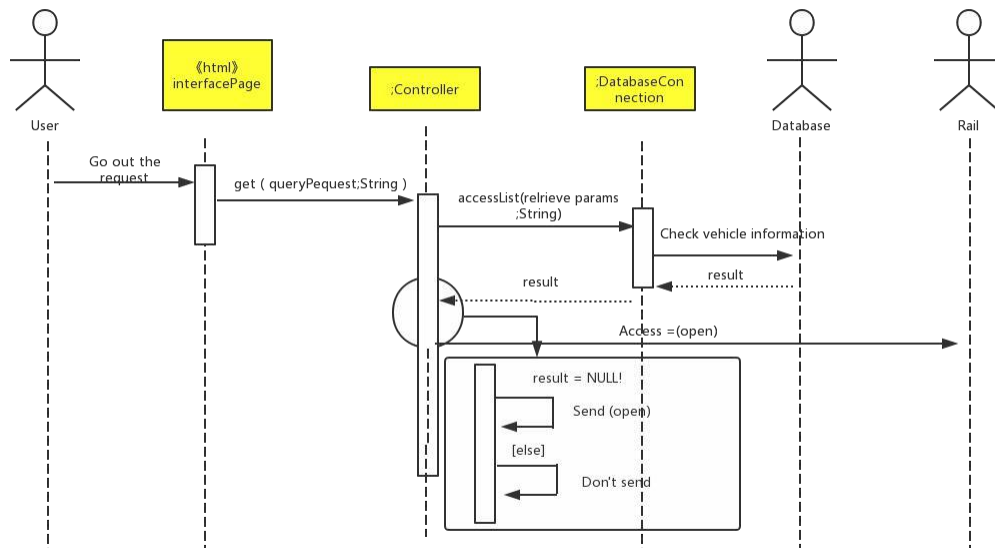
1.Use Case1:Payment



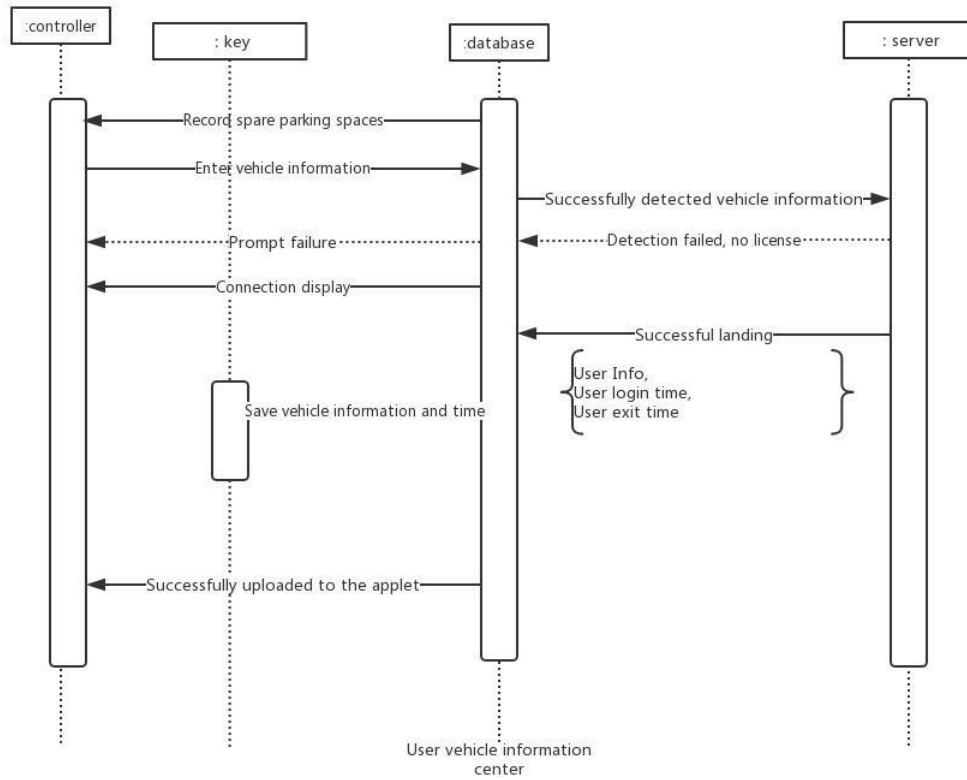
2. Use Case2: Parking



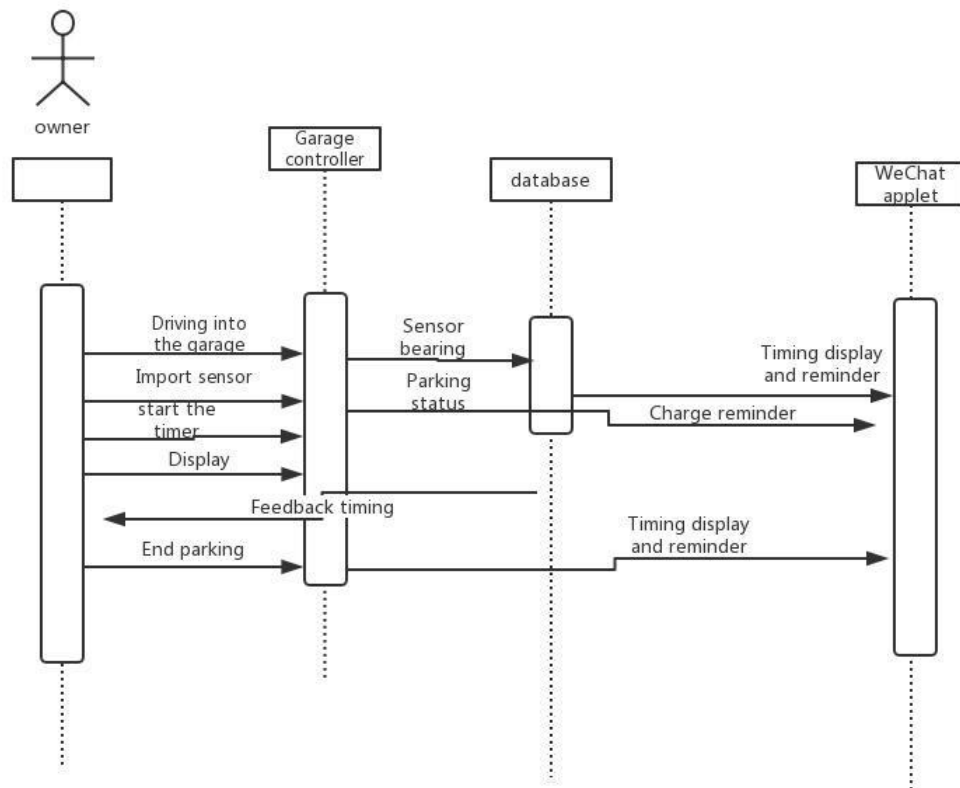
3. Use Case3: Going out



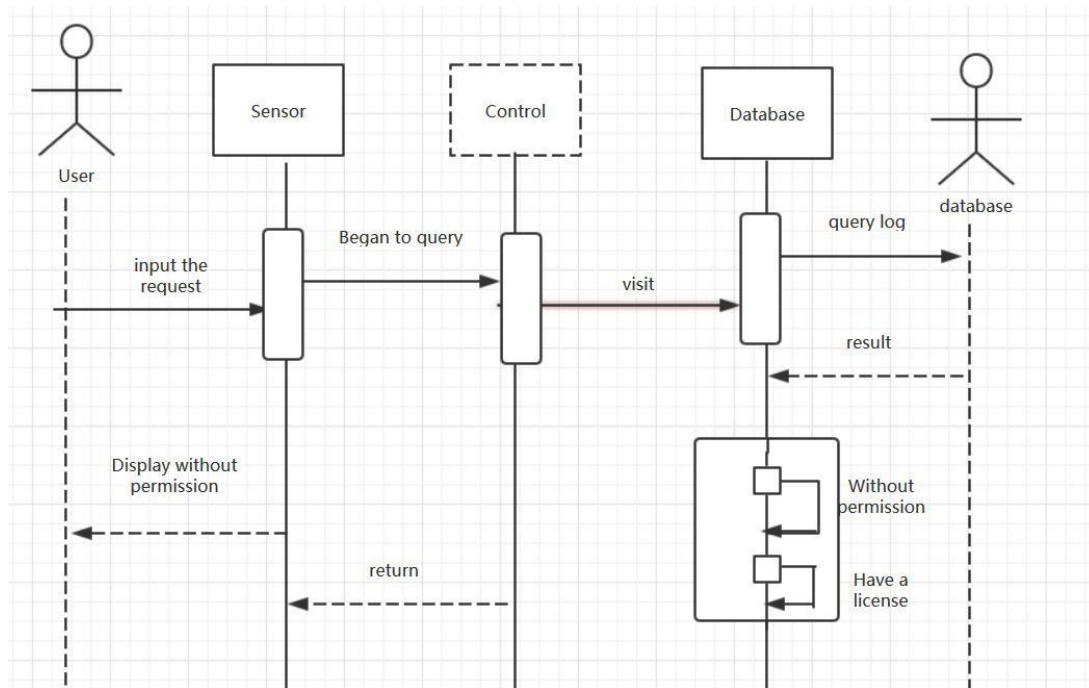
