



**University of Missouri, Columbia
Senior Capstone Design SP2016**

**Reserve It
Final Report**

Group 11

Qintai Liu
Yihua Shi
Xueyan Wu
Chunyang Zhan

Report Date: 04/25/2016

Table of Content

1.	Executive summary.....	3
1.1.	Problem Need.....	3
1.2.	Problem Objectives.....	3
2.	Requirements.....	3
2.1.	Hardware Requirements.....	3
2.2.	User Requirements.....	4
3.	Constraints and applicable standards.....	5
4.	Design.....	5
5.	Testing / data collection and analysis plan.....	18
6.	Project management plan.....	20
7.	Development Plan.....	20
7.1.	Development Strategy.....	20
7.2	How to implement the development strategy.....	20
7.3	Other Alternate Strategy.....	21
8.	Problem analysis and research & Required Independent Learning....	22
8.1.	Group Research Paper 1.....	22
8.2.	Group Research Paper 2.....	27
8.3.	Group Research Paper 3.....	31
8.4.	Group Research Paper 4.....	36
9.	Discussion.....	40
9.1.	Transaction management - Payment buttons.....	40
9.2.	Algorithm of how to determine whether permit is sold out.....	42
10.	Development costs and resource needs.....	43
11.	Results / project outcomes.....	43
12.	Future work.....	55
13.	References.....	56
14.	Appendices.....	57

1 Executive summary

1.1 Problem Need

Mizzou is holding a lot of football games every year. Many people have to arrive at Columbia about 8 hours in advance in order to have a parking spot since the wave is not only from Columbia but also from other cities like St. Louis. Otherwise, it is likely that people who arrive at the game just a couple of hours before game may waste long time finding an available parking spot. Meanwhile, private parking lots around the stadium owned by some individuals or organizations are sold scatteredly for people driving to watch the game. While some people might be lucky to find an available private parking spot, most have no acknowledgement about open private parking lots and will spend a large amount of time searching around.

Our client was looking for a team to develop a website for his parking management business. The website will be used to allow people to reserve and purchase private parking lots ahead of schedule in avoidance of the waste of time. Some private parking lots will be rented by our client.

1.2 Problem Objectives

Our group will build a web application called “Reserve It”. “Reserve It” allows users to make a reservation for a parking spot they prefer with purchase. Additionally, it can show places with available private parking spots near the stadium. In other words, one can easily find a spot to park without searching around. “Reserve It” saves time and eases the trip planning process. Our goal is to build this application as a great alternative for those who would rather pay than waste time searching for a parking spot in advance of the football games.

2 Requirements

2.1 Hardware Requirements

Our web application would be compatible with all IOS, Android and Windows mobile devices as well as all the laptops. Users’ facilities should be able to connect to the network, the additional navigation function would require the support of GPS of the devices. To support functions and assure the information can interact between between users and the system, a database is needed to store data.

2.2 User Requirements

All pages should have back arrows that take the user back to the previous page

1. Access our website through URL or QR code

- Once accessed, the user can sign up, login or search game.

2. The ‘Sign Up’ Interface

- First Name
- Last Name
- Email
- Phone Number
- Password

3. The “forgot password” Page

- If the user forgets the password, our website will send a unique URL to the email address that the user submitted.
- Then the user can change password through that URL.

4. ‘How it Works’ Interface

- What is it? Tired of arriving early just parking? Tired of driving around in traffic look for parking? Want to ensure you have that perfect tailgate spot on game day? Well then, “Reserve It”. “Reserve It” allows you to reserve the parking space of your choice. “Reserve It” saves time and eases the trip planning process.
- Pricing: prices vary based on proximity to the event. Different private lots have different prices.
- Where can I park? Anywhere in the lot where you reserved a spot.
- Am I allowed to cancel my reservations? Sorry, we don’t support refund unless the event is cancelled.

5. Users should be able to search by the following:

- Search by date
- Search by event
- Search by both date and event

* Users can choose to search with or without login. However if the user wants to make purchase, he has to sign up or login first.

6. There will be a menu option “My account” in the top right corner which will redirect page to “Dashboard” which includes:

- User Profile: Allow user to edit their user info and change password
- Order History: Order detail and corresponding google map

7. After users search the game, all the choices will be displayed on the result page..

- It should show the related address , game date, game time, parking lot, address and price.

8 . The Purchase Process

- Users will pay through their PayPal accounts
- Submit and confirm purchase by being prompted to enter the login password
- If the ticket is sold out, after click the “Buy Now” button, the page will redirect to the home page.
- If the purchase is completed and successful, the user will receive confirmation email with confirmation number, event date, event time, parking location and QR code directly.

*QR code is verified on the game to allow user to park.

3 Constraints and Applicable Standards

- The QR code needs to be verified manually instead of the sensor.
- The database is not able to update automatically.
- Order cancel and refund are not supported unless the event is cancelled.

4 Design

This part of documentation shows how our project system will operate. Different kinds of methods including flowcharts, ERD and sequence diagrams have been used to describe it. Especially for the site maps, it is convenient for users and client to know how the design works.

1. Flowcharts of the System:

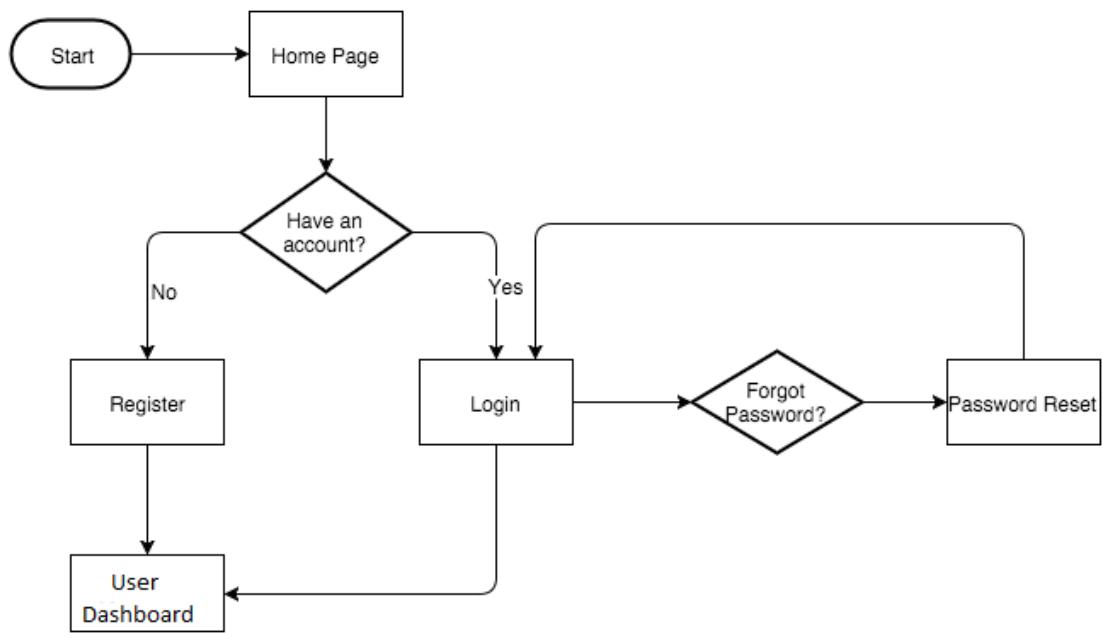


Figure 1, Login

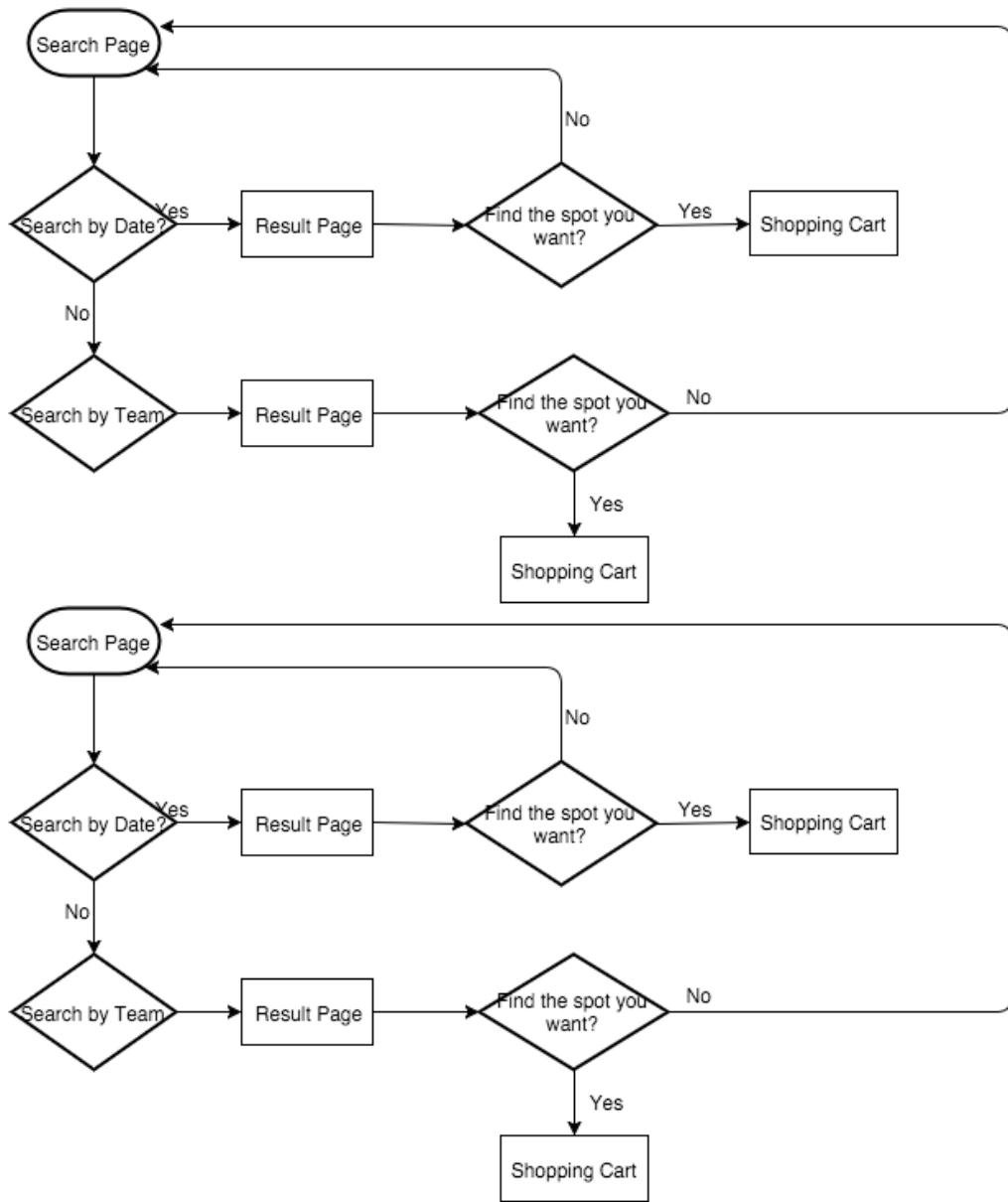


Figure 2, Search the Event

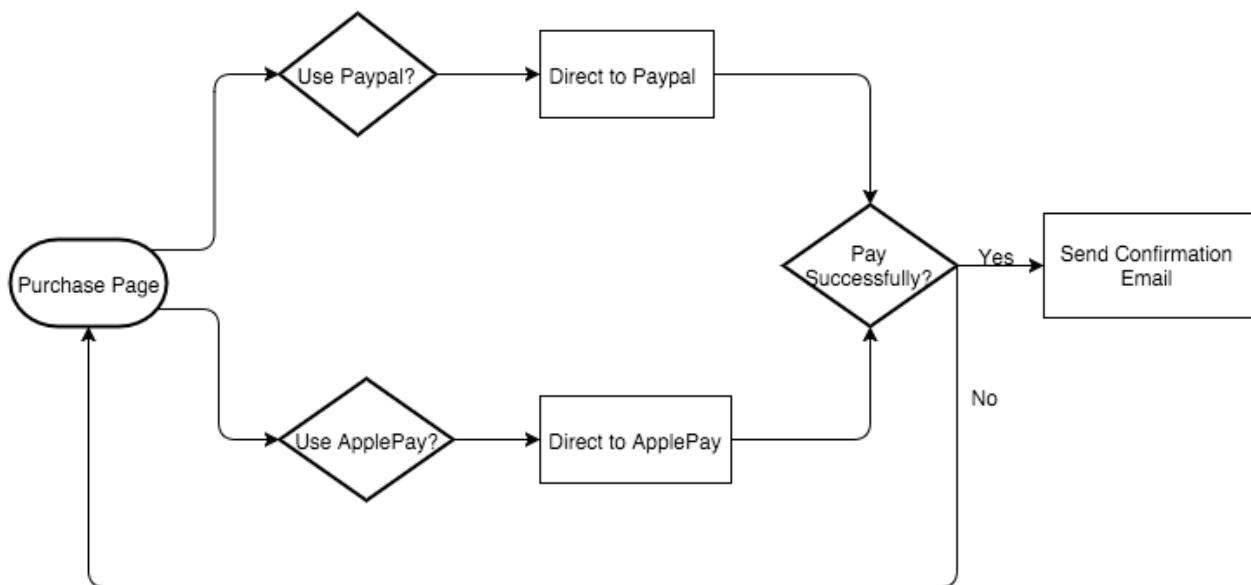


Figure 3, Shopping Cart and Purchase

*ApplePay is only used for Mobile.

