Report

Topic: Parking lot management system

Group members:

Qiu mengke

Xiao zijian

Qiu sixiang

Via Internet Programming

Assignment 3

1. Specification

2. Structure

3. Design

4. Implementation

5. Test

6. Evaluation

1. Specification

1.1 Description

The parking lot management system is, so far, a website application capable of uniting every single parking place into a larger network.

In this case, efforts are devoted to make it more convenient to record or to track the information of the car-parking flow.

1.2 Target audience

The system is created especially to meet the needs of managersin charge of private parking lots with multiple entrances while only one exits.

The system isgiven some of the specialties such as the ability to work with car-owners’ data so that other car-owners cannot park their cars in the lot if they are not on the list.

Also, different managers are able to co-operate the application by log in unique account of their own. Usually, for each entrance and exit there should be one manager. To avoid the concurrent issue, there should be only one exit.

1.3 Product functions

Current parking lot interface:

Cars in/Cars out

Revise parking cars in-time or parking position

Sort parking cars in in-time order

Sort parking cars in parking position order

Search parking cars by car id

Page querying

Owner table interface:

Delete existing car-owners

Update existing car-owners

Add new car-owners

Page querying

Parking history interface:

Delete specific parking history

Search parking cars by car id

Page querying

Other functions:

Account log in/log out

Other information tips and warnings

1.4 Requirements

1.4.1 Functional

1.The software system should record and display the owners’ information to users.

2. Only if the owners’ information is recorded in the system, can they park their cars in the parking lot.

3.The system automatically validates whether the car is in the owner’s information table or not.

4.On the main page which is the “current parking lot” page, the users can search the car by car’s ID, as a result, the car’s entry will be exhibited below.

5.The managing system allows users to delete, update, insert and search for each single entry.

6.Only the users who have registered in thesystem are given the right to manage the parking lot system.

7.Users can sort the entries’ information which was exhibited in the main pages in the order of in-time or the sequence of positions.

8.Every unsuccessful attempt to modify the data should be alerted from thesystem.

9. The current login information will be stored into session, thus the login action is unnecessary before closing the web browser.

10.The data will be sorted initially by time in the current parking lot page.

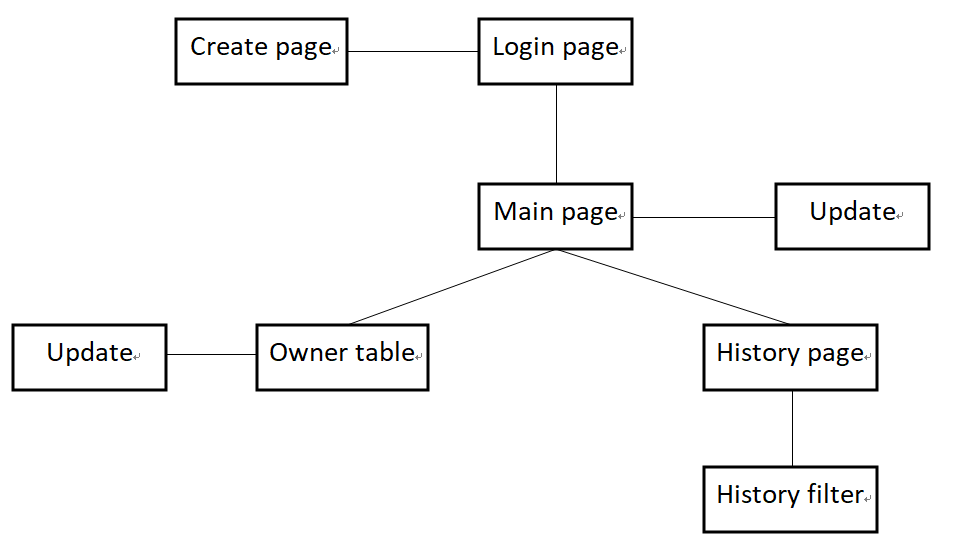
1.4.2 Nonfunctional

1. Password should be invisible at all time.

2. The users interface should be matched to the software functionalities which means each website page should at least contains some cars pictures.