4. Use Case4 : Camera sensor



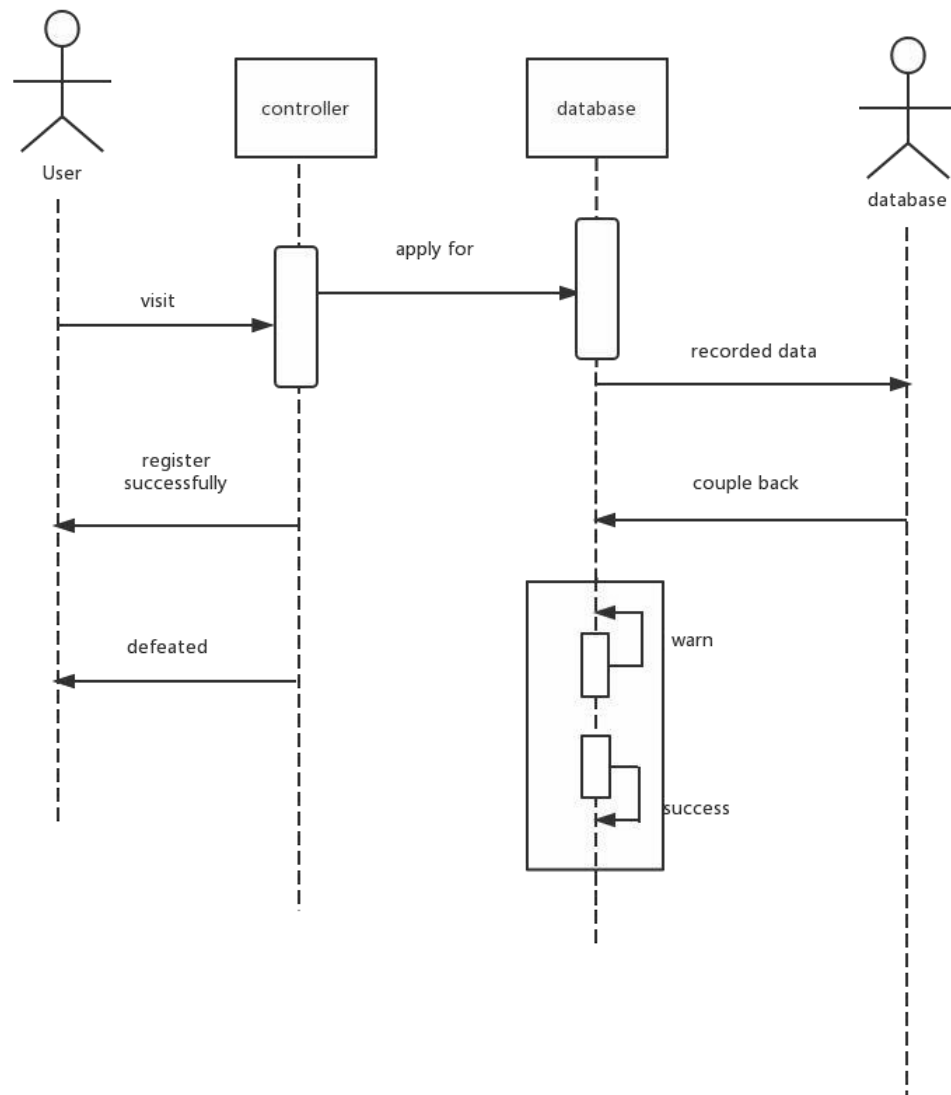
5. Use case 5: Timer



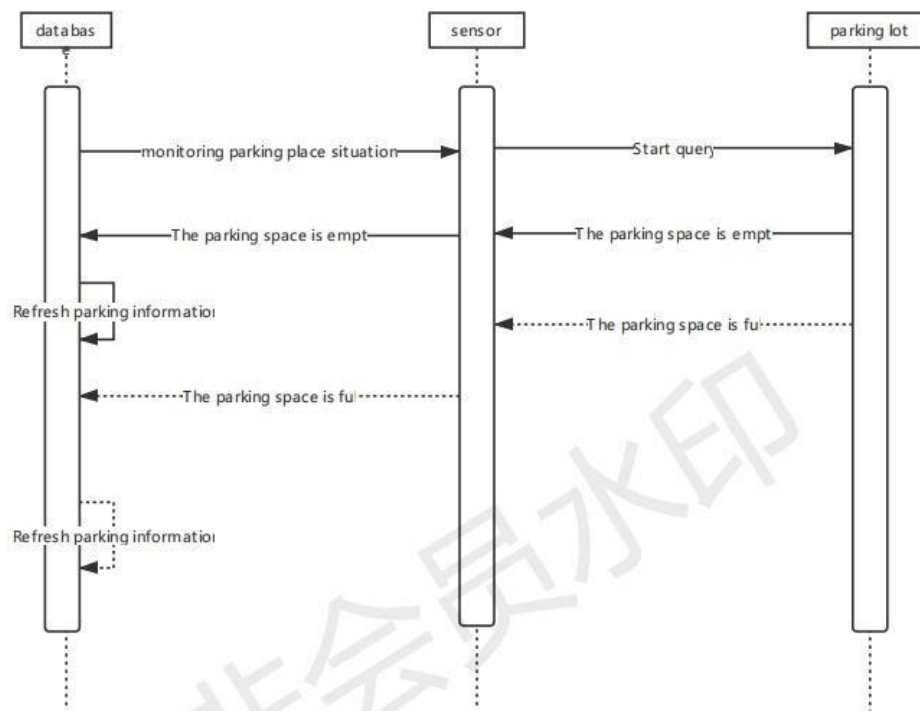
6. Use Case 6 :No Availability



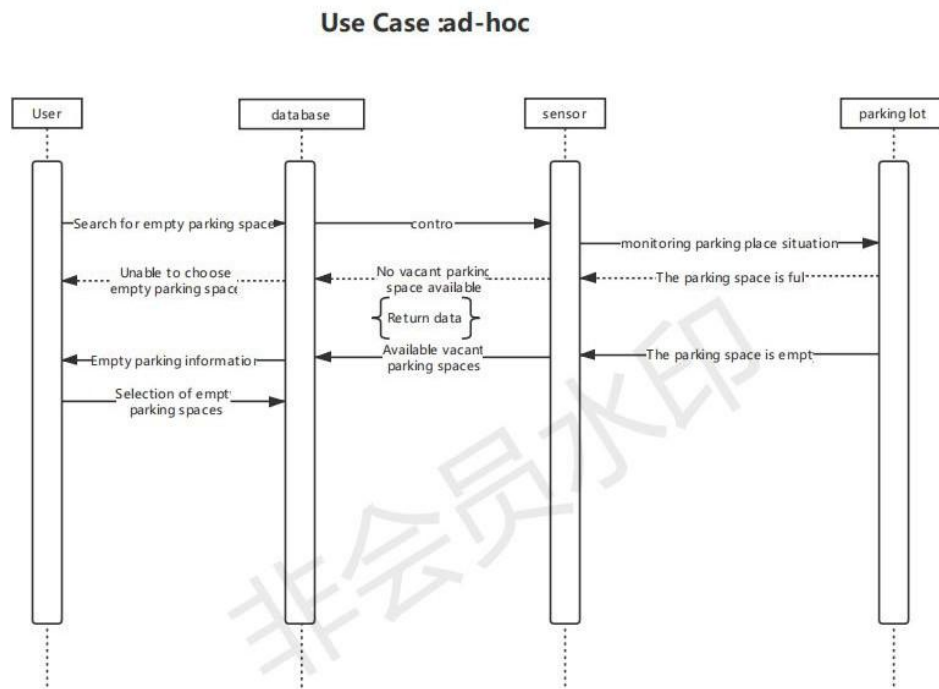
7. Use Case 7: Register