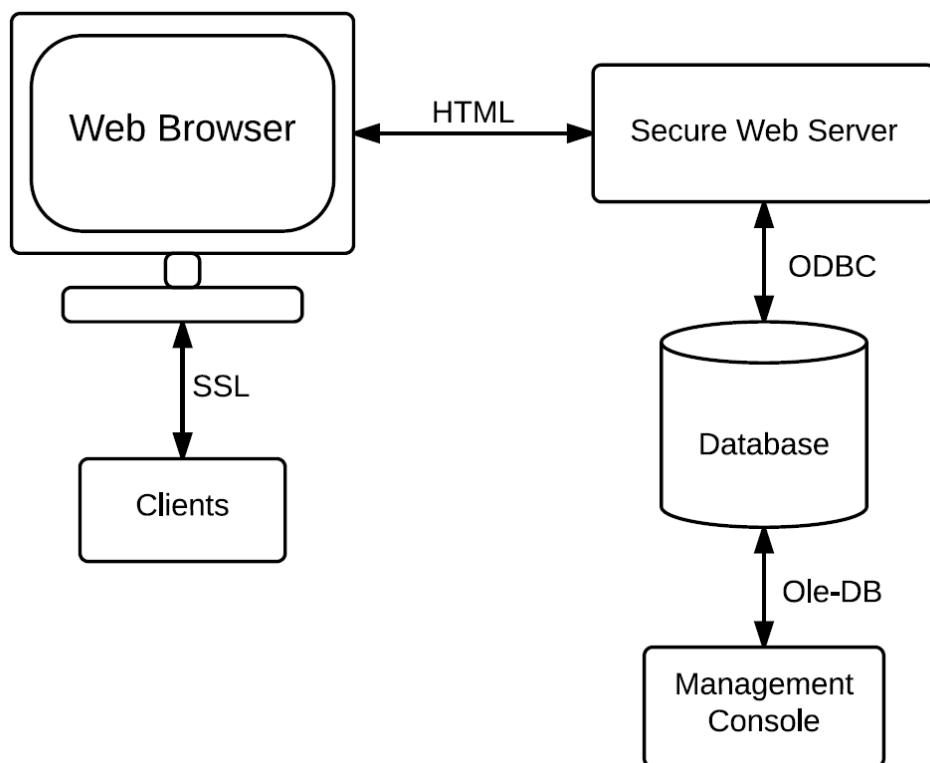


Figure 4, System Architecture

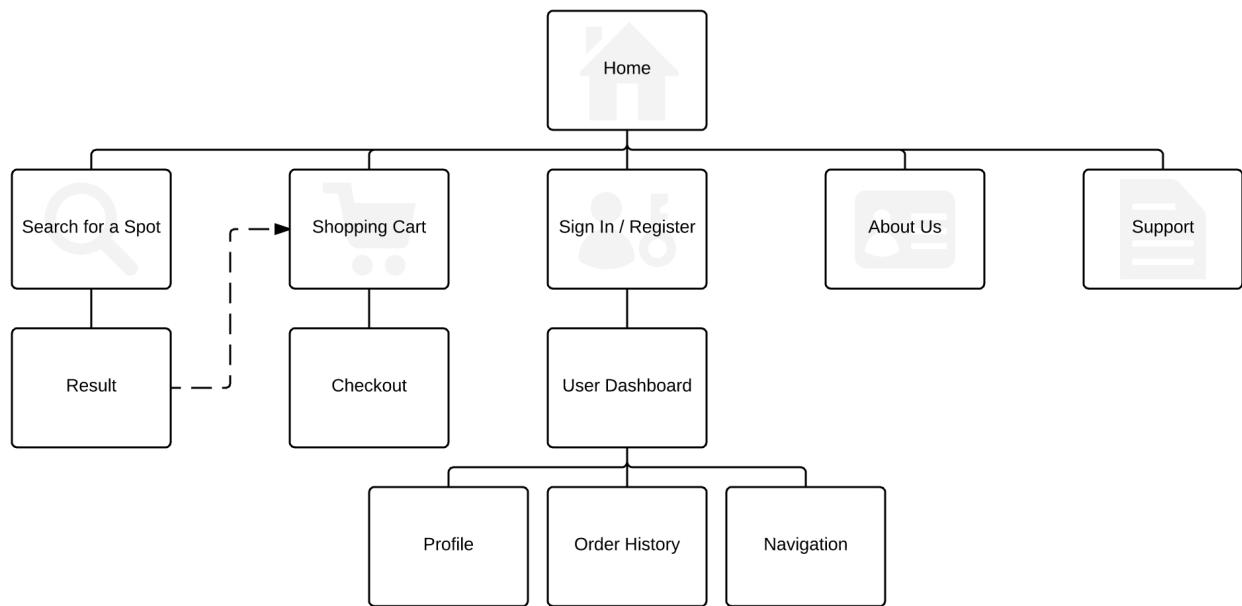


Figure 5, Site Map:

2. Responsive GUI:

- For phones:

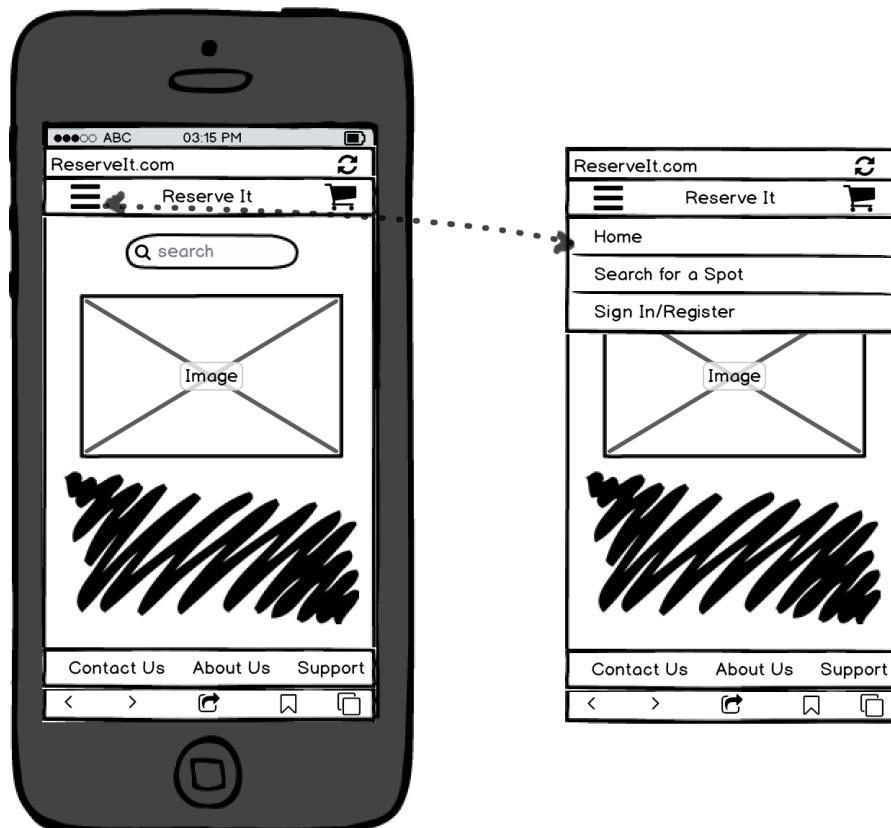


Figure 6, Home Page

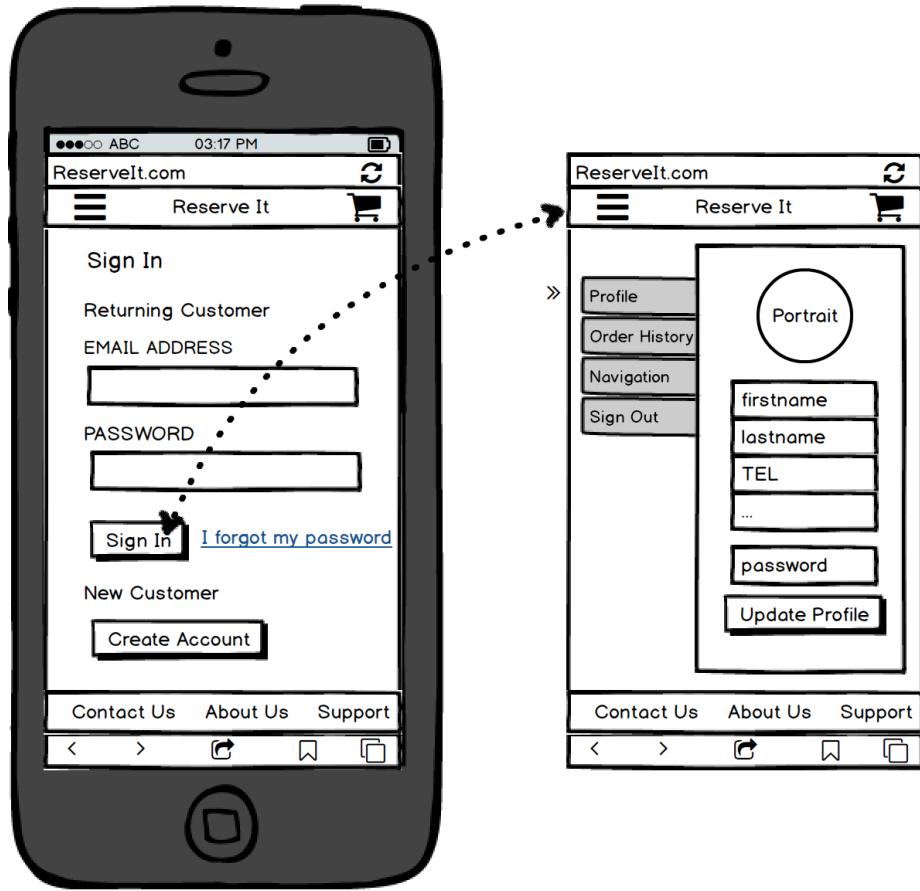


Figure 7-1, User Page

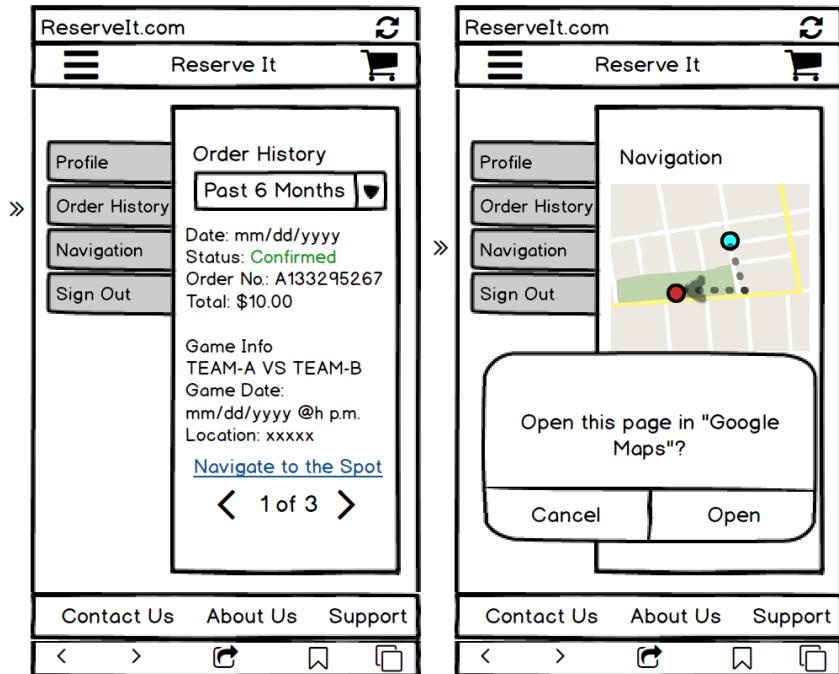


Figure 7-2, User Page

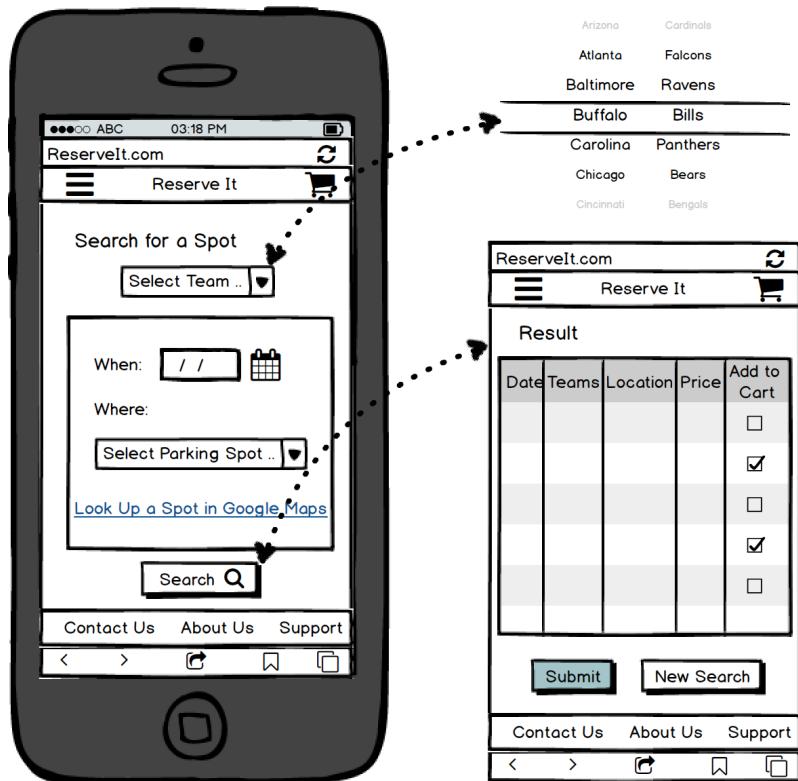


Figure 8, Search Page

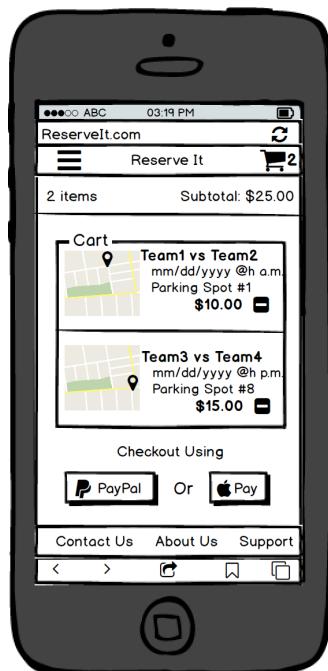


Figure 9, Shopping Cart

- For laptops & tablets:

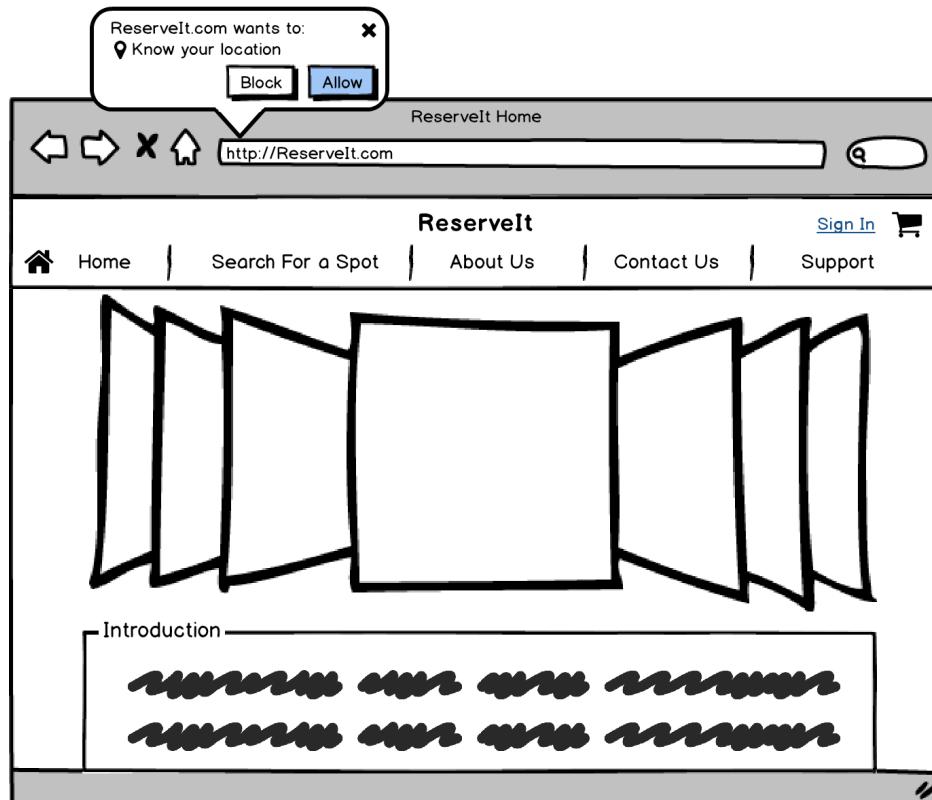


Figure 10, Home Page

Figure 11-1, User Page

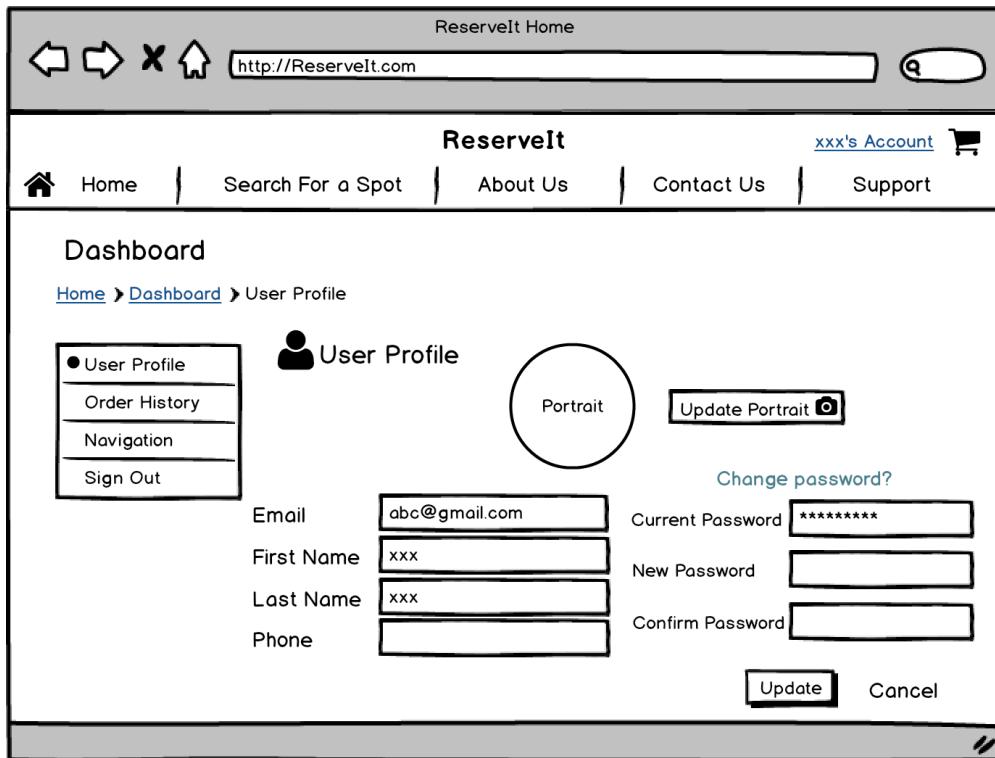


Figure 11-2, User Page

The screenshot shows the 'Order History' section of the ReserveIt website. A sidebar menu on the left shows 'Order History' selected. The main area has a heading 'Order History' with a document icon. It includes a dropdown filter set to 'past 6 months'. Below is a table with columns: Date, Status, and Order Number. Three rows are listed: one confirmed order (A133295267) and two reserved orders (C576233874 and B323986285). Each row has a 'view details' link. To the right, a callout box titled 'Order Details' lists two orders: 'Team1 vs Team2' (mm/dd/yyyy @h a.m., Parking Spot #1, \$10.00) and 'Team3 vs Team4' (mm/dd/yyyy @h p.m., Parking Spot #8, \$15.00). It also shows a subtotal of \$25.00 received via PayPal. Arrows point from the 'view details' links in the main table to the corresponding order details in the callout box. The URL in the browser bar is http://ReserveIt.com.

