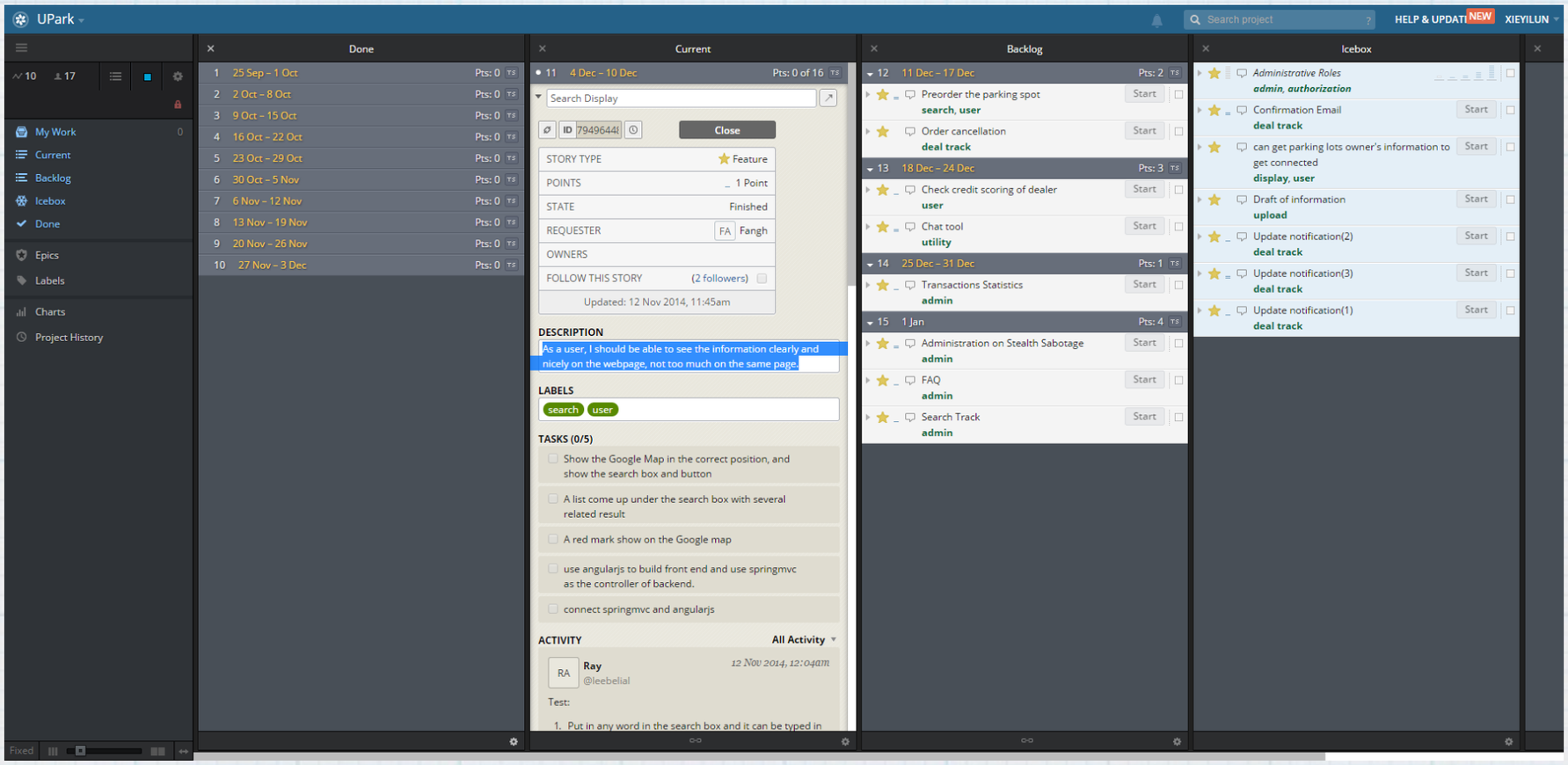
U-PARK.INFO

Find your perfect parking space

* **Customer:** Aim to people who have spare parking places or need to find a parking space in advance.
* **Purpose:** UPark is a information provider for users who want to rent their idle parking spaces to earn some money or find available parking places around specific area in advance.

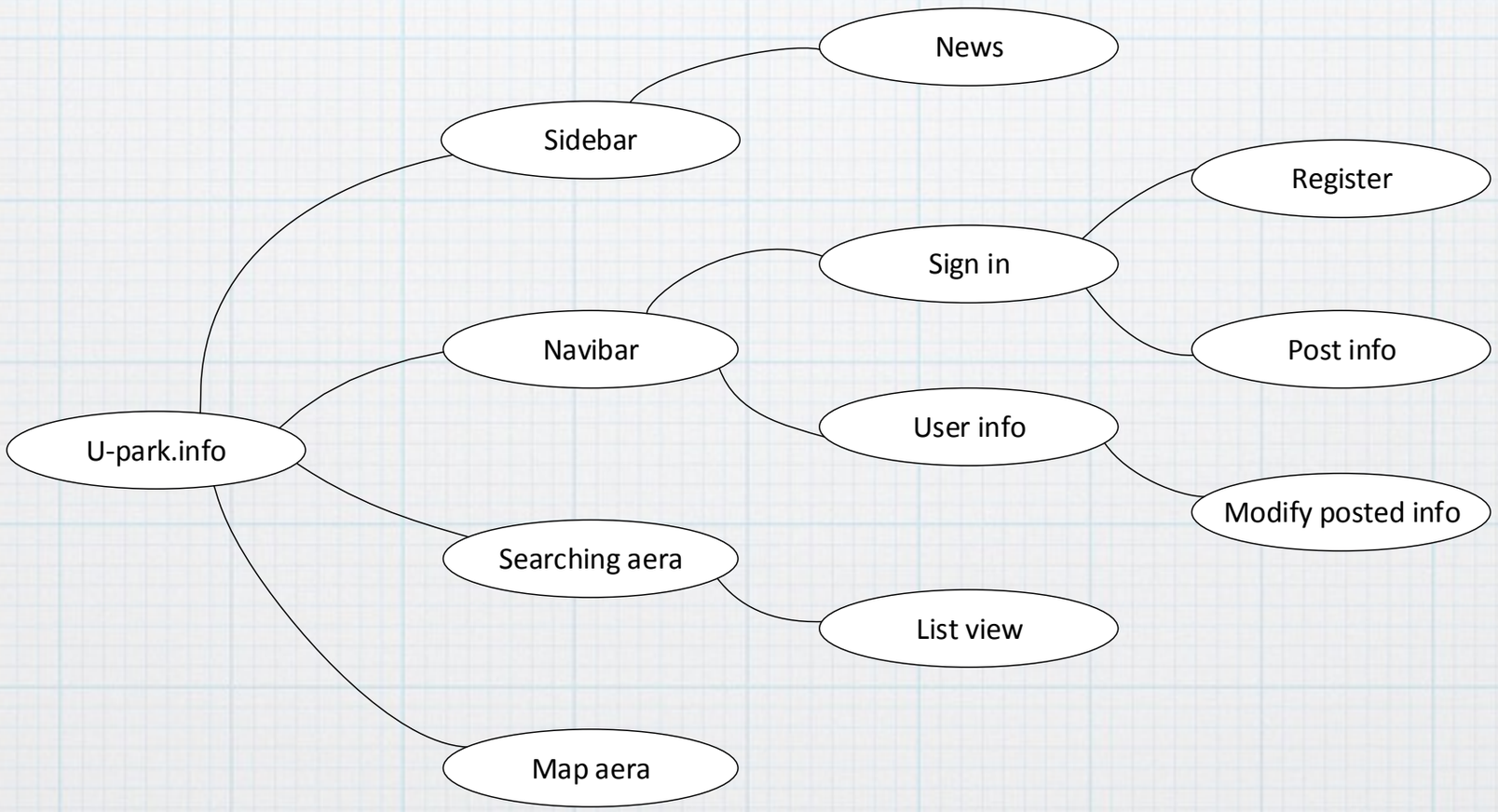
Requirement Control

* We use pivotal track to track and handle the change of requirement.