3. Users shall receive notification while operations.

4. The main page will be redirected to the login page as a result of opening the page before login.

2. Structure

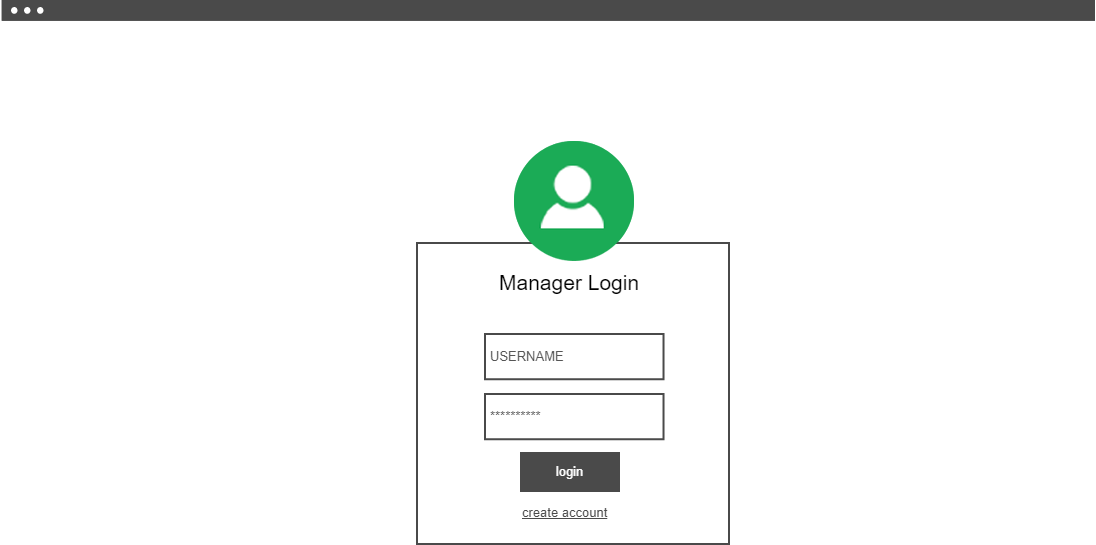
2.1 Sitemap

Reference for background image:

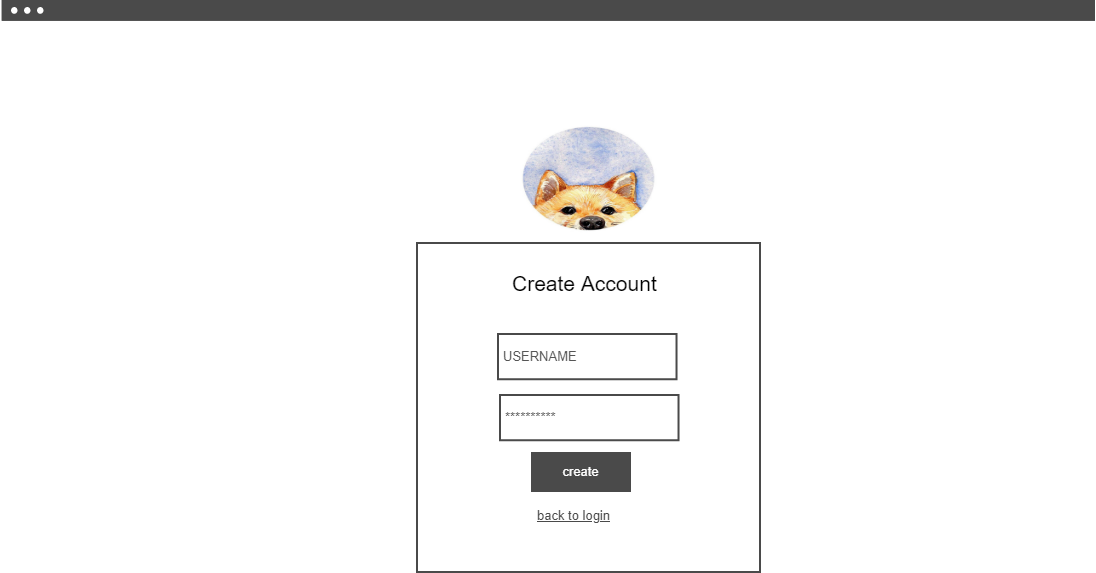
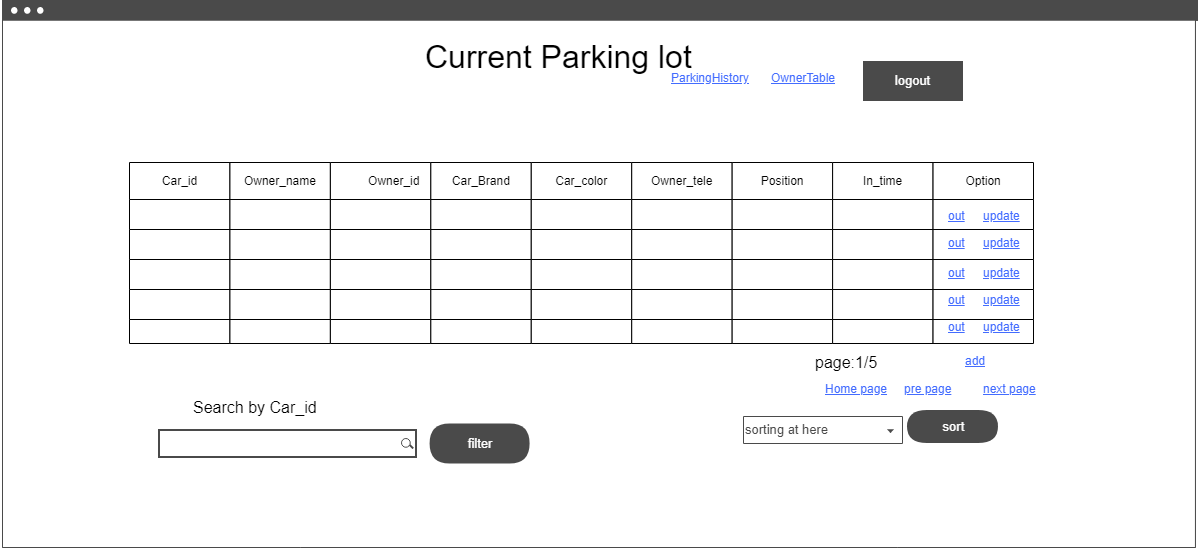
:<http://n.sinaimg.cn/sinacn17/204/w1200h604/20180621/9fb3-hefphqm0800034.jpg>

