4.2 Design Exploration

4.2.1 Design Decisions

In developing our innovative solution, several critical design decisions were pivotal to shaping the final product. Selecting the appropriate sensor for vehicle detection was a foundational step, ensuring our system could accurately and reliably identify vehicles in various conditions. Recognizing the diversity of our user base, we also strategized how to assist users not utilizing our app, aiming for inclusivity and accessibility in our services. The reservation system presented another complex challenge, prompting us to devise a user-friendly, efficient method for managing bookings. Finally, determining the most convenient times and locations for users to make payments was crucial as we sought to streamline the user experience and enhance satisfaction. Each of these decisions was made carefully considering their impact on functionality, user experience, and the overall success of our project.

4.2.2 Ideation

When it came to choosing sensors to detect vehicles, we went through many a few iterations. First we started with infrared sensors because our client wanted to use those if possible. However, after doing some feasibility testing, we learned that we would need an emitter and receiver for each spot. Not to mention that unskilled drivers would likely hit the placement of the emitter and the receiver. Then, the team went through a brainstorming phase where we discussed possible ideas for a solution. Cameras, LiDAR, and Ultrasonic sensors were brought up during the brainstorming. We even discussed a new technology allowing the parking spot to detect when weight is present.

4.2.3 Decision-Making and Trade-Off

Regarding decision-making, we do most of ours through open discussion. When we are considering a decision, we get together, talk about it, and share our opinions and ideas about it. If necessary, we document our concerns and research to address them if we don't currently have the knowledge. Since we are in the early stages of our projects we want to give concerns the time they need so that we can address issues early. Once we have come to a decision, we write another document explaining what we learned and why we came to that decision.

4.3 Proposed Design

4.3.1 Overview

For our smart parking lot system to operate, ultrasonic sensors pointed at each parking space will detect the state of the parking space or whether a space is occupied or not. This information will be sent to our server utilizing an Arduino board connected to the internet. This information will then be updated in a database. This information will be accessible by the mobile application. Our users will interact with our system via a mobile application. This app will allow users to search for a parking lot near their destination, reserve a space in the lot, and pay for parking. The user will be allowed to simply pay for parking without participating in the reservation process. The application will direct the user to the lot with Google Maps. Once the user parks in their reserved parking spot, the application will transition into the payment page, where the user will securely enter their payment information. If the user bypasses the reservation system, they will park in the lot and be instructed to take note of the parking space number where they parked. They will be able to enter this information into the app and pay for parking. If a driver does not have the application downloaded, they can scan a QR code located in the lot to access a "one-time use" version of the application where they can pay.

4.3.2 Detailed Design and Visual(s)

Overall Design:

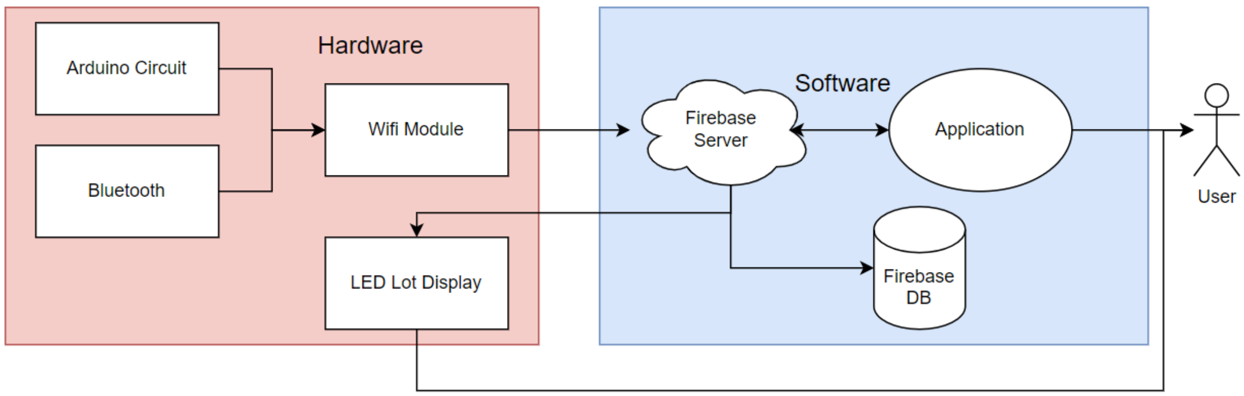

Figure 1: Overall Design Flowchart

Our system is broken into two main components: hardware and software. The hardware component of the system takes the multiple Arduino circuits and connects them together using Bluetooth modules. The wifi module is used to output our hardware information to our server. The software component of our system uses the server loaded with the sensor information and sends it to the application. The application is a way for users to check and reserve parking spaces. The server is also used to send information back to the LED display. This display will represent the parking lot availability, which allows users without the application to use our system.