Figure 11-3, User Page

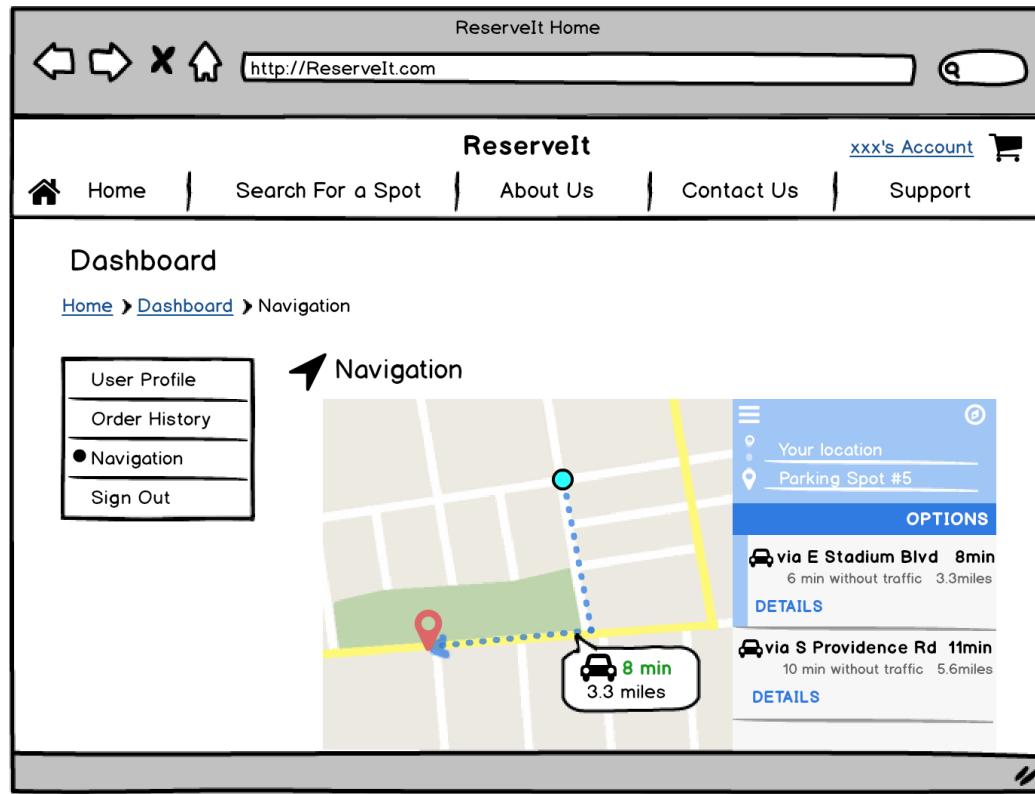


Figure 12, Search Page

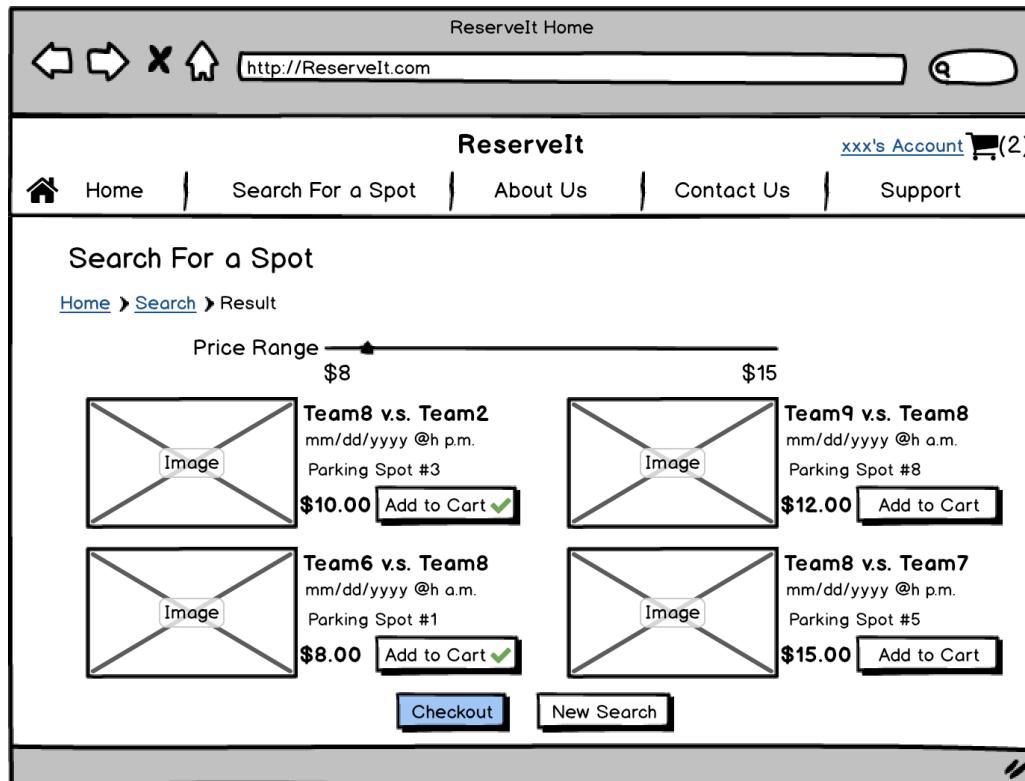


Figure 13, Result Page

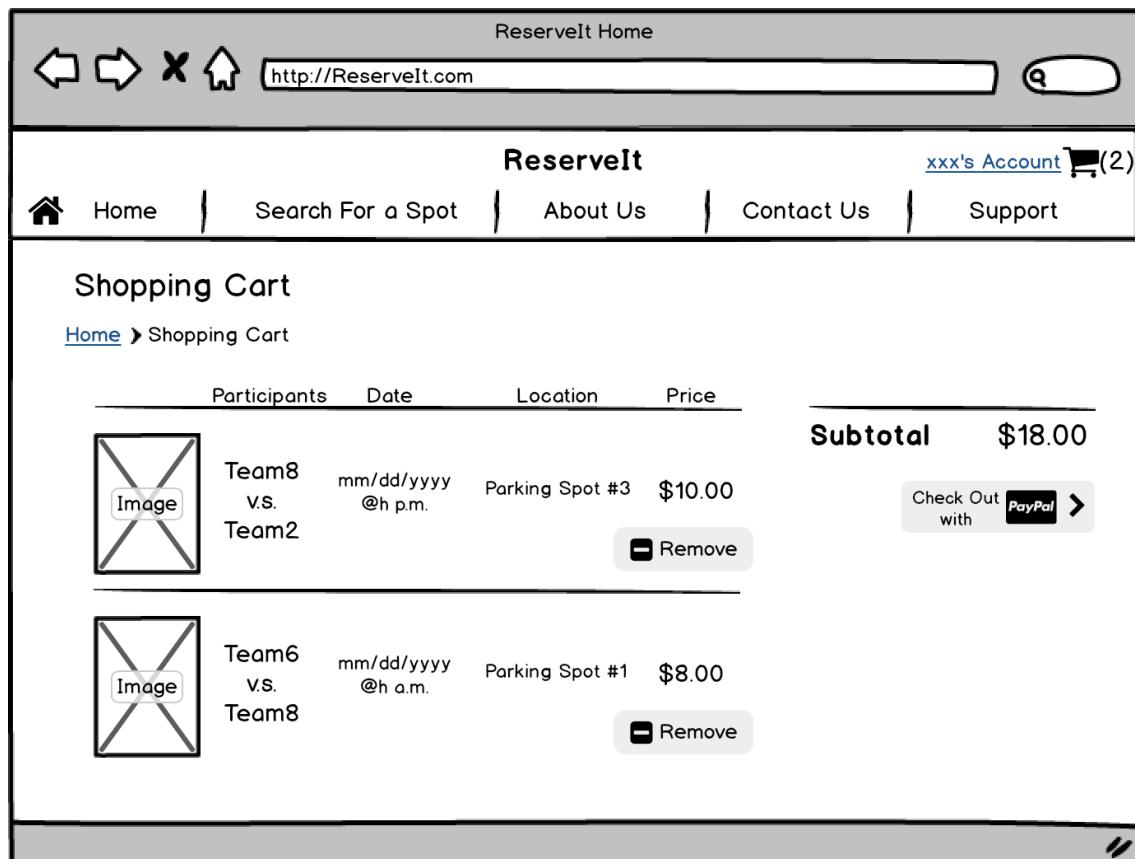


Figure 14, Shopping Cart

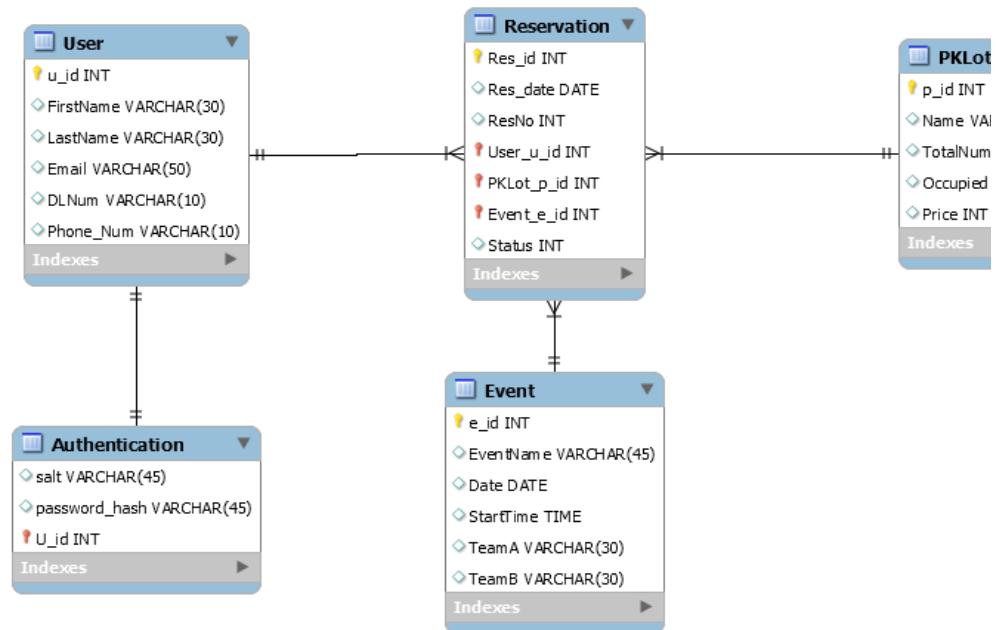


Figure 15, ERD

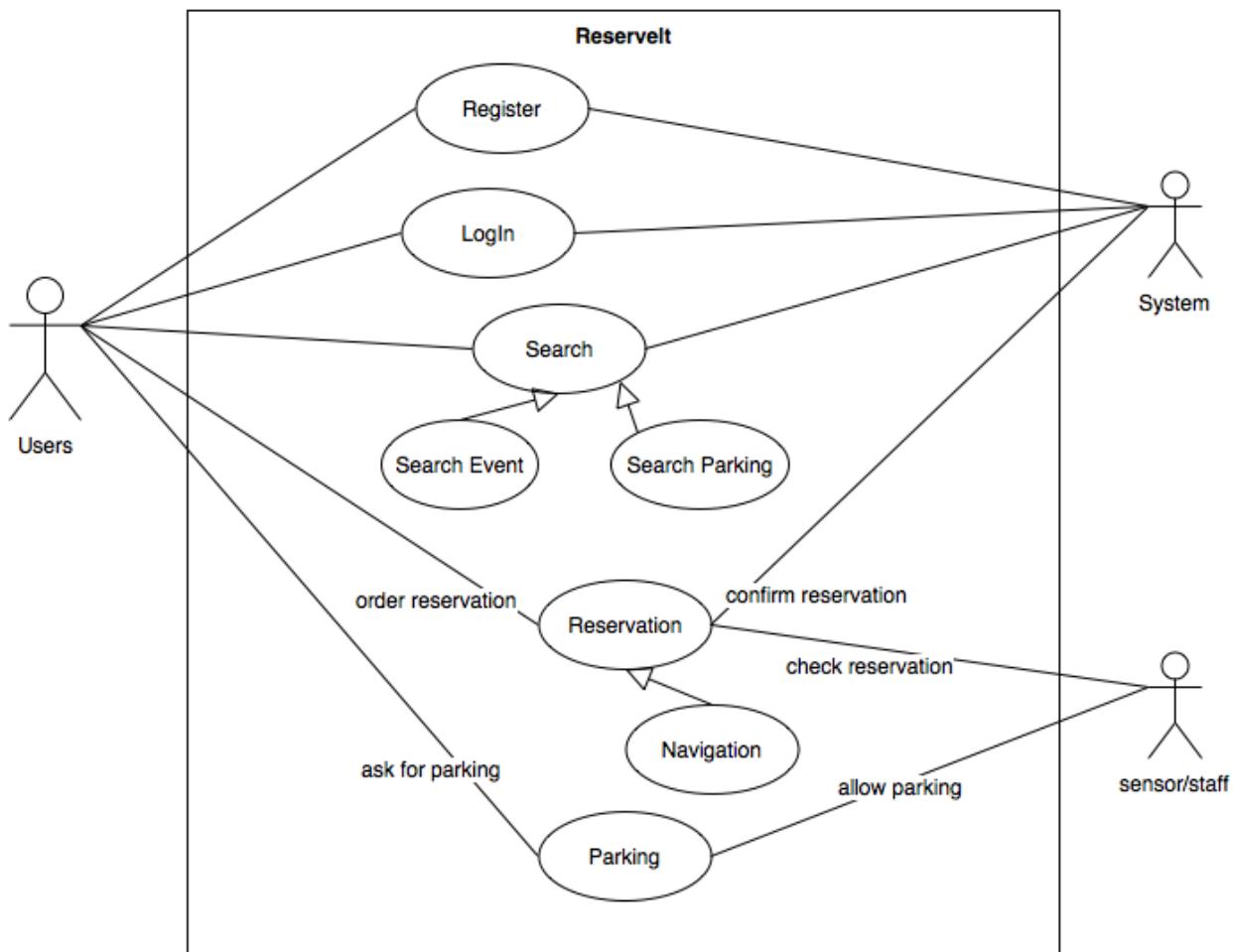


Figure 16, Use Case

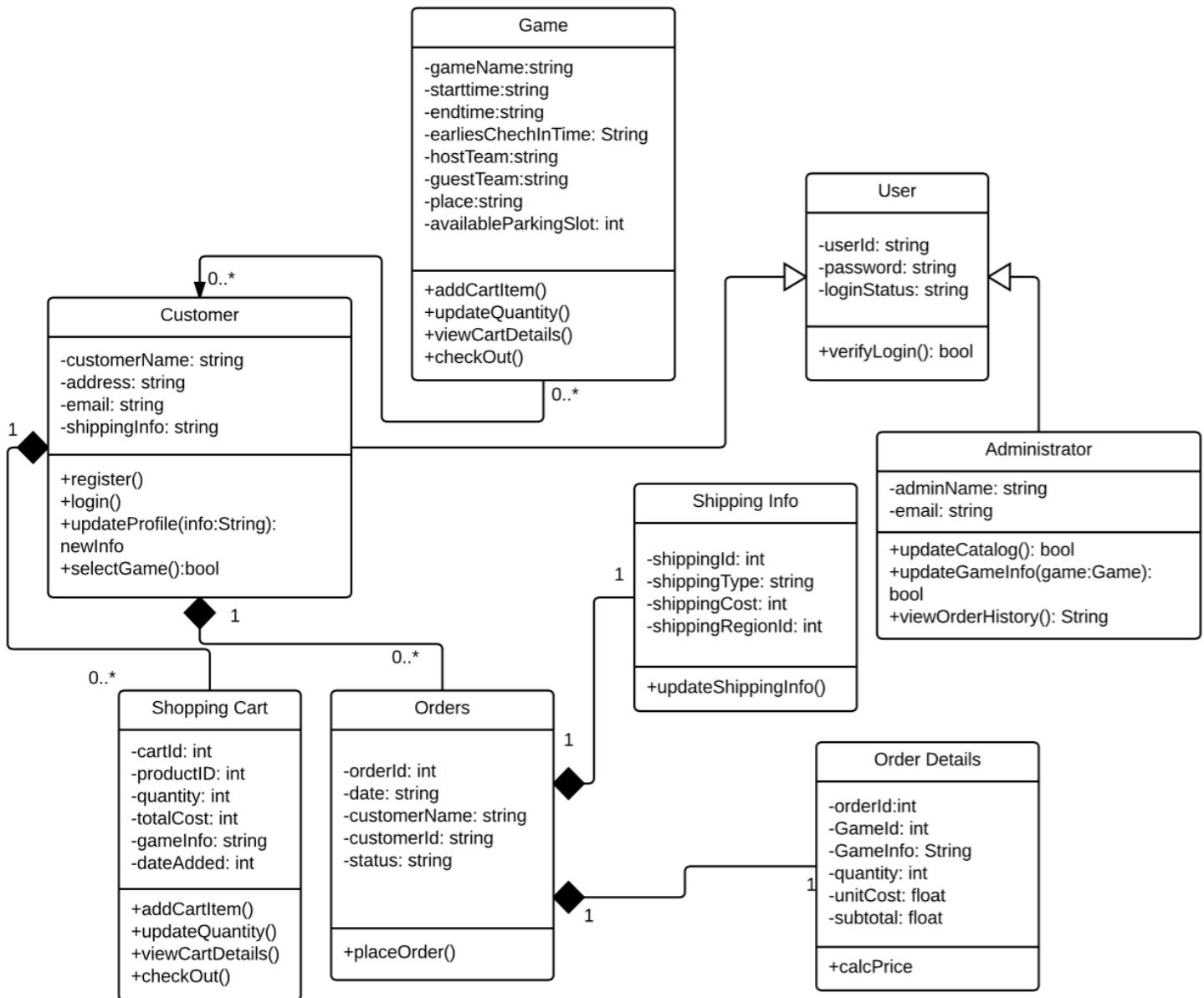


Figure 17, Class Diagram

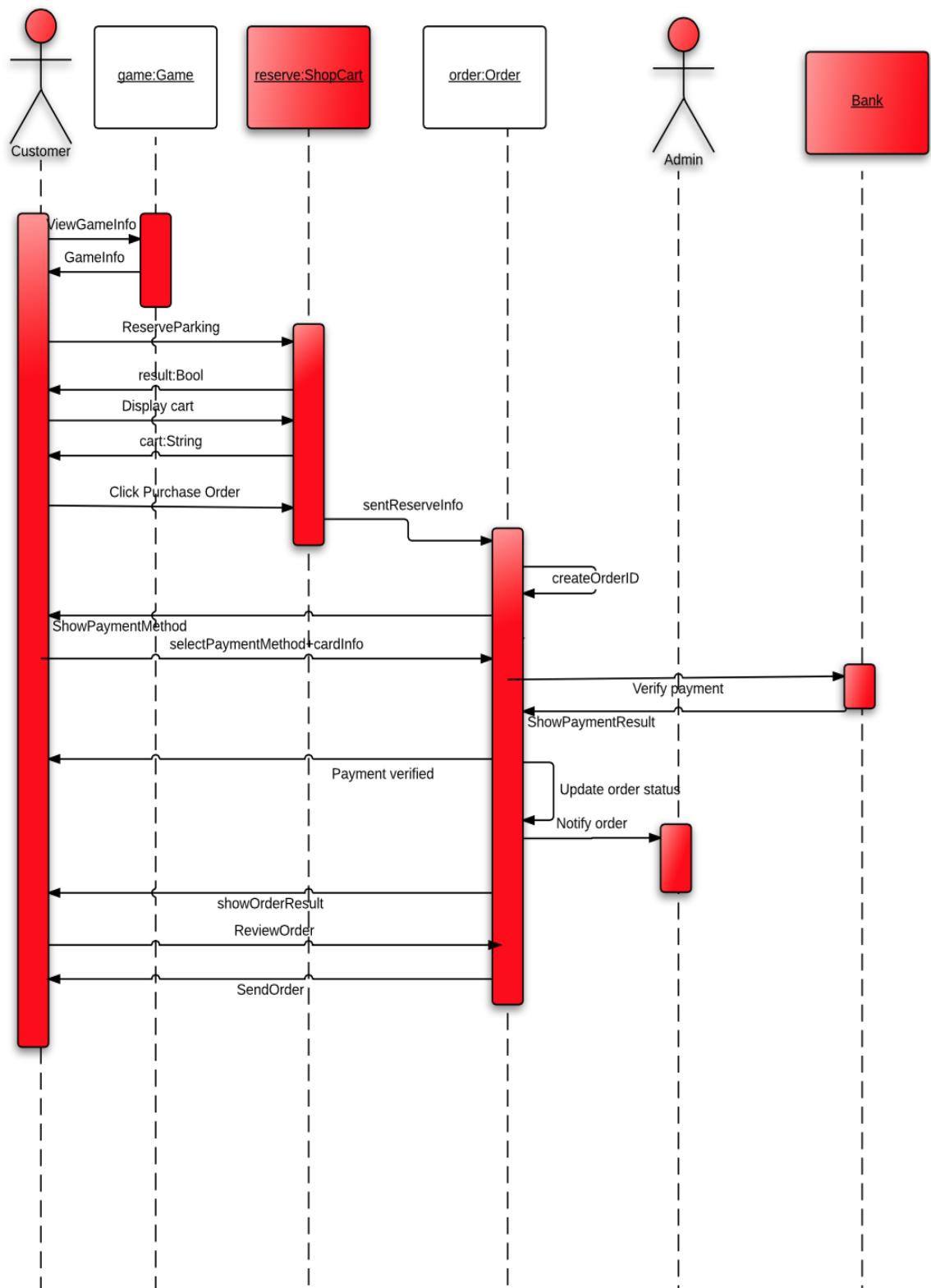


Figure 18, Sequence Diagram

5 Testing/data collection and analysis plan

Design of Testing

- Web and Device Testing:

The first part is to test the reservation system by using various devices, such as personal computers, laptops and smartphones(android or ios system). We will access our application through these devices and attempt to reserve a parking space. Once the payment and the reservation are approved by the system, our reservation system is functionally completed.

Testing Process:

1. Login system:

- a. Registration: We would register a new account and use the account to login. We will check whether the new account can login and whether the data is written into the database.
- b. Password: We will try different wrong passwords try to log in to ensure the security. In addition, we will test “forget password” function, which asks the system for the password and send the password via email if the user forget it.

2. Search

We will test our searching engine based on specific team, parking space, date or any combination of them. We need to check whether the results are user-friendly and easily shown. All returned results should be guaranteed to be correct and exactly the same as our data in the database.

3. Reservation

- a. Place order: We will test reservation function as what exactly same as users will do. We will test parking for a single event at one time and several events at one time separately, then check if the orders added to the database correctly.
- b. Payment: We would attempt to pay a order by ourself to see whether the including app works properly. By doing this, we will check the transition, order details and other issues happening during payment to guarantee the app works well. We do need to avoid any unpredictable or unconsidered error in this process.
- c. Confirmation: We will check whether the system can generate a barcode/QR code correctly after a reservation has been confirmed.

4. Order History

We will use our test account to finish some orders first, then check all orders the visual user have. We will proofread the result shows on the web application with the data in the database to see whether they are correct.

5. Navigation

- a. On PC/Laptop: We will examine the google maps api on our application. We will test several time by using different starting point(maybe where users live) and different parking space we provide

- b. On smart phone: We will use smart phone in different system(android or ios) to check whether our application can request the Google Maps API start and pass the correct starting point and destination to for navigation.
- Sensor Connection Testing:
This part is to test the real-time update. We will use a sensor or scanner to scan our barcode or QR code. If information can interact between the system and the facility, and the parking data can automatically update for users, this part will be considered completed. This process includes following segments: Whether the barcode/QR code convert correctly; Whether the data update correctly;
 - Stress Testing
The third part is stress test, which is to see the capability of our website under the huge traffic flow. Google Analytics will be one of our choices to pinpoint the performance of the website.

6 Project Management Plan

Development Timeline & Work Delegation

Capstone II	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12
Database Table Design	Chunyang											
Database Data Collecting and Dump		Yan & Yihua										
Home Page	Chunyang											
Login/Register		Yan										
Search		Qintai & Chunyang										
User dashboard			Yihua									
PayPal API implementation				Qintai								
Mobile UI Adjustment					Yihua							
Web and Device Testing						Chunyang						
Sensor Connection & Testing							Qintai					
Stress Testing								Yan & Yihua				

Figure 19 , Management Plan

7 Development Plan

7.1 Development Strategy

Each of development strategy has its own recognized strengths and weaknesses while each of them is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations. After much deliberation, we plan to use the spiral model as our development strategy, which combines the idea of iterative development with the systematic, controlled aspects of the waterfall model.

There are several reasons why we choose spiral model.

- There may be some problems, risks and errors through development since it's the first time that we develop a parking app and we are not familiar with it. Spiral strategy has risk analysis preceding each phase of the cascade, which allows us to be able to accurately assess risks and solve problems during development.
- The client gave us a relatively complex requirements and therefore spiral strategy would help us get clear evaluation.
- The specific requirements and demands of this software given by our client might be changed during the development. So this strategy is particularly utilized for adaption to the changeable requirement.
- Using spiral strategy has a great advantage of agile design that it allows for elements of the product to be added in when they become available or known. Fifth, this method enable our client to see the application early.
- The development can be divided into several smaller parts. Moreover, the risky parts are supposed to be developed earlier for better risk management.

7.2 How to implement the development strategy