Reference for user image:

<https://www.58pic.com/tupian/yonghutouxiang.html>

2.2 Wireframes

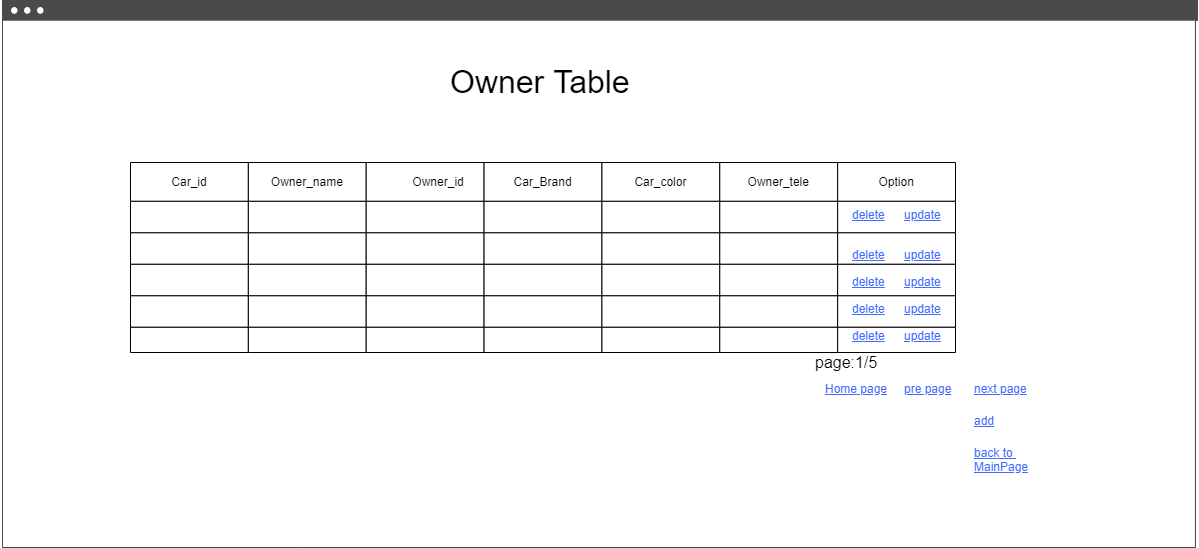
Log in page



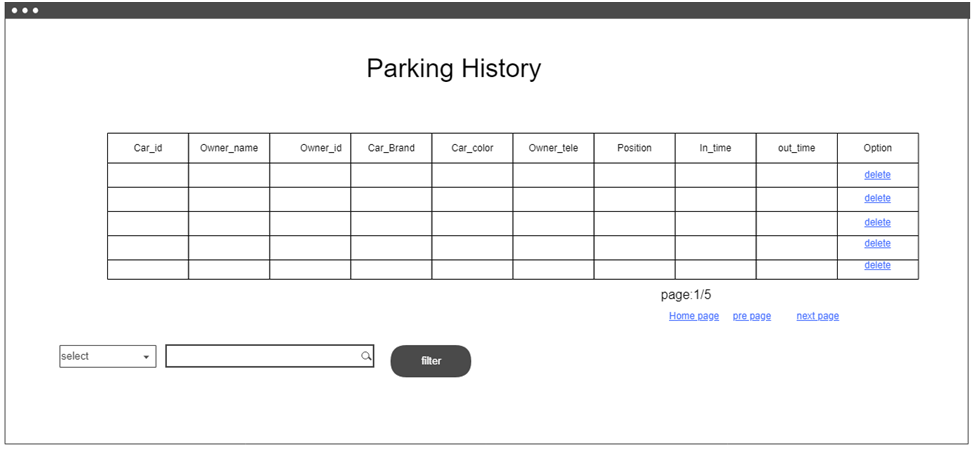
Creating new account

Main page showing the current condition of the parking lot.

