

Testing Strategy:

Scope And Overview

Project overview along with information on team6 members should use this document. Also, customers will review and approve this document. Testing activities and phases in the following to be carried out with timelines.

Test Approach

In this section, we will define the testing process, level of testing, roles, and responsibilities of every team member.

- Shiqi Liu: development testing. In this part, the developer needs to do the unit testing and component testing.
 - ❖ Unit Testing needs to automatically test individual objects and methods in the parking-lot application. We will test functions and its output. In login() method, username and password need to input correctly and matchly. In payMembership() method, users need to click the 'confirm' button to be the membership. In reservation() method, plate number and credit card needs to be input correctly. In findParkLot()method, gps need to find the nearest parklot, and parklot information need to be tested. Payment() method, user bill need be calculated correctly. In add() delete() record()method, developer need to test add() is able to receive a new plate number or not. Delete() is able to delete the selected plate number in the system. Record() is able to store information.Track() method, developer check if admin is able to tack monthly revenue, parking usage, percentage of customers. Define() method needs to check the if system define new parking areas, specify the number of parking lots, and the parking cost per hour and compute the available parking lot.
 - ❖ Component Testing needs to test related groups of objects and system testing in the parking-lot application. We will test the interface between database and system. Check will the message be sent to the end-user when the user needs to get information, available park-lot, bills. Check if the components are compatible or not. Check interact correctly and transfer the right data at the right time.
- Fan Jun : release testing
 - Requirements testing
 - Functional requirements are tested in unit testing. Non-functional requirements testing are in following: security: test card information will be

protected or not. Reliability: test the website run on different browsers.
Efficiency: test the system response time and capacity.

- Scenario testing

In this section, we will verify “realistic” operation.

- ❖ Test Scenario 1: check the login Functionality

Check system behavior when valid username and password is entered.

Check system behavior when invalid username and valid password is entered.

Check system behavior when valid username and invalid password is entered.

Check system behavior when invalid username and invalid password is entered.

Check system behavior when username and password are left blank and Sign in entered.

Check Forgot your password is working as expected

Check system behavior when valid/invalid username number and password are entered.

- ❖ Test Scenario 2: check the Search Functionality

Check system behavior when GPS is not loaded and the internet is not loaded.

Check system behavior when GPS is loaded and the internet is not loaded.

Check system behavior when GPS is loaded and the internet is loaded.

- ❖ Test Scenario 3: check the Available ParkLot Functionality

Check system behavior when parklot is not available and the reservation is received.

Check system behavior when parklot is available and the reservation is not received.

Check system behavior when parklot is available and the reservation is received.

- ❖ Test Scenario 4: check the Reservation Functionality

Check system behavior when a reservation is made.

Check system behavior when a reservation is canceled.

- ❖ Test Scenario 5: check the Payments Functionality

Check system behavior when empty payment is entered.

Check system behavior when invalid card number and valid card cvv is entered.

Check system behavior when valid card number and invalid card cvv is entered.

Check system behavior when invalid card number and invalid card cvv is entered.

- ❖ Test Scenario 6: check the Membership Functionality

Check system behavior when user doesn't have a membership

Check system behavior when user does have a membership
Check system behavior when user have a membership and if the user
able to get guaranteed parking

Test Scenario 7: check the AdminTrack Functionality

❖ Test Scenario 8: check the BackendSystemSetting Functionality

- Jue Huang: user testing. To get our ultimate goal which is user acceptance, we need to involve users during the testing planning and execution. In this part, we would like the external users (people not in the team or organization) to do acceptance testing, the test can evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. Although after the software has undergone Unit, Integration, and System testing the Acceptance Testing may seem redundant, the Acceptance Testing is required because developers have included features on their own understanding.

■ Test cases with scripts:

• Example#1

Purpose	User able to log in with user ID and password
Preconditions	User has signed up, and user has not logged in
Input	Correct user ID Correct password
Process (Actions & Triggers)	1. Enter user ID 2. Enter password 3. Click 'login' button
Output	User logs in and sees the main page of the website for users

• Example#2

Purpose	User can pay the monthly fee to get membership
Preconditions	User has logged in, and isn't membership yet
Input	Correct credit card information
Process (Actions & Triggers)	1. Log in on the website 2. Find "join membership" page 3. Enter car plate number 4. Enter card information

	<ol style="list-style-type: none"> 5. Enter email for getting a confirmation email 6. Enter bill address 7. Click “pay”
Output	User joins the membership and able to enjoy features only for memberships

- Example#3

Purpose	Nonmember can choose a one time reservation for <u>alternative parking space</u> when selected parking is full (when making the online reservation)
Preconditions	The user is nonmember and the chosen parking space is full
Input	Choose one of the provided alternative parking spaces
Process (Actions & Triggers)	<ol style="list-style-type: none"> 1. Log in on the website 2. Make online reservation 3. Select a full parking 4. Choose from the provided alternative parking 5. Enter correct credit card information and other required information 6. Click “pay”
Output	Nonmember user get a one time alternative parking space

- Example#4

Purpose	Member can add a temporary plate
Preconditions	User has membership
Input	Temporary car plate
Process (Actions & Triggers)	<ol style="list-style-type: none"> 1. Log in on the website 2. Go to user's page 3. Click “car plate” 4. Click “add a temporary plate”

	<ol style="list-style-type: none"> 5. Click “done” 6. Check if the temporary plate is deleted on user’s file on second day
Output	Member can add a temporary plate for a day that gets automatically removed the next day