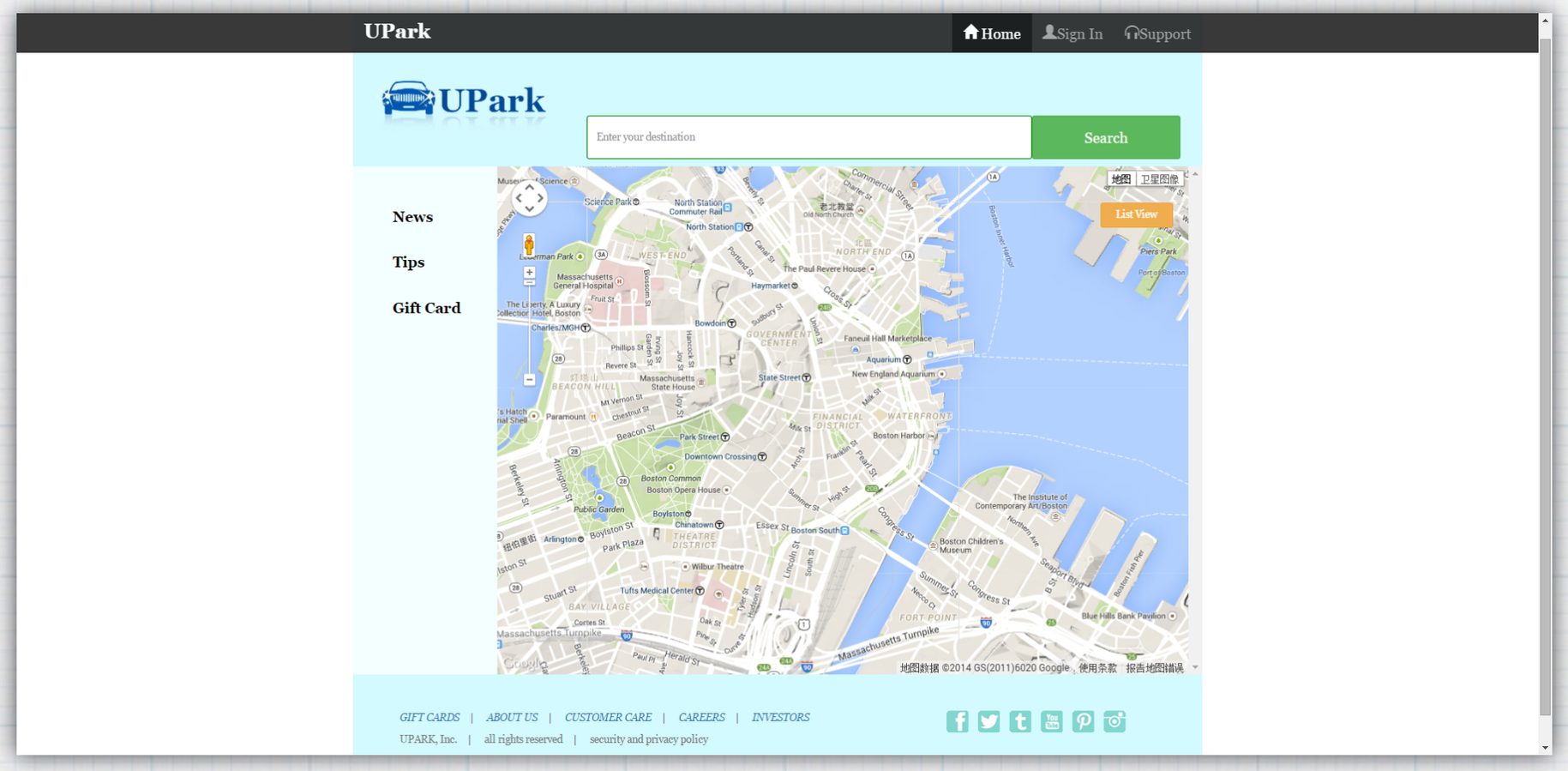
Functional Requirement

* UI flow diagram

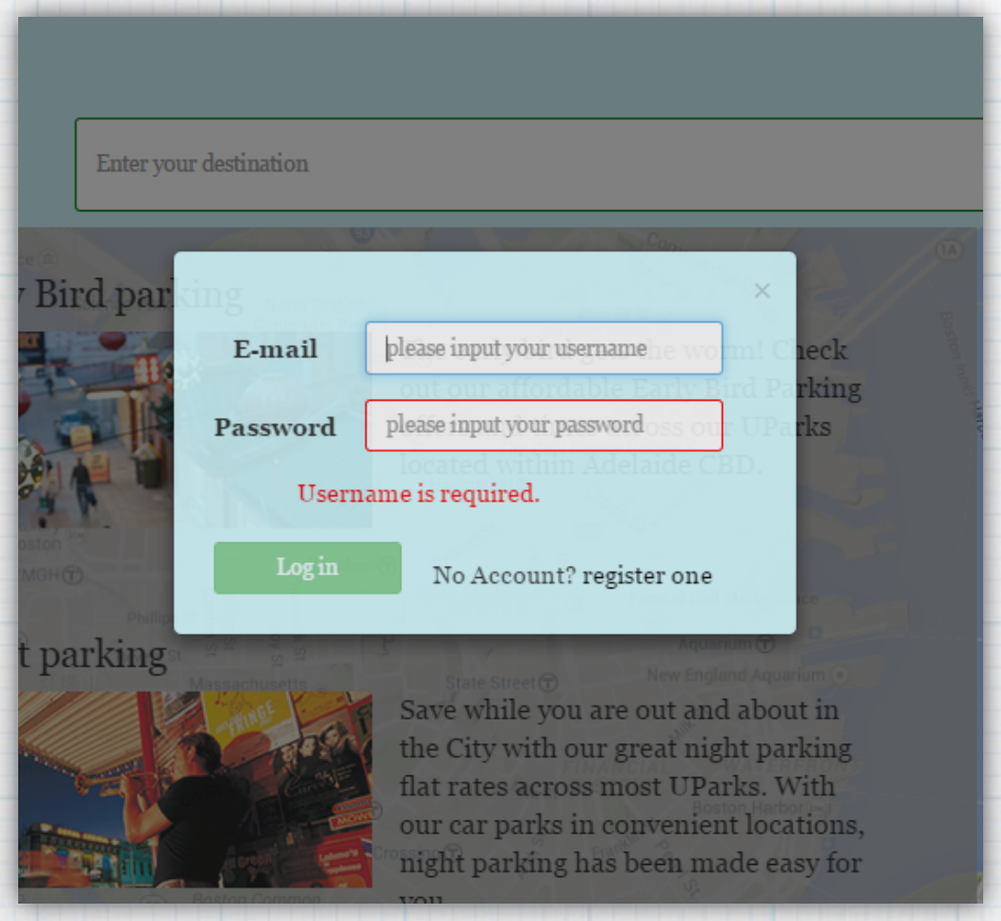


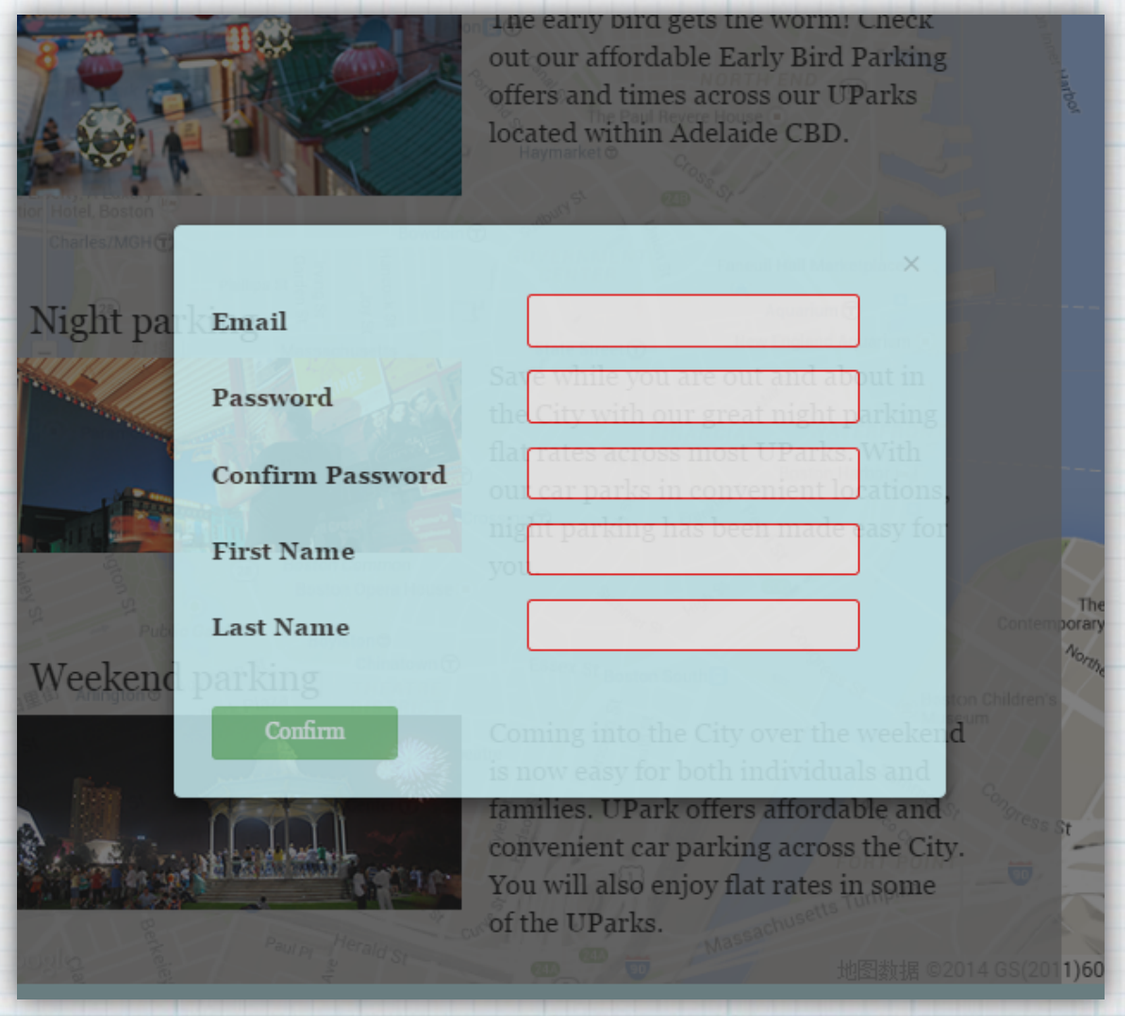
Functional Requirement

* UI Design

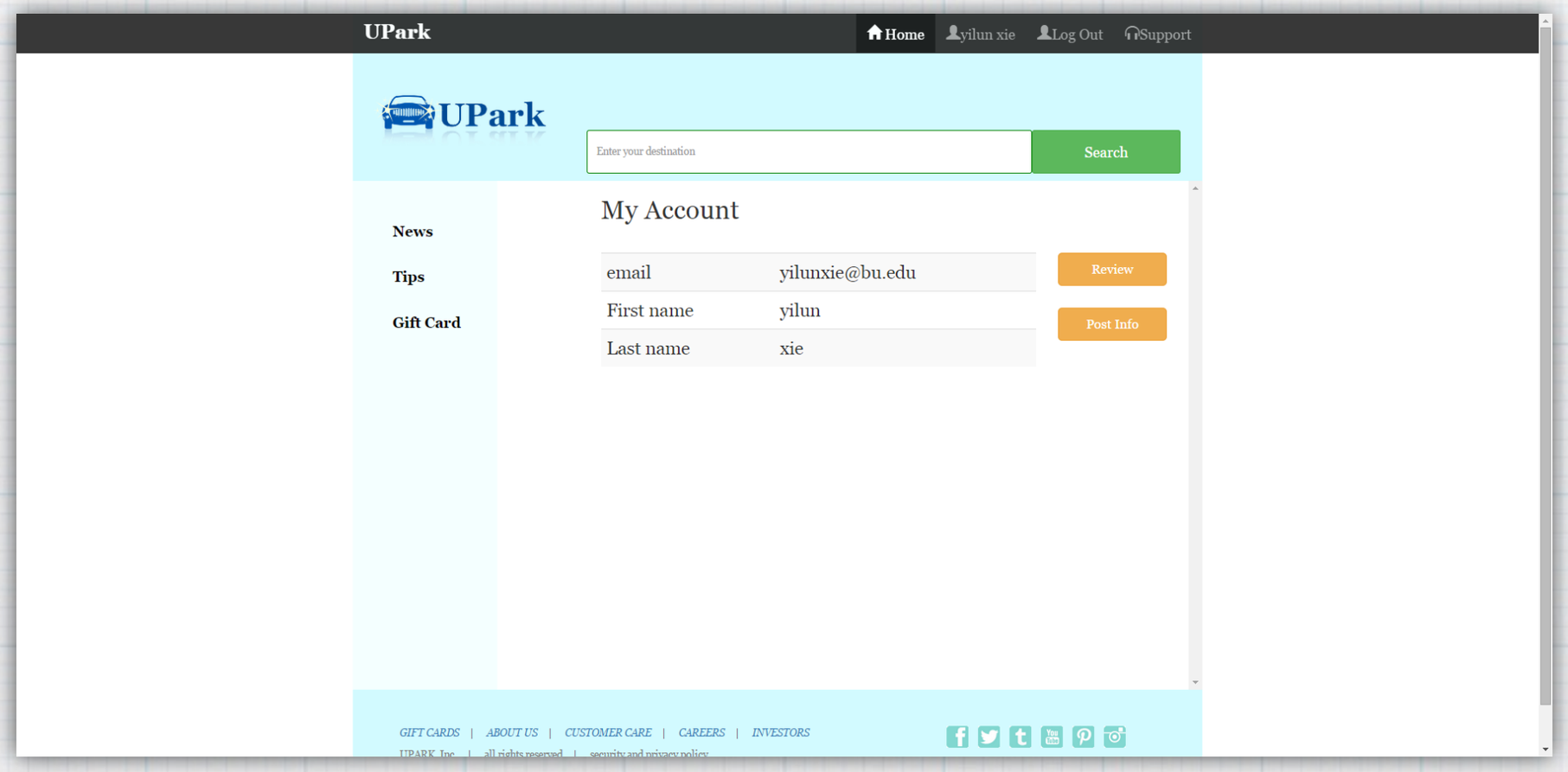


Functional Requirement

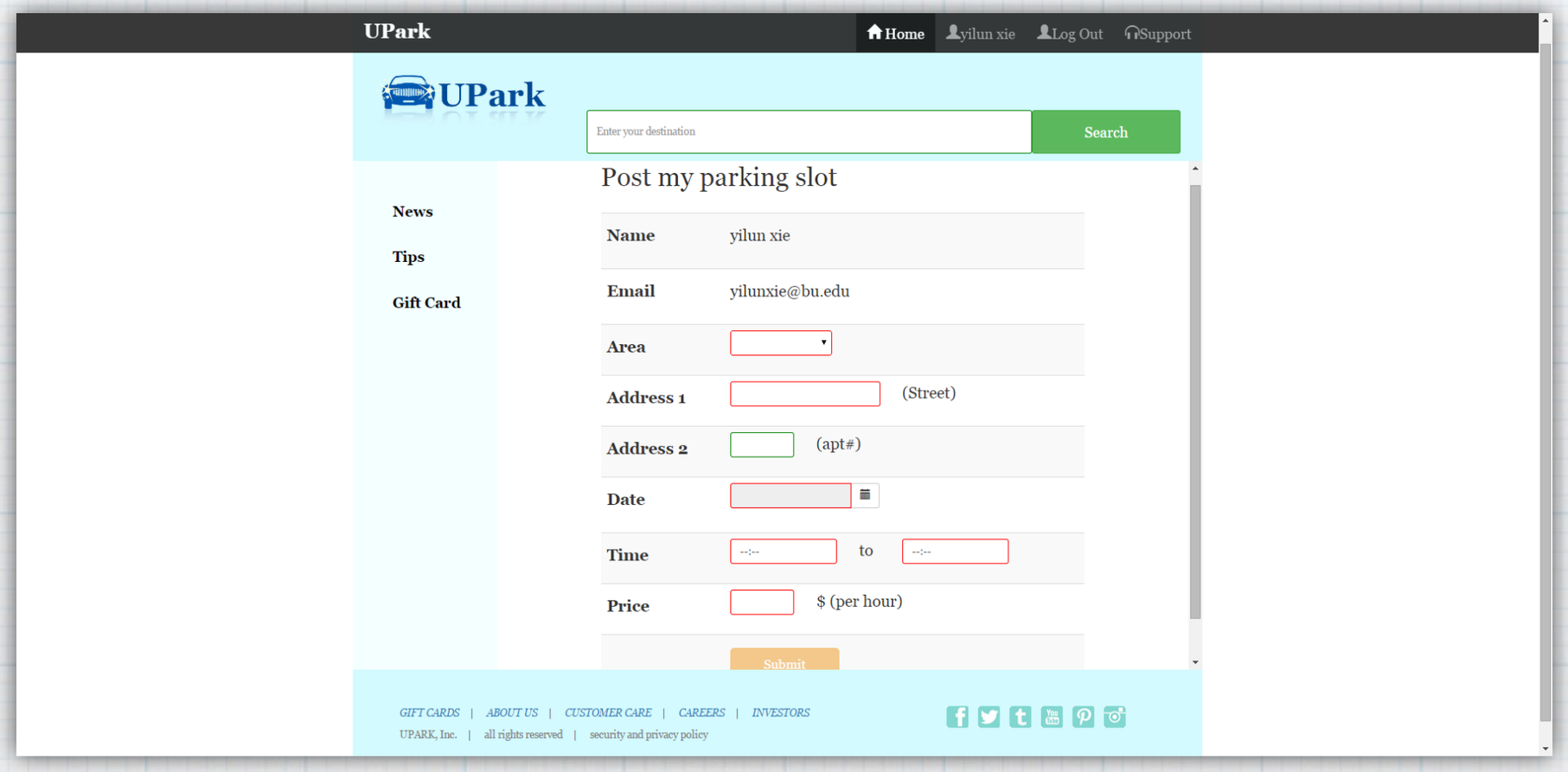




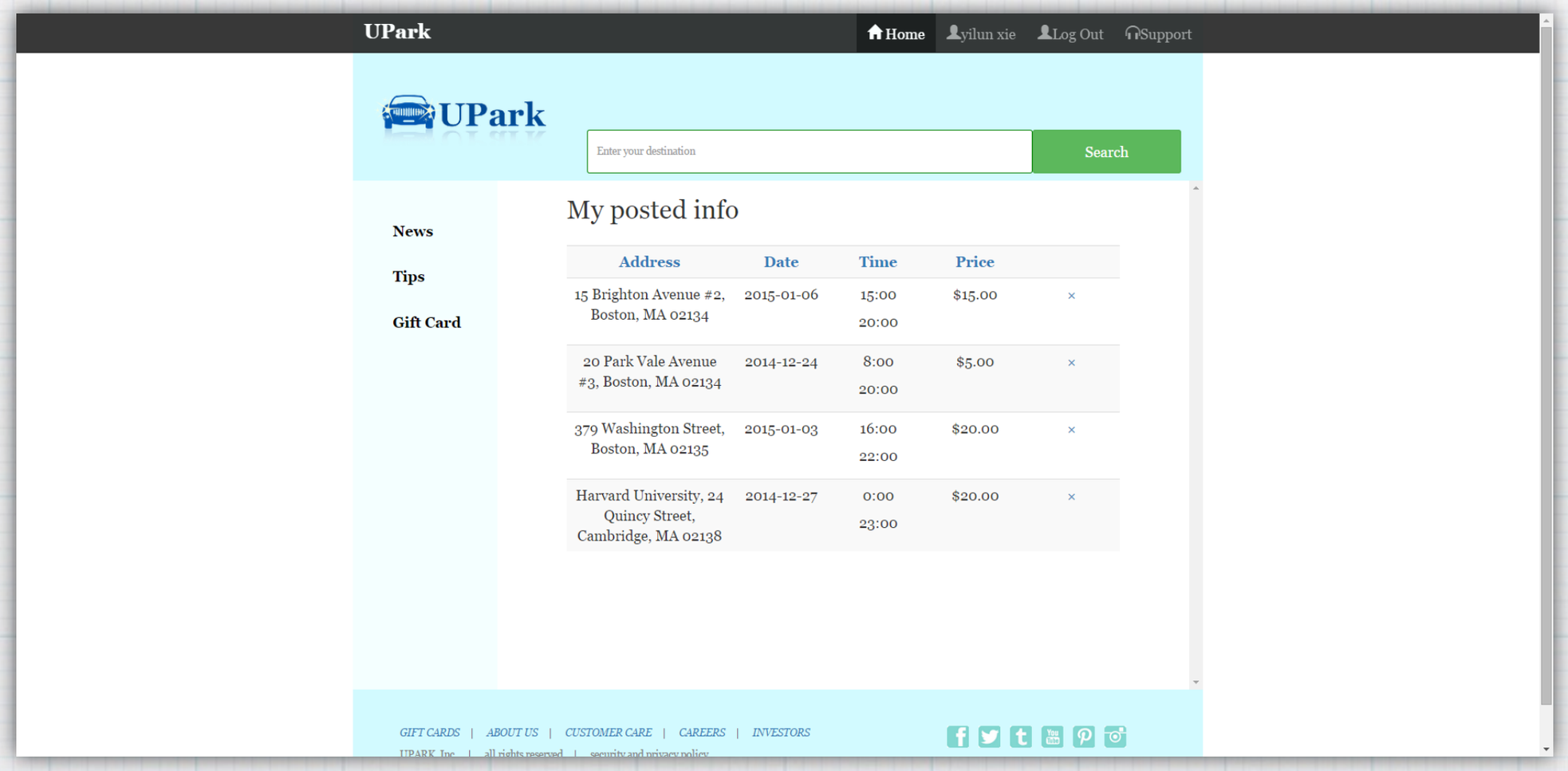
Functional Requirement



Functional Requirement



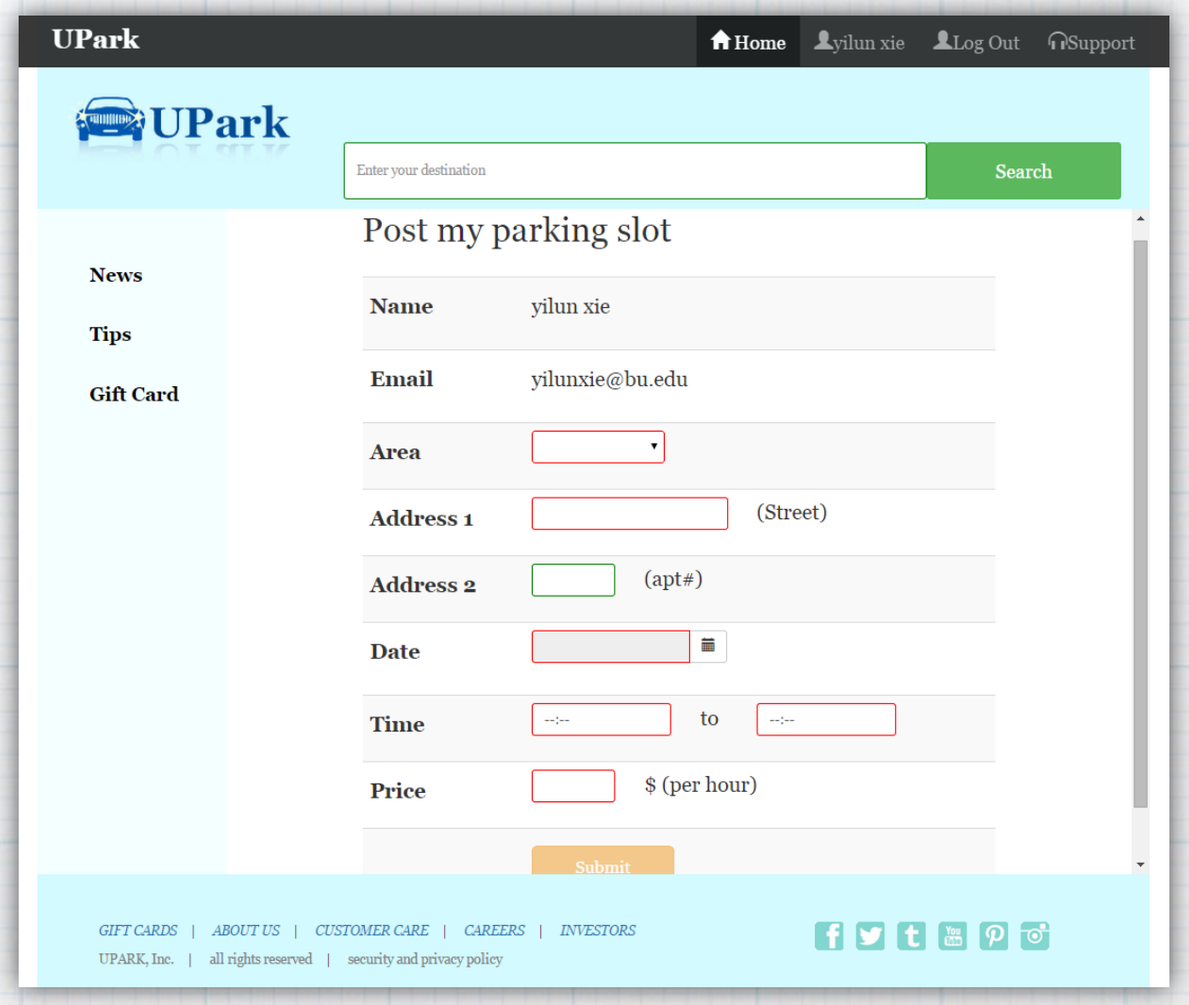
Functional Requirement



Functional Requirement

* User Story - Rental Information

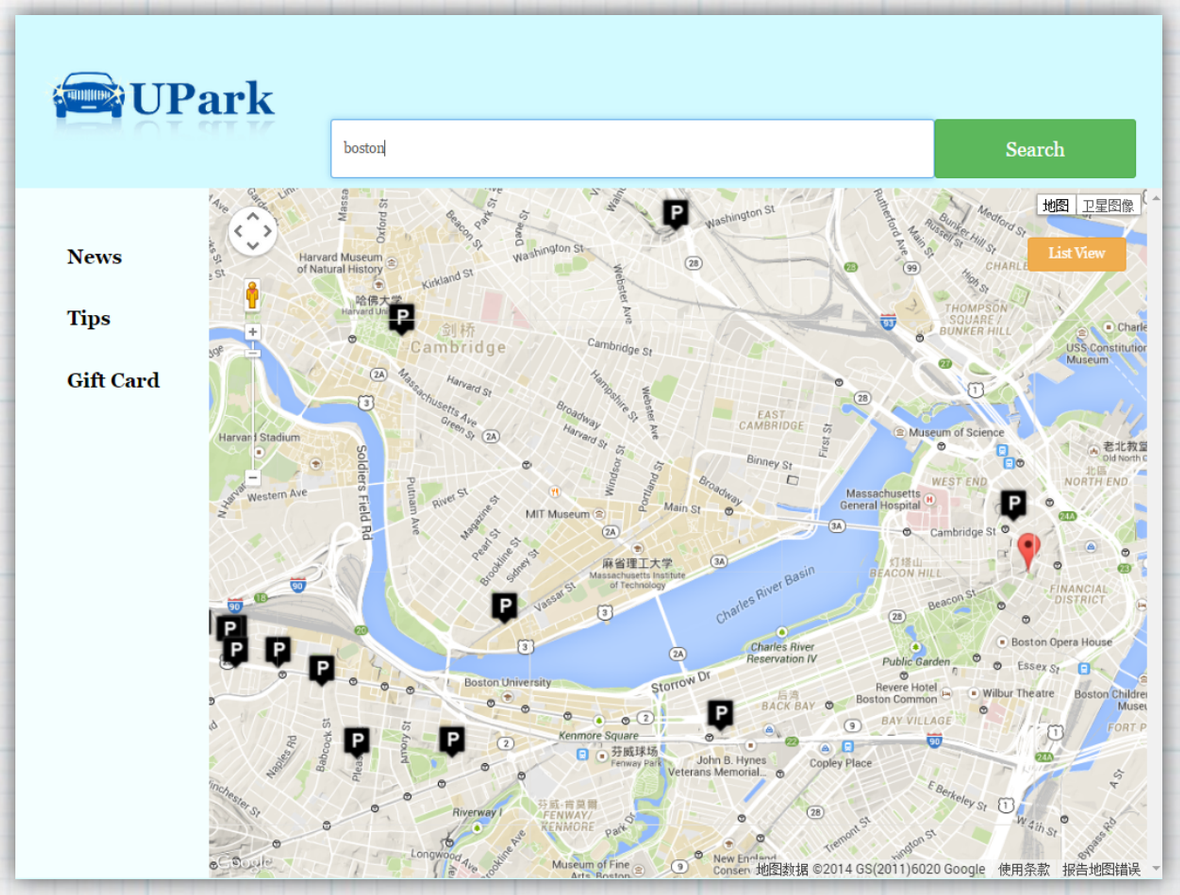
