UML: Car Park Management System

MSc Electronic and Computer Engineering

Chen Kai

Student ID: 1948361

2019-2-4

UNIVERSITY OF BIRMINGHAM

Content

1 Introduction 2
2 Use case model 4
2.1 Use case diagram 4
2.2 Scenario Descriptions 5
2.2.1 User 5
2.2.2 Security staff 5
2.2.3 Distance sensor system 6
2.3 Class identification 6
2.3.1 Nouns identification 6
2.3.2 Stereotypical classes 7
2.3.3 Class diagram 7
2.4 CRC Cards 8
2.5 Collaboration Diagram 10
2.6 Interaction Diagram: Sequence Diagram 11
2.7 State chart diagram 13
3 Analysis Model 14
3.1 Attribute 14
3.2 Methods 15
3.3 Sequence Diagram 17
3.4 Class Diagram 20
3.5 State Chart Diagram 21
3.6 Non-functional Requirements 21
3.7 Packages 21
4 Design Model 22
4.1 Revisit Use-Case Model 22
4.2 Sequence Diagram 22
4.3 Textual Description of Object to Object Interaction 22
4.4 Subsystem 22
4.5 Deployment model 23
4.6 Implementation of Non-functional Requirements 23
4.7 Reconsider the Attributes 23
4.8 State Chart 23
4.9 Class Diagram Showing Visibility 23
4.10 Class Implementation (Psuedo Code) 25
5 Test 27
6 Conclusion 27

1 Introduction

Purpose of the parking management system is to improve operational efficiency, enhance security in all aspects and interact with information from intelligent transportation systems.

In this task, we use UML to design the parking management system. In addition to understanding the functions and concepts of UML, the purpose of this task is to achieve the following functions:

Firstly, the user who needs to enter the parking lot is identified by a personal tagging system, such as an ID card or a transponder. After that, system allows the identified user to enter the designated parking lot. When the car is authorized to enter the park, the user will get the tag. It contains information in the database, such as enter time and vehicle location.

Besides, When the user wants to exit the car park. Users need to use the tag which got before. If the user does not need to pay any parking fee, the barrier will be raised and the user can exit the park directly. If the user needs to pay, they will have to pay the fee then the barrier will be upgraded.

The security staff can control the barrier anywhere in the parking lot and receive feedback from the control system and distance sensor.

When the barrier is raised, the sensor begins to measure the distance between the barrier and the car. It then feeds information back to the control system. When the vehicle is away from the barrier and reaches a safe location, the control system issues a command to lower the barrier.

Above all, the scene of this system is divided into three categories: parking users, parking lot staff and distance sensor. The entire system runs automatically under software and staff control.

Throughout the process, the system can identify users with personal labeling system and control barrier for them. At the same time, the system will record the enter time and save the tag number and vehicle location. After the car passes the barrier, the sensor in the parking lot will measure the distance from the barrier to the car and feed the results back to the control system. The control system staff will check if this distance is safe. If it is a safe distance, the system will lower the barrier after the user reaches a safe distance. When the user want to exit the park, the user needs to leave using a personal tag system (eg, ID card, transponder) and tag. The parking management system will calculate the time the car is parked in the parking lot and calculate the fee that the user needs to pay. If the user does not need to pay for the parking , the barrier will be raised and the user can leave directly. If the user needs to pay a fee, the system will display the amount to be paid and deduct the parking fee from

the user's personal ID card or transponder. After the payment is successful, the barrier will rise and the car can leave the park. When the barrier is dropped, the next user will display the correct ID card or repeater. The control system then stops the descent barrier and lifts it up again.

In case of emergency special circumstances, the parking lot staff can directly perform manual control operations.

2 Use case model

2.1 Use case diagram

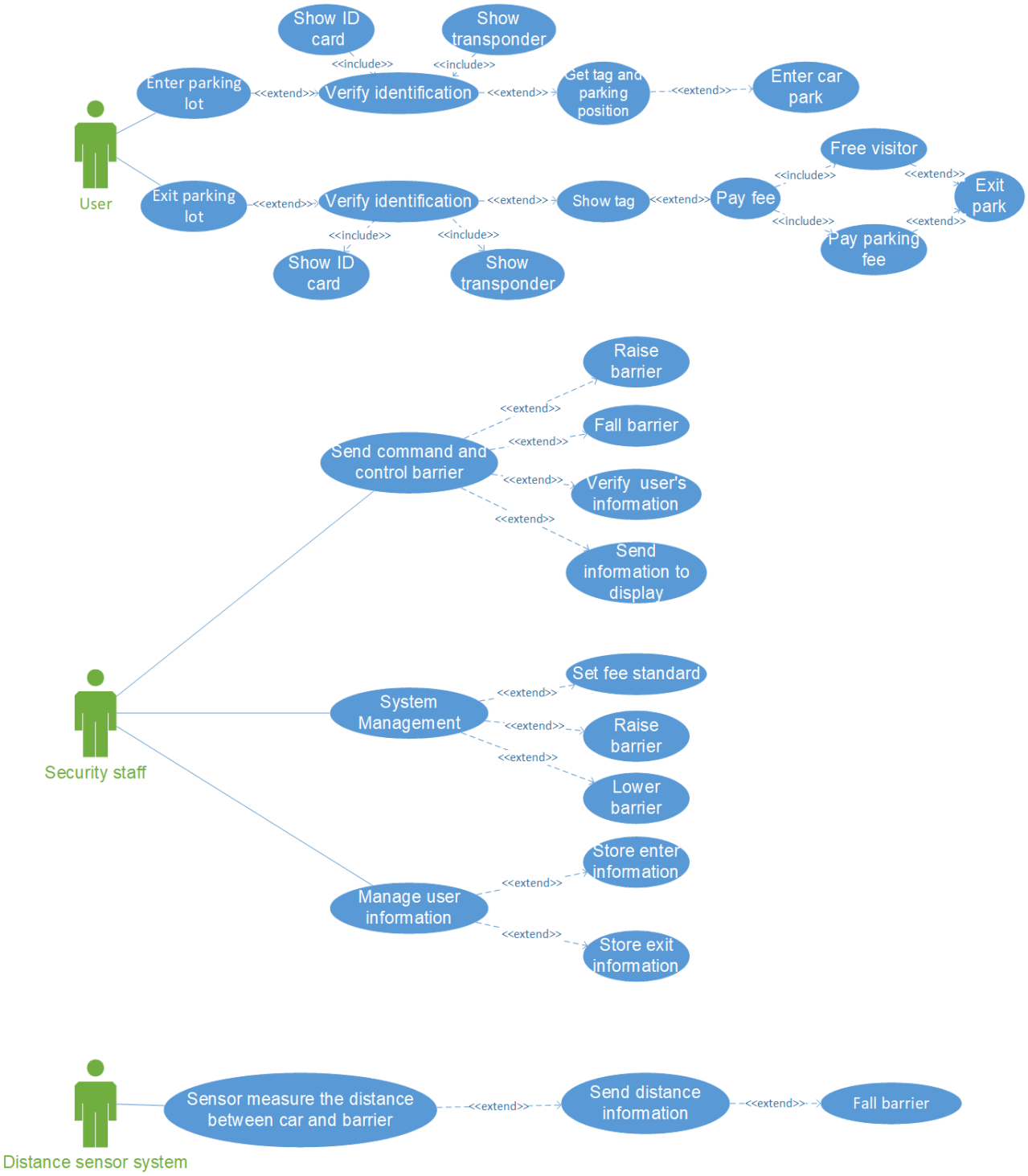

Figure 2.1. Use-Case Diagram

2.2 Scenario Descriptions

2.2.1 User

When the user wants to enter the parking lot, it is necessary to provide the controller with identification documents such as an identity card or a transponder. After successfully verified, the system prints a label/tag for user, which includes database information such as entry time and vehicle location. At the same time the system will record these information. When the user gets the tag, the controller raises the barrier and then the car can pass the barrier.