Software:

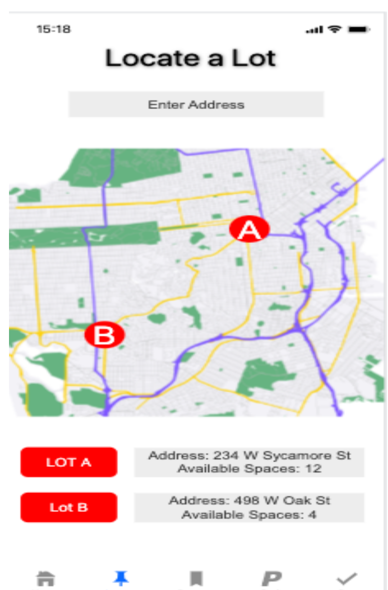

Figure 2: Application User Interface

This is our first design for the app. It might be changed in the future.

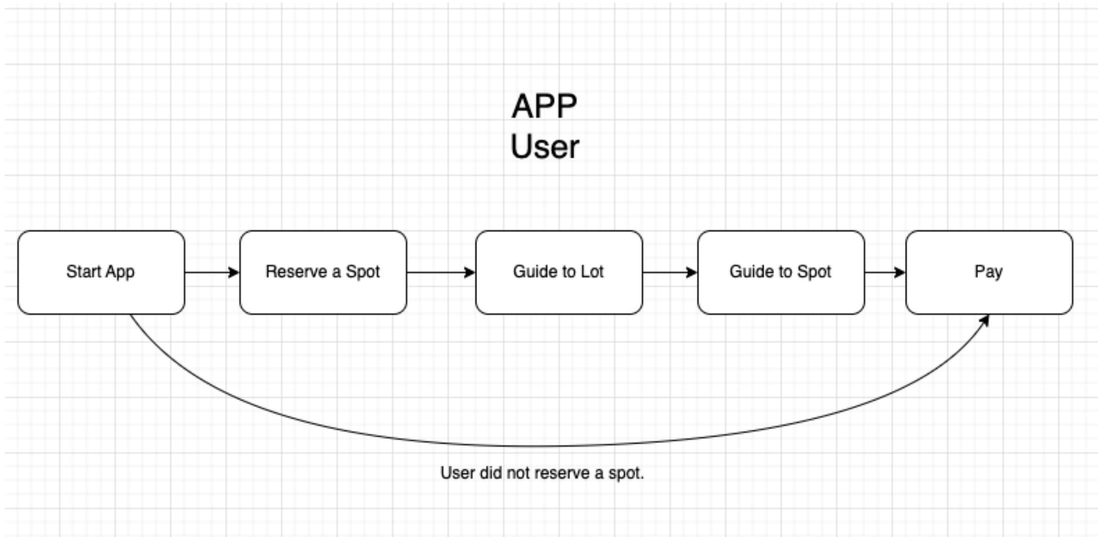

Figure 3: Application Flowchart

Above is the flowchart for our application. The user interface of our application will consist of two different pathways: Reserve and pay.

Reserve:

The user will open the application and choose the reserve tab on the navigation bar on the screen's bottom. The user will then enter their destination in a search box. Our software will search the web for parking lots affiliated with our program in a 2-mile radius of the user's destination. The resulting parking lots will be shown on screen. Each parking lot will have a respective box showing the address, distance from the destination, and number of available parking spots. The user will choose their desired lot to continue to the next step. All of the map functionality will be completed utilizing Google Maps.

After choosing a lot, our application will signal the server to pull an available lot number. This number will be stored in a new variable to ensure actual reservations. Next, the application will direct the user to the parking space with map software upon arrival which the map software with a certain proximity to the parking space will sense. Once parked, the application will continue to the payment function.

Pay:

For the users who arrived at their reserved spot and those who did not participate in a reservation but are still using the lot, the payment process will begin by selecting the app's payment button or scanning a QR code in the parking lot. The payment page will have prompts for spot number, credit card information, and amount of parking time. The users who reserve a spot will have their spot number automatically entered into the spot number prompt. In contrast, the other users will have to take note of their space number, which will be located on a sign in front of the space, and enter it into the app manually. After the necessary information is entered, the app will create a receipt, which will be downloadable for the user. This is the final step of the payment process.

Hardware:

On the hardware side, our team has set out to create an Arduino-based detection system that will be used to sense whether cars are parked or not.