* As a renter, I should post the information of the price, available rental time, location, contact so that the tenant can get his wanted parking information.



Functional Requirement

* User Story - Result display

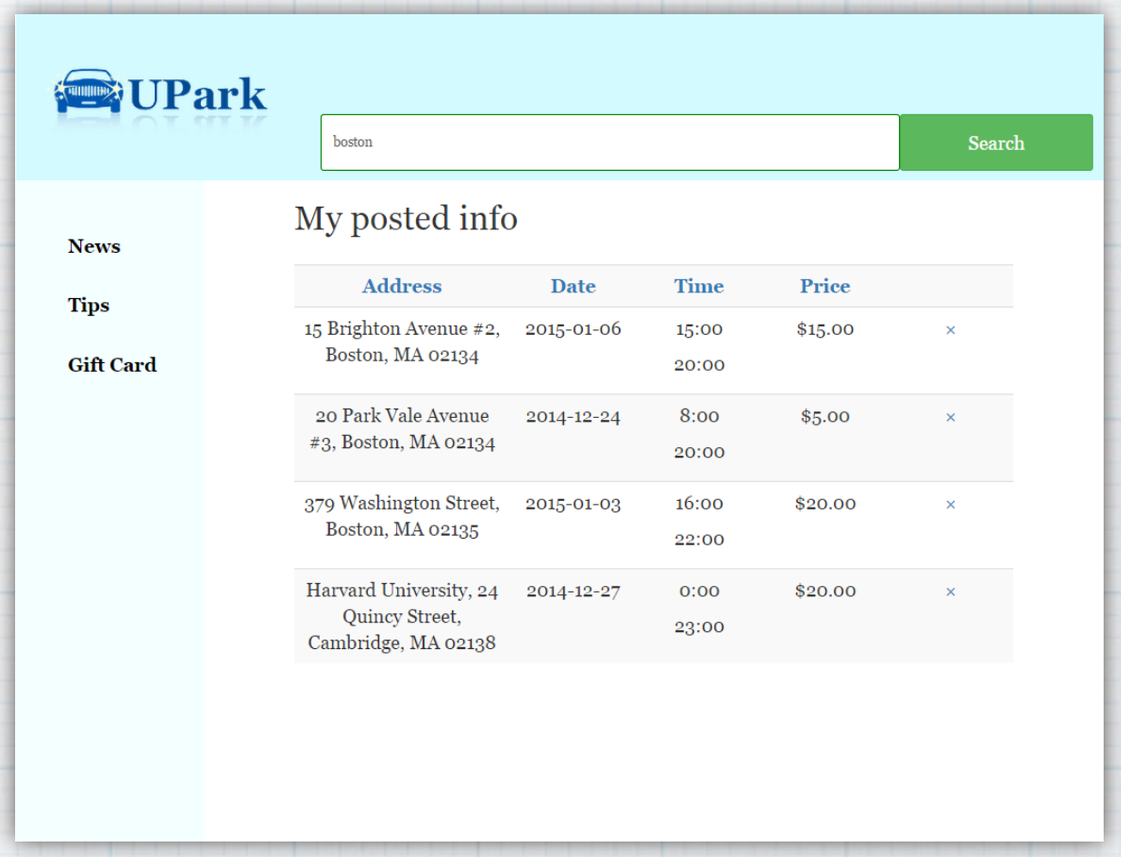
* As a user, I should be able to see the searching results on Google Map so it could be easier for me to find the location.



Functional Requirement

* User Story – Information Review

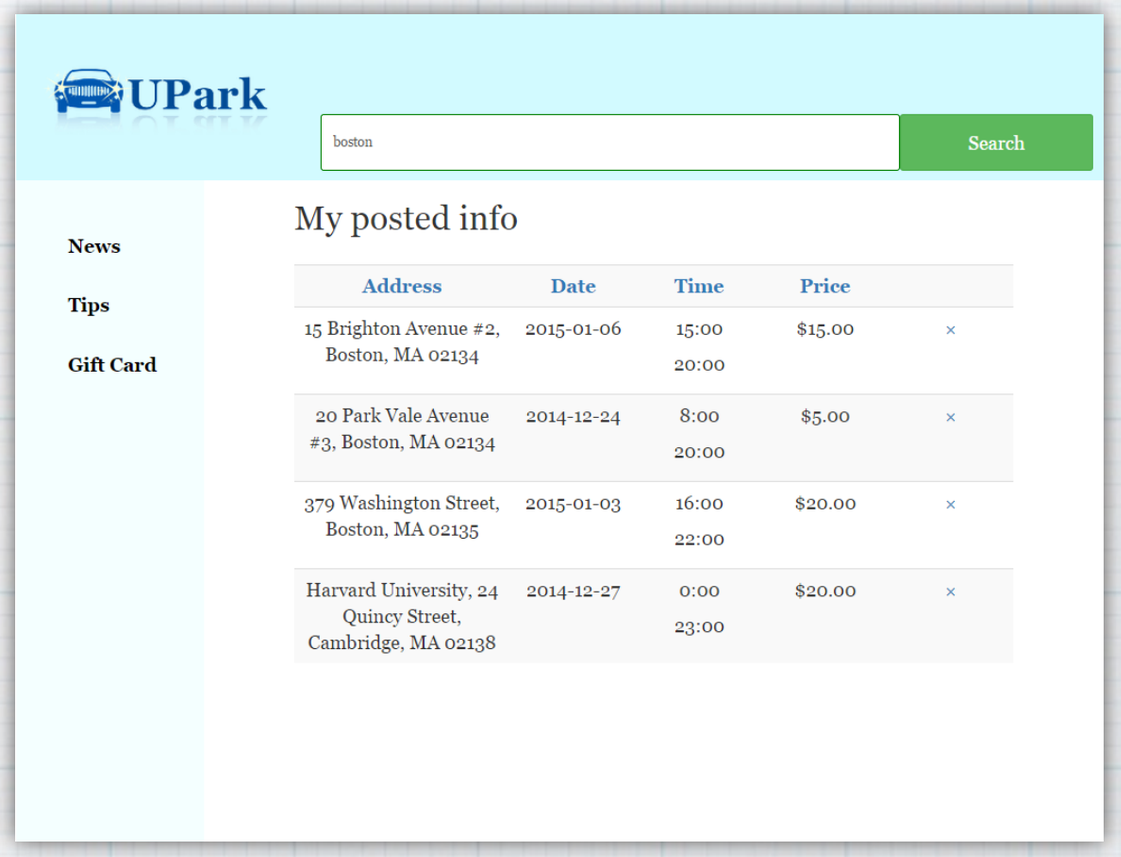
* As a renter, I'll be able to review all the information that I provided about my parking space so I can make some changes if I change my mind.