8. Use Case 8: Sensors



9. Use Case 9 :ad-hoc

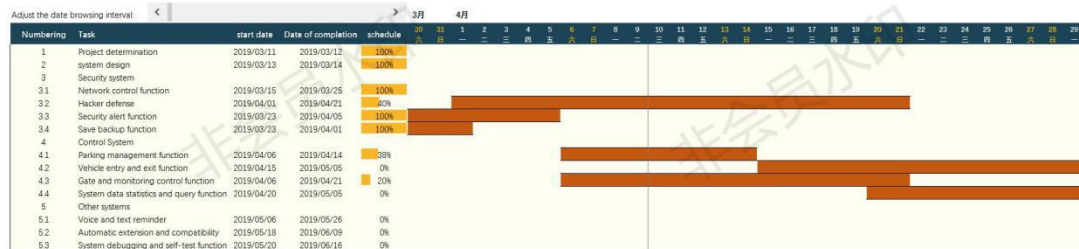


B.Project Management

Smart parking project

Project Name: Intelligent Parking Project
Start date: 2019/3/20
Person in charge: XXXXX

task progress | The same day



References

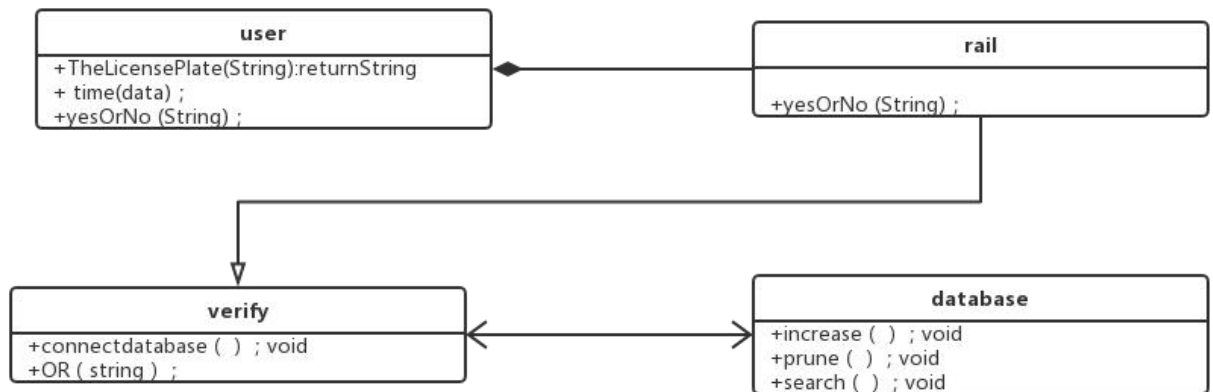
<http://staruml.sourceforge.net/en/>

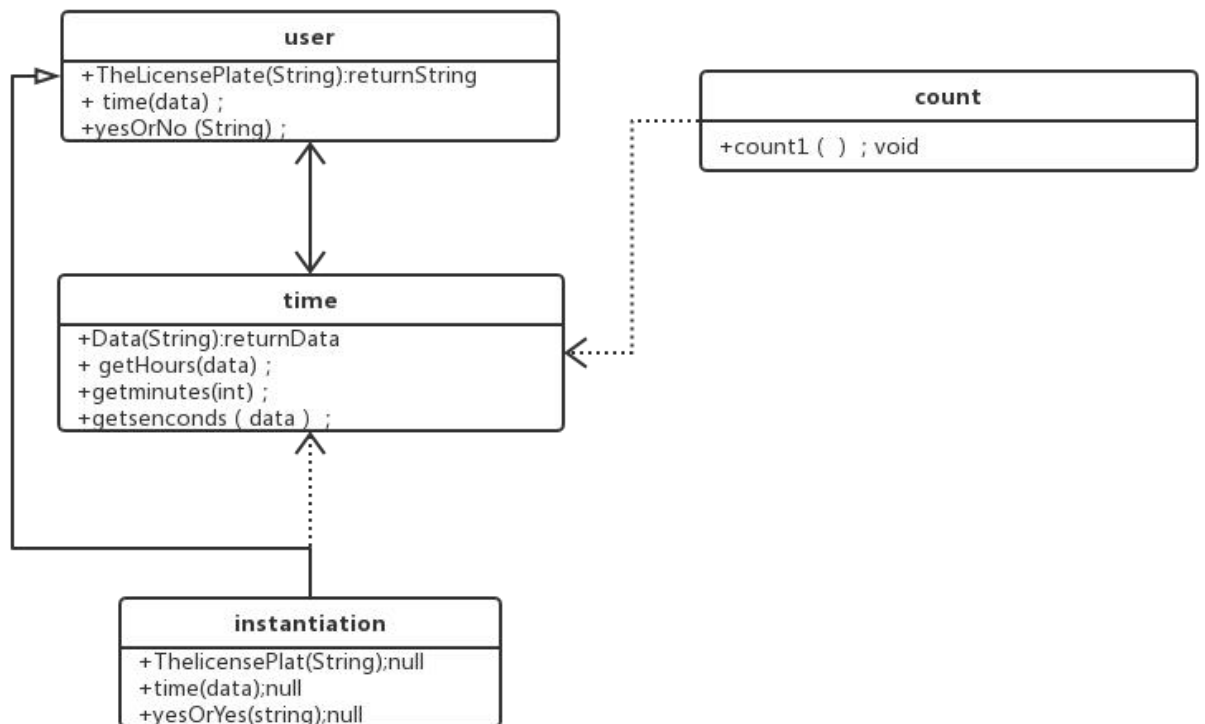
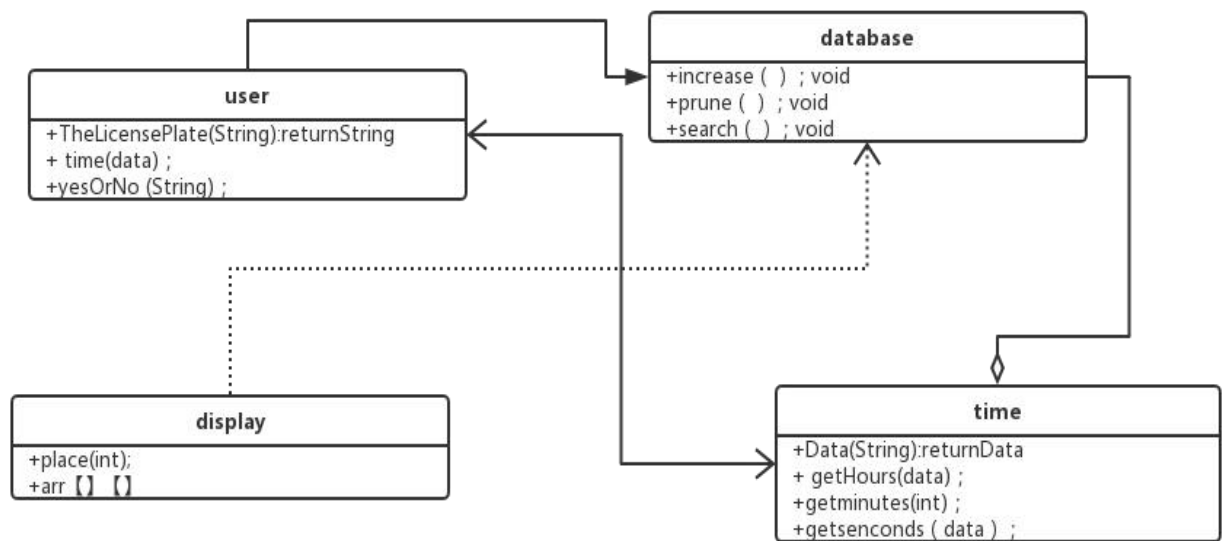
<https://www.processon.com>