When the user wants to leave the parking lot, the user also needs to show the identification to the system. Tags are needed to show at the same time. The system will calculate the duration time of the car in park. If the stay time is within the free range, the user will not be charged. If the time is higher than the range, the system will show the user the fee to be paid. And the user can use the money which already saved in the ID card or transponder to pay.

2.2.2 Security staff

Security staff can perform emergency operations on any barriers in the parking lot from any remote site. In the event of emergency, the security officer can issue commands to the system to control the rise or fall of the barrier. For example, barrier’s malfunction, fire, etc.

After the verification is successful, Staff assigns the parking space to the vehicle and prints the vehicle entry time and parking space on tag. Besides, it will be displayed on the screen. When the user wants to leave the parking lot, the user's information will also be sent to the staff. The staff will get the exit time and calculate the parking fee based on the parking duration time. The system also displays the enter time and cost on the screen. After user pay successfully, the operator will raise the barrier and the car will leave the parking lot. After the car exits, the barrier will decrease.

The fee is set by staff. In addition, if an emergency or unexpected situation occurs, Security staff can directly control the rise and fall of the barrier.

2.2.3 Distance sensor system

The monitor sensor is used to detect the distance between the vehicle and the barrier and send the distance information to the staff. When the distance is less than the safe distance, the staff will raise the obstacle. Conversely, when the distance is greater than the safe distance, the obstacle will fall.

2.3 Class identification

2.3.1 Nouns identification

(1) User scenario:

- User identification, controller
- User identification is a vague class identifier. It can verify the user's information by ID card and transponder.
- The controller controls the entire system. It can control the operation of the barriers and recording vehicle enter and exit time.

(2) Security staff scenario:

- User identification, controller
- Security staff represents the commander of all systems. They not only need to check and verify user's information but also can give command to raise or drop barriers.
- User identification is a vague class identifier. It can verify the user's information by ID card and transponder.
- The controller controls the entire system. It can control the operation of the barriers and recording vehicle enter and exit time.

(3) Distance sensor system scenario:

- Sensor, Controller
- The distance sensor is used to measure the distance between the vehicle and the barrier, and send distance information to the staff
- The controller controls the entire system. It can control the operation of the barriers and recording vehicle enter and exit time.

Conclusion

A database of user information for storing user data about entry time, exit time, vehicle location and parking fee. When the user enters the parking lot, the screen can display the user entry time and parking location, and when the user leaves the parking lot, the duration time and parking fee are displayed. When a security officer or sensor receives a command, the controller should control the barriers to rise or drop. The sensor is used to detect if the distance between the obstacle and the car is a safe distance.

Final classes is listed below:

- User identification
- Controller
- Sensor
- Screen display

2.3.2 Stereotypical classes

- <<Control>>: Controller
- <<Boundary>>: Sensor, User identification, Screen display,
- <<Entity>>: Sensor

2.3.3 Class diagram

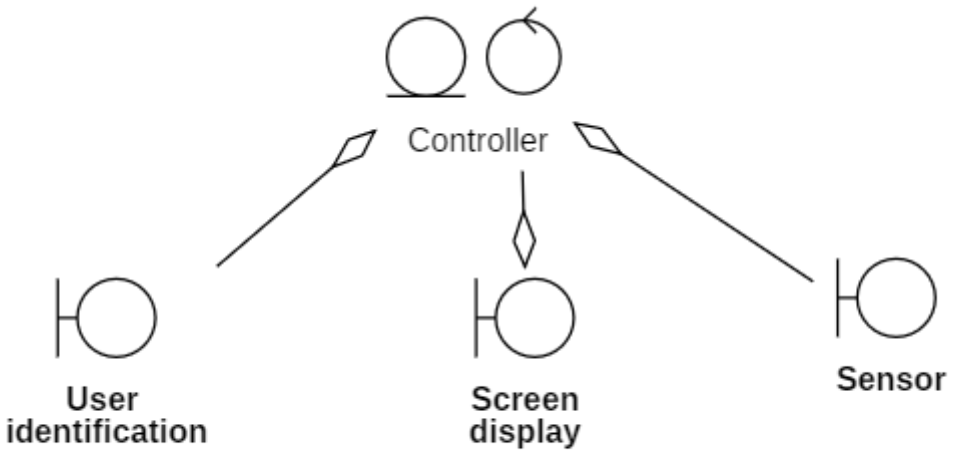

Figure 2.2. Class Diagram (Stereotypes)

2.4 CRC Cards

Table2.1 CRC card of User identification

User identification	
Responsibilities	Collaborators
Class User identification will verify user information by ID card or transponder. Then this class sends user information to controller.	Controller

Table2.2 CRC card of Controller

Controller	
Responsibilities	Collaborators
Class Controller will control all systems in car park. It stores all the information of the user. Therefore, the class controller can check the user information and check if the distance between the user's vehicle and sensor is a safe distance. If the user information is correct, it will print the label and lift the barrier. If the information is incorrect, the controller will send a message to security staff. If the distance is safe, the barrier is controlled to fall. The class controller can also store tag numbers, entry time and vehicle location number. Then, when the car leaves the parking lot, it can calculate and display the cost, the user has to pay fee based on the duration the car is parked in the parking lot. If the parking time is within the free range, the user will exit the parking lot free of charge and the barrier will rise. If the user needs to pay parking fee, the controller will reduce the user's corresponding amount in their ID cards or transponders. After the payment is successful, the controller will lift	User identification, Sensor, Screen display

the barrier to let user pass.	

Table2.3 CRC card of Sensor

Sensor	
Responsibilities	Collaborators
Class Sensor used to detect the distance between car and barrier when user want to leave the car park. If the distance is the safe distance, then sensor will give a instruction to barriers let it drop. When barriers received the order it will be lowered.	Controller

Table2.4 CRC card of Screen display

Screen display	
Responsibilities	Collaborators
Class Screen display used to display the entry time, tag and vehicle location when user entering the parking lot. Besides, after user leaving the car park the screen will display the leaving time, duration time and parking fee.	Controller