Functional Requirement

* User Story – Information Review

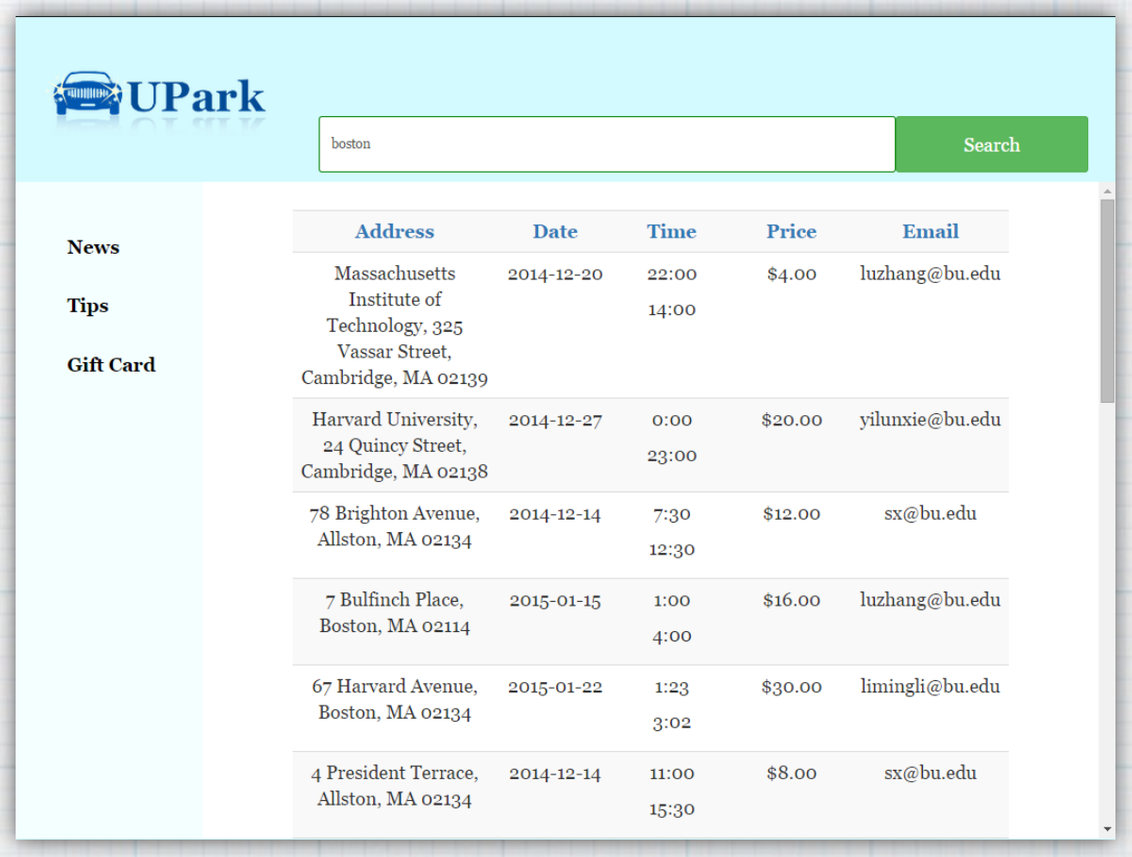
* As a renter, I'll be able to review all the information that I provided about my parking space so I can make some changes if I change my mind.



Functional Requirement

* User Story – Result display

* As a user, I should be able to order the result in different ways so I don't need to deal with too much unnecessary information.



Non-Functional Requirement

* UCompatible with Chrome, IE9+ and other common web browsers
* Need screen resolution higher than 1366*768
* Not compatible with mobile web browsers
* Security guaranteed by https protocol
* Java runtime environment needed

Functional Requirement

* Account Management
  * Sign in/Log out
  * Register
* Information Management
  * User information modification
  * Parking å information modification
* Search Service
  * Search with address
  * Show result in list with different order

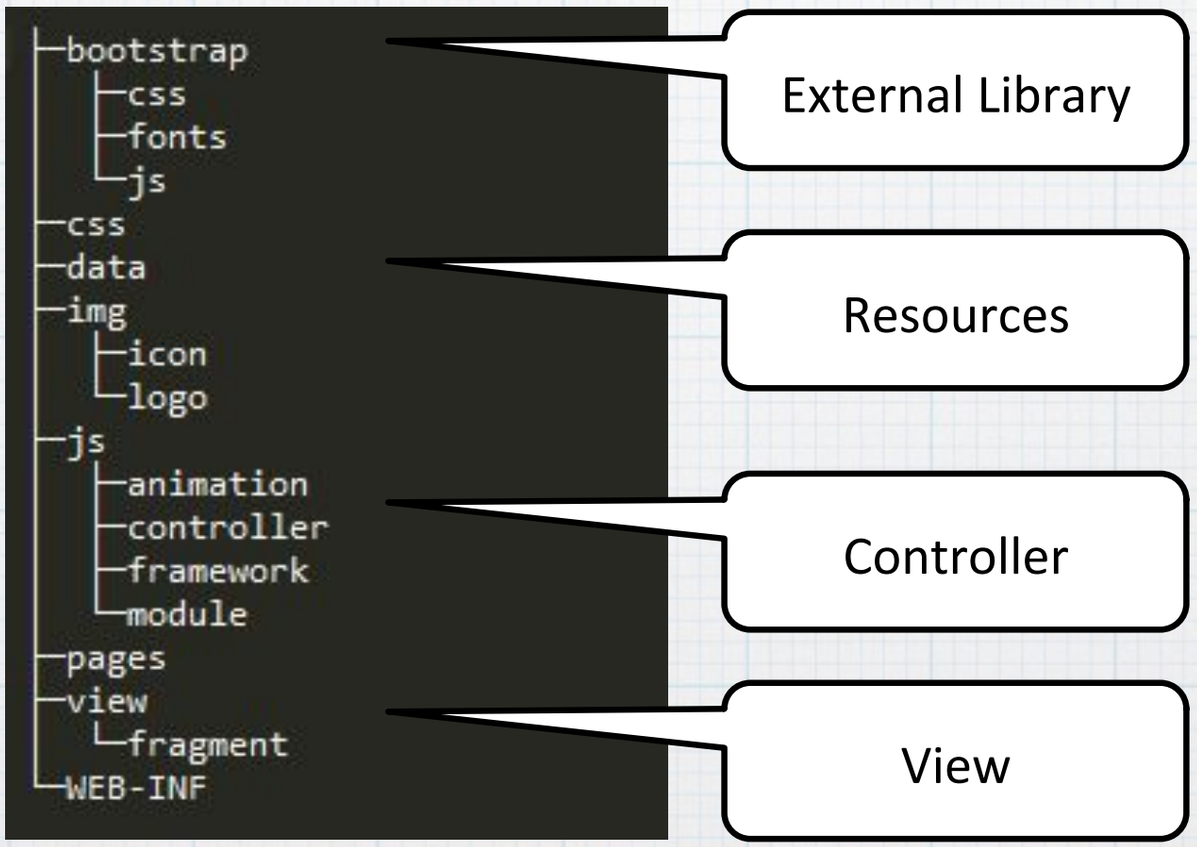
Techniques

AngularJS

Google Maps Javascript API V3

Directory Structure

Standard Structure

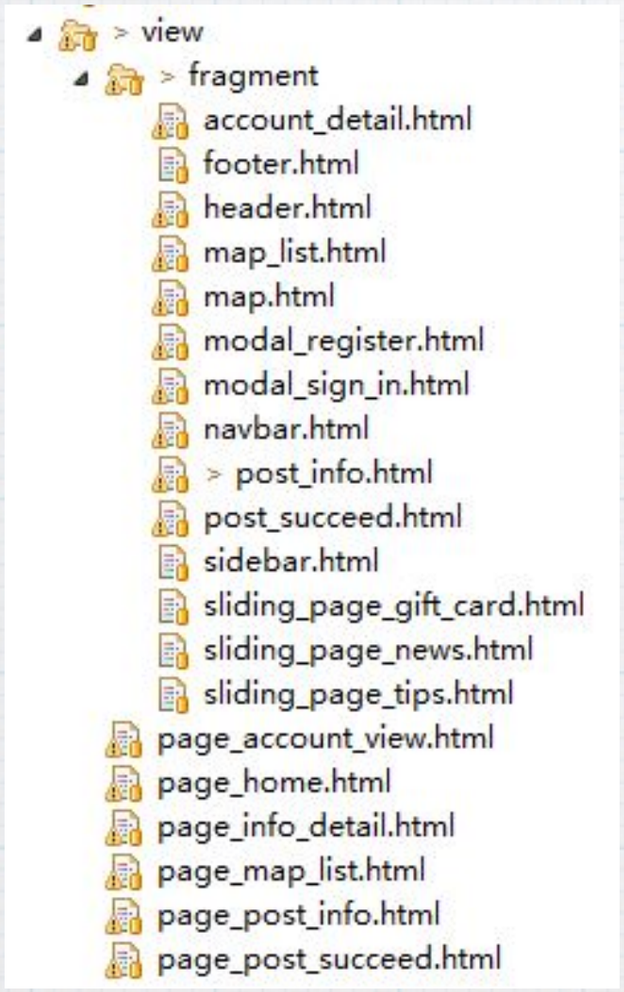


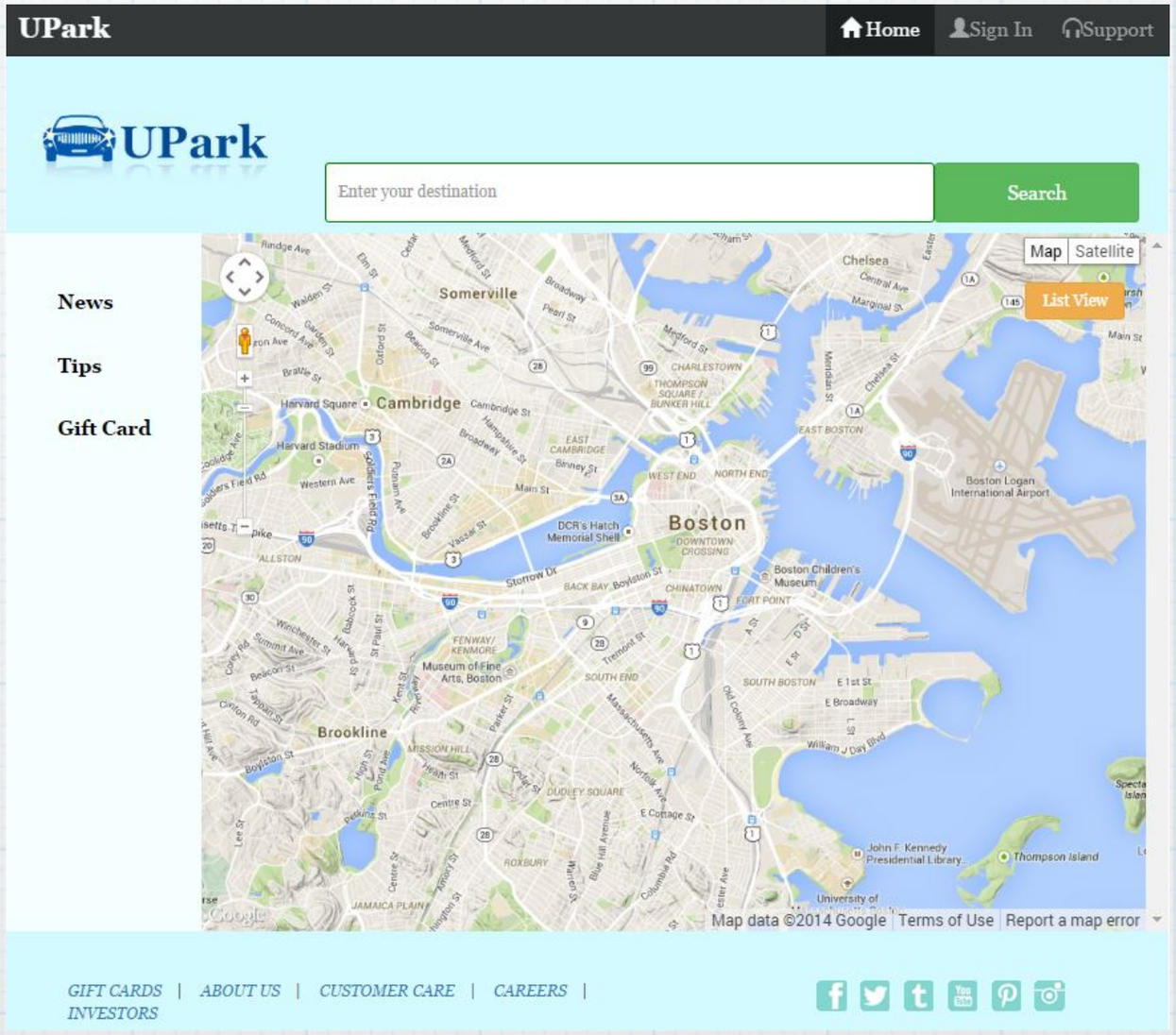
Project Design Pattern

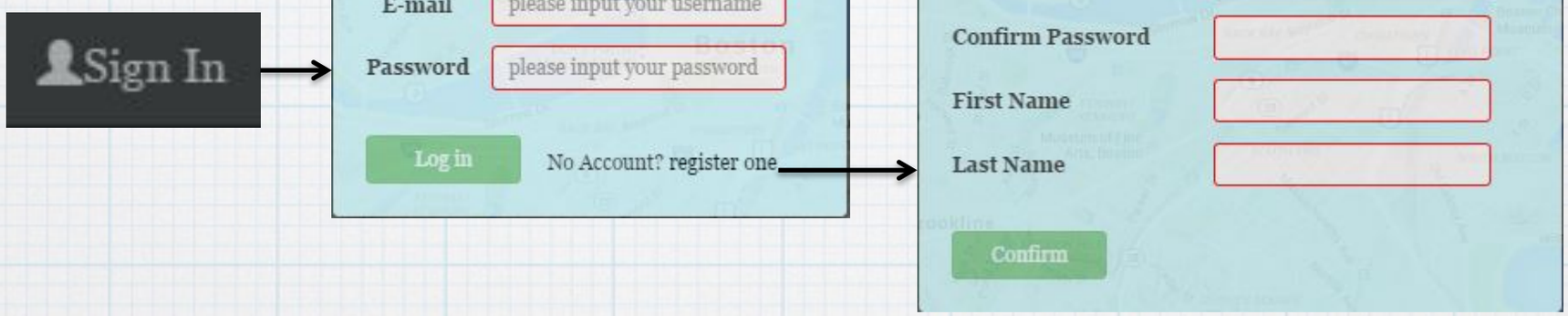
MVW = Model + View + Whatever

MVC or MVVM

View


> view
> fragment
account_detail.html
footer.html
header.html
map_list.html
map.html
modal_register.html
modal_sign_in.html
navbar.html
> post_info.html
post_succeed.html
sidebar.html
sliding_page_gift_card.html
sliding_page_news.html
sliding_page_tips.html
page_account_view.html
page_home.html
page_info_detail.html
page_map_list.html
page_post_info.html
page_post_succeed.html


UPark
Home
Sign In
Support
UPark
Enter your destination
Search
News
Tips
Gift Card
Map
Satellite
List View
Somerville
Cambridge
Boston
Brookline
Chelsea
Map data ©2014 Google
Terms of Use
Report a map error
GIFT CARDS | ABOUT US | CUSTOMER CARE | CAREERS | INVESTORS


Sign In
E-mail
please input your username
Password
please input your password
Log in
No Account? register one
Email
Password
Confirm Password
First Name
Last Name
Confirm