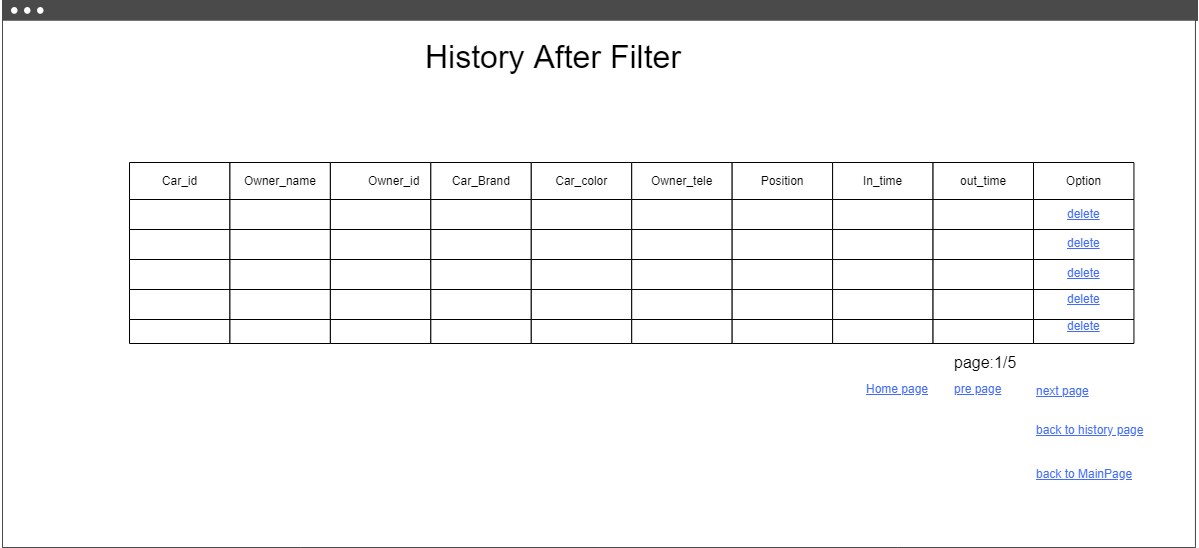
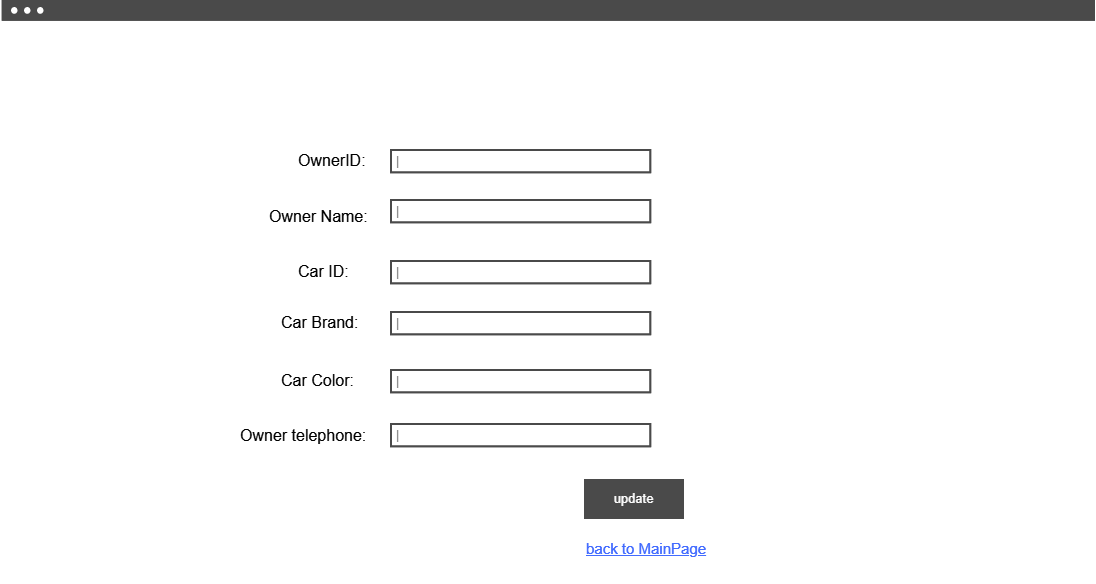
Managers can be facilitated by using the sorting function. They can also target key information using the searching function. Moreover, manager can access sub-pages in themain page or logout to the original login page.