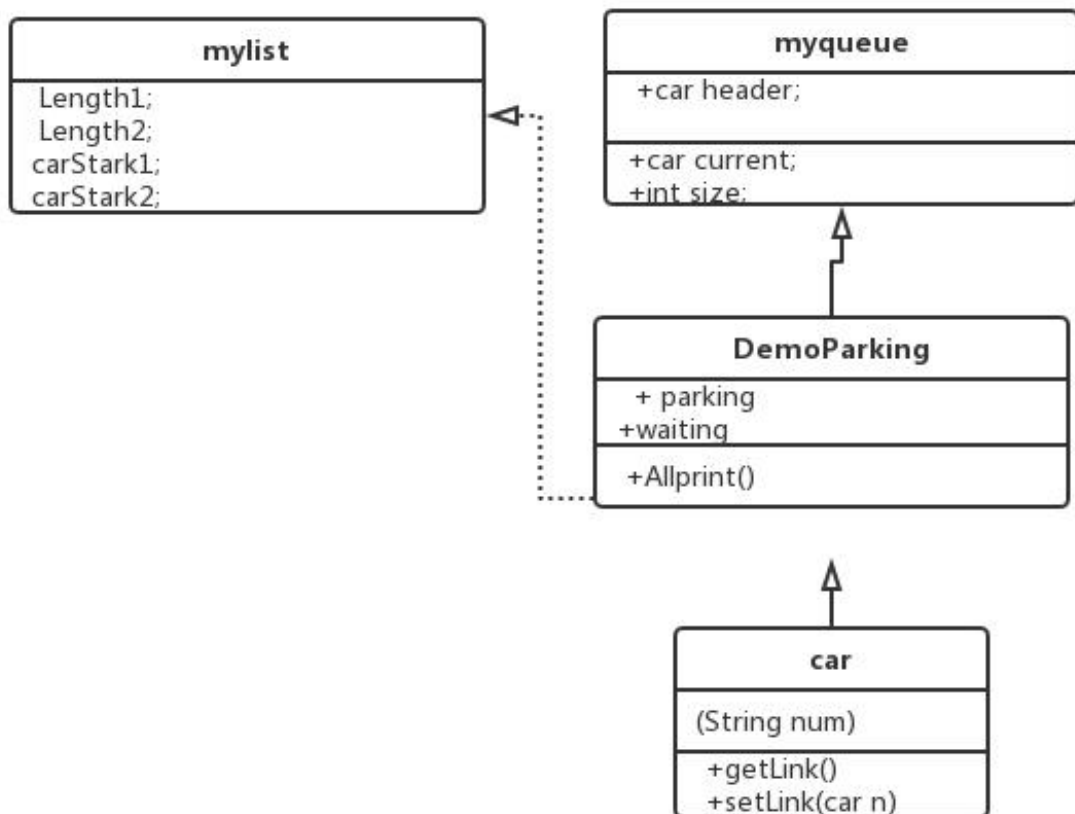
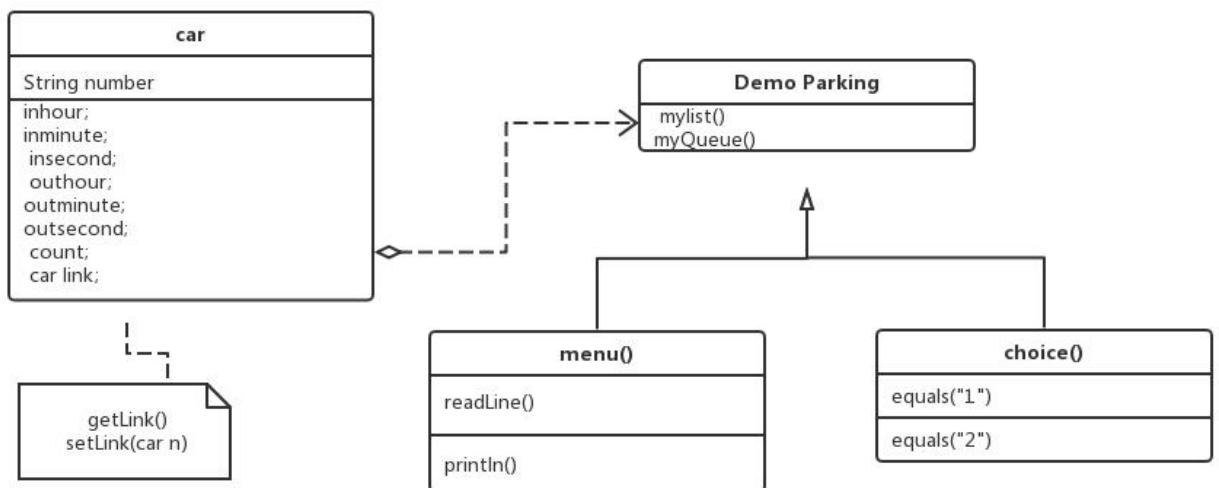
<http://argouml.tigris.org/>