A spiral iteration has 4 phases.

1. Planning phase
 - i. Activities performed : Requirements are studied and gathered. Feasibility study. Reviews and walkthroughs to streamline the requirements
 - ii. Deliverables / Output : Requirements understanding document. Finalized list of requirements.
2. Risk analysis phase
 - i. Activities performed : Requirements are studied and brain storming sessions are done to identify the potential risks. Once the risks are identified , risk mitigation strategy is planned and finalized.
 - ii. Deliverables / Output: Document which highlights all the risks and its mitigation plans.
3. Engineering phase
 - i. Activities performed : Actual development and testing if the software takes place in this phase.
 - ii. Deliverables / Output: Code. And test cases and test results. Test summary report and defect report.

4. Evaluation phase
 - i. Activities performed : Customers evaluate the software and provide their feedback and approval.
 - ii. Deliverables / Output: Features implemented document.

We will go through each phase and repeat it.

7.3 Other Alternate Strategy

In fact, even if our group plans to choose spiral model as our development strategy. There are some disadvantages of choosing this method. The most notable drawback is that this method is not beneficial and costly for smaller projects. It may takes pretty long time to achieve this method; however, our group only has one semester(about 3-4 months) to realize it.

Therefore, our group also has an alternative development strategy which is waterfall model. The benefit is that this method is simple and easy to understand and use, as well as a schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

However, the drawback is fatal. In the waterfall model, each phase must be completed before the next phase can begin and there is no overlapping between phases. This means that any phase in the development process begins only if the previous phase is complete. However, our client may changes his requirement. Additionally, our group can not familiar with the realm of parking app, so risk and uncertainty is high with this process model and cannot accommodate changing requirements.

8 Problem analysis and research & Required Independent Learning

8.1 Group Research Paper 1

Approaches to Consolidating Web Defense

Yihua Shi

The project that our group plans to work on is a web application named “Reservelt”, which allows people far away from the game fields to reserve a parking spot near the stadium before the football games and saves their time. There are several problems

concerning the security issue that call for our attention since our development would be based on web. One of those significant problems is that how to protect our users from potential risks of getting their sensitive information including their payment information stolen. There have been numerous cases annually where large companies get attacks by hackers resulting in serious leaks of their users' private information, and the data breach of credit as well as debit cards on Black Friday 2013 at Target still leaves people with a deep impression. As developers, it should always be part of our responsibility to ensure the safety of our users' private information. Therefore, it's necessary that we take serious consideration of it.

To begin with, we need to perform an in-depth analysis of vulnerabilities of web applications. According to the suggestion of the Open Web Application Security Project (OWASP)¹, there are 10 major concerns regarding the security of a web app as shown in the following table.

OWASP 2013 Top 10 List	
Injection	aws, such as SQL, OS, and LDAP injection occur when untrusted data is interpreted as part of a command or query. The attacker's hostile data can be interpreted into executing unintended commands or accessing data without proper authorization.
Authentication and Session Management	n functions related to authentication and session management are often implemented correctly, allowing attackers to compromise passwords, keys, or tokens, or to exploit other implementation flaws to assume other users' identities.
Site Scripting (XSS)	occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute code in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
Insecure Direct Object References	ect reference occurs when a developer exposes a reference to an internal application object, such as a file, directory, or database key. Without an access check or other protection, attackers can manipulate these references to access unauthorized data.

¹ OWASP 2013 TOP 10

https://www.owasp.org/index.php/Top_10_2013-Top_10

Security Misconfiguration	Security requires having a secure configuration defined and deployed for the system, frameworks, application server, web server, database server, and so on. Secure settings should be defined, implemented, and maintained, as they are often insecure. Additionally, software should be kept up to date.
Sensitive Data Exposure	Web applications do not properly protect sensitive data, such as credit cards, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
Function Level Access Control	Web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same control checks on the server when each function is accessed. If requests are forged, attackers will be able to forge requests in order to access functionality without proper authorization.
Cross-Site Request Forgery (CSRF)	An attacker forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to cause the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can result in serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.
Unprotected Redirects and Forwards	Applications frequently redirect and forward users to other pages and websites, using untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use redirects to access unauthorized pages.