Owner table page, where managers can update or delete information of car owners



History page implemented intentionally to display the former car-parking flow



Result page after filtering in history page

Updating page to update owners’ information

3. Design

3.1 Heuristics

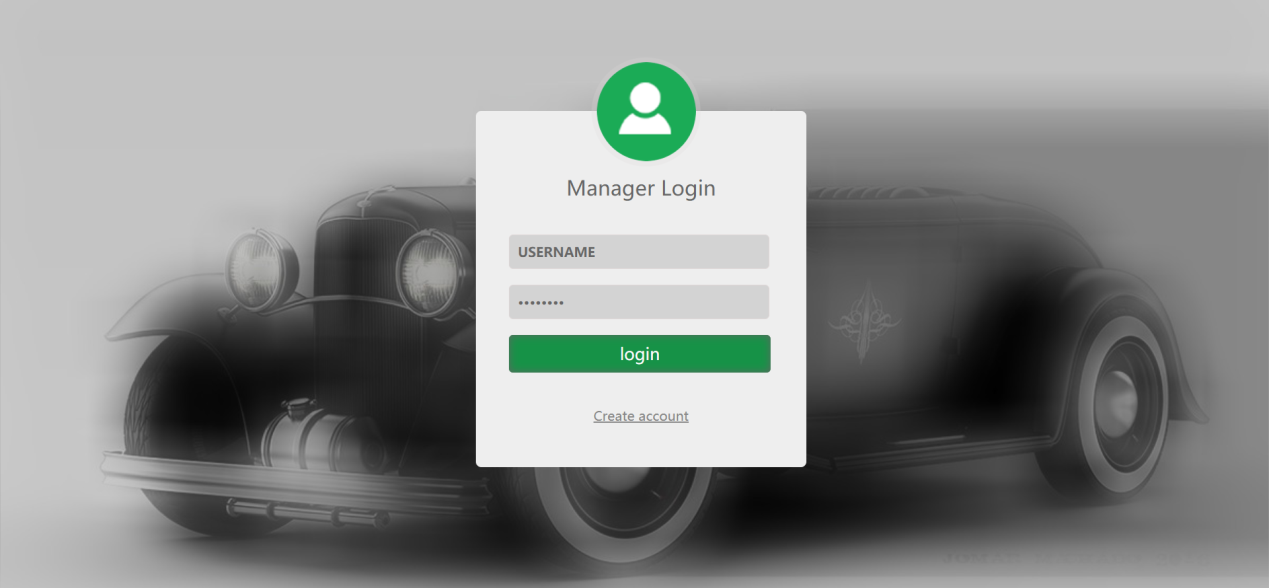
Inspiration, at the beginning, popped up since the willingness to realize a management system.

Considering that cars are becoming the most popular method of traffic whereas some of the private, small parking lots are still using traditional way to record and track the car-parking flow, an Internet integrated application is then well deserved implementing.

Target groups are those private parking lots managers who find it too complex to note the car owners’ information and check every time there’s a car driving in.

Apart from ordinary management system, specific requirement is, without doubt, the function which intelligently checks if the coming car is on the list of permission.