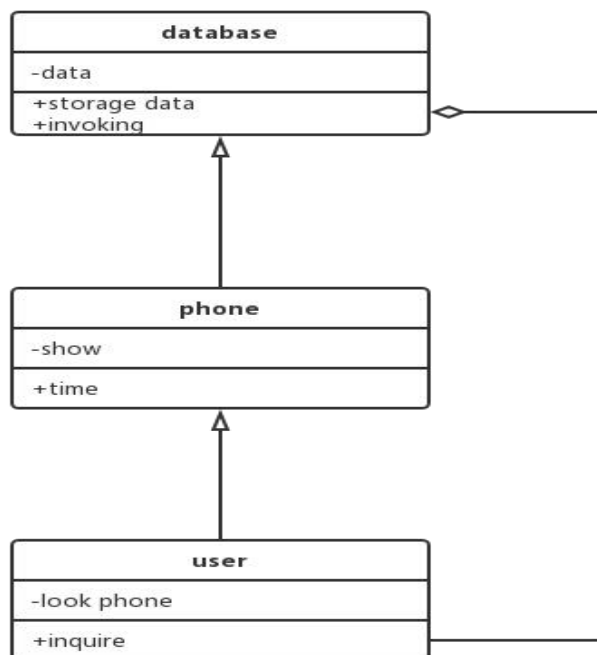
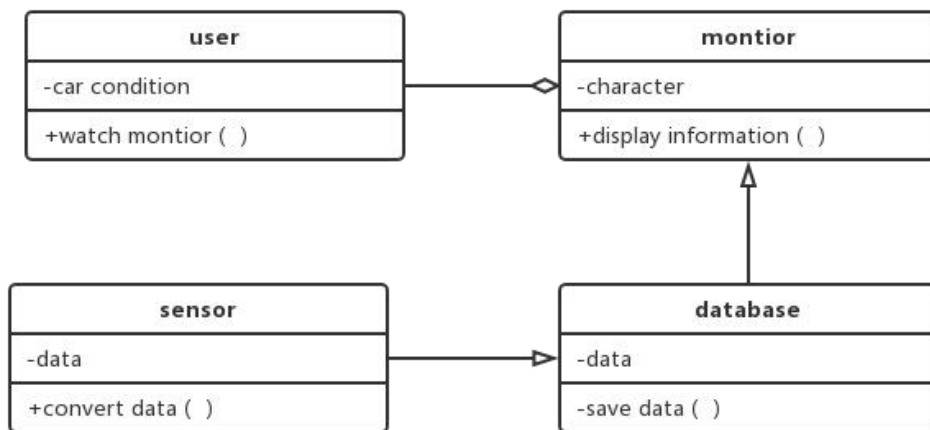
3. Class Diagram and Interface Specification