Physical Parking Lot Setup:

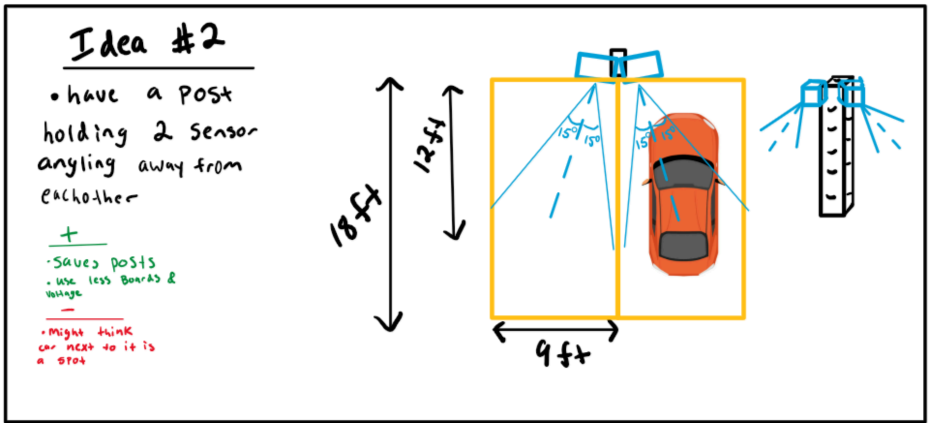

Figure 4: Sketch of Physical Sensor Representation

1 Arduino UNO per post with ultrasonic sensors connected to it. Each sensor will point toward an individual parking spot from an elevated position on the light post.

These sensors will be pointed down at an estimated angle of 30°, as represented by *Figure 4*.

This setup allows for minimal posts to be inserted into the parking lot.

Overall Arduino Circuits:

For every 3 Arduino UNO, there will be 1 Arduino WIFI REV2

The 3 UNO boards that precede the WIFI REV2 board will connect to it via Bluetooth. The data they collect from the parking spots will then be sent to our server and uploaded to the app.

Sensor Arduino Circuit:

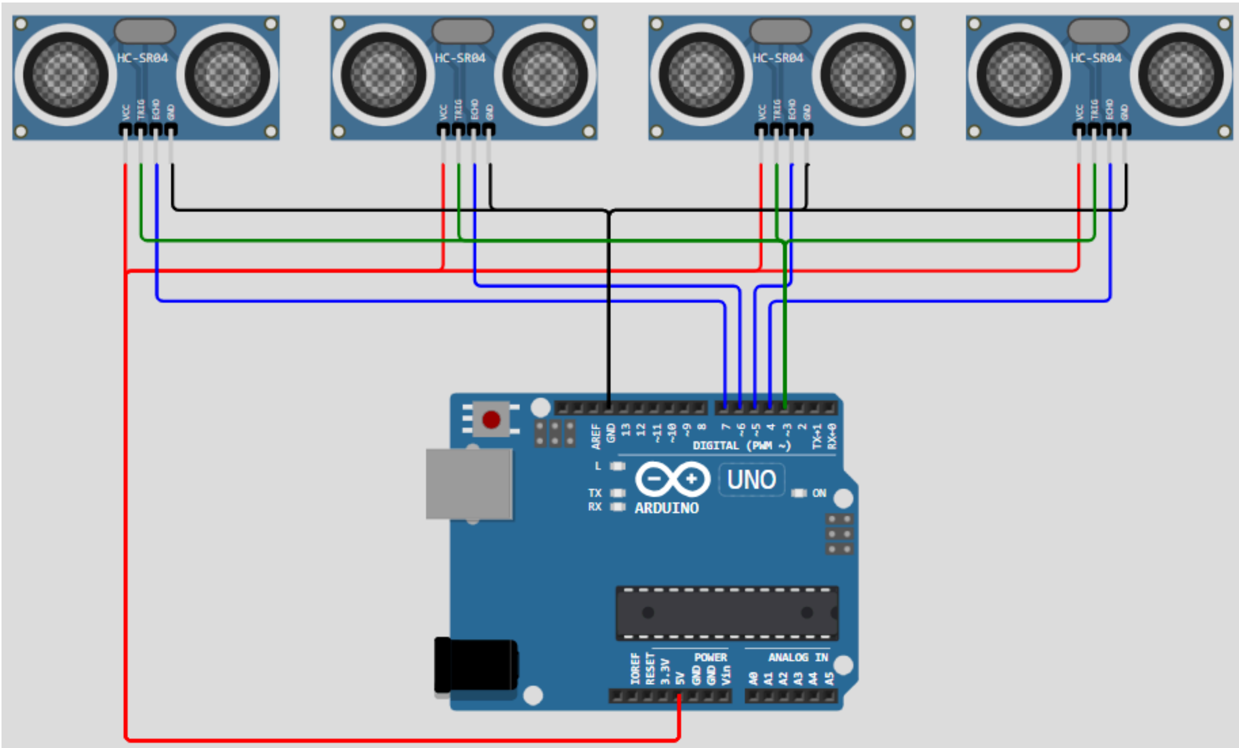

Figure 5: Circuit for Sensors

Each UNO board will have four 5V (Connected to red wire represented in *Figure 5*) ultrasonic sensors connected to it. The sensors will all be connected to the same trigger, so they fire an ultrasonic ping simultaneously (Green wire represented in *Figure 5*).

While the trigger signal is on the same wire, the echo signals are received on their own pins so the board can differentiate them. (Blue wire represented in *Figure 5*).

4.3.3 Functionality

Users can enter our lot regardless of whether they have the app or not. However, they will have different experiences. Our application will direct the users who reserve a spot to their spot, and they will pay upon arrival. For the users who do not reserve a place, they will pay after parking by using the app and entering their spot number which will be posted near their parking space. All necessary data will be taken from ultrasonic sensors, sent to our server, and stored in a database.

4.3.4 Areas of Concern and Development

Our design incorporates all of our user needs and should leave them satisfied if completed as intended. Knowing the impact on users without being finished or without prototyping is challenging. This will be a future focus of the team to come back to. We are concerned with the usability of our application and its friendliness for our users. We are trying to balance the amount of information and detail with simplicity that will be easy to navigate. We plan on holding a usability study when our first app prototype is completed to address this concern.

Our hardware team is focused on a few pending issues. The primary problem is the connectivity of our hardware devices. To update the information about the status of each parking space, our boards must be able to connect to Wi-FI. Based on the spottiness of the internet in some areas of campus, this could cause issues with our connection, and it may require an additional hotspot component. Additionally, we are concerned about the Bluetooth functionality of our boards. How many devices can be connected to our grid by Bluetooth is still being determined. This can only be solved by testing.

Weatherproofing is another issue that our hardware team faces. In the variable weather of Iowa, including rain, snow, ice, and intense heat, our sensors and boards need to survive these conditions. We have been researching for weeks how to accomplish this. Some solutions include creating a case for our sensors and purchasing weatherproofed Arduinos.

For our server, we must carefully choose the correct server for the data type we are working with. Failure to do so can drastically decrease the app's performance and the parking lot. Since we will be working with sensor data, having a database that can handle the constant data stream would be best. However, at this time, we are unsure if Firebase will be able to work for this application.

Finally, for the software, we must ensure that our application is intuitive and safe enough to be used while driving, even in a busy parking lot. We understand that using a phone while driving can be a safety risk. Therefore, we need to minimize the user input while driving and use an acceptable interface for moving applications. At the moment, we are still determining what the best standard of design is for a driving application, so more research is needed.

4.4 Technology Considerations

Describing the distinct technologies we used in our design by highlighting the strengths, weaknesses, and trade-offs made in technology. Also, discussing possible solutions and design alternatives below:

Table 1: Sensor Options

Sensor	Pros	Cons
Infrared	• Preferred by the client • Cheap	• Requires two sensors per parking space • Positioning sensors out of the way of users would be tricky • Sensitive to the color of the car
LiDAR	• Can detect everything in an area	• Expensive • Not entirely appropriate for our use case
Ultrasonic	• Cheap • Only need one sensor per parking space • Familiar to the team	• Little control over where the signal goes •
Pavement	• Discrete	• Super expensive • Will have to renovate the whole lot to implement
Cameras	• Could work over multiple spaces	• Power intensive • Difficult to implement • Possibly expensive

Our final decision was to use ultrasonic sensors because of the team relatively well as their relatively low cost, and ability to function in a variety of conditions that other options may fail in.

4.5 Design Analysis

On the hardware side, we have created a base prototype for the detection system. This system takes an Arduino Nano and connects four ultrasonic sensors to it. We have tested this prototype, and it is working well so far. For the future, we are looking into beginning prototyping for the planned implementation of Arduino Wifi boards to connect to our server. We will start looking into weatherproofing options for the system because it is mainly outdoors. The system must be encased in something waterproof so it does not break down due to water damage.

We have just begun designing the application prototype in Figma for the software side. We reached this after discussing with the client what the user experience should be and doing some whiteboard to create a baseline for how our application will look. We have also started experimenting with Firebase and React Native for backend and frontend development, respectively.