Several of these concerns are encountered at especially high frequency in the daily maintenance of many web applications, such as SQL injection, Cross-Site Scripting and Cross-Site Request Forgery. To some extent, it can be said that any web application with user input is under the potential risk of being attacked as long as hackers find out a way to give some malicious input. Once those malicious contents get embedded on the web page, any user accessing the site will be put under the risk. Take the example of the famous “Samy worm”², which was once widely spread on the social-networking

² Samy worm

website MySpace. Even though the script written by Samy itself was relatively harmless and merely displayed one extra message on users' profile pages saying that "but most of all, samy is my hero", it succeeded in spreading at an incredibly fast rate and aroused people's attention. In the design of our application, it will be sure that we also allow user inputs since it's required that each user has a registered account in order to reserve a parking spot and make the purchase. As a result, it appears especially important how to protect our users' information from leakage throughout the process, and this may require a combination of several methods.

One basic strategy for partially solving this problem that will definitely be applied in the construction of our application is to have all sensitive information salted and hashed before storing it into the database. In other words, some random number will be generated and combined with user's original input and then hashed when handling with sensitive inputs like passwords. Meanwhile, the salt value will be stored into our database so that the authentication mechanism can be maintained using the hash of users' original input combined with their corresponding salt. Having sensitive information salted and hashed would greatly improve the security defense since it doesn't require the storage of the original input.

Another solution that may be used while building the application is to apply some existing well written templates such as angular.js or mustache. Many templates have already taken into consideration the processing of unsafe inputs and thus could provide better protection. As shown in the following screenshots, we attempted to inject some scripts into the page, which is a simple to-do application utilizing angular.js. However, the script tag didn't trigger the execution and was finally displayed as HTML source code on the page. To find out the things happening underneath, we used Chrome's developer tools to check the HTML code after the failed injection, which reveals that angular has actually escaped the unsafe characters with HTML entities. This can also be achieved by using the htmlspecialchars() function in php, which acts in a similar way. Nevertheless, there're still more advantages and convenience of using templates as

they manage to make the code appear more organized and neater. Angular.js also provides a service named Strict Contextual Escaping (SCE)³ that assists in writing code in way that is secure by default and makes auditing for security vulnerabilities much easier. This service is enabled by default and allows developers to make their own whitelists where they're free to define what contents are safe enough to be rendered as HTML, CSS or URL, etc.



```

▼ <ul>
  <!-- ngRepeat: todo in todos -->
  ▼ <li ng-repeat="todo in todos" class="list-unstyled ng-scope">
    <input type="checkbox" ng-model="todo.done" class="ng-pristine ng-valid">
    <span class="done-false">&lt;script&ampgtalert('error!');&lt;/script&ampgt</span>
  
```

A third approach to introduce is the utilization of some third-party APIs to fulfill the purchase process instead of keeping users' payment information such as credit card numbers on our own. The two APIs that we would possibly use are PayPal and Apple Pay, which are two of the nowadays most popular and well-known payment methods. Third-party APIs support a variety of credit card companies and are more experienced in taking care of users' payment information and thus the risk of sensitive data breach is avoided.

As indicated by OWASP's XSS (Cross Site Scripting) Prevention Cheat Sheet⁴, developers are also encouraged to escape untrusted data, encode HTML and JSON values, etc. However, some template systems are already well prepared with these

³ Angular.js SCE

[https://docs.angularjs.org/api/ng/service/\\$sce](https://docs.angularjs.org/api/ng/service/$sce)

⁴ OWASP XSS Cheat Sheet

https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet

features and require less efforts for developers. We believe that the security issue of our application would be largely relieved with a combination of salting and hashing, using templates and third-party APIs.

8.2 Group Research Paper 2

Research on Solving the Payment Processing Problem

Xueyan Wu

Introduction

The university of Missouri – Columbia is holding a great number of football games every year. Many people have to arrive at Columbia about 8 hours in advance in order to have a parking spot since the wave is not only from Columbia but also from other cities like St. Louis. Therefore, our client had a plan about renting some parking lots from the university and then let the customers to reserve the parking spot with our reserve system during game days.

Our group will build a web application called Reservelt, which allows users to make a reservation for a parking spot they prefer with purchase. Here comes the most important problem – how to process the customer payments with security. Multiple methods will be analyzed in this paper later.

Background

There are two types of payment method can be chosen - a true merchant account, or a third-party processor. A true merchant account is always used by the business online store. The merchant apply with a processing bank then obtain the dedicated account for the business. Basically, the merchant account is a direct account with Visa, Mastercard, American Express or Discover Card, so the merchant must abide by their

rules. The third-party processor like PayPal is a way for businesses to accept credit card payments through its own merchant account.⁵

Analysis

When our client brought up the idea that the reservation should be paid by credit cards, we realized it was not easy to implement it. First of all, the four major credit card groups require that all applicants are legally registered businesses. However, neither the client nor our group is a registered business during building the website. Second, our group does not have the ability to guarantee the security of the customers' credit card information. Therefore after the investigation, it is preferred to use the third-party processor to process the payment, which is more secure and easier.

PayPal is the method we brought up first. As the merchant, the client just has to handle the sales and PayPal will handle everything from the checkout process and ensure the security. Besides PayPal payment, credit cards can be used as well. The rate for "PayPal Standard" is (2.9% + \$0.30) per transaction.⁶ If the business wants to design his own checkout pages and let customers pay without leaving his website, "PayPal Pro" can be considered with extra \$30 monthly fee.

⁵ <http://www.sitepoint.com/merchant-account-review/>

⁶ <https://www.paypal.com/us/webapps/mpp/paypal-payments-standard>

Features	PayPal Payments Standard	PayPal Payments Pro
	Get Started	Apply Now
Monthly fee (See all merchant fees)	\$0	\$30
Design and host your own checkout pages for full control	✓	✓
Customers pay without ever leaving your website	✓	✓
Accept credit cards via phone, fax, and mail (Virtual Terminal)	✓	✓
Accept credit cards (your buyers don't need a PayPal account)	✓	✓
Accept PayPal payments	✓	✓
Send invoices online for fast payment	✓	✓
PayPal Here card swiper for your iPhone, iPad, or Android device	✓	✓
Get paid on eBay and Etsy	✓	✓

Comparison of PayPal Standard and Pro

Therefore, PayPal is a pretty good method that can be considered. And later our client suggests another payment provider called “2checkout” that allows companies to accept online and mobile payments from buyers worldwide. There is no monthly fees and setup fees and the rate of this provider in United States is (2.9%+30¢) per successful transaction , which is the same as PayPal Standard’s.

The setup procedure is pretty similar to PayPal. Once someone decides to start his business, he has to finish the application with providing the information about the business (industry category, product description, address, website URL, legal registration), company’s owners (contact and identification details) and business policies including Refund and Privacy statement.⁷

⁷ <https://www.2checkout.com/>

The screenshot shows the 2CHECKOUT Review Cart Page. At the top, there's a header with the ACME logo and a 'Settings' button. Below the header, a breadcrumb navigation bar shows the current step: 'Review Cart' (highlighted in green), followed by 'Shipping Info', 'Shipping', 'Billing Info', and 'Payment'. A progress indicator '1 / 5' is also present. The main content area displays a table of items:

Item	Unit Price	Quantity	Subtotal
ACME Product	\$14.79	1	\$14.79

Below the table, a note says: "Shipping: Price is determined by your shipping info below." To the right, it shows "Current Total (USD): \$14.79". A blue 'CONTINUE' button is centered at the bottom of the page. At the very bottom, a summary bar shows: Cart Summary, shipping \$ 0.00, all items \$14.79, and total \$14.79.

2CHECKOUT Review Cart Page

The screenshot shows the CHECKOUT Payment Page. The breadcrumb navigation bar is identical to the one above: 'Review Cart', 'Shipping Info', 'Shipping', 'Billing Info', and 'Payment' (highlighted in green). A progress indicator '5 / 5' is shown. The main content area has a heading 'Submit payment' and fields for card information: Credit/Debit Card Number (4242424242424242), Expiration Date (12/2018), and CVV Code (123). It also features a 'PayPal' payment method option. A green 'SUBMIT PAYMENT' button is at the bottom. At the very bottom, a summary bar shows: Cart Summary, shipping \$ 11.00, all items \$14.79, and total \$25.79.

CHECKOUT Payment Page

In fact, there is no big difference between these two methods. Since right now our group has no idea what the monthly volume will be, so the long-term cost of these

provides can't be calculated. What to think about is the accessibility and implementation of these methods. At the end, the client will pick one method from these two since their APIs are both online and not hard to implement.

8.3 Group Research Paper 3

How to implement Google Maps API into Reservalt

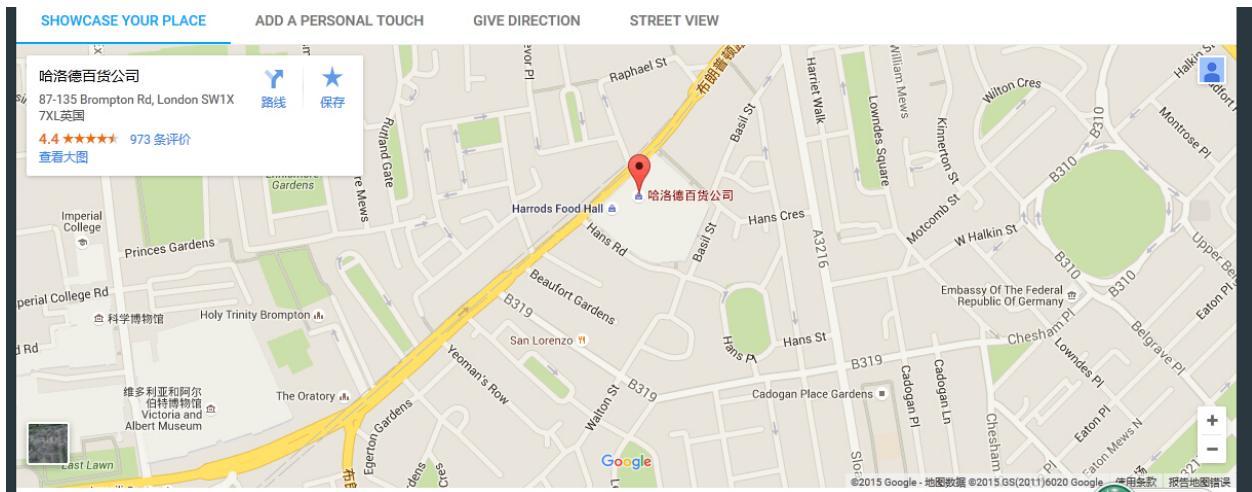
Chunyang Zhan

For this project, our group aims to build a reservation system for parking. In the system, our group supposes to be able to help users to locate where the parking spaces are and how they can get there. For this goal, it is a good choice to implement Google Maps API. Google Maps API allows the application to show a real map and a real-time navigation for our users. Moreover, our group needs to make sure our design can be used in either mobile devices or traditional desktops. To meet the functional requirements, our group needs to figure out how to implement Google Maps API into our specific pages.

To implement Google Maps API, some basic skills are needed, because Google Maps uses JavaScript extensively and the site also uses JSON for data transfer rather

than XML, for performance reasons⁸. Also, to finalize the navigation function, an API KEY may be required for loading and using information of millions of locations.

The first step is to add the Google Maps API into our site. There are two ways. One is to use Google Maps JavaScript API, another is to use Embed API. The Google Maps Embed API is a well-designed package, which helps users discover places using their own personalized map, tapping into Google's rich database of over 100 million places worldwide². By viewing the demo on Google Developer, I think it is not a proper way.



Although it is fine to show a specific location (figure 1), it is not

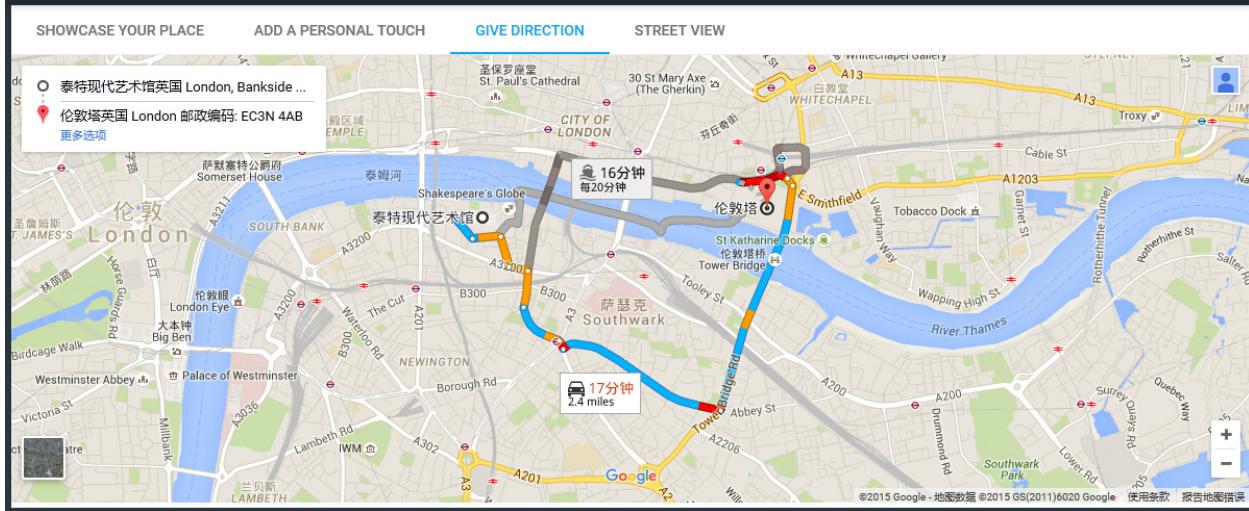
(Figure 1)⁹

convenient to show how to go to the location. When users click the direction button, it will redirect to the maps.google.com. This would give our users an inconvenient experience. There is an alternative way to show the direction by giving the origin and

⁸ https://en.wikipedia.org/wiki/Google_Maps#Google_Maps_API

⁹ <https://developers.google.com/maps/documentation/embed/>

destination(figure 2). However, if the users want to change either location, they need to go back and input them again or click “more opinion” to the google map



(Figure 2)¹⁰

site. Admittedly, this is an easy way to finalize but, I think, users may suffer an uncomfortable experience when they use this.

Another method is to use the JavaScript version which can perform more friendly to users but a little bit tough to code. The Google Maps JavaScript API can build a custom map for your site using styled maps, 3D buildings, indoor floor plans, multi-modal directions and more⁴. Since it uses JavaScript to function, it would be much easier to change locations if users need. In the demo code(figure 3) on Google Developer, the location can be changed by passing a value to the variable “myLatLng”.

¹⁰ <https://developers.google.com/maps/documentation/embed/>

The following example adds a simple marker to a map at Uluru, in the center of Australia:

```
function initMap() {
  var myLatLng = {lat: -25.363, lng: 131.044};

  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: 4,
    center: myLatLng
  });

  var marker = new google.maps.Marker({
    position: myLatLng,
    map: map,
    title: 'Hello World!'
  });
}
```

(Figure 3)¹¹

It is the same when the application perform the navigation function. So, an input box can be added allowing users update their location then the direction can be updated. This would be much similar with users do on the google map website. In this way, it can give users a better using experience comparing with the Embed API. Besides, there is a tutorial on w3school¹² which shows basic functions and UI of the API. This tutorial can help us develop our functions.