My Account

email	luzhang@bu.edu
First name	Lu
Last name	Zhang

Review

Post Info

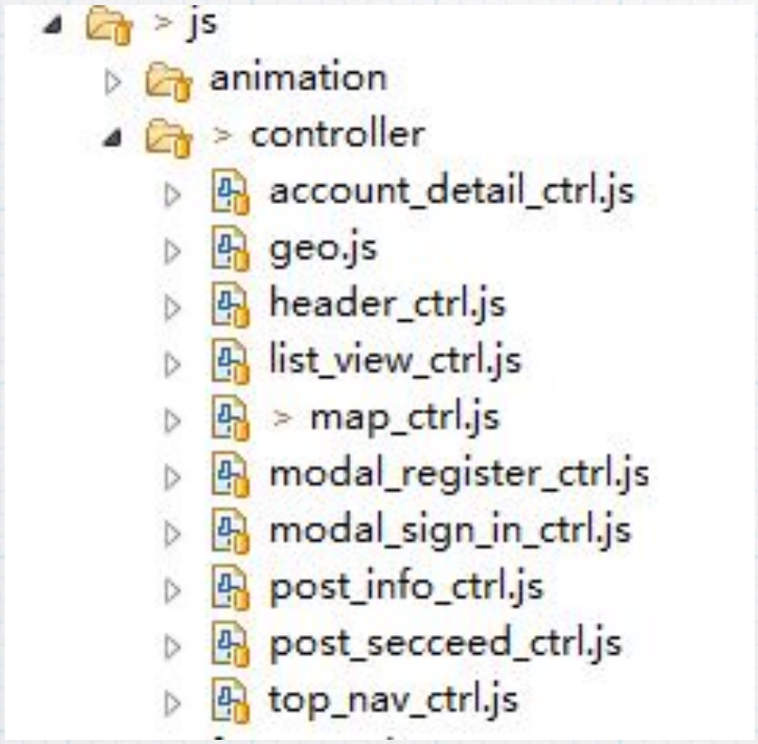
My posted info

Address	Date	Time	Price	
12 Boston Street, Somerville, MA 02143	2014-12-06	15:00 16:00	$5.00	×
10 Jamaicaway, Jamaica Plain, MA 02130	2014-12-07	12:00 15:00	$10.00	×
344 Newbury Street, Boston, MA 02115	2014-12-10	9:00 12:00	$12.00	×
Massachusetts Institute of Technology, 325 Vassar Street, Cambridge, MA 02139	2014-12-21	1:00 17:00	$4.00	×
7 Bulfinch Place, Boston, MA 02114	2015-01-16	4:00 7:00	$16.00	×

Post my parking slot

Name	Lu Zhang
Email	luzhang@bu.edu
Area	
Address 1	(Street)
Address 2	(apt#)
Date	
Time	--:-- to --:--
Price	$ (per hour)

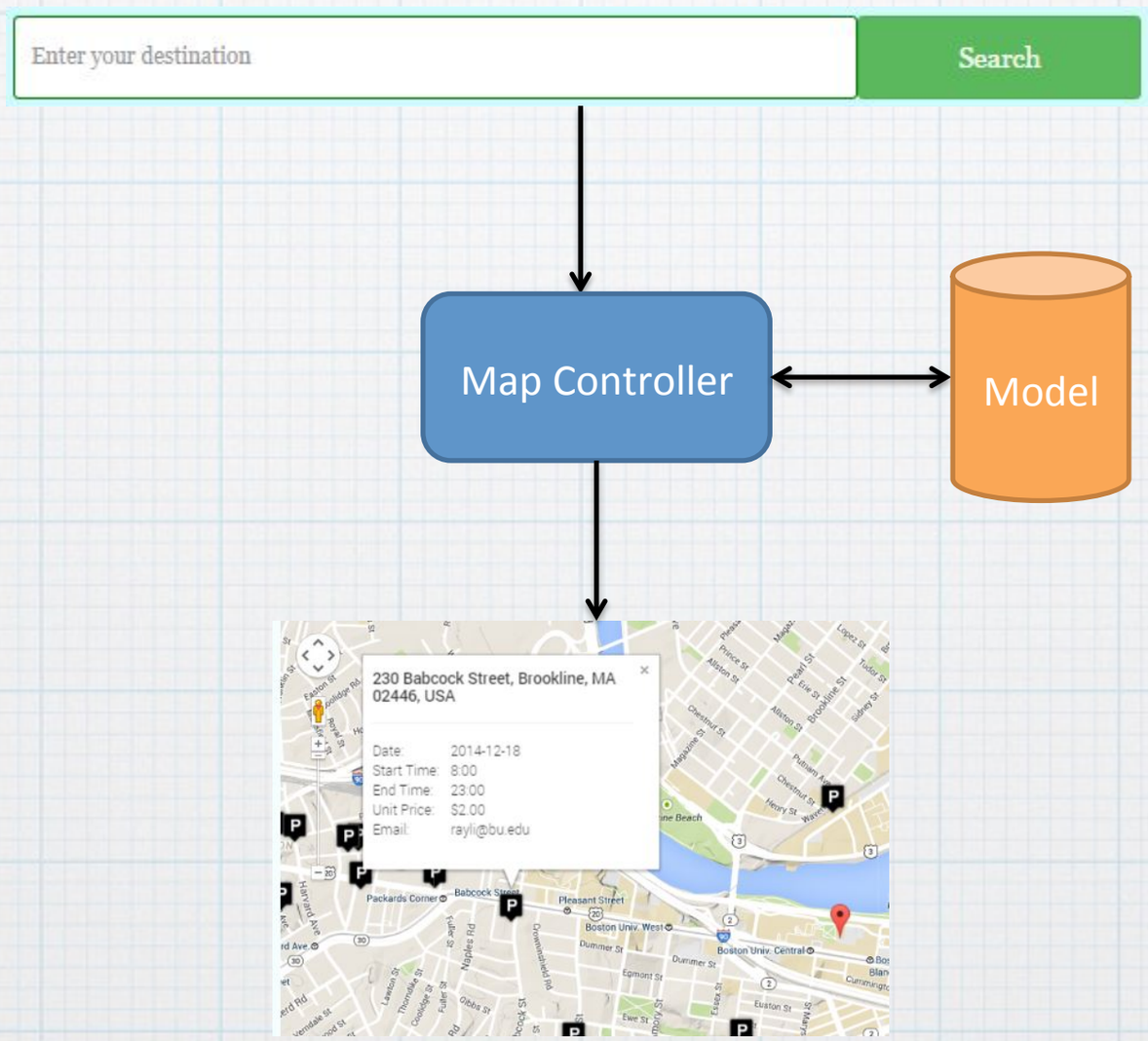
Controller



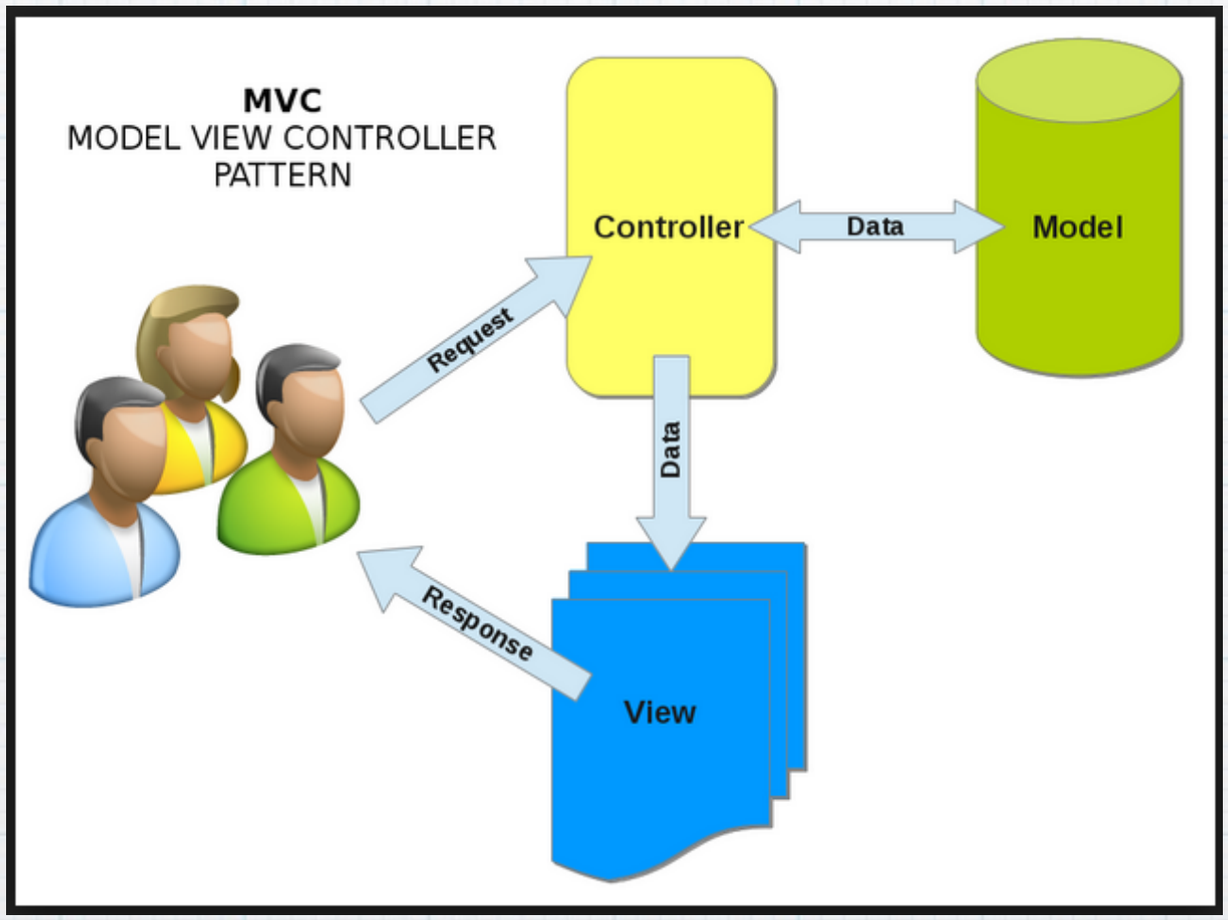
$Scope

$Resource

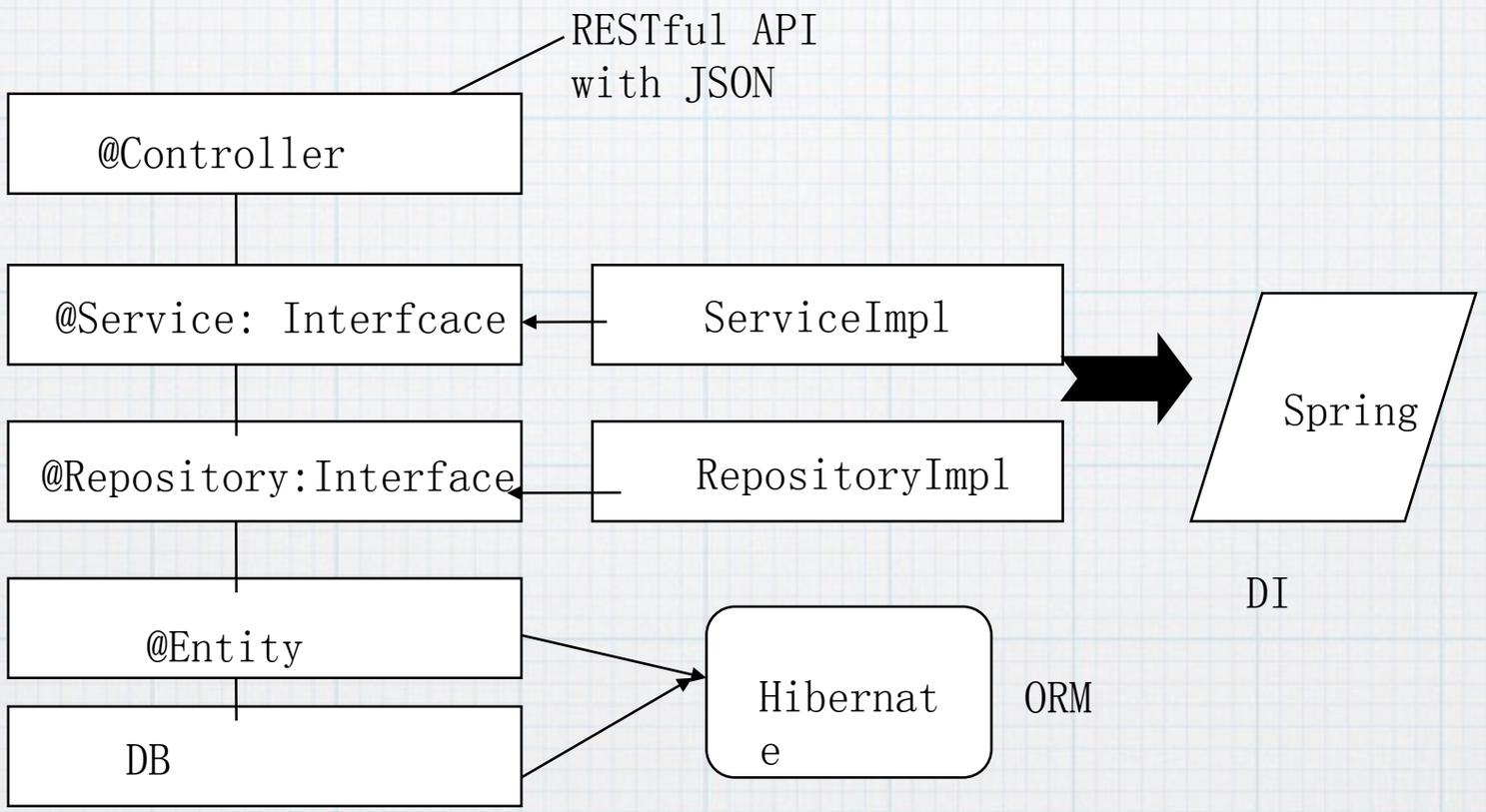
Map Controller



Spring MVC

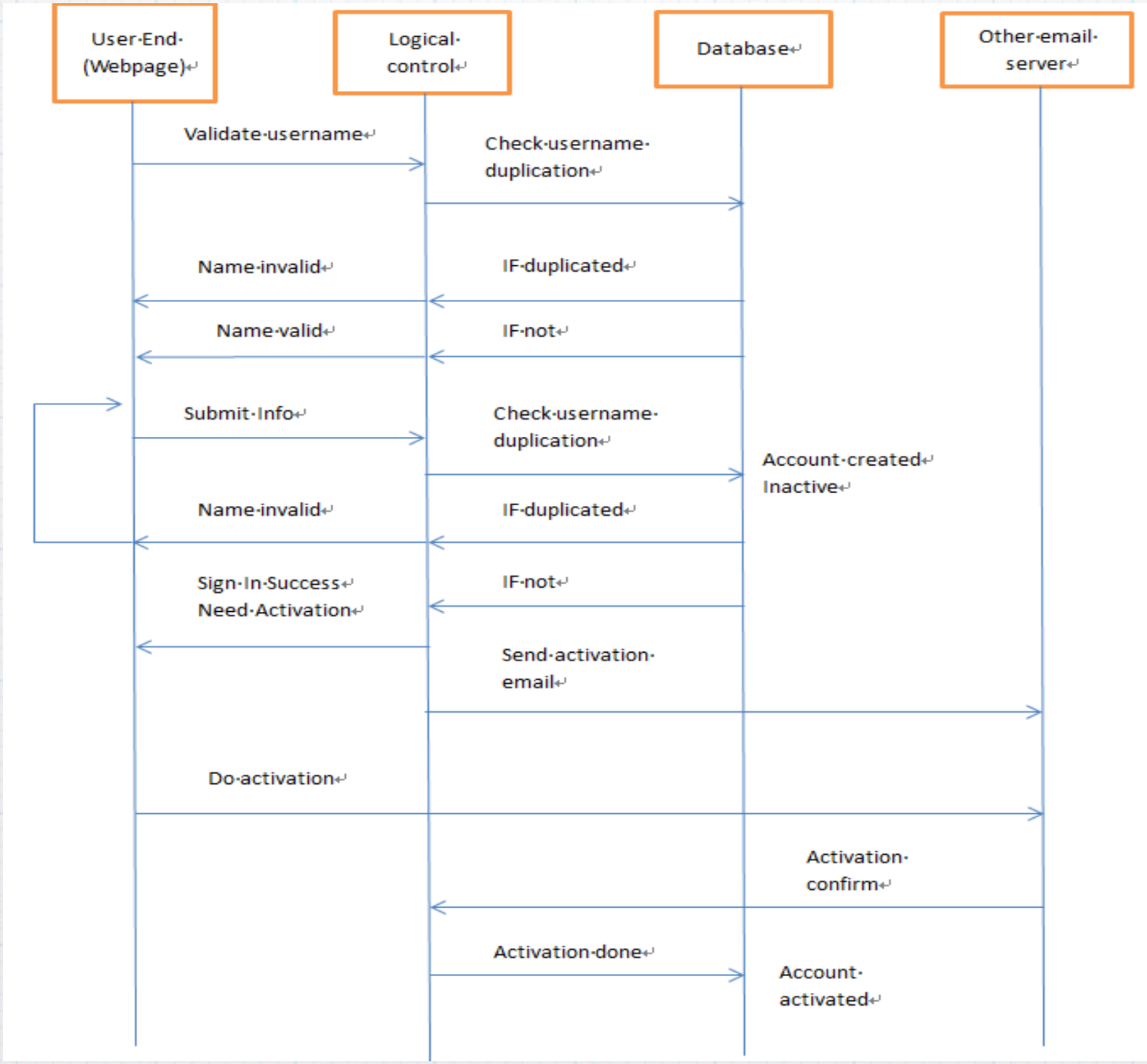

MVC
MODEL VIEW CONTROLLER
PATTERN
Controller
Data
Model
Request
Data
Response
View