3.2 Screenshots



Manager login

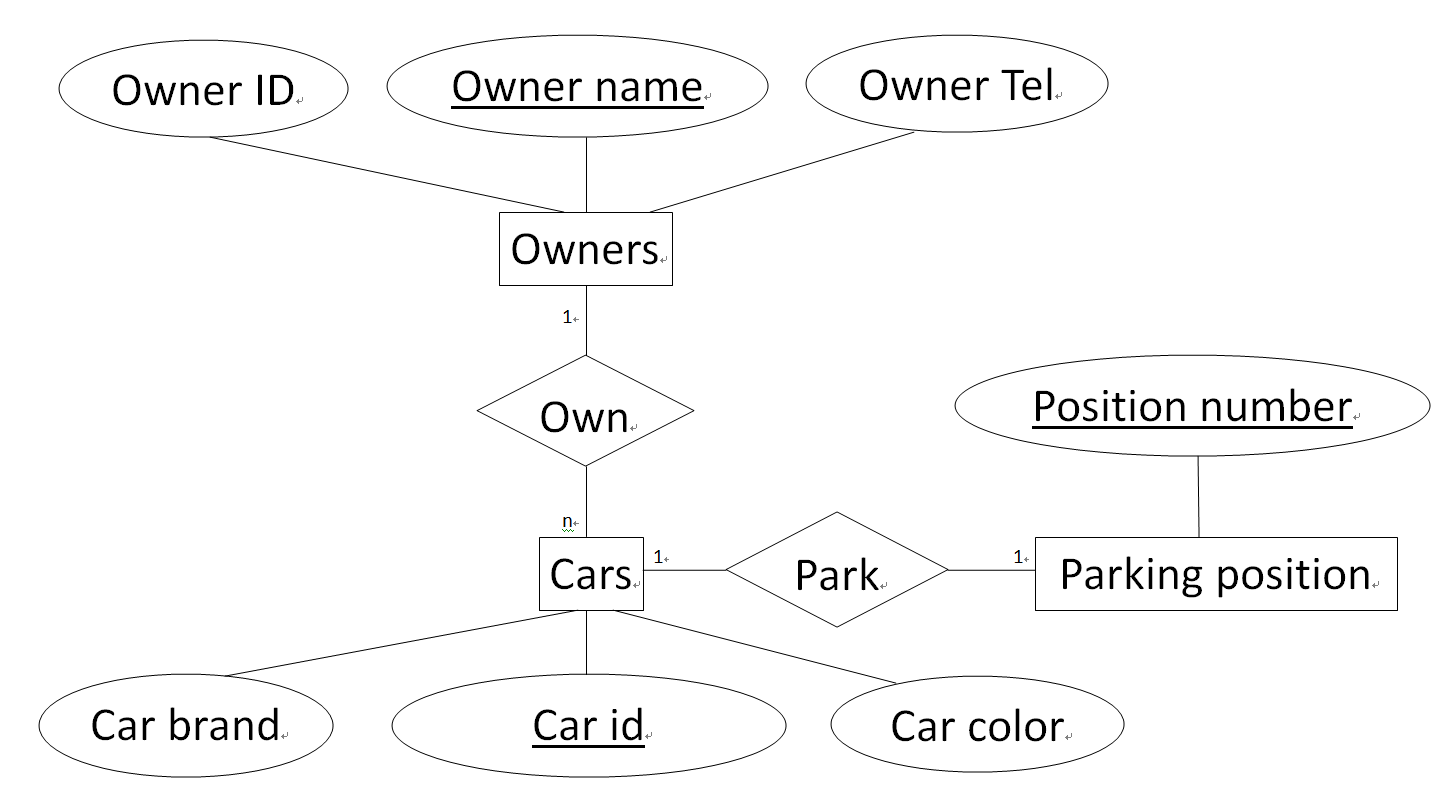


List of permission

4. Implementation

4.1 Class definition

The structure of the application mainly follows the MVC (Model, View, Controller), each part assume its own responsibility. In model part, the class “car” and “owner” were responsible as the data’s carriers. And the “DB Connector” class is responsible as a DAO (Data access object).

4.2 E-R model

4.3 Relative technologies

4.3.1 AJAX

In the current parking lot page and history page, AJAX was implemented into the filter and search functions. For instance, in the current parking lot page, filtering request is handled by “handleAjax.php” asynchronously after which the response will return in type of “JSON” and display the server-side evaluated result below.