2.5 Collaboration Diagram

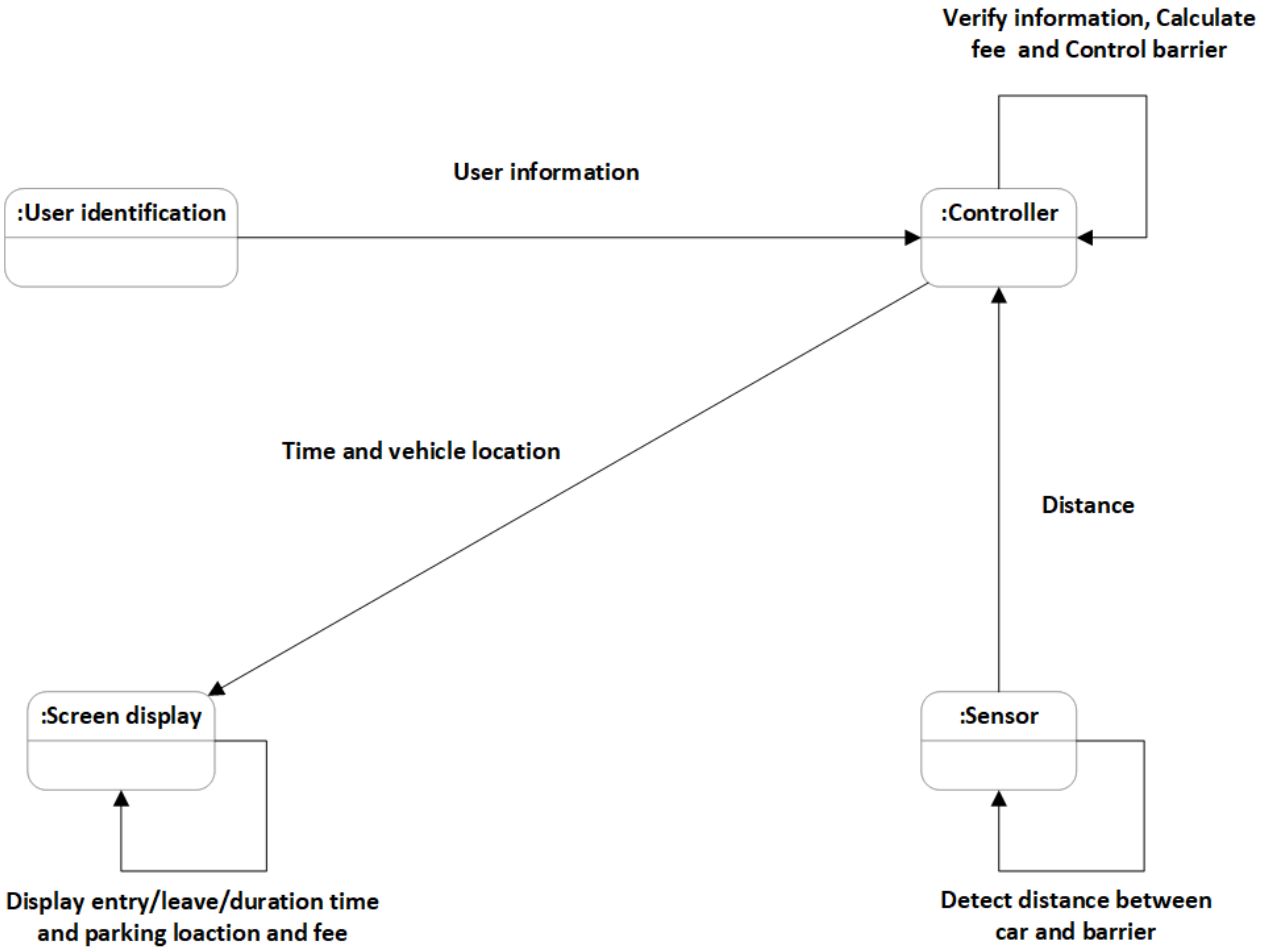

Figure 2.3. Collaboration Diagram

2.6 Interaction Diagram: Sequence Diagram

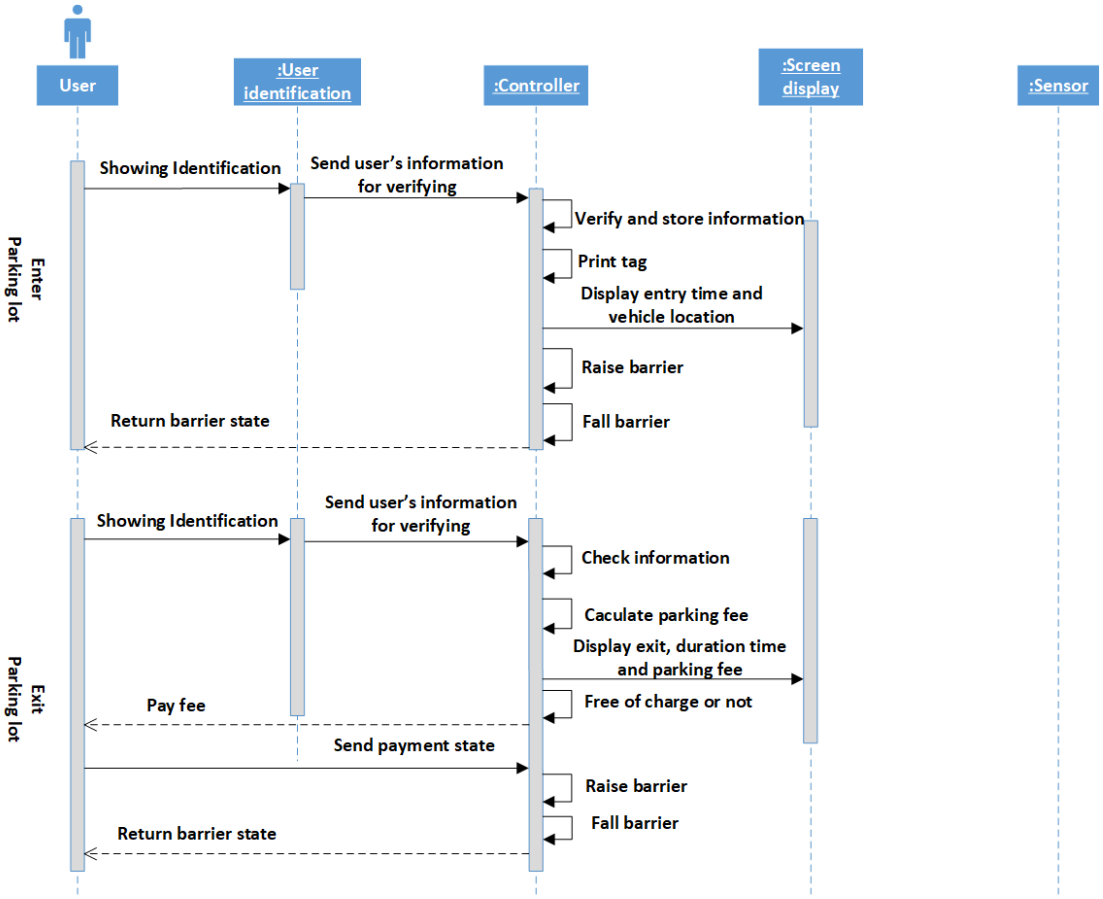

Figure 2.4. Interaction Diagram: Sequence Diagram of User

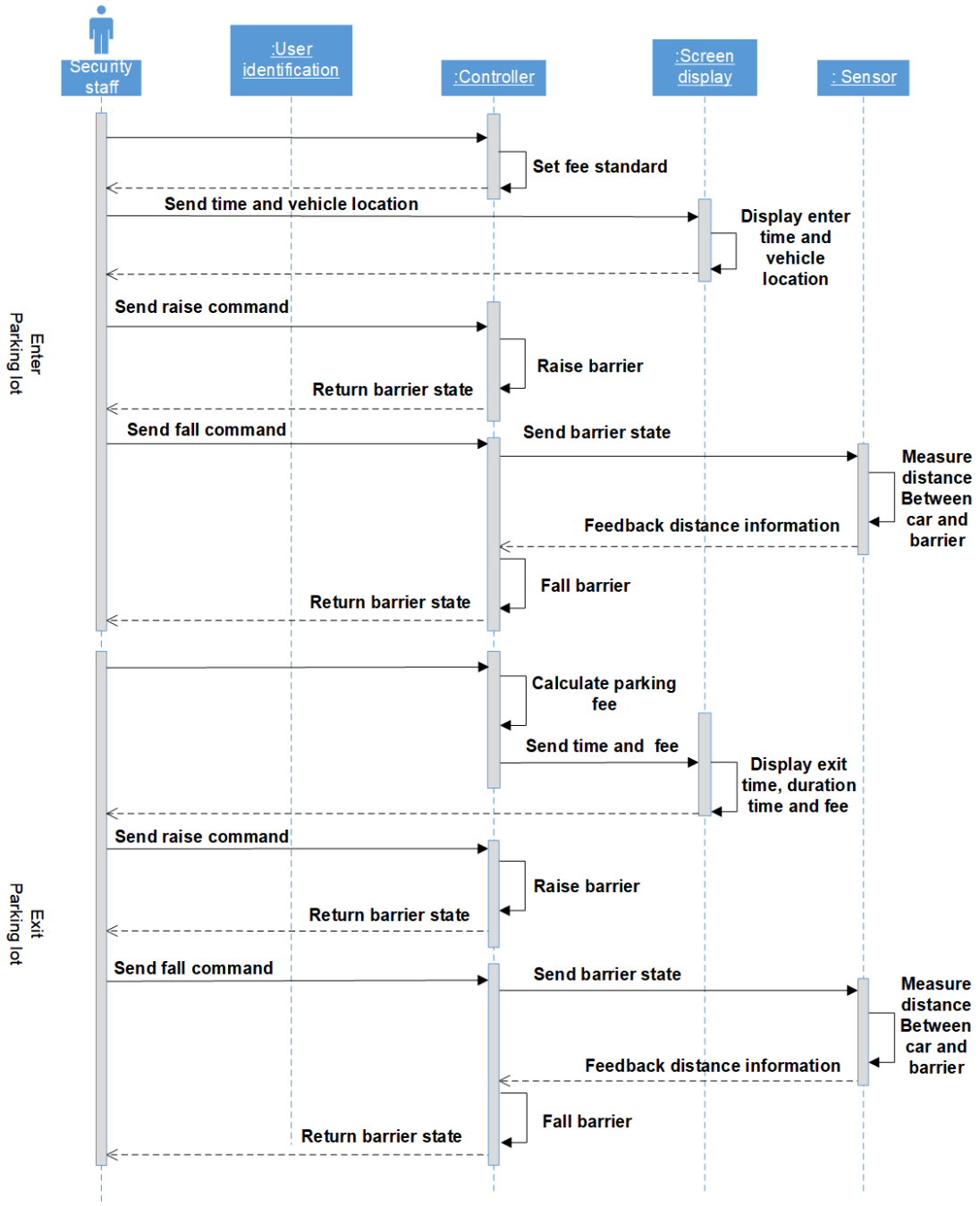

Figure 2.5. Interaction Diagram: Sequence Diagram of Security staff

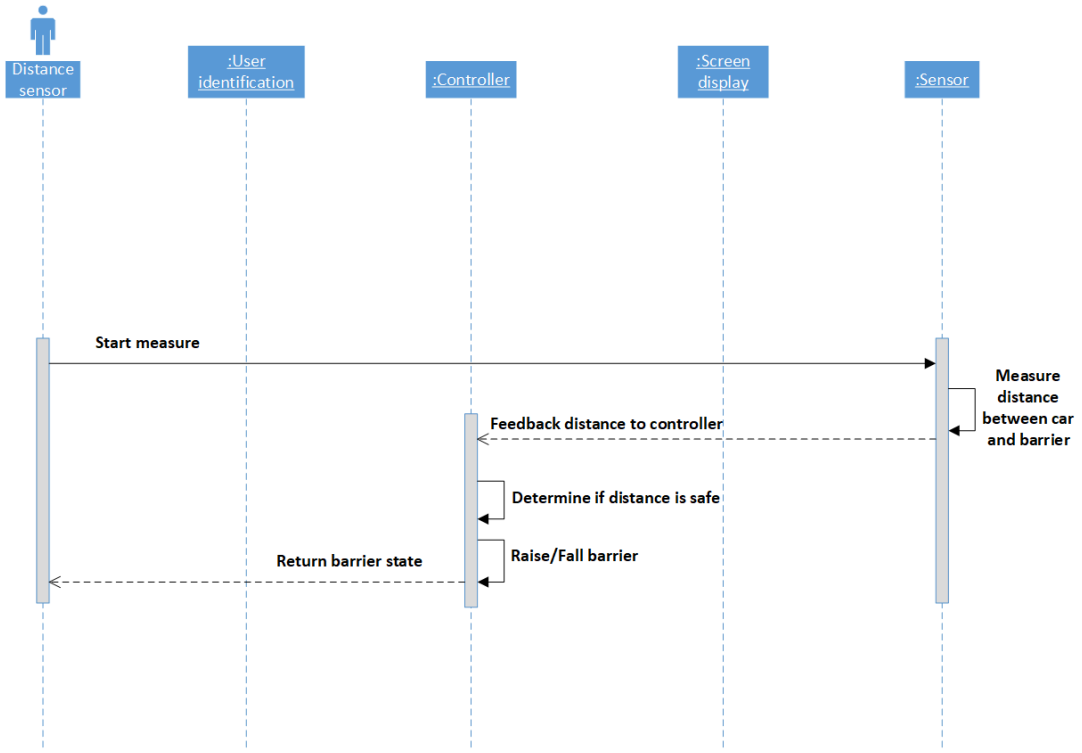

Figure 2.6. Interaction Diagram: Sequence Diagram of Distance sensor

2.7 State chart diagram

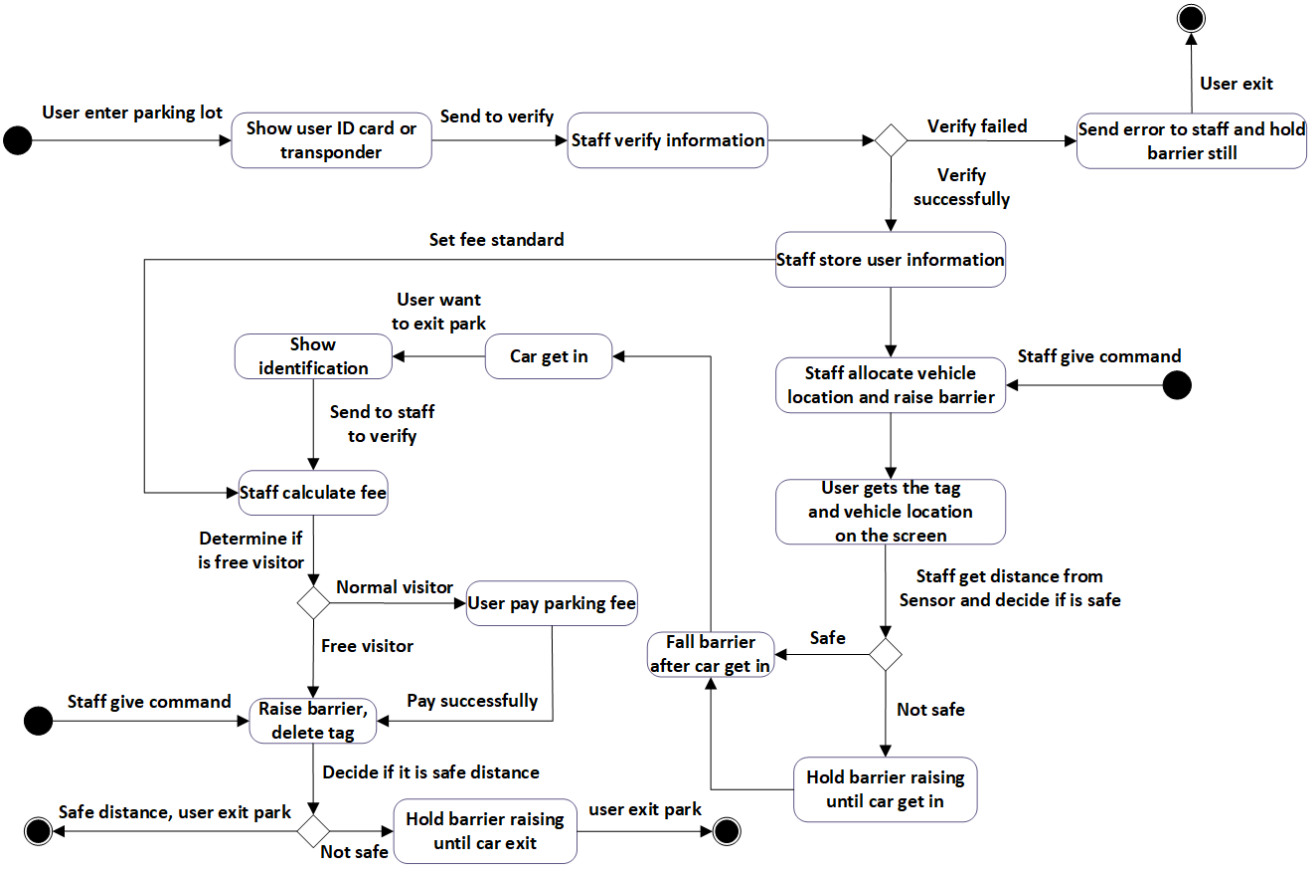

Figure 2.7. Initial State Chart Diagram

3 Analysis Model

3.1 Attribute

Table 3.1. Attribute

Class	Attribute	Comment
User Identification	Information reading	Boolean
	Information transmission	Boolean
Controller	Barriers state	State
	User's information verify	State
	Fee paying state	State
	Tag number	Number
	Enter time	Number
	Leave time	Number
	Duration time	Number
	Park location	Integer park number
Sensor	Distance	Integer array
	Information transmission	Boolean
Screen display	Enter time	Number
	Leave time	Number
	Duration time	Number
	Tag number	Number
	Parking fee	Number
	Park location	Integer park number

3.2 Methods

Table 3.2. Methods

Class	Method	Comment
User Identification	GetUser'sIDcard()	Get information about user
	GetUser'sTransponder()	
	GetTagInformation()	
	SendInformation()	
Controller	RaiseBarrier()	Control barrier by using information from sensor
	FallBarrier()	
	VerifyUserInformation()	
	CalculateFee()	
	PayFee()	
	SetTagInformation()	
	PrintTag()	
	DeleteTagInformation()	
	CheckDistance()	
	CheckPaymentState()	
	SendEnterInformation()	
	SendParkingLocation()	
	SendBarrierState()	
	SendParkingFee()	
	SetFeeStandard()	
	SetEnterTime()	
	SetLeaveTime()	
	SetDurationTime()	
	SetParkingLocation()	
Sensor	GetDistance()	
	SendDistanceInformation()	Feedback information to controller

Screen display	DisplayEnterTime()	Display time, fee and vehicle location
	DisplayLeaveTime()	
	DisplayDurationTime()	
	DisplayTagNumber()	
	DisplayParkingFee()	
	DisplayParkingLocation ()	

