

A Parking Sharing System

Design Document

Team Number: 20

Client & Advisor: Dr. Ahmed Kamal

Team Members:

Jeremy Galang

Gabrielle Johnston

Jorden Lee

Ethan McGill

Jason Neville

Lorenzo Zenitsky

Team Email: sddec20-20@iastate.edu

Team Website: <http://sddec20-20.sd.ece.iastate.edu/>

Executive Summary

Development Standards & Practices Used

- Agile project development

Summary of Requirements

- Our application will require a dedicated frontend and backend that supports iOS and Android applications. We will need a platform that would host our server 24/7 once we have the application up and running. The apps will need to be published on the Google Play and Apple App Store so that anyone could download our application.

Applicable Courses from Iowa State University Curriculum

- COMS 227 + 228 (Object-Oriented Programming and Data Structures)
 - Knowledge of object-oriented programming and data structures are essential, and will most likely be utilized.
- COMS 252 (Linux Operating Systems)
 - It could be useful if we're working with Unix systems or a Linux server.
- COMS 309 (Software Development)
 - Knowing how to research unfamiliar technologies would be very applicable to this project.
 - Utilizing Agile development practices.
- COMS 311 (Algorithms)
 - Knowledge of algorithms would be beneficial for our application's performance.
- SE 319 (Construction of User Interfaces)
 - Teaches basic of how to design an attractive and useable user interface through a variety of languages and tools
- COMS 362 (Object-Oriented Analysis and Design)
 - Some project management techniques would be useful such as CRC cards and knowledge of UML. This would be helpful for planning.
- COMS 363 (Introduction to Database Management Systems)
 - Experimented with SQL and other databases

New Skills/Knowledge acquired that was not taught in courses

- How to build an iOS app (Swift)
- Using Google APIs and Android SDK for Android development
- (We will put the backend technologies here once we finalize it)

Table of Contents

1 Introduction	4
1.1 Acknowledgment	4
1.2 Problem and Project Statement	4
1.3 Operational Environment	4
1.4 Requirements	4
1.5 Intended Users and Uses	5
1.6 Assumptions and Limitations	6
1.7 Expected End Product and Deliverables	6
2 Specifications and Analysis	6
2.1 Proposed Approach	6
2.2 Design Analysis	7
2.3 Development Process	7
2.4 Conceptual Sketch	7
(Everything past here is tentative, per Canvas)	9
3 Statement of Work	9
3.1 Previous Work and Literature	9
3.2 Technology Considerations	9
3.3 Task Decomposition	9
3.4 Possible Risks and Risk Management	9
3.5 Project Proposed Milestones and Evaluation Criteria	9
3.6 Project Tracking Procedures	9
3.7 Expected Results and Validation	9
4 Project Timeline, Estimated Resources, and Challenges	10
4.1 Project Timeline	10
4.2 Feasibility Assessment	10
4.3 Personnel Effort Requirements	10
4.4 Other Resource Requirements	10
4.5 Financial Requirements	10
5 Testing and Implementation	11
5.1 Interface Specifications	11
5.2 Hardware and software	11
5.3 Functional Testing	11
5.4 Non-Functional Testing	11

5.5 Process	11
5.6 Results	11
6 Closing Material	12
6.1 Conclusion	12
6.2 References	12
6.3 Appendices	12

1 Introduction

1.1 Acknowledgment

Outside of our own team members, Dr. Ahmed Kamal, the client who proposed this project, has been, and will continue to be a significant contributor to the overall lifecycle of the project.

1.2 Problem and Project Statement

The problem that our project hopes to solve is event parking. In large events, such as Iowa State football games or the Iowa State Fair, designated parking can be limited, expensive, and overall a hassle.

The Parking Sharing System aims to provide an alternative for finding parking at busy events. This application will enable two kinds of users: to either find parking spaces, or host parking spaces. The host will be able to post a parking spot to the application. They will provide an address, size of space, time frame the space is available, and a price for that time. The guest will be able to search through the existing posts to find a spot close to the event they want to go through. All transactions will be processed through the application. Similar to ridesharing apps such as Uber and Lyft, this application is crowdsourced and will heavily rely on individuals willing to host and find parking spaces. We hope to create an application that makes event parking easier, cheaper, and overall better client experience.

1.3 Operational Environment

The application will function at its best in large cities or local events such as concerts, football games, etc. It is not limited to specific events and large cities, but it is available wherever users can host and find available parking. The application will also need a large user base to make the product viable. Urban, rural and suburban areas are suitable areas for this application.