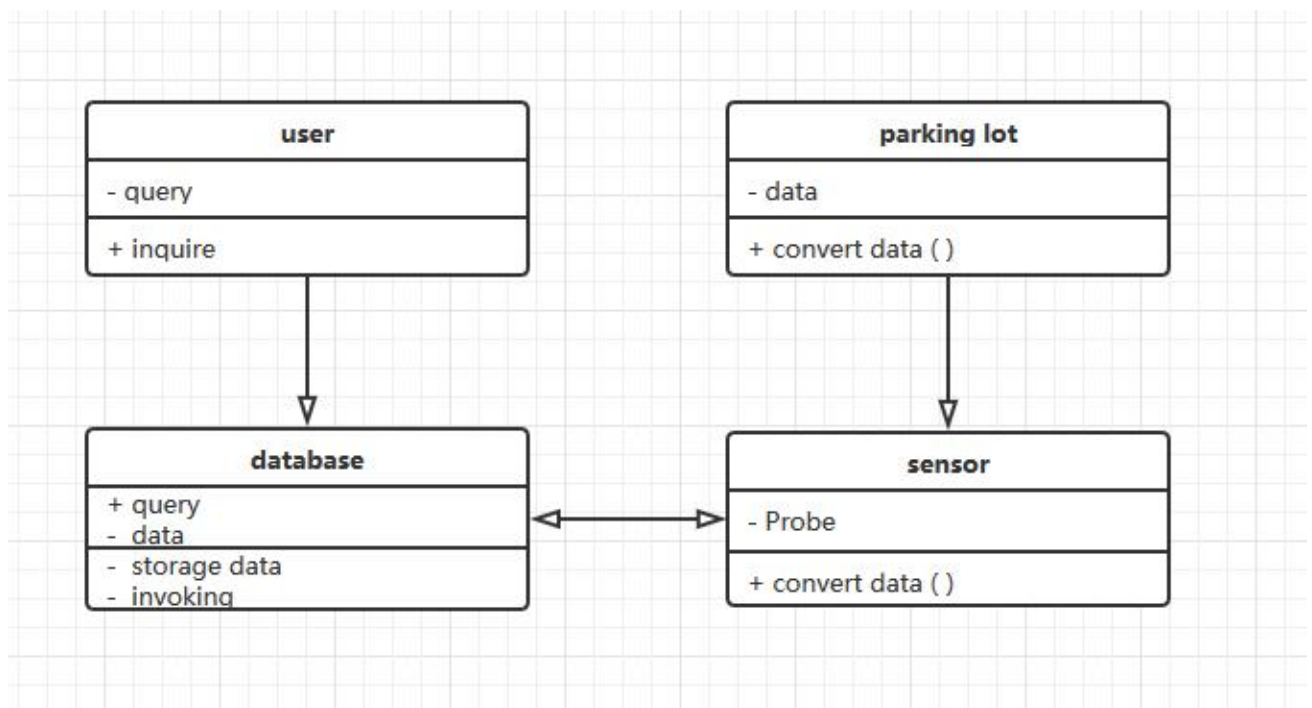
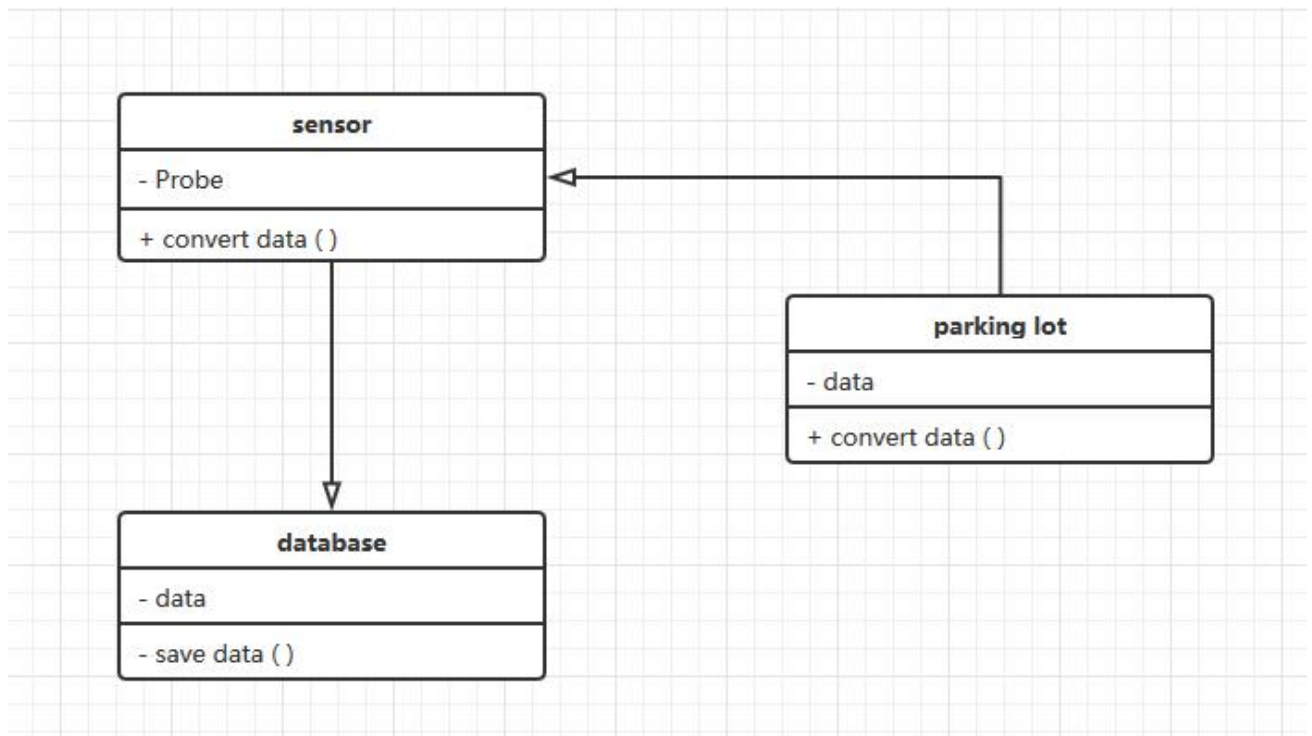
a. Class Diagram











4. System Architecture and System Design

a. Architectural Styles

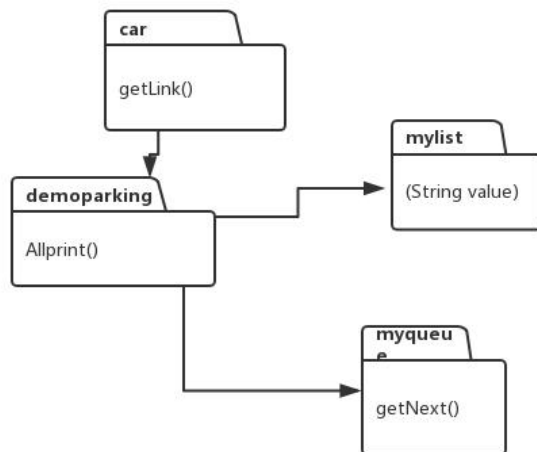
Data abstraction and object-oriented style

The abstract data type concept plays an important role in software systems. Currently, the software industry has generally turned to object-oriented systems. This style is based on data abstraction and object-oriented, and the

representation of data and their corresponding operations are encapsulated in an abstract data type or object. This style of artifact is an object, or an instance of an abstract data type. An object is a component called a manager because it is responsible for maintaining the integrity of the resource. Objects interact through calls to functions and procedures.