3.3 Sequence Diagram

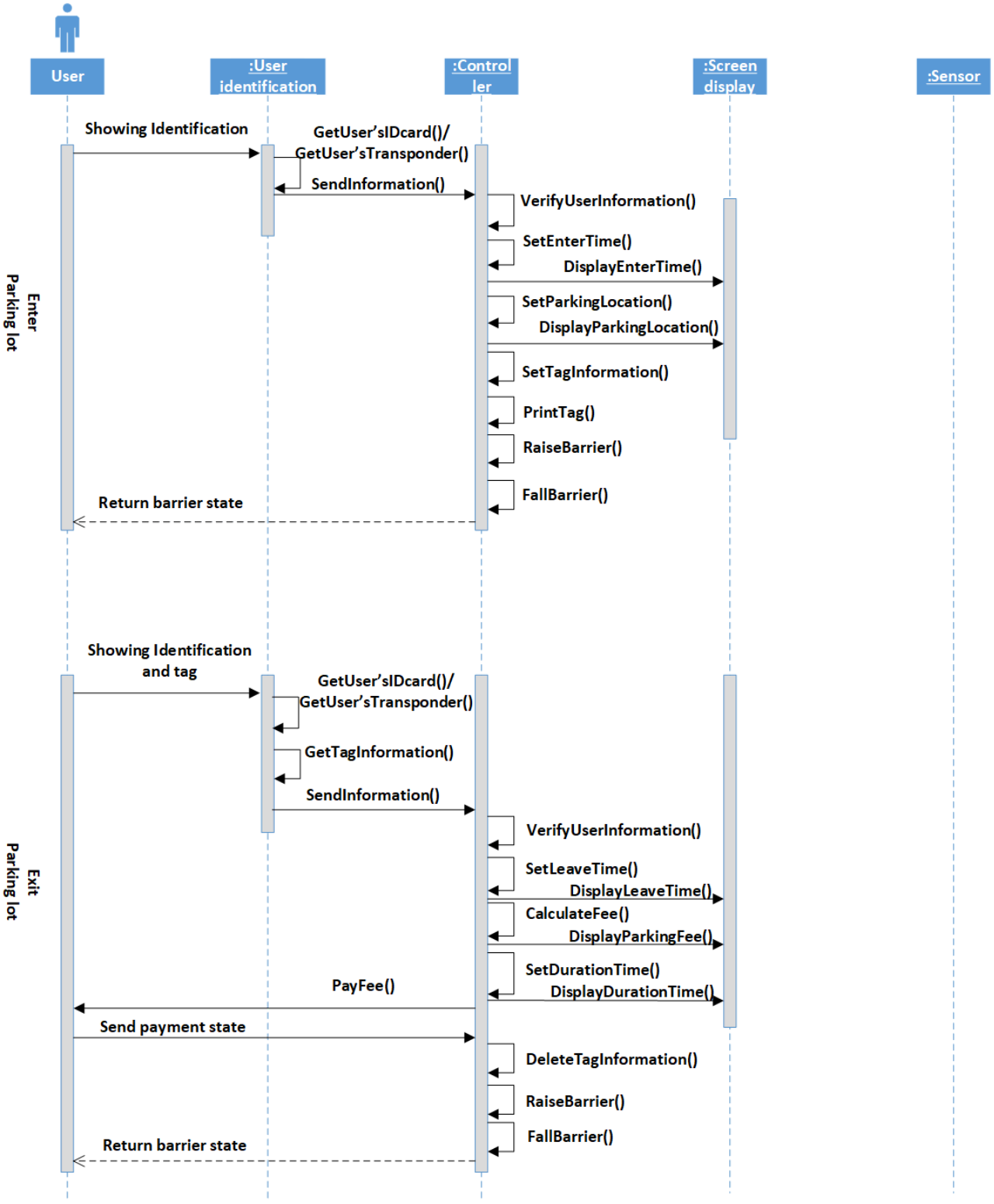

Figure 3.1. Analysis model sequence diagram of User

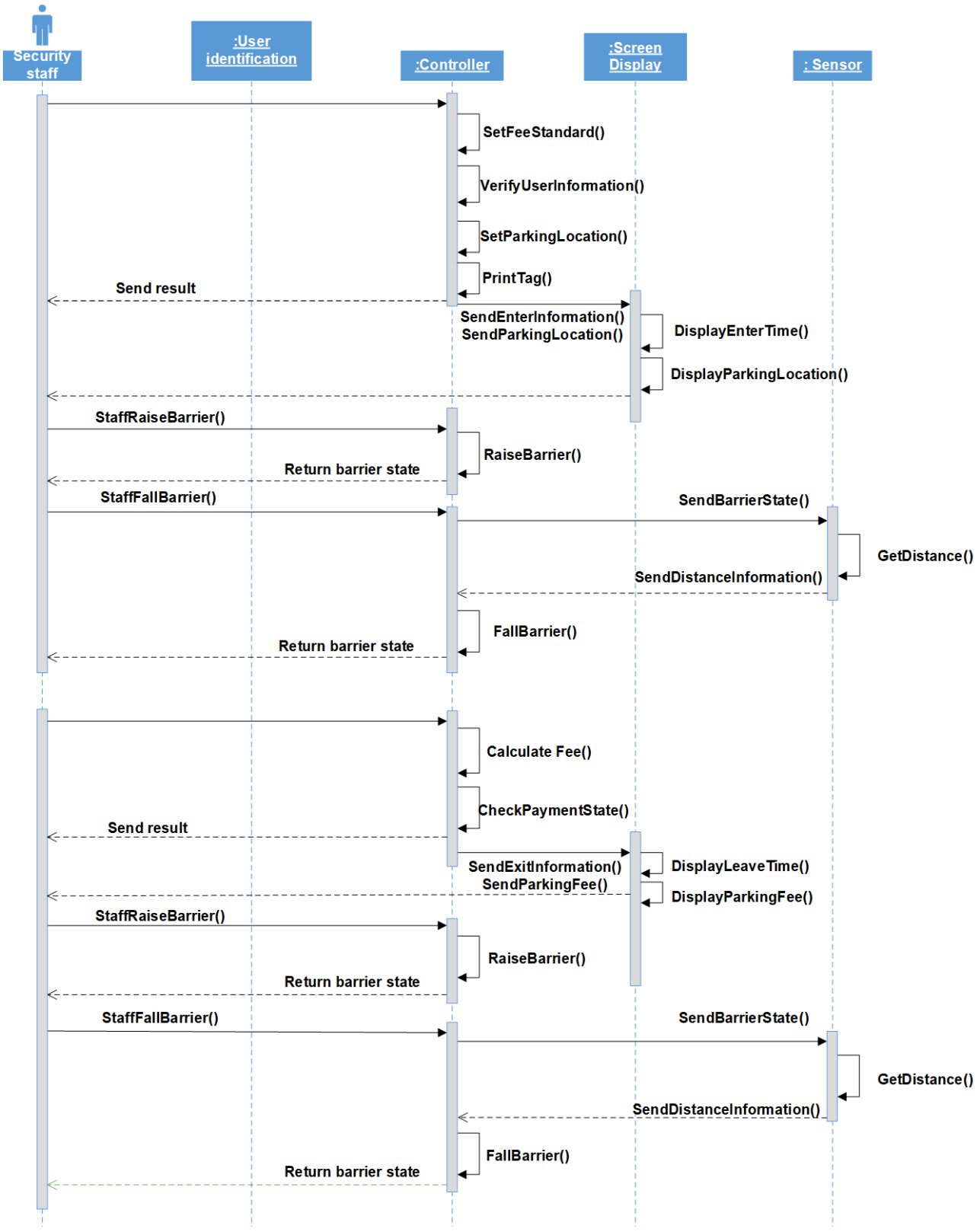

Figure 3.2. Analysis model sequence diagram of Security staff

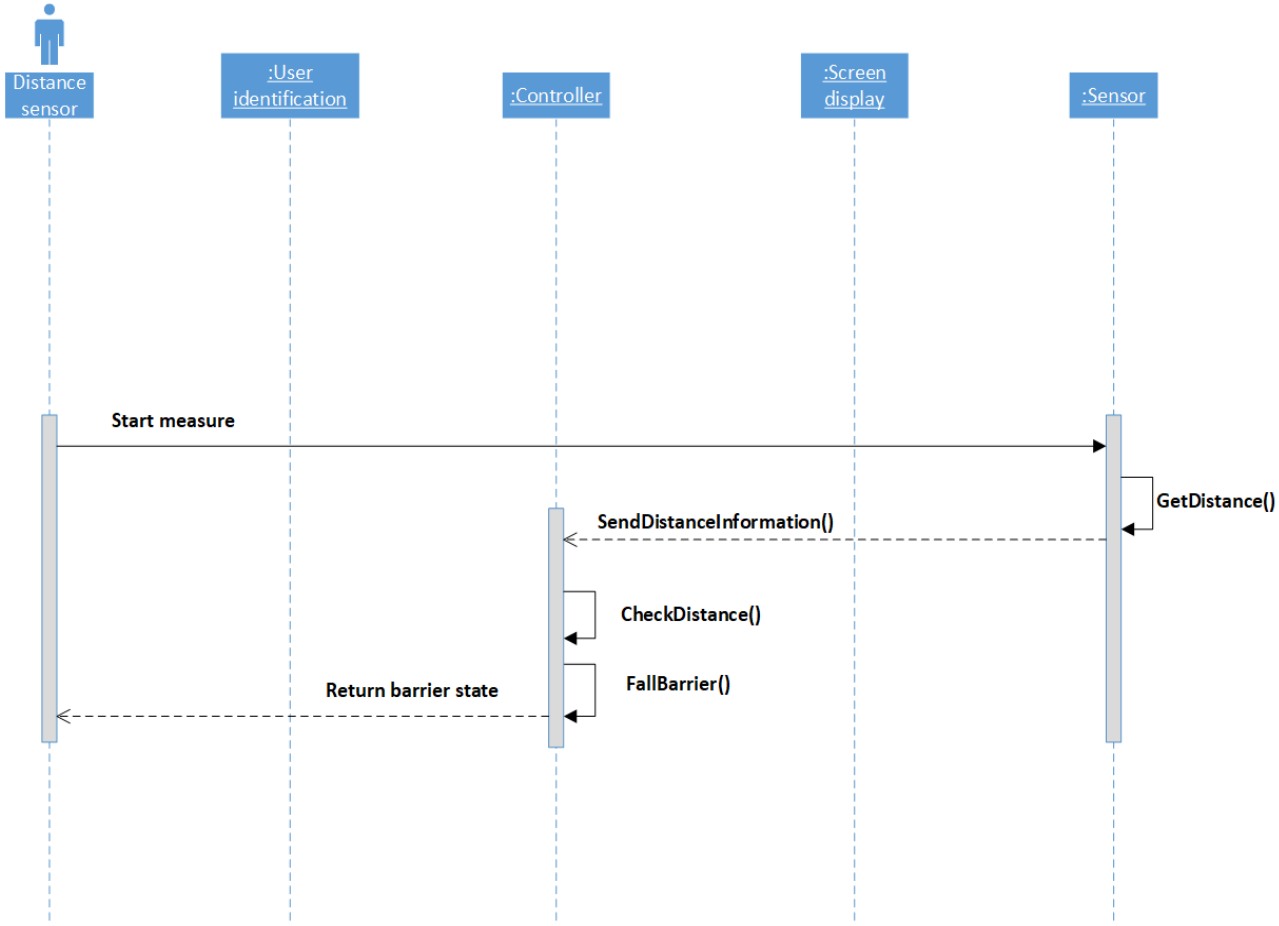

Figure 3.3. Analysis model sequence diagram of Distance sensor

3.4 Class Diagram

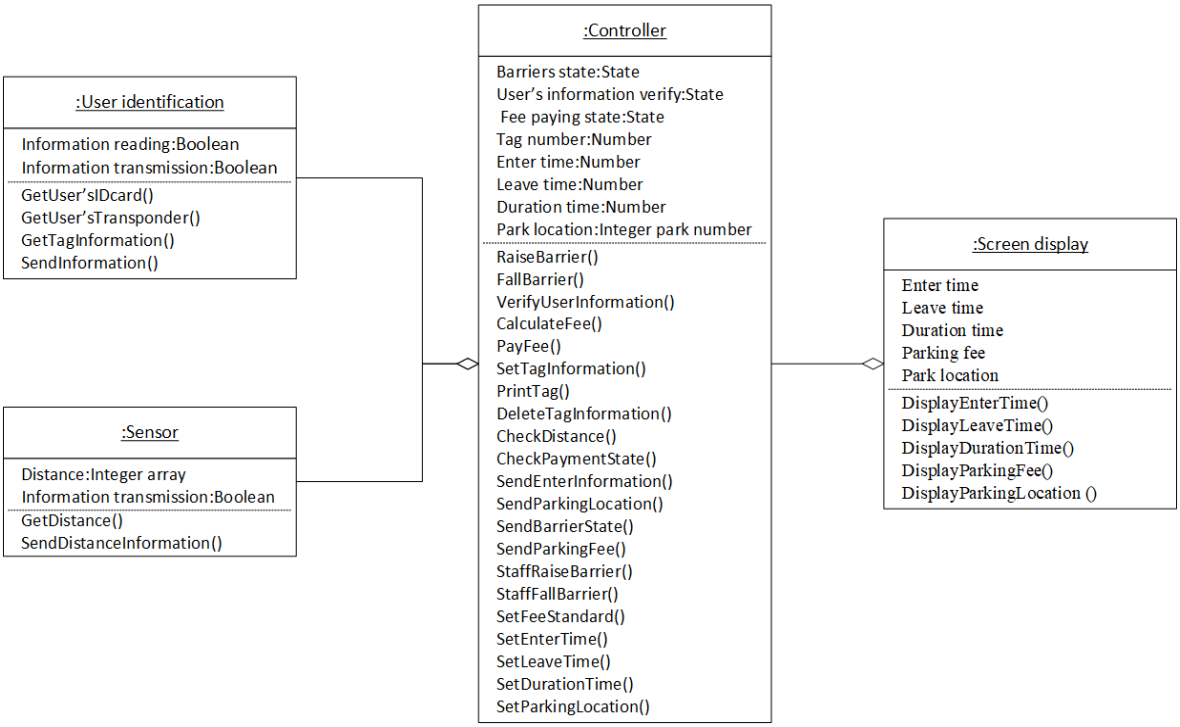

Figure 3.4. Analysis Model Class Diagram

3.5 State Chart Diagram

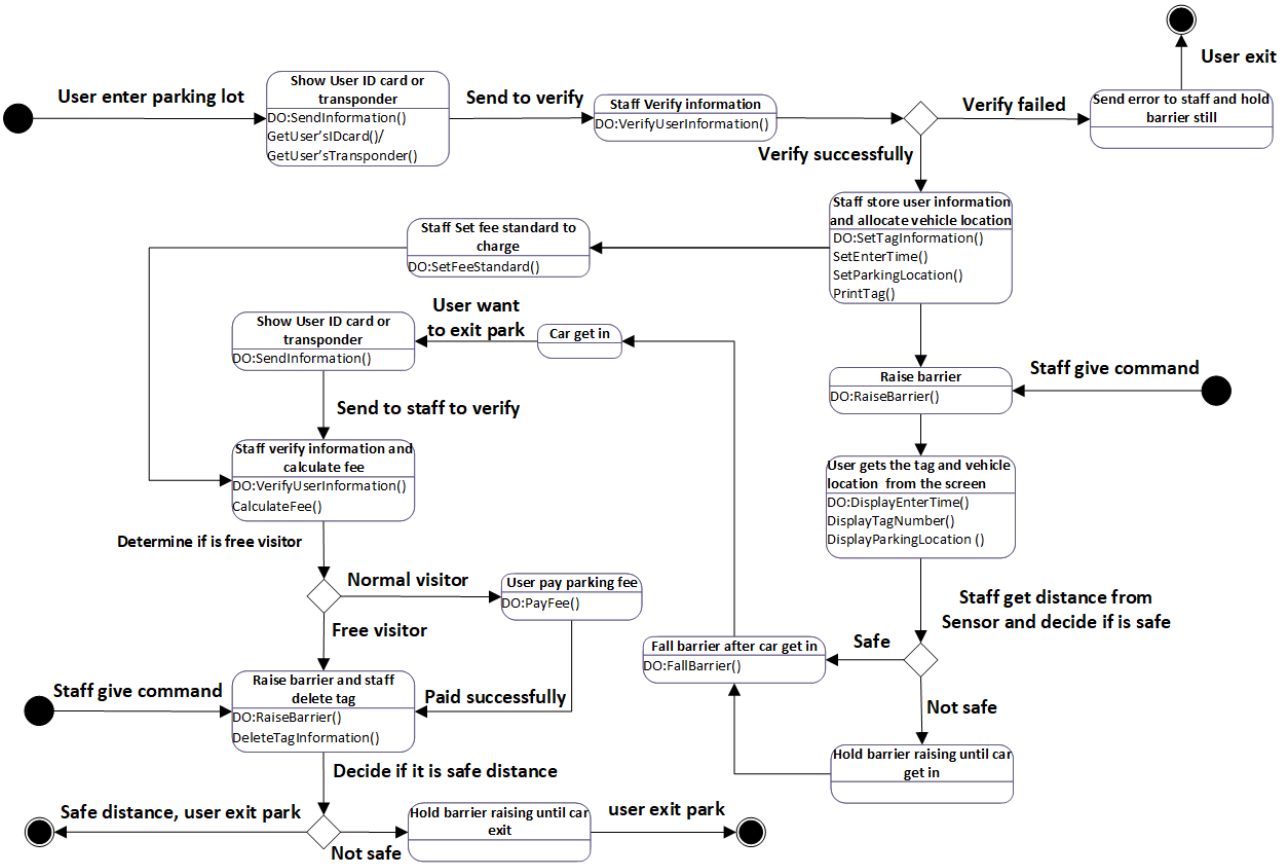

Figure 3.5. Analysis Model State Chart Diagram

3.6 Non-functional Requirements

When there is no position in the parking lot, the operator will not let the vehicle in either.

Besides, the controller operated by the staff with the highest administrative authority which controls the entry and exit of the vehicles.

3.7 Packages

In this design, there are only four classes. It doesn't include so many classes and functions. Thus, it does not require packages.