1.4 Requirements

The application will require two types of users:

- Individuals or groups who are willing to host parking spaces.
- Individuals looking for parking.

A Parking Sharing System will heavily rely on its user base in order to function. In some areas, the demand for hosts/host-seekers can vary. As long as there is a consistent ratio of hosts and host-seekers, the application should eventually have an increase in users.

For functional requirements, the application will need to be available on Apple's App Store and the Google Play store. Both of these stores charge a fee in order for the application to be published. On top of that, both stores will test and approve our application if it is allowed to be published. An in-app payment system will be required in order for users to pay and receive money. Users will also need to list and reserve parking spots.

From an economic standpoint, the project will have to include the functionality to carry out monetary transactions for the users of the two mobile applications. To facilitate these transactions, all of which cost developers a tiny fee per, we will need to integrate with popular payment options like PayPal. Also, the integration of software like Google Maps will eventually cost us a small fee to use because more often than not, companies charge based on the number of calls to their APIs. Second, the cost of publishing our app to the Google Play Store for Android devices includes a one-time fee of \$25. On the other hand, Apple charges a yearly fee of \$299 for a group of developers to publish their app to their App Store. Lastly, we will be using servers for hosting purposes. For now, we are only concerned with hosting our product in Iowa, so our server costs will be next to nothing. However, if we ever choose to expand our scope to accompany other regions of the country and/or the world, we will be subjected to much higher server costs.

1.5 Intended Users and Uses

The intended users include, but not limited to individuals who are:

- Guests that will be seeking parking spaces
- Hosts that are Willing to rent out parking spaces on their property
- Admins that will authenticate hosts and moderate posted parking spots

The intended uses include, but are not limited to:

- Making listings for parking spaces
- Making payments
- Reserving parking spaces
- A feedback system
- Proof of availability (picture)
- Authentication of host

1.6 Assumptions and Limitations

One of the more prevalent limitations when it comes to mobile applications of this nature includes regional support. The more popular parking apps tend to serve only a specific state or region of the US whereas a majority of the parking apps akin to what we are trying to build tend to try and support all users from any location. More often than not, a user will download a residential parking app only to find out that it features little to no support in their area because they may not live in a busy and well-populated city like Los Angeles, New York City, or Boston.

1.7 Expected End Product and Deliverables

The end product should result in a functional iOS and Android application that has a simple UI that users can understand. We will also deliver a working backend environment with associated API endpoints.

2 Specifications and Analysis

2.1 Proposed Approach

As stated in the problem statement when attending events it can be very difficult to find parking and at a reasonable rate. After meeting with the client, the team received the research paper that this idea came from. The only stipulation was that this project would be 100% software. A full list of requirements is available in section 1.4.

See also section 3.2 for more details on the proposed technologies we will be using.

2.2 Design Analysis

- Frontend:
 - **Android:** The team has chosen to use Java with Android SDK for developing the Android application for the project because it is one of the Google supported languages for Android.
 - **iOS:** The team has chosen to build the iOS portion of the project in Swift as opposed to Objective-C, the older language alternative supported by Apple for developers to write and publish apps to their app store. Objective-C is a significantly lower-level language in the same vein as C when compared to the modern look and style of Swift, a language more reminiscent of JavaScript and Python than its lower-level ancestor.

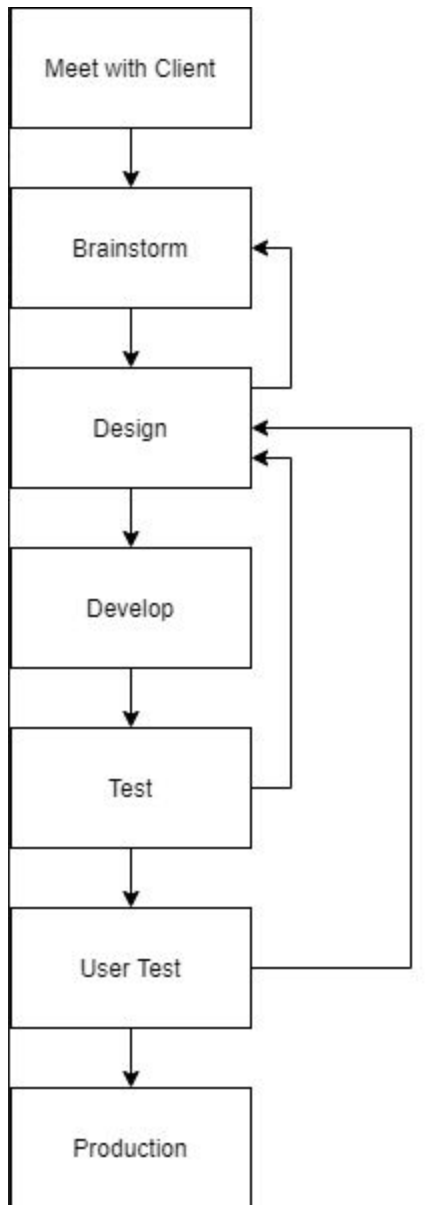
- Backend:
 - **Node w/ Express:** The team is considering NodeJs because it is a very popular backend language to run on servers. There are plenty of resources for team members who are unfamiliar with Node and the Express library.
 - **Database:** The team is discussing the use of MySQL, MongoDB, or PostgreSQL because each of these database's can easily be connected to the server-side code using express. Members of the team have experience working with MySQL and MongoDB.