The problem with using JavaScript version is how to convert the location into geographic coordinate. Google provides a useful tool called The Google Maps Geocoding API. Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map.⁵ The service will sent requests and receive responses. Accessing

¹¹ <https://developers.google.com/maps/documentation/javascript/>

¹² <http://www.w3schools.com/googleapi/default.asp>

the Geocoding service is asynchronous, since the Google Maps API needs to make a call to an external server. For that reason, you need to pass a callback method to execute upon completion of the request. This callback method processes the result(s).

The Geocoding service requires a callback method to execute upon retrieval of the geocoder's results. Following are the fields that request and response contain.¹³

The `GeocoderRequest` object literal contains the following fields:

```
{  
  address: string,  
  location: LatLng,  
  placeId: string,  
  bounds: LatLngBounds,  
  componentRestrictions: GeocoderComponentRestrictions,  
  region: string  
}
```

Required parameters: You must supply one, and only one, of the following fields:

- `address` – The address which you want to geocode.
- `location` – The `LatLang` (or `LatLangLiteral`) for which you wish to obtain the closest, human-readable address. The geocoder performs a *reverse geocode*. See [Reverse Geocoding](#) for more information.
- `placeId` – The place ID of the place for which you wish to obtain the closest, human-readable address. The place ID is a unique identifier that can be used with other Google APIs. For example, you can use the `placeId` returned by the [Google Maps Roads API](#) to get the address for a snapped point. For more information about place IDs, see the [place ID overview](#). If you pass a `placeId`, the geocoder performs a *reverse geocode*. See [Reverse Geocoding](#) for more information.

¹³ <https://developers.google.com/maps/documentation/javascript/geocoding#GeocodingResponses>

The `GeocoderResult` object represents a single geocoding result. A geocode request may return multiple result objects:

```
results[]: {  
  types: string,  
  formatted_address: string,  
  address_components[]: {  
    short_name: string,  
    long_name: string,  
    postcode_localities[]: string,  
    types[]: string  
  },  
  partial_match: boolean,  
  place_id: string,  
  postcode_localities[]: string,  
  geometry: {  
    location: LatLng,  
    location_type: GeocoderLocationType  
    viewport: LatLngBounds,  
    bounds: LatLngBounds  
  }  
}
```

(figure 4 and 5)¹⁴

With these APIs and functions, the Google Maps API would work properly.

8.4 Group Research Paper 4

Research Paper about QR Code

Qintai Liu

Our group project aims to build a website for customers to reserve parking before Mizzou football game starts. Almost every football spectator has trouble in finding a park spot unless they present several hours in advance. Our client told our teammates that he would have the authority to charge every customer who parks in a specific open car park before football game starts.

¹⁴ <https://developers.google.com/maps/documentation/javascript/geocoding#GeocodingResponses>

The issue being discussed and explored is related to what procedures should have during a car enter our open park entrance. More specifically, there are several subquestions which need to be solved.

1. How to guarantee that only people who have already reserved parking can enter this open park and others can not do that?
2. How to track how many car in our open park? In other word, the system should have the ability to count how many cars enter the park and how many cars exit the park. The reason is that the client asked for having the function that the spectators could park their car in this open park for free if there are still available parking slots after the start of the game.
3. How to minimize our investment in achieving this goal to maximize our client's earnings.

Overview

Basically, all of the solution can be classified into two major categories — hiring employee or not hiring employ. And there are several different solutions within the category of not hiring employee.

Alternative one

The first solution to resolve this issue is that hiring an employee to help verify whether customers have reserved parking spot or not as well as count how many cars enter and exit the open park. More specifically, after each customer reserved a parking spot successfully, a unique string or number responding to this transaction will be created. This unique string serving as a primary key combines with this purchase information, which forms <key,value> pair. Then this pair will be inserted into database.

Additionally, customer will be provided with this string which is available to print out. When each customer drive at the open park entrance, the employee will check the validity of the string this customer provides by search this string in database. If the information associated with this string match today's game information, the employee allows this customer to park as well as updates the number of cars existing in the open park and the number of parking spots are available.

Alternative two

The rest of all solutions which would be discussed later are based on no hiring employee. The first resolution came up with about the certification of reserving is QR codes (abbreviated from Quick Response Code). A QR code consists of black modules (square dots) arranged in a square grid on a white background, which can be read by an imaging device (such as a camera, scanner, etc.)¹⁵

Like the previous method, after each customer makes an reservation successfully, the system will create a unique string or number related to each purchase information to form <key,value> which will be insert into database. The information about this purchase will include the purchasing date, the price, and reserving time for parking. Then the system will encode this string into QR code and include this QR code in receipt.

The customer can save this receipt locally and print it out. When the customers at the entrance of park, there will be a scanner or reader to read and decode the QR code into string. Then the system will query whether there is a purchase associated this string.¹⁶

¹⁵ https://en.wikipedia.org/wiki/QR_code

¹⁶ How to Use QR Codes as Mobile Boarding Passes, <http://www.farecompare.com/travel-advice/the-qr-code-for-airport-check-in-how-does-it-work/#/>

The most challenging points are how to encode and decode QR code, and use what device to scan it. After lots of searches are done online, there are several QR code generator online and also available for mobile, like Scan¹⁷, QR Code Generator¹⁸. At entry of open park, a mobile, which downloads responding App to scan the QR code and get the decoded string, will be placed. Finally using this string to verify the validity of purchase.

Alternative Three

Similar to alternative two, this solution still use QR codes to be the verification of purchase. The difference to alternative two is that the new method is more reliable and secure, which is implemented by well-developed check in application called Event Check-in¹⁹. There are several famous company, like Apple, ITV and Oracle²⁰, have already utilized this application to check in guests onsite and then analyze attendance data afterwards. Event Check-in Professional(ECi Pro) can generate a unique QR code for each of customers. These QR codes are encrypted (AES 256-bit) and contain no personal information. They cannot be used or read by any other system except for ECi Pro. What's more? They are one-time-use QR codes; once scanned, they cannot be re-used for other events. Additionally, Event Check-in Professional uses industrial-strength laser scanners from industry-leader SocketMobile. These lightweight devices have a long battery life. Compared to this method, the alternative two can only use mobile to be a scanner. Finally, ECi Pro provide the ability to see real-time data about how many and who arrived at the open parking.²¹

Apparently, there is a tradeoff to this amazing check-in application that the price is relatively more expensive than the alternative two. There are also several other well-

17 <https://scan.me/>

18 <http://www.qr-code-generator.com/>

19 <http://www.geteventcheckin.com/>

20 Clients, <http://www.geteventcheckin.com/event-check-in-pro>

21 *Id.*

developed application about online event registration like Event Espresso²² and Tickera²³.

Solution about vehicle counter

Basically, what our group think is that utilizing sensor counts vertices. As a vehicle enters the parking garage or lot it passes between a pair of sensors or over loop detectors that senses their direction of travel. When vehicles enter the parking garage, the number of open spaces is reduced by one for each vehicle entering. Conversely as each vehicle exits, the number of open parking spaces is increased by one²⁴.

In conclusion

The final method which be chosen is alternative three, which is achieved by well-developed check-in application. First, alternative one, which is hiring a person, will not be chosen. The reason is that even if some money is saved at the beginning of our project realized by alternative one, money is wasted in the end. At the begin, our group could avoid buying QR code generator or check-in application and sensors that count vehicles; however, the cost of hiring people would increases with the increasing numbers of football games which are held. The reason why our group prefer the alternative 3 to alternative 2 is that well-developed check-in application like Even Check-in(ECi Pro) is more reliable and secure. Not like alternative 2, unique string related to purchase information do not need to be created, because ECi Pro gains purchase information directly by scanning QR codes. Additionally, Eci Pro provide function that track how many vehicles enter the open park dynamically. What's more? ECi Pro can use the SocketMobile CHS-7xi scanner²⁵ to scan the QR codes, instead of only mobile or iPad.

22 <http://eventespresso.com/>

23 <https://tickera.com/>

24 RedStorm™ Parking Guidance System, http://www.signal-tech.com/products/parking/redstorm_parking_guidance_system

25 <https://www.socketmobile.com/>

9 Discussion

9.1 Transaction management - Payment buttons

Instead of using “Payments REST API”, a quicker and easier way for accepting payments is utilized, which is “PayPal Payments Standard - Payment buttons”. We use the PayPal Button Creation Tool to create the HTML code and then place it on our website, then we have the “Buy” button. Different combinations (event & parking lot) have different button code. However, in order to track the customer information(who has made this purchase) and verify whether the payment is successful, it is necessary to get the response message from the PayPal.

Solution: Instant Payment Notification(IPN)

Instant Payment Notification (IPN) is a message service that notifies you of events related to PayPal transactions. The service is used to automate back-office and administrative functions, including fulfilling orders, tracking customers, and providing status and other information related to transactions. According to the Figure 1, these actions prevent our website from accepting POST messages sent by third party.

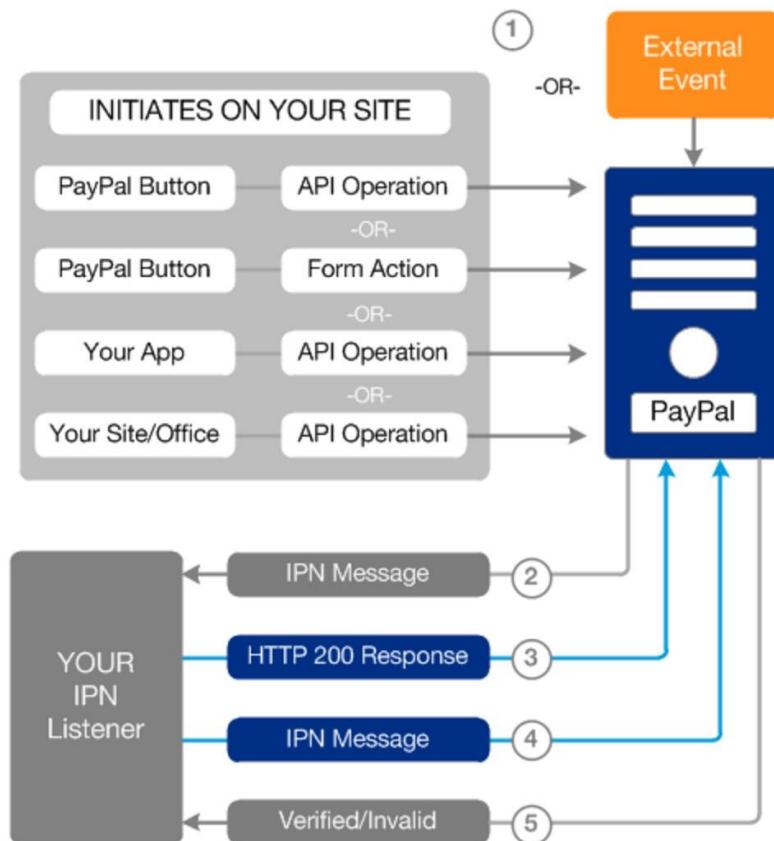


Figure 20, IPN Flow

The numbers in the diagram correspond to the following actions:

1. A user clicks a PayPal button to kick off a checkout flow; our website makes an API call; our back-office system makes an API call; or PayPal observes an event.
2. PayPal posts a message to our listener, notifying us of this event, which starts the request-response process.
3. Our listener returns an empty HTTP 200 response.
4. Our listener performs an HTTP POST to send the complete, unaltered notification back to PayPal, completing the initial request-response handshake, and allowing PayPal to verify that the IPN message is being sent to the correct location.
5. Note: This message must contain the same fields, in the same order, as the original notification, all preceded by cmd=_notify-validate. Further, this message must use the same encoding as the original.
6. PayPal sends a single word back - either VERIFIED (if the message matches the original) or INVALID (if the message does not match the original).²⁶

```
// Assign payment notification values to local variables
$item_name      = $_POST['item_name'];
$item_number    = $_POST['item_number'];
$payment_status = $_POST['payment_status'];
$payment_amount = $_POST['mc_gross'];
$payment_currency = $_POST['mc_currency'];
$txn_id         = $_POST['txn_id'];
$receiver_email = $_POST['receiver_email'];
$payer_email    = $_POST['payer_email'];
```

Figure 21 ,Extract variables from the notification for later processing

Figure 2 shows the variables should be got from the completed payment. Since the \$payer_email is the email address of the PayPal account rather than the one used to log in on our website. We use the code below to track the “logged in” email of on our website.

```
$(".paypal_form").submit(function( event ) {
    <?php $email=$_SESSION['Email'];?>
    $(this).find('input[name="custom"]').val("<?=$email?>");
});
```

9.2 Algorithm of how to determine whether permit is sold out

26 https://developer.paypal.com/docs/classic/ipn/gs_IPN/

Assumption:

1. Have multiple parking lots.
2. In the database, there will be a table named ParkNum with three attributes (ParkName, Date, avaNum). ParkName represents the name of the parking lots. The avaNum represents the number of available spots in that parking lot. The Date is used to distinguish the reservation for different events but at the same location.

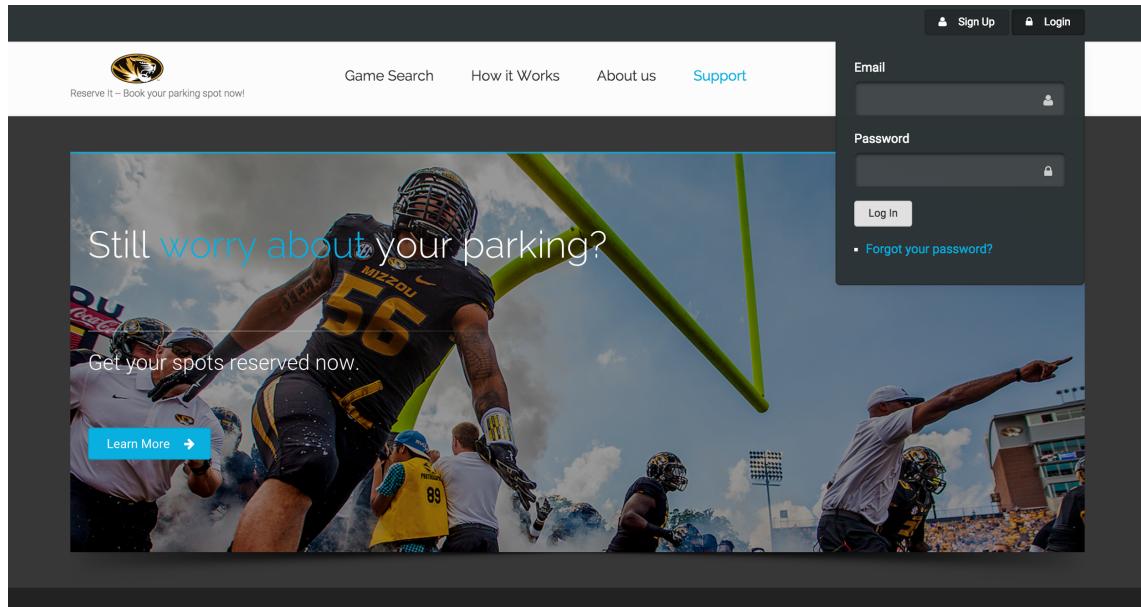
```
function demAvaLot(date){  
    query table ParkNum based on data to get avaNum and corresponding Parking  
    name.  
    Store avaNum and ParkName in associated array arr[‘ParkName’][‘avaNum’]  
    for each arr[‘ParkName’]{  
        if(arr[‘ParkName’]>0){  
            return true;  
        }else{  
            do nothing;  
        }  
    }  
    return false;  
}
```

10 Development Costs and Resource Needs

School funded / provided :

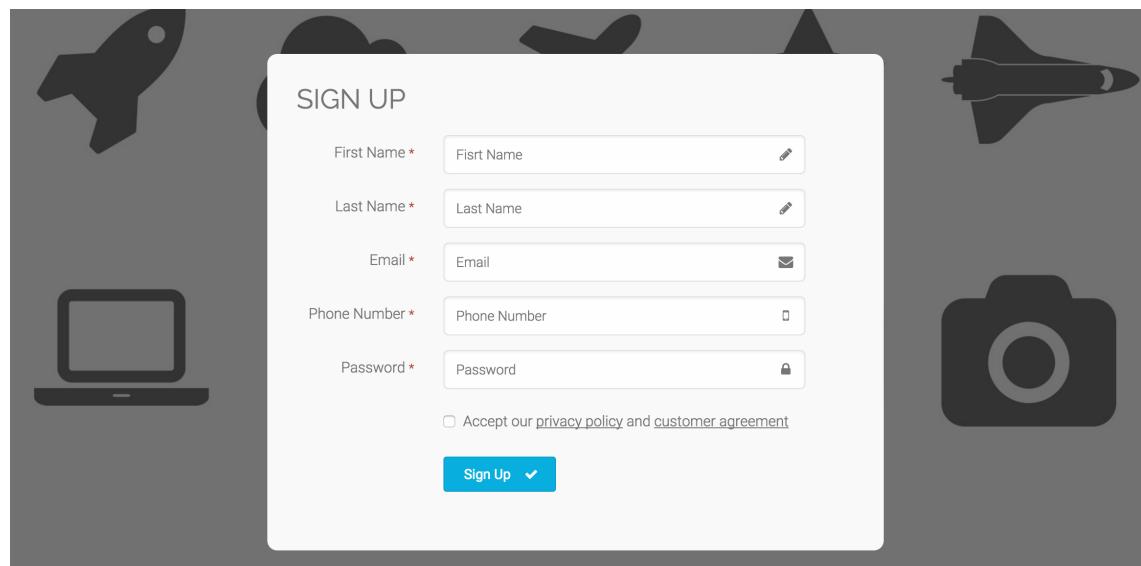
- Azure server
- Bootstrap for website application (\$15)

11 Results / Project Outcomes



(figure 22-1)

The homepage (figure 22-1) displays a brief introduction about our reservation system and the navigation bar above appears in each page prompts users to log in or sign up. If a registered user forgets the original password, the blue link below would direct the user to enter an email address used for registration to receive a reset link that is uniquely generated for this user. Once the link is clicked, the email address needs to be verified again so as to reset the password.