Figure 2-2 is a schematic diagram of data abstraction and object-oriented style.

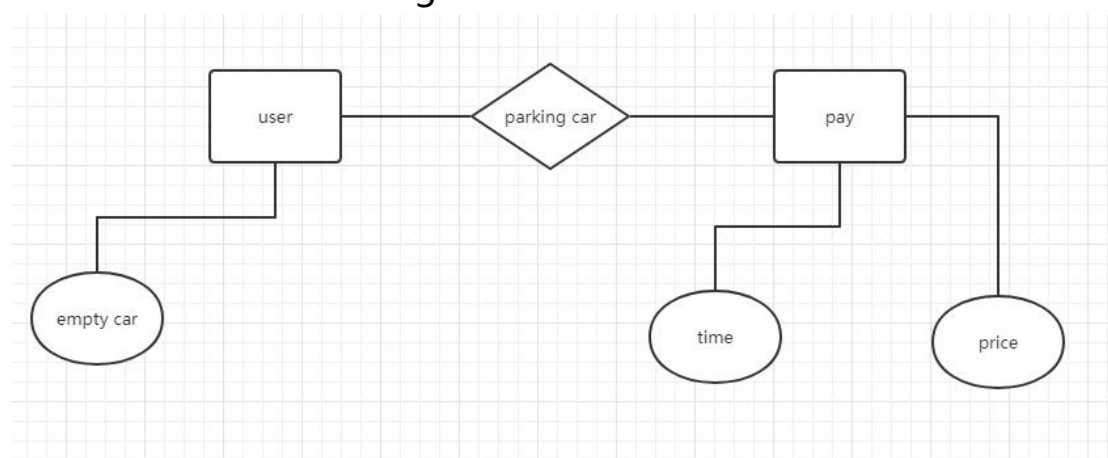
B. Identifying Subsystems



C. Mapping Subsystems to Hardware

Our software has its own server, then you can call it on other websites.

D. Persistent Data Storage



E. Network Protocol

We choose HTTP protocol because the main protocol used by WWW server is HTTP protocol, namely hyper style transport protocol. Since the services supported by HTTP protocol are not limited to WWW, but also other services, HTTP protocol allows users to use different protocols to access different services under a unified interface.

F. Global Control Flow

Execution orderness: Our system is an event-driven system.

Time dependency: Our system is event-response type.

Concurrency: our system does not use multiple threads.

G. Hardware Requirements

Server and client computers: Processor Intel (R) Core (TM) i3, Memory 8 GB or more, A color display with a minimum resolution of 640x480 pixels is required, minimum network bandwidth 10 Mbps, At least 100GB of hard disk space and Uninterruptible Power Supply.

Other hardware: Multiple LED Electronic Display Screens, Digital Vehicle Detector, Control Card Reader, Gangting Card Reader, Card Distributor, Horn, Automatic Card Distributor, Infrared Detector, Video Card, Serial Port Extension Card, Intelligent Gate, Surveillance camera.

5. User Interface Design and Implementation

@ Javadoc 声明 控制台 ✕
DemoParking [Java 应用程序] E:\myeclipse\Common\bin

```
    欢迎进入停车场系统
*****
*   本停车场每秒收费1元 停车场内有5个车位   *
*   倒车一次返回2元(最少不收费!)           *
*****

    1:开进
    2:开出
    3:车位情况
    4:退出

请选择:
```

car.java DemoParking.java mylist.java myQueue.java ✕

@ Javadoc 声明 控制台 ✕
DemoParking [Java 应用程序] E:\myeclipse\Common\binary\com.sun.java.jdk.win32.x

```
    欢迎进入停车场系统
*****
*   本停车场每秒收费1元 停车场内有5个车位   *
*   倒车一次返回2元(最少不收费!)           *
*****

    1:开进
    2:开出
    3:车位情况
    4:退出

请选择:1
请输入车牌号:123456
*****
您的车牌号:123456
进入停车场时间:11:36:46
*****

    欢迎进入停车场系统
*****
*   本停车场每秒收费1元 停车场内有5个车位   *
*   倒车一次返回2元(最少不收费!)           *
*****

    1:开进
    2:开出
    3:车位情况
    4:退出

请选择:
```

```

        4:退出

请选择:3
*****

停车场中:
1号车位:123456
*****

        欢迎进入停车场系统
*****
*      本停车场每秒收费1元 停车场内有5个车位      *
*              倒车一次返回2元(最少不收费!)              *
*****

        1:开进
        2:开出
        3:车位情况
        4:退出

请选择:
```

```

        欢迎进入停车场系统
*****
*      本停车场每秒收费1元 停车场内有5个车位      *
*              倒车一次返回2元(最少不收费!)              *
*****

        1:开进
        2:开出
        3:车位情况
        4:退出

请选择:2
请输入车牌号:123456
*****

        您的车牌号:123456
        进入时间:11:38:22
        离开时间:11:38:28
        倒车次数:0
        您的费用:6元
        欢迎下次再来! @_@再见
*****
```

```

        欢迎进入停车场系统
*****
*      本停车场每秒收费1元 停车场内有5个车位      *
*              倒车一次返回2元(最少不收费!)              *
*****

        1:开进
        2:开出
        3:车位情况
        4:退出

请选择:
```