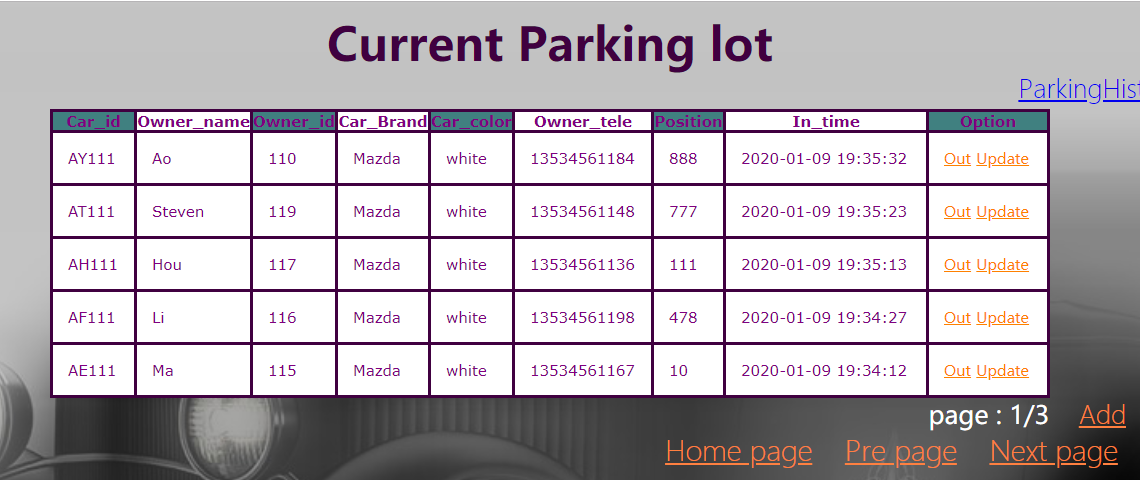


The system will throw an error message in case that the inputs do not exist.



4.3.2 Page querying function

Page querying function is implemented to restrict the number of shown entries in the current page. Otherwise, all the entries or information would be shown in one page which is against the original intention of user-friendly.



Only five entries are allowed to be shown in one page.

5. Test

5.1 Task assessment

Before going into the testing part, the website is ran several times internally in order to make sure that, from programmers’ angle, there’s no bugs or problems during general using.

The testing part mainly focuses on how the website reflects on the users’ practices upon the following factors.

readability, operability, efficiency…

5.2 Tasks

5.2.1 The log in interface

This session measures if the users can successfully create and log in their own account which is the very first step of knowing how to use the application.

5.2.2 The current parking lot interface

This session shows if the main page is readable enough for users to get knowledge of current parking lot situation as to operate in accordance with the car-parking flow.

5.2.3 The parking history interface

The parking history page is merely for confirmation apart from complicated manual stuffs. Therefore, more emphasis is placed on collecting feedbacks to improve the readability.

5.2.4 The owner table interface

This page is a feature of private parking lots. Testees were told to substitute the manager’s role and point out hardship they encountered during interacting with the functions included in.

5.3 Testing process

5.3.1 Foreword

Altogether, five testees were invited to the test. They were asked to run through each interface of the application and give the overall feedbacks afterward.