Architecture

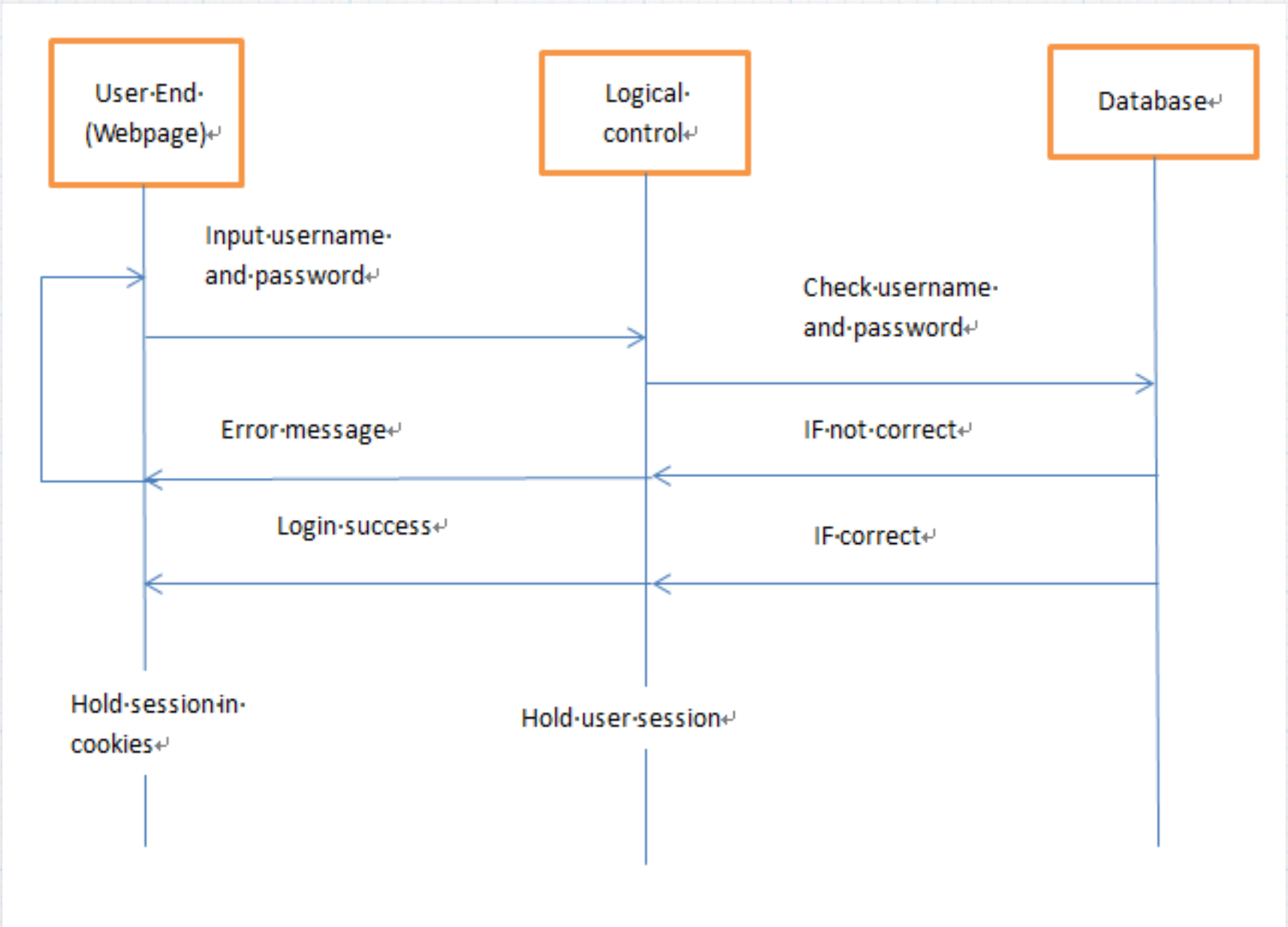


Main Function Sequence Diagram

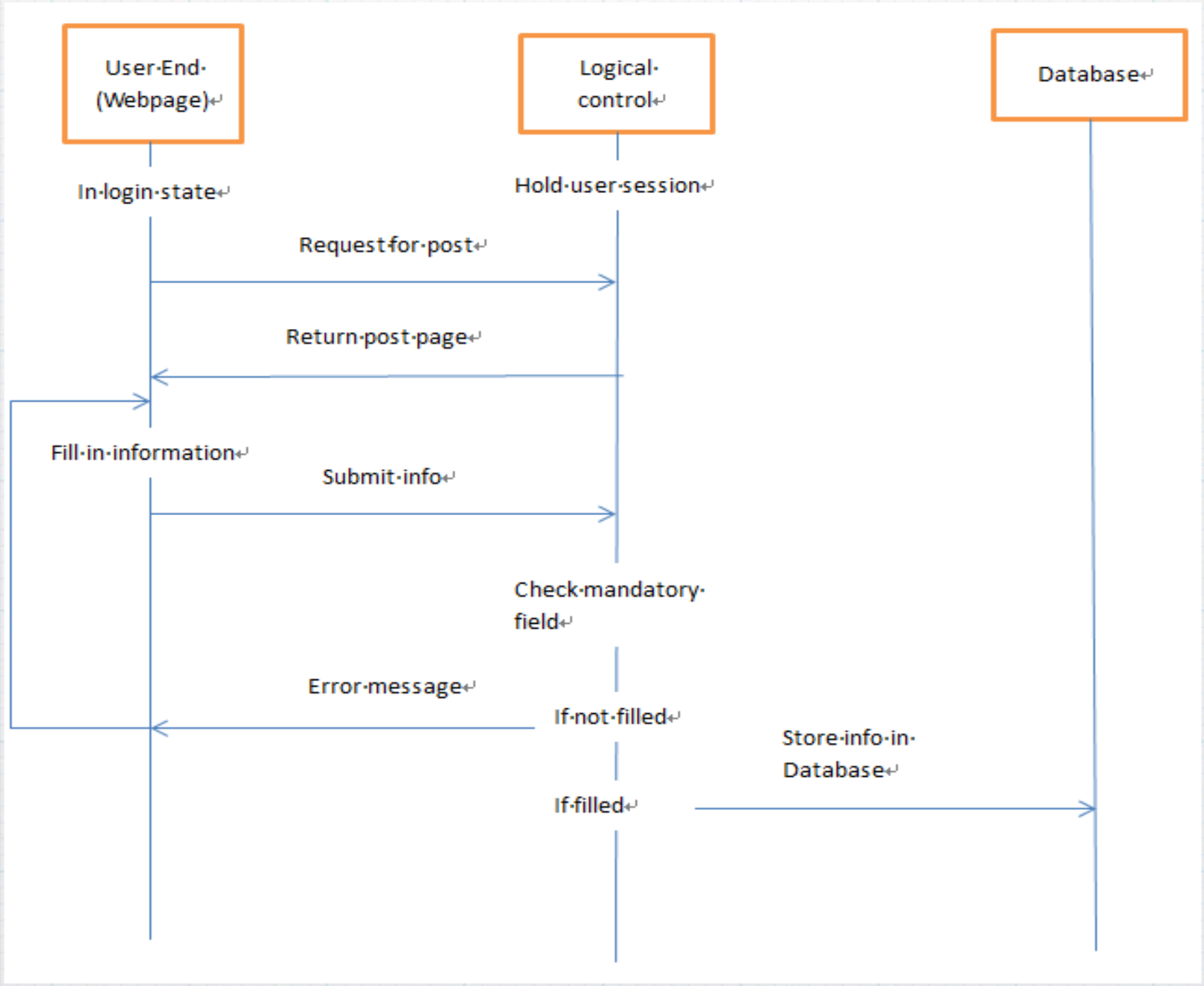
Register


User·End·(Webpage)
Logical·control
Database
Other·email·server
Validate·username
Check·username·duplication
Name·invalid
IF·duplicated
Name·valid
IF·not
Submit·Info
Check·username·duplication
Account·created
Inactive
Name·invalid
IF·duplicated
Sign·In·Success
Need·Activation
IF·not
Send·activation·email
Do·activation
Activation·confirm
Activation·done
Account·activated

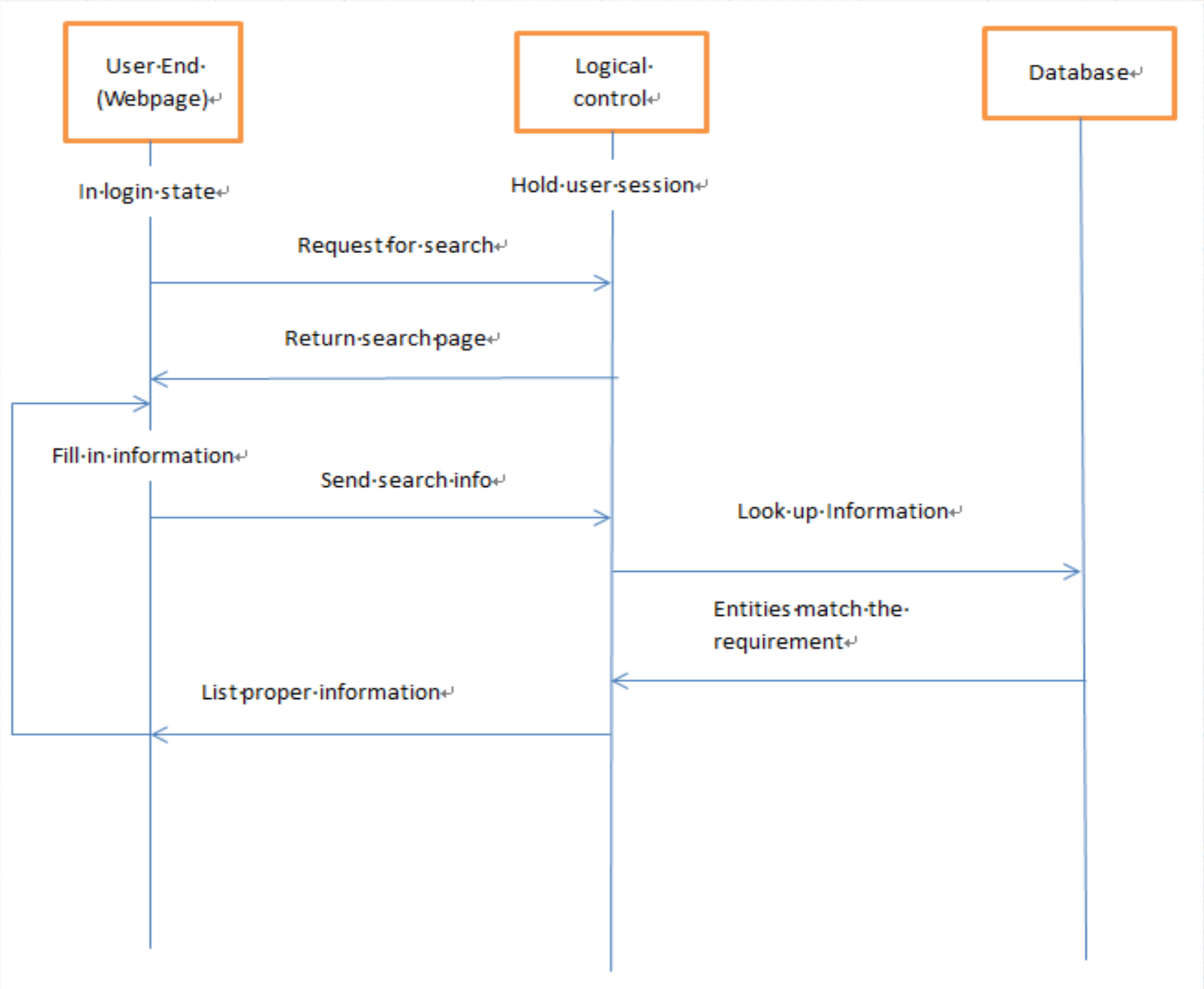
Login


User·End·(Webpage)
Logical·control
Database
Input·username·and·password
Check·username·and·password
Error·message
IF·not·correct
Login·success
IF·correct
Hold·session·in·cookies
Hold·user·session

Post


User·End·(Webpage)
Logical·control
Database
In·login·state
Hold·user·session
Request·for·post
Return·post·page
Fill·in·information
Submit·info
Check·mandatory·field
Error·message
If·not·filled
Store·info·in·Database
If·filled

Search


User·End·(Webpage)
Logical·control
Database
In·login·state
Hold·user·session
Request·for·search
Return·search·page
Fill·in·information
Send·search·info
Look·up·Information
Entities·match·the·requirement
List·proper·information

UPark Schema
Created By: Liming Li, Feiyu
Created On: 10/3/2014

CUSTOMER_DATA

USER
- ID VARCHAR(32)
- FIRST_NAME VARCHAR(45)
- LAST_NAME VARCHAR(45)
- SEX VARCHAR(15)
- BIRTHDAY DATE
- ACCOUNT_ID VARCHAR(32)
- CREATE_DATE TIMESTAMP
- LAST_UPDATE TIMESTAMP
- Indexes

