Senior Design CSC 198S
Project - Campark
Kavleen Kaur and Jumana Rahman
Advisor - Mr. Thomas Re

I. Things We Removed

1. Parking Lot Closures
2. Admin View of Parking Lots and Interaction of Admin with Parking Lots
3. Robust Application Security (Input sanitation, User Type Verification with list of pride emails)
4. Limited to 3 Maps
5. Hosting

Due to limited time and the fact that we are a two person team, we didn't have enough time to do the features accessible to users of the user type admin. Our current web application completed the main functionalities initially set for general type users. Although our application ensures user security through email verifications and correct login and password, the web applications would have been more secure for users. Users currently only have access to view information pertaining to their user and vehicle information once they are logged into their account with the use of unique session uids. In order to test and ensure full functionality of each and every parking spot, we limited the maps we used to 3 parking lot maps.

II. Things We Added

1. Google Javascript API, Google Maps API v3 Tool, Admin SDK with session cookies/tokens

In order to make our web application more user friendly, we decided to incorporate and utilize the Google Javascript API along with the Google Maps API v3 Tool when presenting the parking lot views in the web application. Originally, we were going to simply use a grid system where users would've been more likely to accidentally check into a wrong spot because all the parking spots look similar. With the Google API's we were able to make the parking lot view so that users are able to recognize exactly where they are since it shows the names of nearby buildings, roads, along with provides a satellite view of the parking lot. We also used Admin SDK with session cookies/tokens in order to track user and parking spot information with respect to a specific user and specific parking spot.

III. Things We Changed and How We Changed Them

1. Framework (React to Svelte)
2. Geolocation function just checks if you are on campus not your location in individual lots
3. Guests can park anywhere right now

During the first Sprint, we decided to switch the framework we were using for our web application from ReactJS to Svelte. The reason for this was because we were facing routing issues when it came to our code and decided to switch to a framework that was familiar to both of us. We originally wanted to have the Geolocation function check if users are in individual lots to allow them to check into a point. However, we decided to change this to be so that the web application only checks if a user is on campus. One of the main reasons we decided to do this was for testing purposes.

IV. Overview

Overall, we accomplished the main goals of our project which are: 3 map views, check-in based on user type, check-out, create-account, email-verification, view-account, edit-account, report-complaint, forgot-password. We also engaged with Firebase Admin SDK, Firestore database, Firebase Authentication as well as the Google API's. Figure 1 shows the an updated system design diagram.

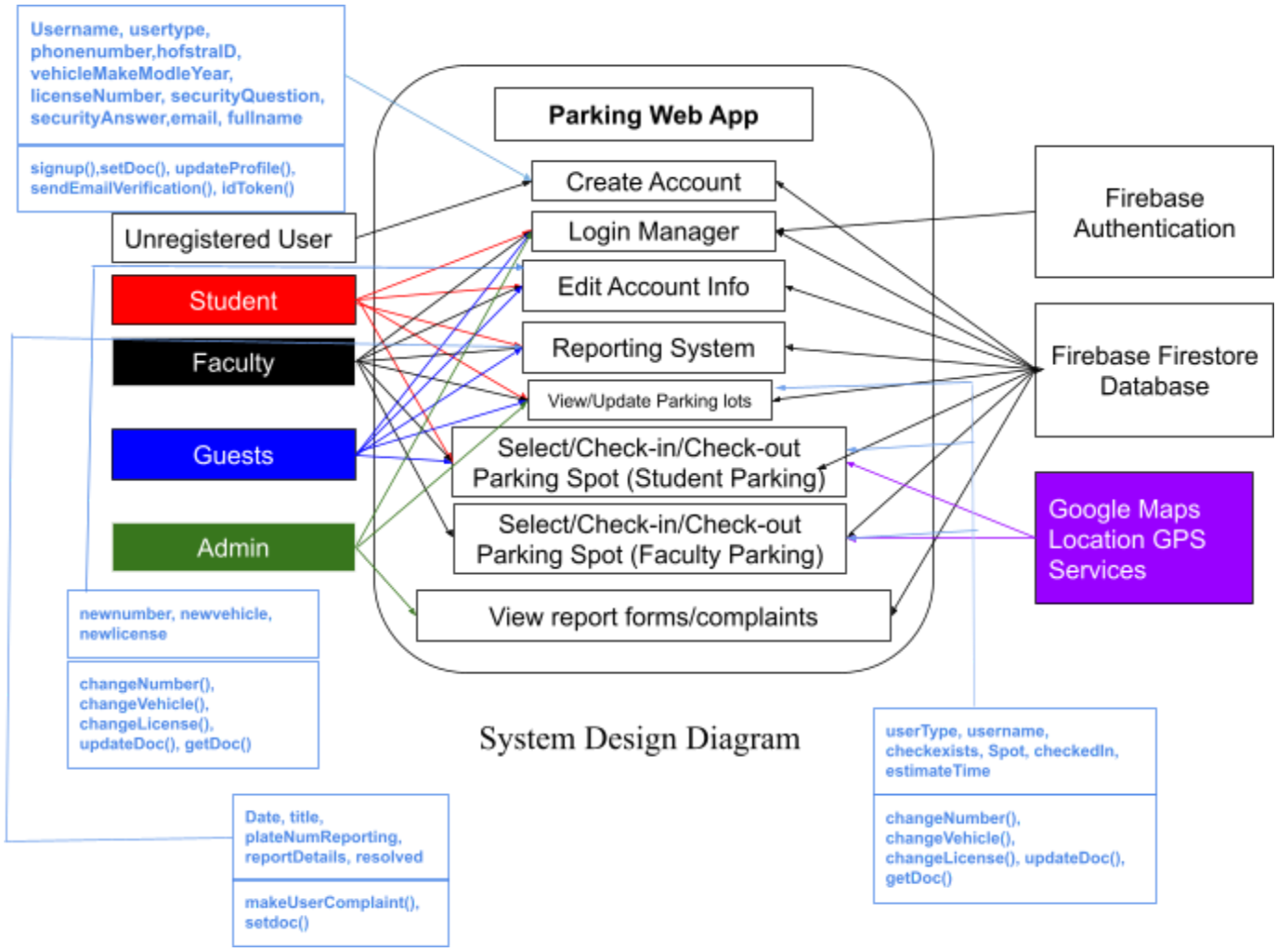

Figure 1