4 Design Model

4.1 Revisit Use-Case Model

The design requirements have been reached in task book.

4.2 Sequence Diagram

The controller class has the highest administrative privileges. It controls the entire system, including verifying user information, assigning parking locations, raising and lowering obstacles, calculating parking fees and sending information to security staff.

The screen display class is used to display entry and exit times, tag number, parking locations and parking fees.

Sensors are used to measure distances to control the time it takes for barrier to be reduced and to avoid damage to the vehicle.

4.3 Textual Description of Object to Object Interaction

(1) User identification class is used to get the user's information and transfer the information to controller.

(2) Controller class controls the entire system, including verifying user information, assigning vehicle locations, raising and lowering barrier, calculating parking fees and sending information to security staff.

(3) Screen display class is used to display entry, exit times and duration time, tag number, parking locations and parking fees.

(4) Sensor are used to measure distances to control the time it takes for barrier to be reduced and to avoid damage to the vehicle.

4.4 Subsystem

Because it is not a difficult and complex design. Thus, there are no subsystem in design.

4.5 Deployment model

The car park controller only needs a processor. Thus, there is no need to use extra processor.

4.6 Implementation of Non-functional Requirements

(1) Controller class has the highest administrative authority which controls the entry and exit of the vehicles. It can get and store the user's information from user identification, and get the distance information from sensor to control the raise and fall of barrier.