(figure 22-2)

The sign-up page (figure 22-2) mainly serves for user registration and has a form validation mechanism that only accepts appropriate user information.

SEARCH FOR A SPOT

Participant ▼

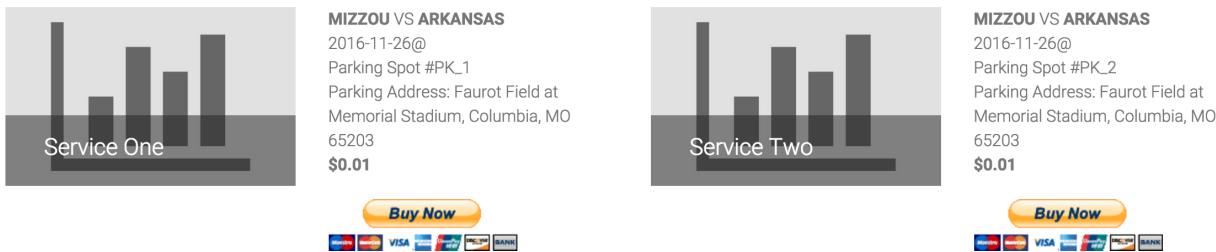
Pick the Date

May 2016						
Su	Mo	Tu	We	Th	Fr	Sa
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

(figure 22-3)

The search page (figure 22-3) is open for both our current users and unregistered guests with two searching fields supported - by participant, by date or a combination of both.

SEARCH RESULTS



New Search

(figure 22-4)

All fetched results will be displayed in the result page (figure 22-4) with detailed information about the upcoming games and available parking spots. The 'buy now' buttons for PayPal would

only show up after a user has signed in to enable the purchase, however, all website visitors have access to the information about search result.

MizzouGroup11

The screenshot shows a payment interface. On the left, a sidebar titled "Your order summary" displays a single item: "MIZZOU VS EASTERN MICHIGAN P1" with an amount of "\$0.01". It also shows the item number (3), price (\$0.01), and quantity (1). Below this, the "Item total" is listed as "\$0.01" and the "Total \$0.01 USD" is shown. A "PayPal Purchase Protection" section is present, with a link to "See details". A message encourages users to "Shop around the world with confidence". On the right, the main area is titled "Choose a way to pay". It features a "Pay with my PayPal account" section with fields for "Email" and "PayPal password", a checkbox for "This is a private computer.", and a "Log In" button. A "Forgot email or password?" link is provided. Another section, "Pay with a debit or credit card, or PayPal Credit", offers an optional link to "Join PayPal for faster future checkout". At the bottom, a link to "Cancel and return to MizzouGroup11." is visible.

(figure 22-5)

On click of the ‘Buy Now’ buttons, the page will redirect to the PayPal (figure 22-5) and require payment information to finish the purchase. Once the purchase is successful, our server gets a response from PayPal and generates a new order for the user. A confirmation email with detailed information and a QR code included is automatically sent to the email address used for registration for future verification.

This is helvetica bold.

NAME:	Yan W
GAME:	MIZZOU VS GEORGIA
DATE:	2016-09-17
LOCATION:	Faurot Field at Memorial Stadium, Columbia, MO 65203
PRICE:	\$0.01

This is helvetica bold.



(figure 22-6)

Figure 22-6 is a sample confirmation email that a user will receive after a successful purchase.

DASHBOARD

The screenshot shows a user's dashboard interface. On the left, there is a sidebar with two items: "User Profile" and "Order History". The "User Profile" item is currently selected, indicated by a blue background and white text. The main content area has a light gray background. At the top, it says "User Profile" with a person icon. Below that is a form with fields for "Email" (containing "xwu3@cougars.ccis.edu"), "First Name" (containing "Yan"), "Last Name" (containing "W"), and "Phone" (containing "5738180000"). There is also a "Change Password?" button with a magnifying glass icon. To the right of the password button, there is a dropdown arrow icon. Below the password fields, there are fields for "Current Password", "New Password", and "Confirm Password", each with a lock icon. At the bottom of the main content area is a blue "Update" button.

(figure 22-7)

User dashboard consists of three main functions - profile update, order review and navigation. Figure 22-7 is the part of profile update, which also supports password change with form validation.

DASHBOARD

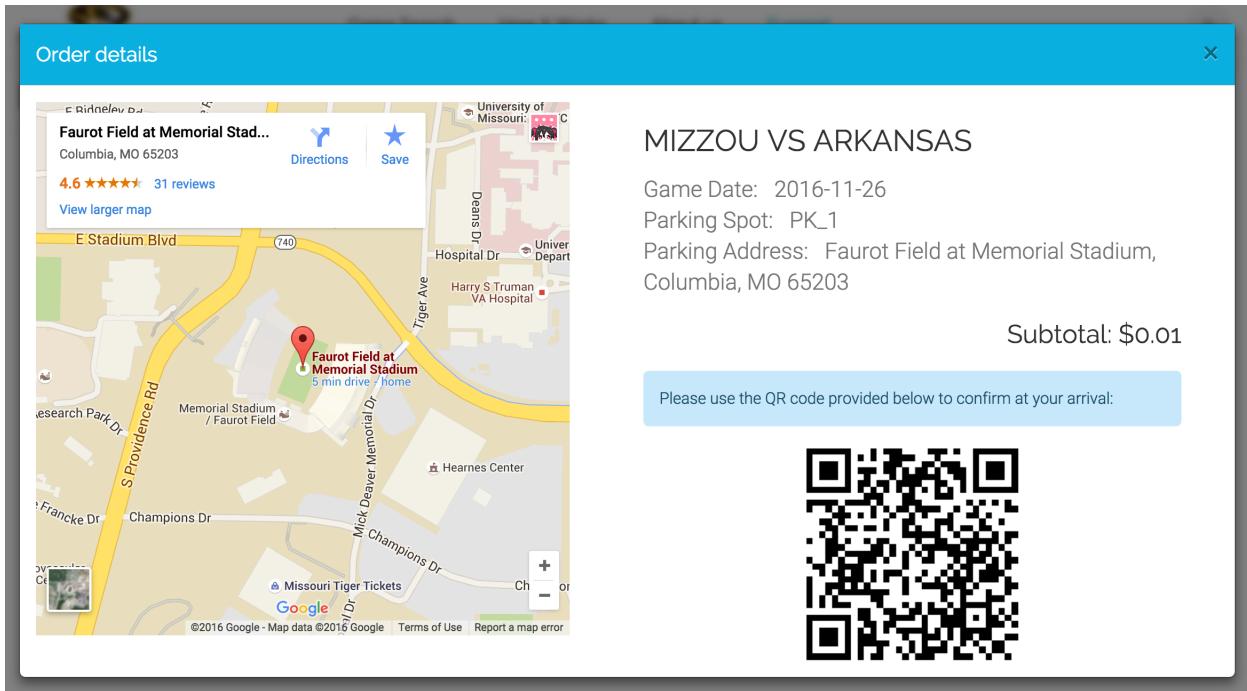
The screenshot shows the 'User Profile' section of a dashboard. On the left, there's a sidebar with 'User Profile' (selected) and 'Order History'. The main area is titled 'Order History' and contains a table with two entries:

Order Number	Status	Order Date	Detail
4G294244YX072132H	Completed	2016-04-26 01:42:29	View details
83C89712CW650530W	Completed	2016-04-26 01:48:24	View details

Below the table, it says 'Showing 1 to 2 of 2 entries'. At the bottom, there's a 'Read more' button.

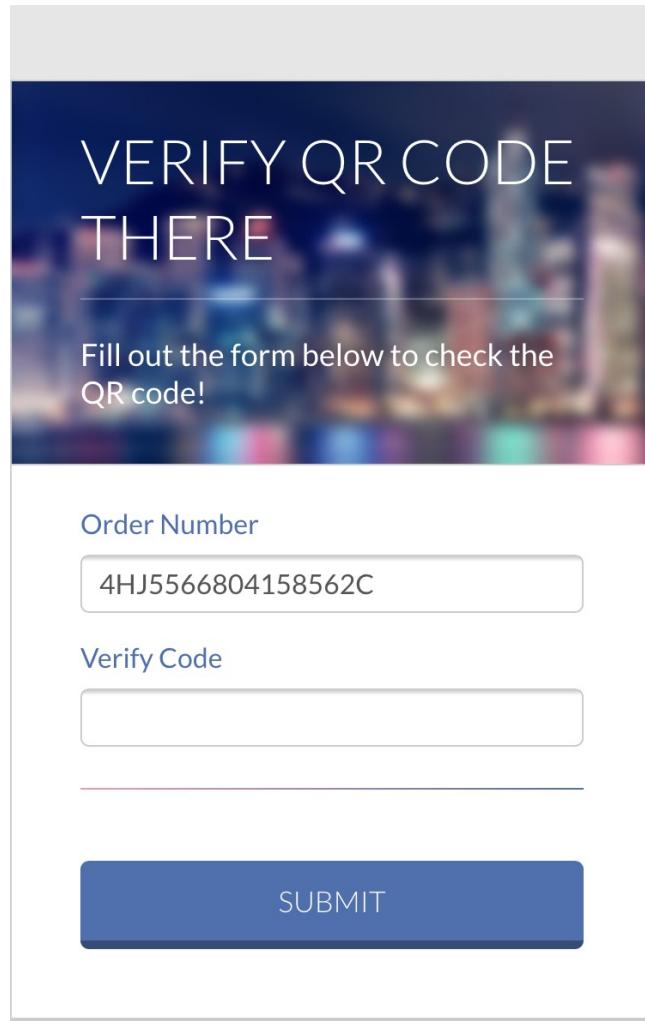
(figure 22-8)

The order history section (figure 22-8) allows user to search for all orders placed in the past with its status, order number and date. More detailed information about the game and parking lot will be shown only when the ‘view details’ button is clicked.



(figure 22-9)

Figure 22-9 is an example order details for a past order, which includes the name of both game participants, game date, parking lot location, QR code, subtotal and a map via Google Maps API. A user will get the navigation to the parking spot when hitting ‘Directions’ on the map.



(figure 22-10)

A QR code verification mechanism is utilized upon the user's arrival after a successful reservation. A user can also access this interface by scanning the code to check the order number, whereas a verify code below must be entered by a staff at the parking lot to activate the code and fulfill the confirmation of arrival.

Testing Result:

This document provides a description of the final testing, results and the specific functionalities that were taken into account.

I. Software

A. Performance

This test is performed to measure the scalability and performance of the web application. To test the software performance, the expected response time is less than 3 seconds. The

average load time across web browsers is 5.2841 seconds. According to www.webpagetest.org, the response time for different web browsers is described below.

						Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In
First View	7.243s	0.205s	1.986s	5494	1683	7.243s	77	3,443 KB	7.987s	78	3,445 KB
Repeat View	1.886s	0.150s	1.487s	2288	1683	1.886s	1	10 KB	1.886s	1	10 KB

Speed test in IE11

						Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In
First View	7.622s	0.234s	4.385s	7327	1529	7.622s	73	3,286 KB	8.042s	74	3,288 KB
Repeat View	4.789s	0.154s	2.786s	4780	1529	4.789s	1	10 KB	4.789s	1	10 KB

Speed test in Chrome

						Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In
First View	6.353s	0.219s	2.956s	5462	1529	6.353s	73	3,280 KB	6.353s	73	3,280 KB
Repeat View	2.503s	0.201s	2.439s	3032	1529	2.503s	1	10 KB	2.503s	1	10 KB

Speed test in Firefox

						Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In
First View	8.380s	0.250s	2.722s	6632	1529	8.380s	78	3,415 KB	8.380s	78	3,415 KB
Repeat View	3.497s	0.200s	3.499s	4268	1529	3.497s	1	10 KB	3.497s	1	10 KB

Speed test in Safari

						Document Complete			Fully Loaded		
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In
First View	6.301s	0.292s	5.554s	6798	1529	6.301s	77	3,087 KB	6.247s	77	3,087 KB
Repeat View	1.355s	0.330s	4.020s	4176	1529	1.355s	1	10 KB	0.624s	1	10 KB

Speed test in iphone 6 iOS9

B. Compatibility

This test was performed on the platforms and Internet browsers mentioned below.

- Mac OS X 10.11 Chrome 22

- Mac OS X 10.11 Safari 9.0.1
- Mac OS X 10.11 Firefox 45.02
- Windows 7 IE11
- Android Sony Xperia Z3 Android Browser 4.4
- Android Sony Xperia Z3 Android Chrome 49.0
- iPhone 6 iOS 9.3.1 Safari

C. Reliability

Since the web app resides on the Azure server, the web app will be subject to the availability of the server.

II. Implementation

A. Performance

To get the result for this area, these are the aspects that were tested

- Response time
- Web behavior with concurrent users
- Web behavior when loading
- Simultaneous connection to the database

B. Usability

To measure the usability of the web. We follow the next criteria to measure the user's experience while navigating through the web.

- Responsive GUI for both mobile devices and laptops.
- Web is user friendly with clear instructions.
- Web has a navigation bar in each page
- Subjective user satisfaction
- Testing web appearance, easy to understand, images placed with proper size The web is user friendly enough so a novice can utilize it without instructions.

C. Functionality

Test all links:

- Testing internal Links
- Testing external Links
- Testing sending receipt / email
- Testing reset password email
- The issues we found after this test are:
- User should not be able to access dashboard page without login

Test all forms:

- Field Validation
- Error message for wrong input when register / update profile
- Optional and mandatory fields
- Minimum requirement for user password
- Post and Get (submitting and getting information from user) work properly and safely.
- The issues we found after this test are:
Password not strong enough should not be allowed to be used for registration.