5.3.2 Results

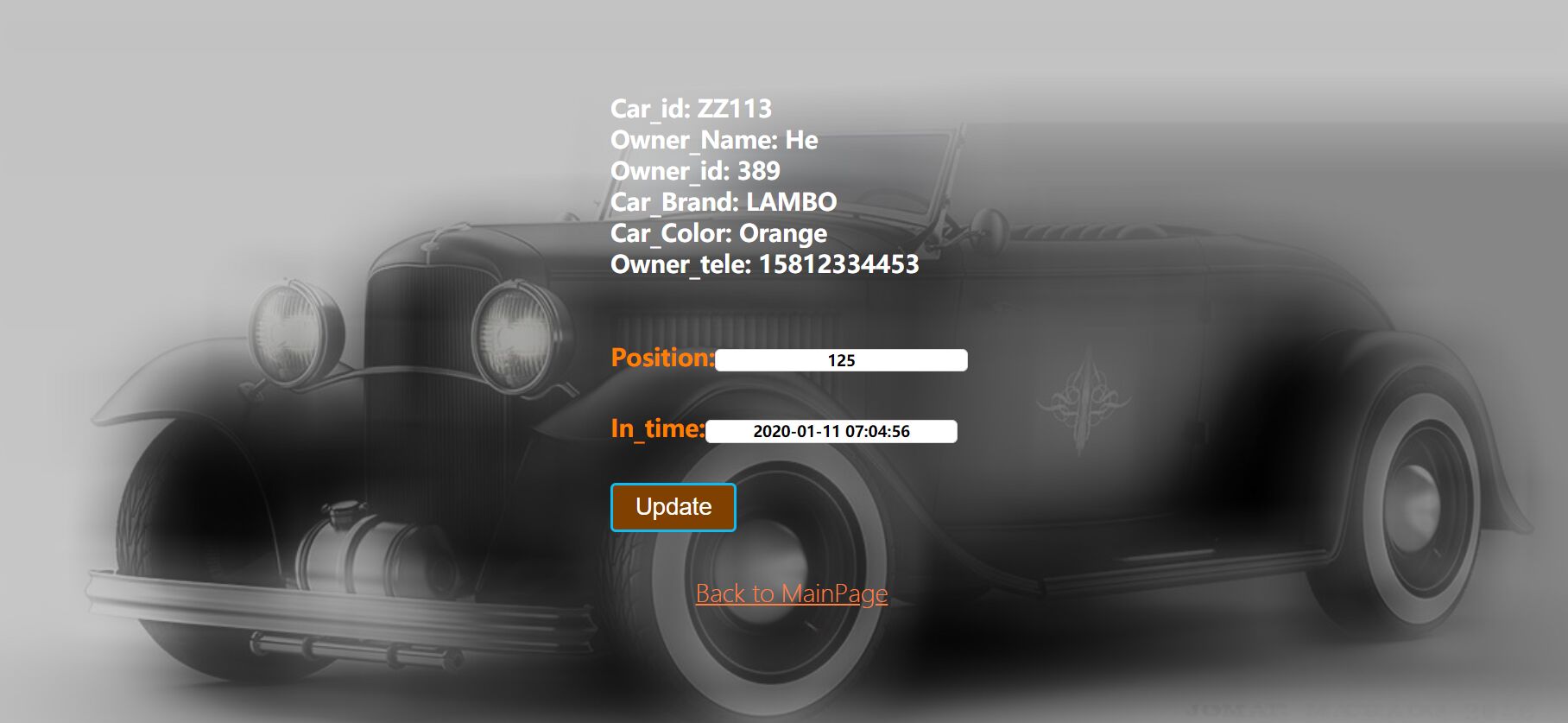
1. Optimization of the login interface:

Now, as soon as the new manager succeeds to create his own account, the system will redirect right to the login page instead of the account creating page.

Also, the login page improves the user-friendliness by retaining their input content instead of deleting it after mouse moving out of the text field.

2. Modification upon the current parking lot interface:

As one of the testees claimed, now, from the main page, only the position and the entry time is accessible for updating as shown below.



3. General revision:

Form validation is presently applied on several places including position adding and updating ranging from 000 to 299.

5.4 Boundary test

5.4.1 Log in interface (creating account)

|  |  |
| --- | --- |
| Input username | aaaaaaa |
| Output | Alert(“id must be a string of 6-16 characters which contains one letter and one digit at least”) |
| Input password | 123 |
| Output | Alert(“password must be a string contains 6 characters at least”) |

5.4.2 Current parking lot interface (adding cars)

|  |  |
| --- | --- |
| Input position | 32432 |
| Output | Alert(“position must be set between 000 to 299”) |

5.4.3 The owner table (updating and adding)

|  |  |
| --- | --- |
| Input owner ID | 3ssssss |
| Output | Alert(“owner ID must be a 0-8 bits digit”) |

5.4.4 General updating and adding

|  |  |
| --- | --- |
| Input | null or ‘’ |
| Output | Alert (“cannot be empty!”) |

6. Evaluation

6.1 Further outlook

Further works to be carried out are focused on improving our functionality and security.

1. In light of the system is based on the internet and database, therefore some of the protections like SQL injection mechanism need to be carried out in further works.

2. More sorting functions could be implemented next time, such as sorting by the parking time.

3. The log-function is required for an integral project. Therefore, a log utils (Aspect Orient Programming) could be carried out to log necessary information. For example, if users fail to connect the database or somebody tries to open the website with several illegal actions, actions will be logged based on which the developers can improve specific functionalities.

4. The structure of the system could be more decoupling in the future, such as dividing the works from each controller more specifically. The more decoupling the structure of the system is the easier for developers to improve functionality.

6.2 Responsibility

Qiu mengke is responsible for:

1. The construction of the web app.

2. Design and realize the structure of the software.

3. Realize the Model, Controller and View (MVC model) in our project such as our main page, the history page, owner table page, and all the controllers between each page.

4. Design and construct the database.

5. Part of the report.

Xiao zijian is responsible for:

1. CSS files.

2. The login function.

3. The alert window of inputting check.

4. Boundary test.

5. Part of the report.

Qiu sixiang is responsible for:

1. Tests part and problems fixing.

2. Most of the report, information gathering and summary.