VEHICLE
- ID VARCHAR(32)
- YEAR YEAR
- MAKE VARCHAR(20)
- MODEL VARCHAR(20)
- COLOR VARCHAR(20)
- DEL_FLAG TINYINT(1)
- ACCOUNT_ID VARCHAR(32)
- CREATE_DATE TIMESTAMP
- LAST_UPDATE TIMESTAMP
- Indexes

ACCOUNT
- ID VARCHAR(32)
- NAME VARCHAR(45)
- EMAIL VARCHAR(45)
- PASSWORD VARCHAR(60)
- PHONE VARCHAR(45)
- ADDRESS_ID VARCHAR(32)
- DEL_FLAG TINYINT(1)
- BLOCK_FLAG TINYINT(1)
- PASSWORD_FAILED_TIMES TINYINT(3)
- LOCK_FLAG TINYINT(1)
- CREATE_DATE TIMESTAMP
- LAST_UPDATE TIMESTAMP
- Indexes

PAYMENT_METHOD
- ID VARCHAR(32)
- ACCOUNT_ID VARCHAR(32)
- DEL_FLAG TINYINT(1)
- CARD_HOLDER_NAME VARCHAR(60)
- CARD_NUMBER VARCHAR(45)
- CARD_EXPIRE_DATE DATE
- CREATE_DATE TIMESTAMP
- LAST_UPDATE TIMESTAMP
- Indexes

ADDRESS
- ID VARCHAR(32)
- STREET VARCHAR(45)
- STREET2 VARCHAR(45)
- CITY VARCHAR(45)
- STATE VARCHAR(45)
- COUNTRY VARCHAR(45)
- ZIP VARCHAR(20)
- DEL_FLAG TINYINT(1)
- CREATE_DATE TIMESTAMP
- LAST_UPDATE TIMESTAMP
- Indexes

BUSINESS

RENTAL_HISTORY
- ID VARCHAR(32)
- OWNER_ID VARCHAR(32)
- RENTER_ID VARCHAR(32)
- PURCHASE_ORDER_ID VARCHAR(32)
- COMMENT_FOR_OWNER TEXT(1000)
- COMMENT_FOR_RENTER TEXT(1000)
- RATING_FOR_OWNER FLOAT
- RATING_FOR_RENTER FLOAT
- CREATE_DATE TIMESTAMP
- DEL_FLAG TINYINT(1)
- Indexes

PARKING_LOT
- ID VARCHAR(32)
- NAME VARCHAR(45)
- ACCOUNT_ID VARCHAR(32)
- ADDRESS_ID VARCHAR(32)
- AVAILABILITY TINYINT(1)
- AVAILABLE_FROM DATETIME
- AVAILABLE_UNTIL DATETIME
- PRICE DECIMAL(16,2)
- POST_IT TINYINT(1)
- PHOTO_URL VARCHAR(256)
- DESCRIPTION TEXT(1000)
- DEL_FLAG TINYINT(1)
- Indexes

PURCHASE_ORDER
- ID VARCHAR(32)
- ORDER_ID VARCHAR(45)
- CONFIRMATION_FROM_OWNER TINYINT(1)
- TOTAL DECIMAL(16,2)
- STATUS VARCHAR(15)
- PAYMENT_METHOD_ID VARCHAR(32)
- Indexes

PURCHASE_ORDER_ITEMS
- ID VARCHAR(32)
- ITEM_LINE_NUMBER SMALLINT
- PURCHASE_ORDER_ID VARCHAR(32)
- PARKING_SPOT_ID VARCHAR(32)
- Indexes

Quality Assurance

* Metrics
* Standard
* Inspection / review
* Testing
* Defects management
* QA Team: Feiyu Shi, Rui Li

Metrics

* MProduct metrics
    * LOC: 14827
    * #defects per KLOC: 0.27
* Process Metrics
    * Project effort: 785 man-hours
    * Productivity: around 80 man-hours per main functionality
* More on next page…

Metrics

Metric	Value
Abstractness	18.7%
Average Block Depth	0.84
Average Cyclomatic Complexity	1.41
Average Lines Of Code Per Method	7.95
Average Number of Constructors Per Type	0.00
Average Number of Fields Per Type	2.03
Average Number of Methods Per Type	4.25
Average Number of Parameters	0.62
Comments Ratio	1.8%
Efferent Couplings	30
Lines of Code	1,486
Number of Characters	59,245
Number of Comments	27
Number of Constructors	0
Number of Fields	69
Number of Lines	1,883
Number of Methods	136
Number of Packages	14
Number of Semicolons	930
Number of Types	32
Weighted Methods	193

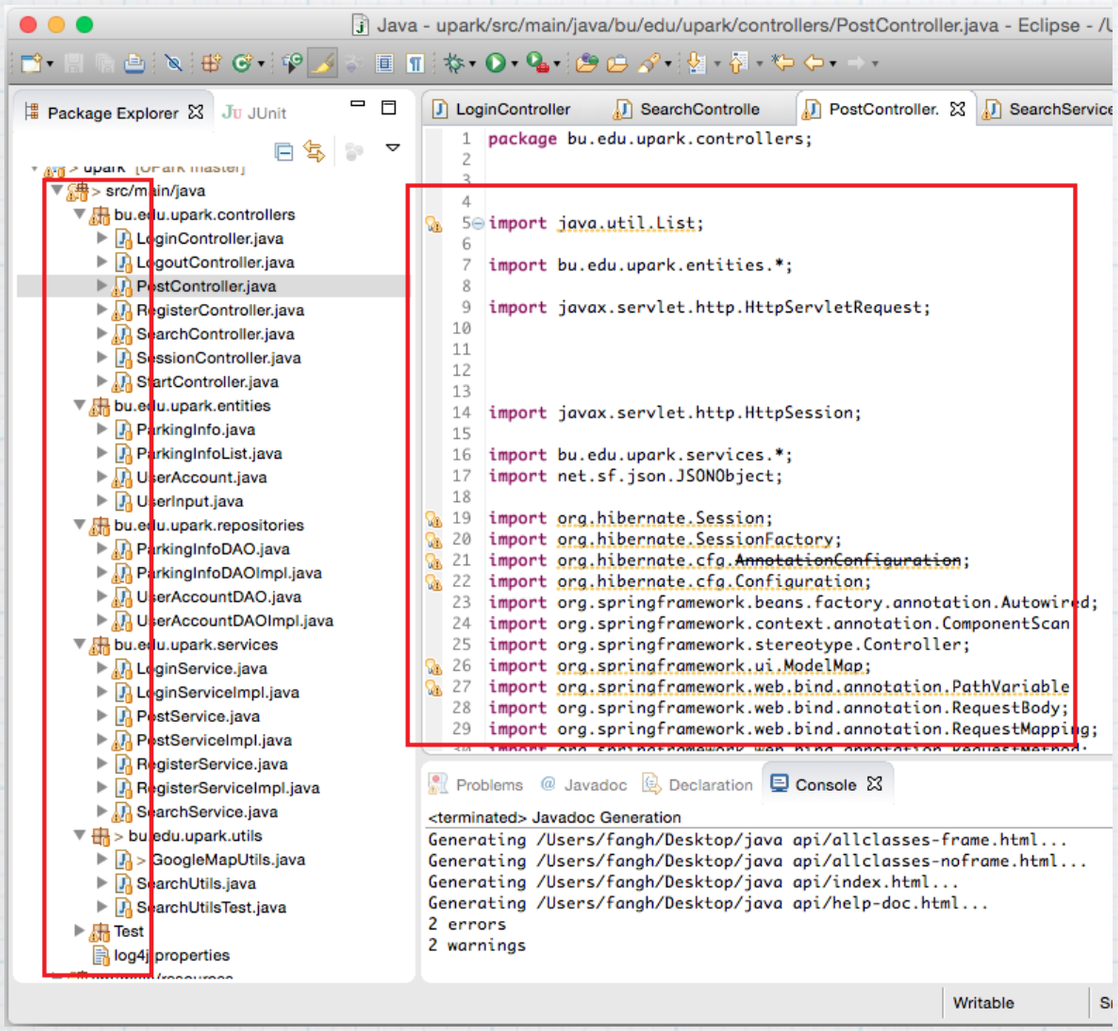
Generated by Google CodePro

Standard

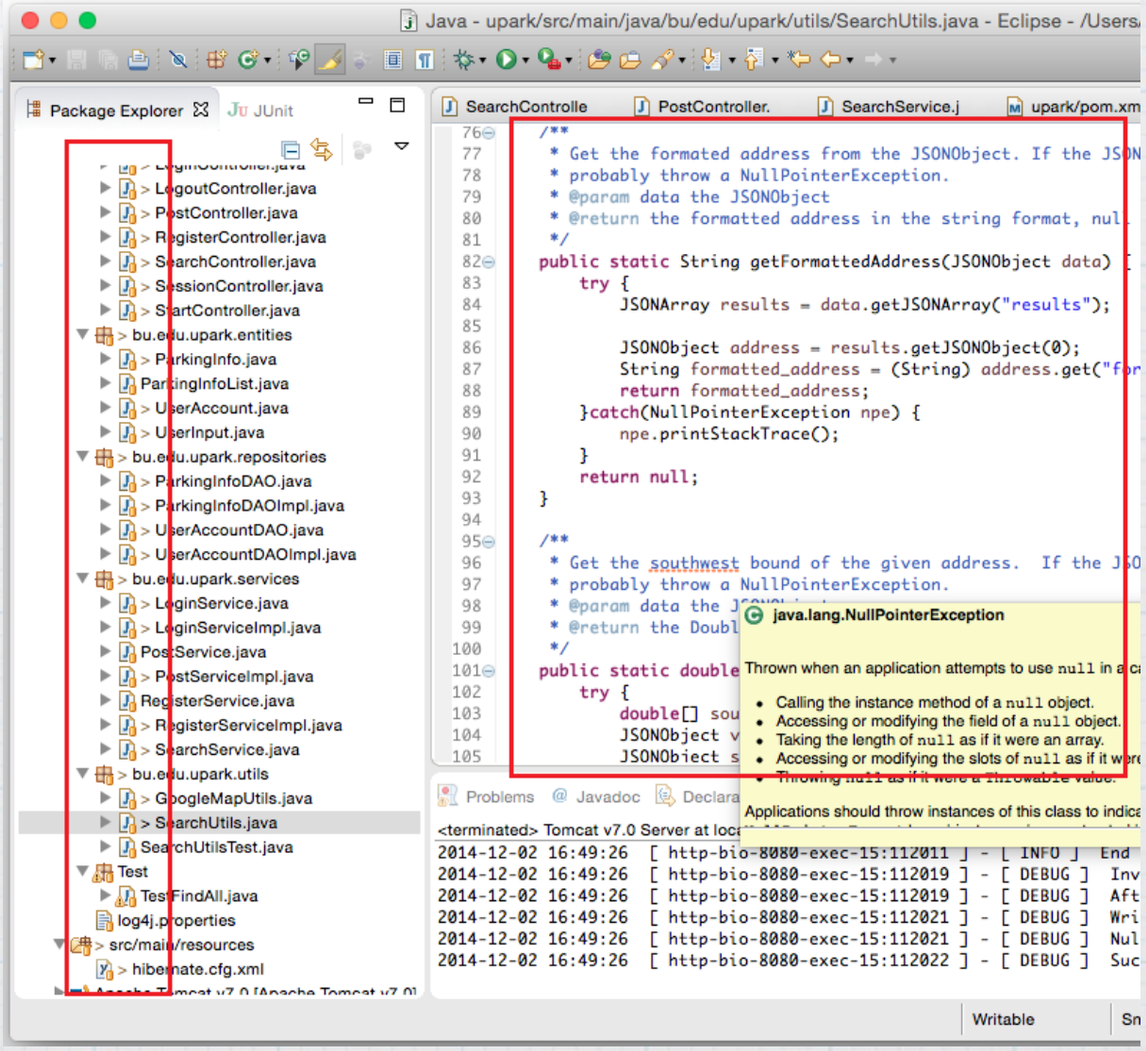
* Meeting and Friday Coding Time
* Documentation: project plan, software design, requirement, weekly report, others on Git, database, spring framework, coding style, etc
* Coding style: checkstyle plugin for eclipse

Coding Style

Before



After



Inspection/Review

* MCode review: developers review each other's
* Story Review/Test: once each functionality's done, reviewed and tested by QAs
* Iteration Review: peer reviewing each other's performance

Testing

* Unit Test
    * JUnit, by developer
* Integration Test
    * Grouped functionalities (Log in/out, search / display, registration/ log in, etc)
    * Selenium, manual
    * Test cases

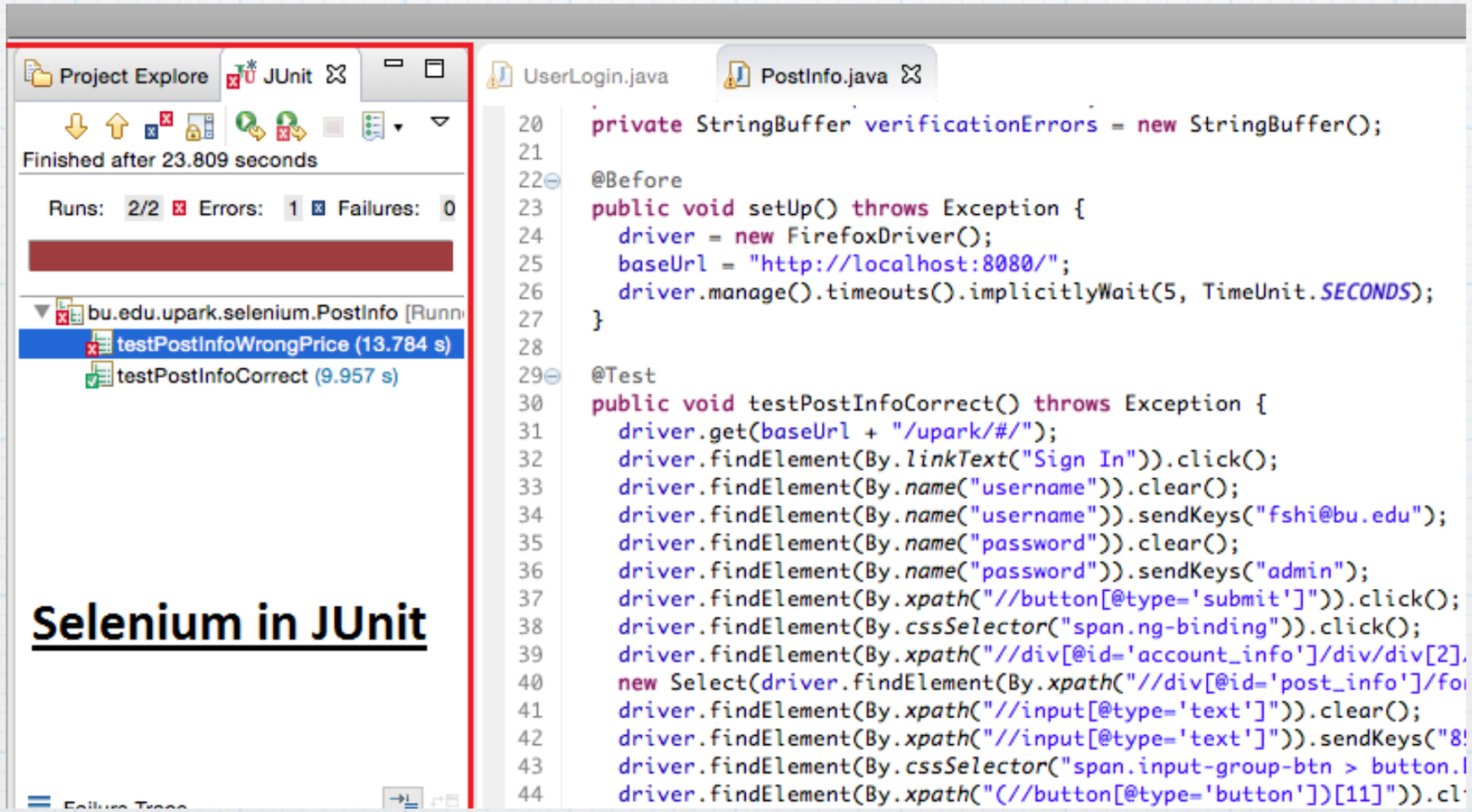
Testing

* Regression Test
  * Test after each iteration and before deployment
* Acceptance Test
  * Final round before “release” to check if requirements are met (last minute features)