2.3 Development Process

Our group has chosen to follow the Agile methodology. Routine meetings and frequent testing will be implemented for the application. This also includes frequent collaboration with our client to ensure that we are meeting his needs and the timeline that he has set forth.

2.4 Conceptual Sketch

This is a rough idea of what our proposed plan of action for this project is. However we will most likely make changes to this in the future as we are still finalizing how we are going to approach this project.



(Everything past here is tentative, per Canvas)

Our group cannot fill out some of the sections because we are still finalizing and researching technologies to narrow down for our project. We are concerned that we may need to either change or add new technologies as we work on the project. Our advisor/client has been away for two weeks on business trips and we have not had a chance to talk more in depth with him about some aspects of our project. We are in the process of finalizing what technologies we are using, and once we find that out we can figure out what we can use for testing and other requirements.

3 Statement of Work

3.1 Previous Work and Literature

Since all of us have taken COMS 309, we find that class will be incredibly useful for this project. Having done research on technologies not traditionally used in our classes will prove to be beneficial in this project. The resources we used in that course may also help us learn the new technologies, such as W3Schools.

3.2 Technology Considerations

Since the project will be made up of two separate mobile applications (iOS and Android), there will, of course, be two uniquely different frontend interfaces. The iOS frontend application will be written in Swift by members of the team who own an Apple device. The Android frontend application will be written in Java with an Android SDK.

Both front end applications will be connected to a single backend hosted on a web server. The team has not committed to one language, but plan on using some variant of Javascript that is good for a large number of concurrent users. The backend will also be connected to a database to securely store information needed for our application. Like the backend technology, the team has not committed to a single database language, although technologies like MySQL, MongoDB, Postgresql are all being considered.

3.3 Task Decomposition

3.4 Possible Risks and Risk Management

A few risks include:

- Privately securing user payment information
- Location sharing
- Liability of having vehicles parked for longer periods of time than allocated.

3.5 Project Proposed Milestones and Evaluation Criteria

One milestone will be the prototype availability during the football season, but everything else is TBD

3.6 Project Tracking Procedures

(Will most likely be along the lines of something dealing with Agile)

3.7 Expected Results and Validation

Our application must be able to:

- Successfully allow a guest to download the app to their mobile phone.
- Allow said guest to either host or find nearby parking spaces via the app.
- Provide a reservation to the host saying the guest wishes to park at their location
- Allow the host to verify the guest and their identity and set certain guidelines for said guest.
- Securely exchange and validate payment between host and guest
- (More to be added possibly)

4 Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

First, our team will do research on what technologies and options that are available to us. We had the choice of either doing an API or building a new app from the ground up.

(Everything here is still tentative)

...

By Fall of 2020, our client had requested to possibly have a working prototype that we can test our during the football games.

4.2 Feasibility Assessment

4.3 Personnel Effort Requirements

4.4 Other Resource Requirements

4.5 Financial Requirements

A few financial requirements include:

- Server costs
- Publishing our app to the Google Play and App Store

(We cannot do this section until we have fully narrowed down the technologies. Once we narrow it down we can figure out what testing options/frameworks it offers)

5 Testing and Implementation

5.1 Interface Specifications

5.2 Hardware and software

5.3 Functional Testing

5.4 Non-Functional Testing

5.5 Process

5.6 Results

6 Closing Material

6.1 Conclusion

6.2 References

6.3 Appendices