Testing Database:

- Database queries are executed correctly.
- Database insert, update, delete correctly.
- Database is retrieved correctly.
- Database need manually update match information each year. No other issues were found after performing this test

Cookies:

- Enable or disable cookies.
- Session cookies (expired after log out or expiration date)
- No issues were found after performing these tests

Test Register/Sign in/Log out:

- Same email can not be used to create account more than once.
- The user can receive and reset password successfully when forgets password.
- No issues were found after performing these tests

Test QR Code:

- Make sure the QR code can be generated successfully.
- The QR code can be verified by scan.
- Can retrieve the information of the ticket by scanning the code.
- The QR code can be scanned only by phone, other devices can not.
- No other issues were found after performing these tests

Test Dashboard:

- Users can update their profile and password properly.
- Navigation page can show corresponding event location correctly.
- Navigation page can show correct route from user current or input address to the parking spot.
- No issues were found after performing these tests

Game Search:

- · The page can return the accurate result efficiently.
- · The page will not return the result with past event.
- No issues were found after performing these tests

Check Out:

- · Successfully turn into payment pages
- · The process can update reservation info into the database
- · Immediate update on all web pages
- No issues were found after performing these tests

Payment:

- The payment can be finished successfully
- Testing returning information of the payment can be accessed
- Testing confirmation email can be sent immediately
- The QR code can be generated and sent through email after the payment
- No issues were found after performing these tests

Security:

- Testing unauthorized access to user specific pages
- Testing SQL injection to all pages include forms
- Testing session management
- Testing Cross-Site Scripting (XSS)
- Salted and hashed passwords
- Testing secure connection with Paypal
- No issues were found after performing these tests

12 Future Work

In the future, we would add a function for administrator to update game information through their account. And we will make the reservation can be canceled and refundable. Besides, instead of verify the QR code manually, Security Equipment Car Parking Access Control System with Car Parking Barrier can be setup to let customers scan their QR code by themselves.

13 References

1. OWASP 2013 TOP 10 https://www.owasp.org/index.php/Top_10_2013-Top_10
2. Samy worm [https://en.wikipedia.org/wiki/Samy_\(computer_worm\)](https://en.wikipedia.org/wiki/Samy_(computer_worm))
3. Angular.js SCE [https://docs.angularjs.org/api/ng/service/\\$sce](https://docs.angularjs.org/api/ng/service/$sce)
4. OWASP XSS Cheat Sheet https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet
5. <http://www.sitepoint.com/merchant-account-review/>
6. <https://www.paypal.com/us/webapps/mpp/paypal-payments-standard>
7. <https://www.2checkout.com/>
8. https://en.wikipedia.org/wiki/Google_Maps#Google_Maps_API
9. <https://developers.google.com/maps/documentation/embed/>
10. <https://developers.google.com/maps/documentation/javascript/>
11. <https://developers.google.com/maps/documentation/embed/>
12. <http://www.w3schools.com/googleapi/default.asp>
13. <https://developers.google.com/maps/documentation/javascript/geocoding#GeocodingResponses>
14. <https://developers.google.com/maps/documentation/javascript/geocoding#GeocodingResponses>
15. https://en.wikipedia.org/wiki/QR_code
16. How to Use QR Codes as Mobile Boarding Passes, <http://www.farecompare.com/travel-advice/the-qr-code-for-airport-check-in-how-does-it-work/#/>

17. <https://scan.me/>
18. <http://www.qr-code-generator.com/>
19. <http://www.geteventcheckin.com/>
20. Clients, <http://www.geteventcheckin.com/event-check-in-pro>
21. Clients, <http://www.geteventcheckin.com/event-check-in-pro>
22. <http://eventespresso.com/>
23. <https://tickera.com/>
24. RedStorm™ Parking Guidance System, http://www.signal-tech.com/products/parking/redstorm_parking_guidance_system
25. <https://www.socketmobile.com/>
26. https://developer.paypal.com/docs/classic/ipn/gs_IPN/

14 Appendices

14.1 Team information:

- “You only get to live once, so you’d better do what you enjoy.”
What we are chasing for is the enjoyment of coding and our group name is “Mad Potatoes”.
- What we will build is a parking reservation web application.
- Group members:



Chunyang Zhan
Senior
BS Computer Science

I'm super senior in CS major and math minor. I work in soykb lab of agriculture department with experience using C, HTML, PHP, JS, SQL etc.. I enjoy trying any new idea. I like games and ,of course, food.

In the project, I mainly take charge of database design, Google Maps API, testing and some other stuffs.



Qintai Liu
BS Computer Science

I am senior student in CS. I am proficient in developing website like html, javascript and server side like php and database design. I like basketball and watching movies.

I am taking participate in Undergraduate Student Honor Search in this semester called Digital Biology Laboratory. I focus on developing website.

In this project, I am responsible for developing QR code and generate e-ticket. Also I developed how to verify e-ticket. I also figured out how to send e-ticket to customer. Additionally, I succeeded in setting up paypal business account and accomplished how to let customer pay fee through paypal by paypal button. And I learned how to utilize Instant Payment Notification (IPN) which is a message service that automatically notifies merchants of events related to PayPal transactions. I used IPN to update our database and tracked the status of payment.



Xueyan Wu

Major BS Computer Science
Minor in Mathematics

I am currently a member of Digital Biology Laboratory and focusing on website building of showing description of gene products across databases and gene analyses.

I especially enjoy working on the front and back end of web development, which is exactly what I am doing right now.

Skills: C, PHP, Java , HTML, JavaScript, Database Management

In this project, I'm in charge of database design & modification and data collection & analysis. At the beginning, I implemented the functionalities of most UI including login, sign up, forgot password and game search. Later I was working on user dashboard with Yihua, PayPal buttons with Qintai. Working with them is pretty fun and challenging for me, which makes me also learn a lot.



Yihua Shi

Senior in CS major with a Math minor

I enjoy graphic designs, games & horror movies in my spare time. I've been doing undergraduate research about Subcellular Localization in Digital Biology Lab since 2015.

Have experience with C, JAVA, HTML, JS, CSS, PHP, MYSQL, Postgres and quite familiar with Bootstrap and jQuery. More experienced in web development and database design.

My main responsibility for this project is UI adjustment to a uniform and desired style. I also helped improving search page with jQuery plugins, integration of backend data and front end design for order details. I have utilized the Google Maps API in the display of order details in dashboard and added form validation for pages with form submission. I was also involved in minor bug fixing.

The development process was difficult yet rewarding, it convinced all of our team members that we could accomplish any goal when we set our mind to it. The problems we've encountered and the corresponding solving techniques that we've developed help bringing us to a higher level.

14.2 Acronyms and terminology

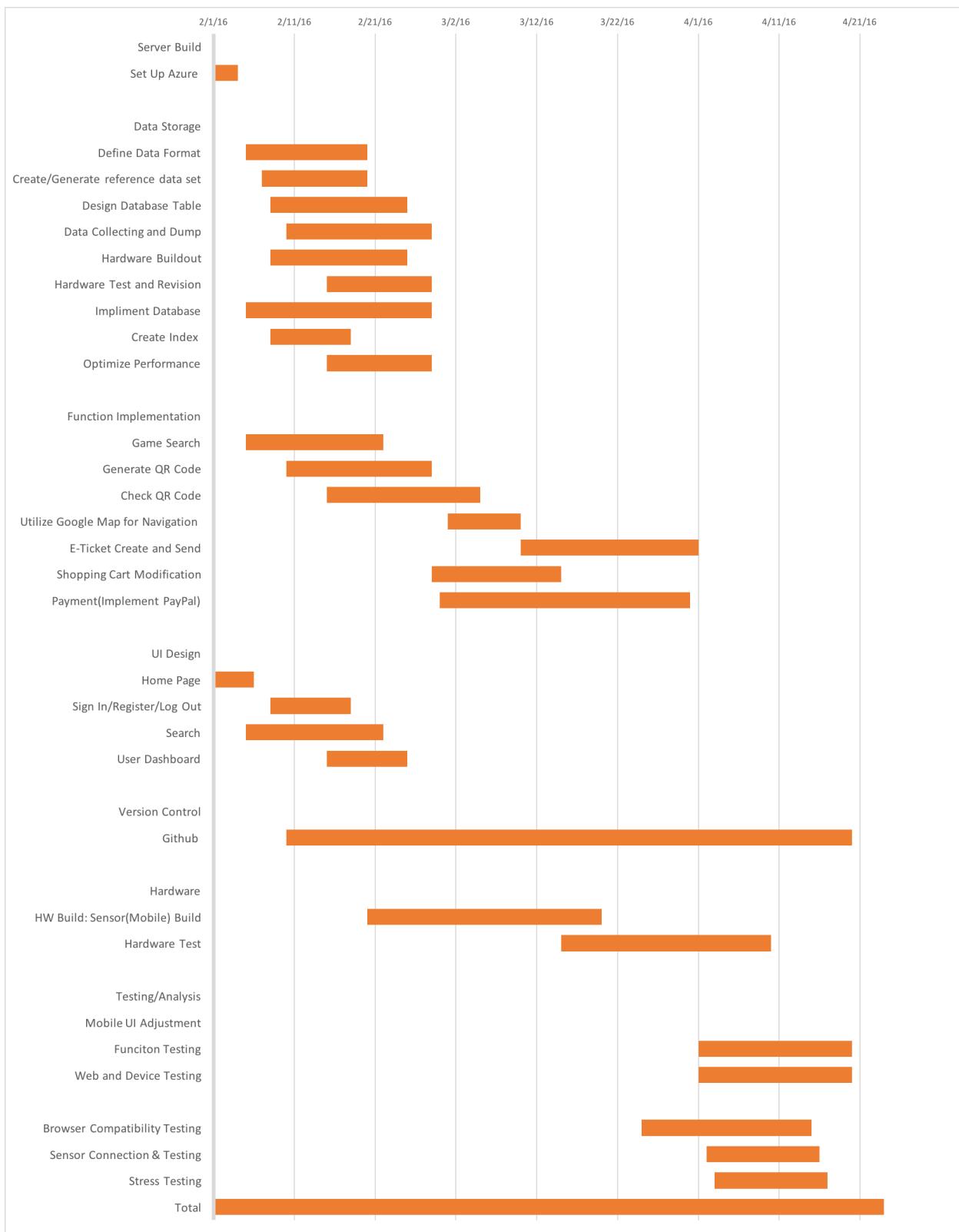
- URL : Uniform Resource Identifier, in HTTP/HTTPPs protocol means “Web address”.
- QR code: Quick Response code.
- API: Application programming interface
- REST: Representational state transfer
- HTTP 200 response: Standard response for successful HTTP requests.
- Back office: The back office is generally considered to be the technology, services and human resources required to manage a company itself.
- HTTP POST: In computing, POST is one of many request methods supported by the HTTP protocol used by the World Wide Web.
- ERD: Entity relationship diagrams, used to describe the database.
- GUI: Graphic user interface.
- Use case: In software and systems engineering, a use case is a list of actions or event steps.
- SQL Injection: SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution.

14.3 Progress report

Tasks	Start Date	End Date	Duration	Assignee
Server Build				
Set Up Azure	16/2/1	16/2/4	3	Zhan
Data Storage				

Define Data Format	16/2/5	16/2/20	15	Zhan
Create/Generate reference data set	16/2/7	16/2/20	13	Zhan
Design Database Table	16/2/8	16/2/25	17	Team
Data Collecting and Dump	16/2/10	16/2/28	18	Team
Hardware Buildout	16/2/8	16/2/25	17	Zhan
Hardware Test and Revision	16/2/15	16/2/28	13	Zhan
Implement Database	16/2/5	16/2/28	23	Team
Create Index	16/2/8	16/2/18	10	Qintai
Optimize Performance	16/2/15	16/2/28	13	Team
Function Implementation				
Game Search	16/2/5	16/2/22	17	Yihua/Yan
Generate QR Code	16/2/10	16/2/28	18	Qintai
Check QR Code	16/2/15	16/3/5	19	Qintai
Utilize Google Map for Navigation	16/3/1	16/3/10	9	Zhan
E-Ticket Create and Send	16/3/10	16/4/1	22	Qintai
Shopping Cart Modification	16/2/28	16/3/15	16	Yihua
Payment(Implement PayPal)	16/2/29	16/3/31	31	Zhan/Yan
UI Design				
Home Page	16/2/1	16/2/6	5	Yihua
Sign In/Register/Log Out	16/2/8	16/2/18	10	Yihua/Yan
Search	16/2/5	16/2/22	17	Yihua/Yan
User Dashboard	16/2/15	16/2/25	10	Yihua/Yan
Version Control				
Github	16/2/10	16/4/20	70	Team

Hardware				
HW Build: Sensor(Mobile) Build	16/2/20	16/3/20	29	Qintai
Hardware Test	16/3/15	16/4/10	26	Qintai
Testing/Analysis				
Mobile UI Adjustment	16/2/5	16/3/5		Yihua
Funciton Testing	16/4/1	16/4/20	19	Team
Web and Device Testing	16/4/1	16/4/20	19	Team
Browser Compatibility Testing	16/3/25	16/4/15	21	Team
Sensor Connection & Testing	16/4/2	16/4/16	14	Team
Stress Testing	16/4/3	16/4/17	14	Team
Total	16/2/1	16/4/24	83	Team



14.4 Documentation, tutorials, API's

- Google Maps Embed API
- Paypal API
- Google Maps Tutorial: <http://developers.google.com/>
- Paypal API Tutorial: <https://developer.paypal.com/docs/integration/web/>

14.5 Event Logs

- Database is modified based on functional and nonfunctional requirements.
- Shopping cart removed for convenience and real process.
- Navigation merged into order history with the Google Maps API which shows location of the destination.
- Remove ApplePay because of the difficulty of implementing the API.
- Give up using sensor because of the funds and hardware limitation.

14.5 Difficulties and attempted solutions

When we tried to use the Paypal API, we met some difficulties using the API directly. So we tried some alternative ways. We finally find Paypal API buttons is a easier way for our functions. The button can help us interact to the Paypal. After the payment is done, we can get some returning values from Paypal, which includes Payment information and can help us finish reservations and generate QR code. The result shows below:



(figure 22-4)

All fetched results will be displayed in the result page (figure 22-4) with detailed information about the upcoming games and available parking spots. The ‘buy now’ buttons for PayPal would only show up after a user has signed in to enable the purchase, however, all website visitors have access to the information about search result.

MizzouGroup11

The screenshot shows a PayPal payment interface. On the left, there's a summary of the purchase:

Your order summary	
Descriptions	Amount
MIZZOU VS EASTERN MICHIGAN P1	\$0.01
Item number: 3	
Item price: \$0.01	
Quantity: 1	
Item total	\$0.01
Total \$0.01 USD	

Below the summary, there's a section for **PayPal Purchase Protection**, which is optional for eligible purchases. It also mentions that the user can shop around the world with confidence.

The main right section is titled **Choose a way to pay**. It features a **Pay with my PayPal account** option, which requires logging in. There's a checkbox for marking it as a private computer. Below this is a **Log In** button and a link for forgot email or password.

At the bottom, there's another option for **Pay with a debit or credit card, or PayPal Credit**, which is optional for faster future checkout. A link to cancel and return to the previous page is also present.

(figure 22-5)

On click of the ‘Buy Now’ buttons, the page will redirect to the PayPal (figure 22-5) and require payment information to finish the purchase. Once the purchase is successful, our server gets a response from PayPal and generates a new order for the user. A confirmation email with detailed information and a QR code included is automatically sent to the email address used for registration for future verification.

This is helvetica bold.

NAME:	Yan W
GAME:	MIZZOU VS GEORGIA
DATE:	2016-09-17
LOCATION:	Faurot Field at Memorial Stadium, Columbia, MO 65203
PRICE:	\$0.01

This is helvetica bold.



(figure 22-6)