Testing


Runs: 6/6
Errors: 0
Failures: 0
bu.edu.upark.utils.SearchUtilsTest [Runner: JU
getZipcodeTest (0.001 s)
getFormattedAddressTest (0.000 s)
getSouthwestTest (0.000 s)
getNortheastTest (0.000 s)
getNewSouthWestTest (0.000 s)
getNewNorthEestTest (0.000 s)
Failure Trace
JUnit Test
4
5 import net.sf.json.JSONObject;
6
7 import org.junit.After;
8 import org.junit.AfterClass;
9 import org.junit.Before;
10 import org.junit.BeforeClass;
11 import org.junit.Test;
12
13 public class SearchUtilsTest {
14
15 public static JSONObject data;
16 public static double[] southwest;
17 public static double[] northeast;
18
19
20 @BeforeClass
21 public static void before() {
22 System.out.println("global");
23 String path = "./src/test.json";
24 String str = SearchUtils.ReadFile(path);
25 data = JSONObject.fromObject(str);
26 southwest = SearchUtils.getSouthWest(data);
27 northeast = SearchUtils.getNorthEast(data);
28 }
29
30 @AfterClass
31 public static void after() {
32 System.out.println("global desotry");
33 }
34

Testing


Project Explore
JUnit
Finished after 23.809 seconds
Runs: 2/2 Errors: 1 Failures: 0
bu.edu.upark.selenium.PostInfo [Runn
testPostInfoWrongPrice (13.784 s)
testPostInfoCorrect (9.957 s)
Selenium in JUnit
Failure Trace
UserLogin.java
PostInfo.java
20 private StringBuffer verificationErrors = new StringBuffer();
21
22 @Before
23 public void setUp() throws Exception {
24 driver = new FirefoxDriver();
25 baseUrl = "http://localhost:8080/";
26 driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
27 }
28
29 @Test
30 public void testPostInfoCorrect() throws Exception {
31 driver.get(baseUrl + "/upark/#/");
32 driver.findElement(By.linkText("Sign In")).click();
33 driver.findElement(By.name("username")).clear();
34 driver.findElement(By.name("username")).sendKeys("fshi@bu.edu");
35 driver.findElement(By.name("password")).clear();
36 driver.findElement(By.name("password")).sendKeys("admin");
37 driver.findElement(By.xpath("//button[@type='submit']")).click();
38 driver.findElement(By.cssSelector("span.ng-binding")).click();
39 driver.findElement(By.xpath("//div[@id='account_info']/div/div[2]
40 new Select(driver.findElement(By.xpath("//div[@id='post_info']/fo
41 driver.findElement(By.xpath("//input[@type='text']")).clear();
42 driver.findElement(By.xpath("//input[@type='text']")).sendKeys("8
43 driver.findElement(By.cssSelector("span.input-group-btn > button.
44 driver.findElement(By.xpath("(//button[@type='button'])[11]")).cl

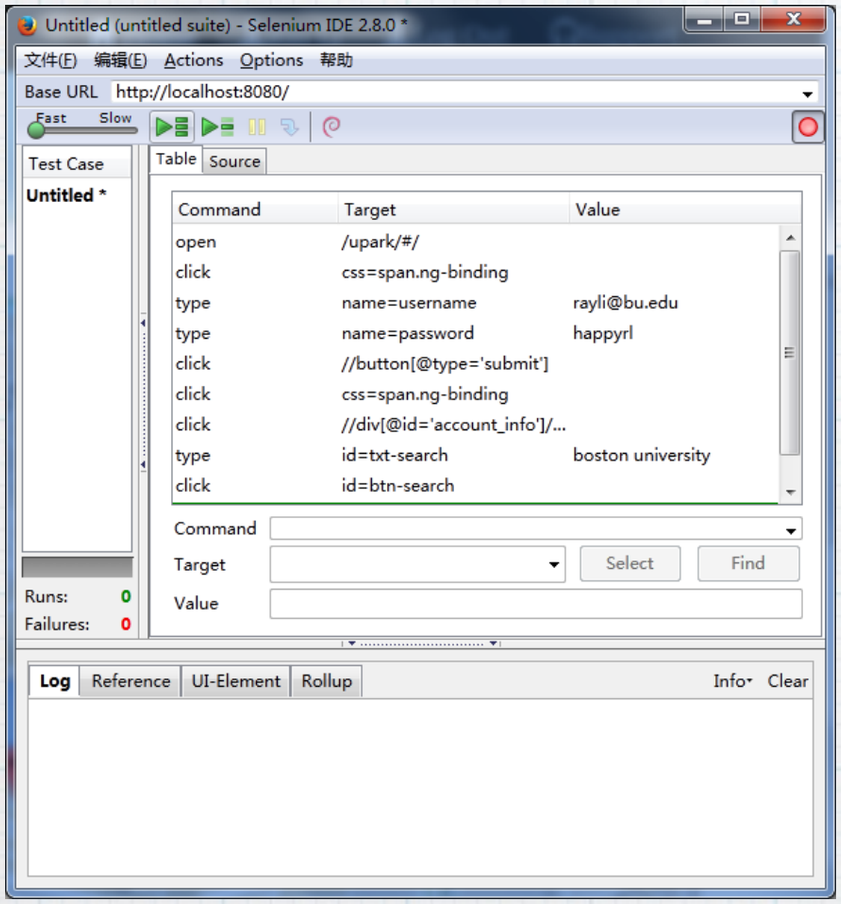
Test Cases

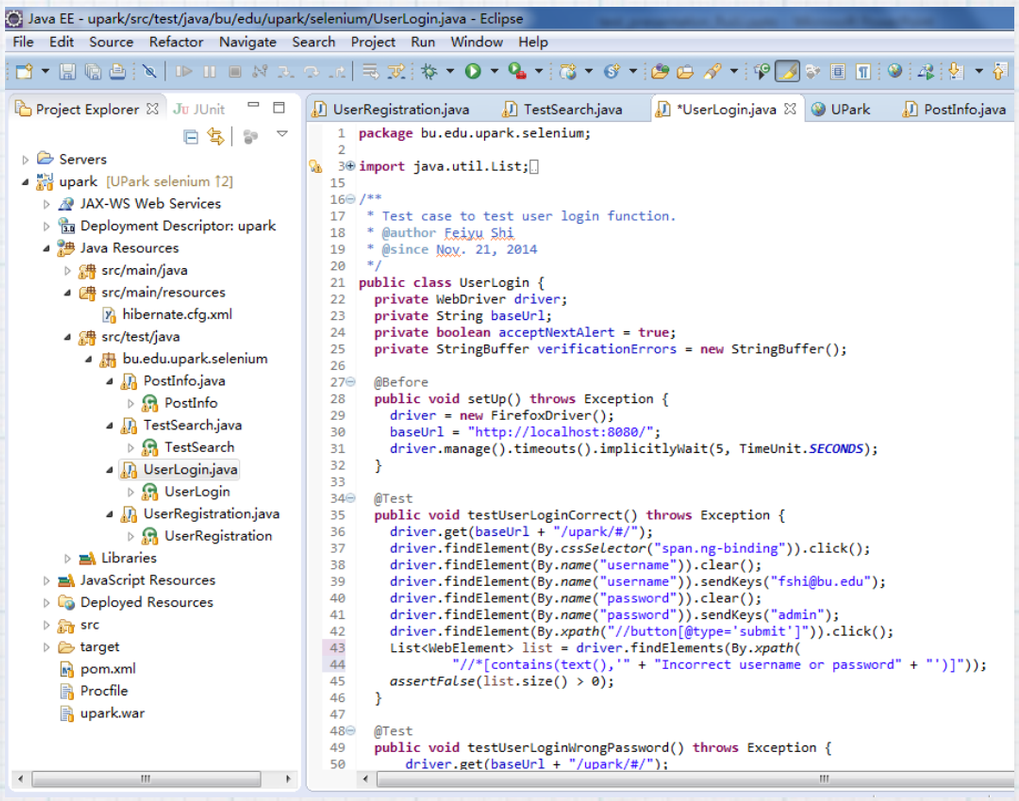
- MLog in/out: 4
    - Selenium: 4 (test illegal inputs)
- Account registration: 5
    - Selenium: 5(test illegal inputs)
- Information posting: 4
    - Selenium: 4 (test illegal inputs)
- Search locations: 2
    - Manual: 2 (test info display)
- Result: at least 4 bugs reported

Defect Management

* Degree: low, medium, major
* Defect follow-up: report to developer immediately; follow-up frequency depends on degree
* So far, all defects should be fixed.

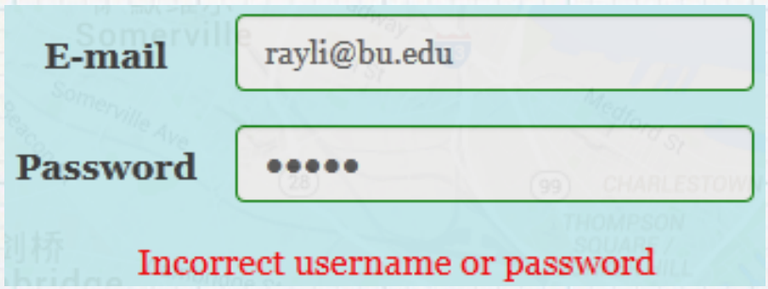
Selenium





Selenium IDE exported as Junit 4

Verification




```
List<WebElement> list = driver.findElements(By.xpath(
        "//*[contains(text(),'" + "Incorrect username or password" + "')]"));
assertFalse(list.size() > 0);
```

Video of Automatic Test

Risk Management

General Cause of Risks

* Lack of Information
* Lack of Control
* Lack of Time

An example for a failure

By the end of Iteration2, we were behind course schedule

We should set up most develop and test tasks

Consider

Risk Identification

* Scope and Quality
* Resources
* Schedule

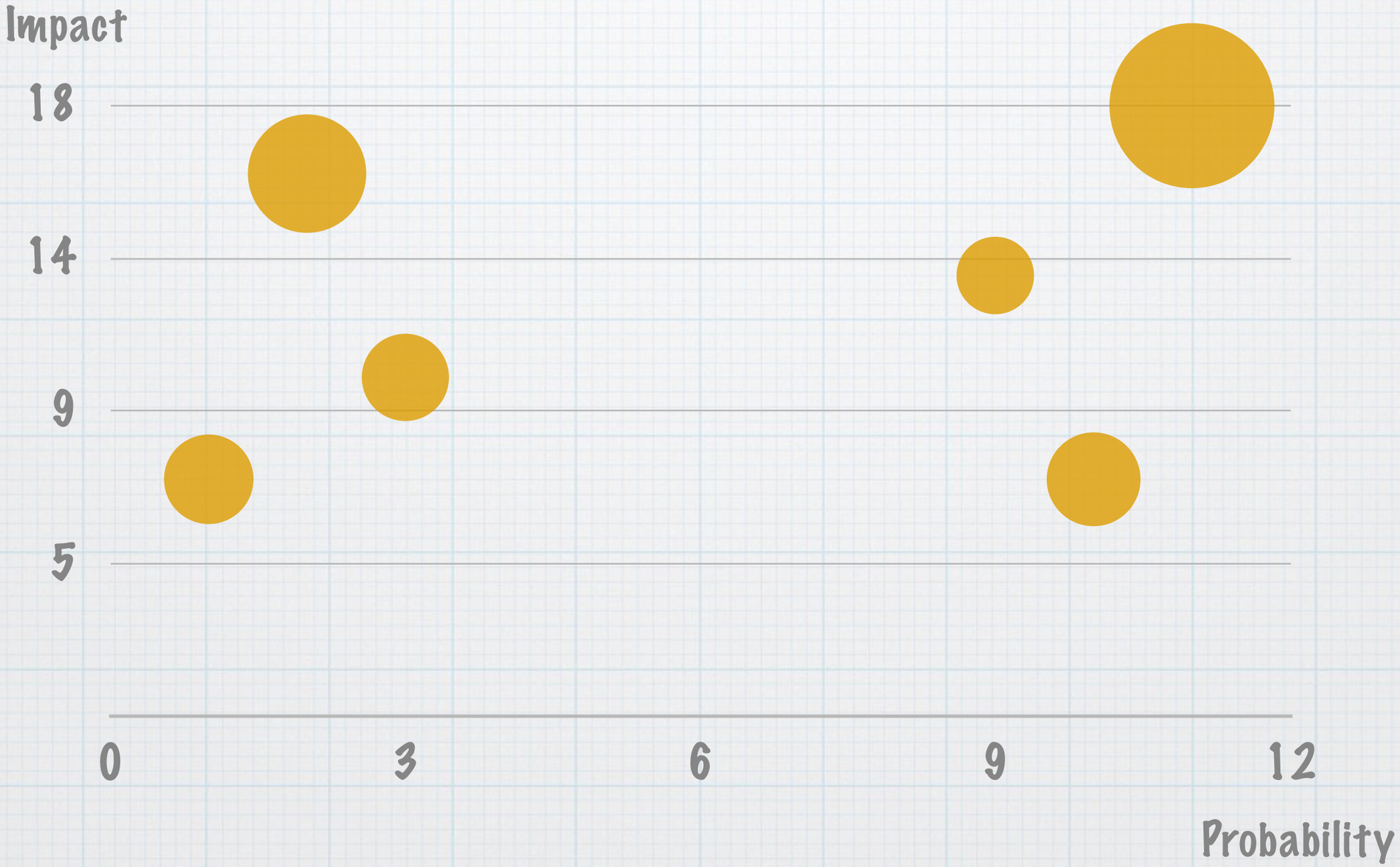
Scope and Quality Risks Remedies

	Remedies
Unrealistic goals	do research, estimate, communicate
Complexity of task	Decompose complex tasks into smaller tasks
New tools and techniques	spend time and early use
…	…

Schedule Risks Remedies

	Remedies
Delays in critical path	Most experienced team members for these tasks
Complexity of task	Early Feedback

Risk Matrix



DEMO