(2) Use connection-function to send the message.

4.7 Reconsider the Attributes

It will be shown in the class diagram.

4.8 State Chart

It has been shown on above.

4.9 Class Diagram Showing Visibility

Class	Attribute	Comment
User Identification	+Information reading	Boolean
	+Information transmission	Boolean
Controller	+Barriers state	State
	-Information verify	State
	-Fee paying state	State
	-Tag number	Number
	+Enter time	Number
	+Leave time	Number
	+Duration time	Number
	+Park location	Integer park number

Sensor	+Distance	Integer array
	+Information transmission	Boolean
Screen display	+Enter time	Number
	+Leave time	Number
	+Duration time	Number
	+Tag number	Number
	+Parking fee	Number
	+Park location	Integer park number

Class	Method	Comment
User Identification	+GetUser'sIDcard()	Get information about user
	+GetUser'sTransponder()	
	+GetTagInformation()	
	+SendInformation()	
Controller	+RaiseBarrier()	Use all information from sensor
	+FallBarrier()	
	-VerifyUserInformation()	
	-CalculateFee()	
	-PayFee()	
	+SetTagInformation()	
	-PrintTag()	
	+DeleteTagInformation()	
	+CheckDistance()	
	-CheckPaymentState()	
	+SendEnterInformation()	
	+SendParkingLocation()	
	+SendBarrierState()	
	-SendParkingFee()	
	-SetFeeStandard()	
	+SetEnterTime()	

	+SetLeaveTime()	
	+SetDurationTime()	
	-SetParkingLocation()	
Sensor	+GetDistance()	
	+SendDistanceInformation()	
Screen display	-DisplayEnterTime()	
	-DisplayLeaveTime()	
	-DisplayDurationTime()	
	-DisplayTagNumber()	
	-DisplayParkingFee()	
	-DisplayParkingLocation ()	

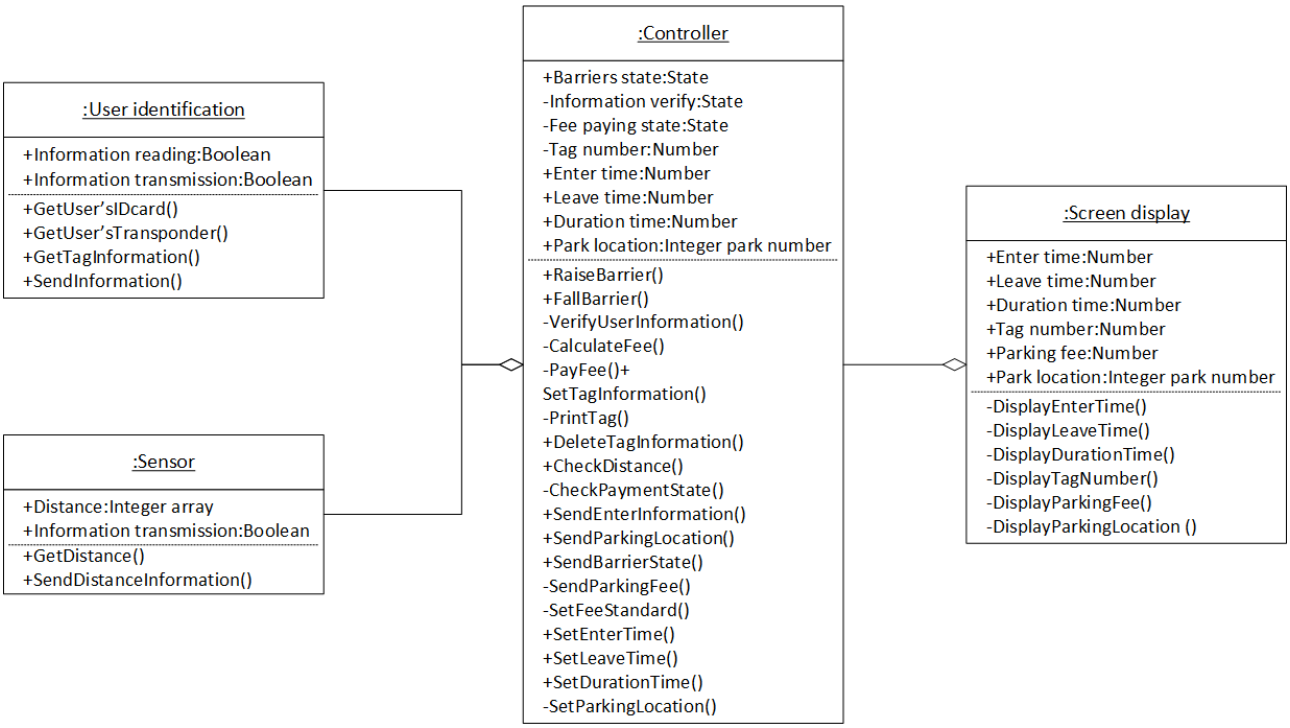

Figure 4.1. Design Model Class Diagram showing visibility

4.10 Class Implementation (Psuedo Code)

Pseudo code for managing the system list:

```
Parking_Limit()
Begin
If(carpark_user(number)=max(number))
    Then BarrierState = Fall
    Printf("There are no extra locations here right now.")
Return (BarrierState)
End

User_Enter()
Begin
If (User_ID == Database_UserID)  //Get and store user information
    BarrierState = Raise
    Then Position = StaffAllocate(Location)
Else
    BarrierState = Fall
Return (BarrierState)
End

Sensor_Distance()
Begin
If (distance <=1)   //Define safe distance is 1
    Then BarrierState = Raise
Else
    BarrierState = Fall
Return (BarrierState)
End

```

Figure 4.2. Design Model Code_1

```
User_Exit()
Begin
If (User_ID == Database_UserID)
    If (TagNumber == Database_UserID)
        Then BarrierState = Raise
        Fee = (Time/60)*3
        Pay_Fee(Fee)
    Else
        BarrierState = Fall
Else
    BarrierState = Fall
Return (BarrierState)
End

Staff_Calculate(Fee)
Begin
If (time <= 30)      //Define free time range is 30
    Then fee = 0
Else
    fee = Fee
Return (Fee)
End

Sensor_Distance()
Begin
If (distance <=1)    //Define safe distance is 1
    Then BarrierState = Raise
Else
    BarrierState = Fall
Return (BarrierState)
End
```

Figure 4.2. Design Model Code_2

5 Test

This is a simple design and further test development is not considered necessary.

6 Conclusion

The task adopts the engineering idea of object-oriented design, and uses UML as the design tool to carry out detailed requirements analysis design for the parking lot management system. Therefore, it can meet the basic functional requirements. The management software can control access to the

parking lot and automatically charge to the paid roll.

It's simple, with good performance and scalability, high response speed and high efficiency. Besides, we have basically implement our goals in introduction. User enter the parking lot is identified by a personal tagging system. System allows the identified user to enter the parking lot. When the car get in, the user will get the tag. It contains information in the database, such as enter time and vehicle location. When user wants to exit the car park. User needs to use the tag which got before. If the user does not need to pay any parking fee, the barrier will be raised and the user can exit the park directly. If the user needs to pay, they will have to pay the fee then the barrier will be upgraded. In this system, the security staff can control the barrier anywhere in the parking lot and receive feedback from the control system and distance sensor. When the barrier is raised, the sensor begins to measure the distance between the barrier and the car. It then feeds information back to the control system. When the vehicle is away from the barrier and reaches a safe location, the control system issues a command to lower the barrier.

Above all, the system can meet the actual campus management system and provide a fast, simple and standardized management platform. However, it also needs to consider some details and should be added in the future.