

CamPark

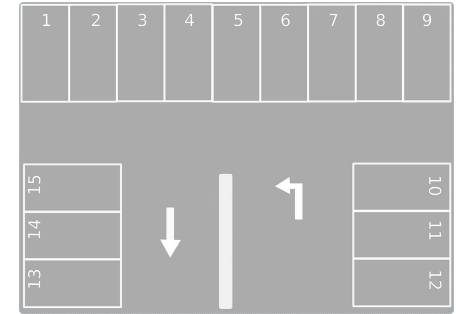
Jumana Rahman & Kavleen Kaur

With Mr. Thomas Re



PROBLEM

- Hofstra Commuters : 40% - 50%
- Parking has always been an issue when it comes to campus parking
- Students and Faculty arrive late to class due to driving around in search of a parking spot
- Incoming freshman commuters aren't familiar with parking lots available to them
- Commuter vs non - commuter parking



SOLUTION (GOALS)

- **Interactive and easy view of available parking for commuters**
 - **Allow commuters to check-in/check-out parking spots**
 - **Report complaints**
 - **GPS Location sharing**
1. Create a web application that people can use to find parking easily and efficiently on Hofstra Campus.
 2. The app will allow students and faculty to see if there is a parking spot available at the South or North Campus parking lot by showing maps with color coordination and numbering. These maps will be constantly updated.
 3. This app will allow people to check-in by being in the vicinity of the spot. Location sharing/GPS will be used to counter reservations of spots.
 4. The app will not allow students to check-in to a parking spot that is not available to their current user type along with parking closures.
 5. The app will allow secure login to all users and will keep information on the user and their vehicle for admin personnel such as Public Safety in the case they need to contact the vehicle owner immediately or verify if the user is a registered hofstra.edu person (students, faculty, administrators). This will also let Public Safety know the guests that are visiting by car on campus.
 6. The web app will provide a reporting system to resolve any issues users face while using the app.
 7. The web app will require users to provide an estimated check-out time when checking-in in order to automatically check-out users that might forget to check-out of their parking spot.
 8. The web app will be responsive for all phone types and sizes.

SYSTEM PERSPECTIVE

- User has access to different features depending on user type
- 4 user types
 - Control where a user can park and features they are exposed to
- System will interact with database to store info
- System will interact with Google Location Services to regulate check-in

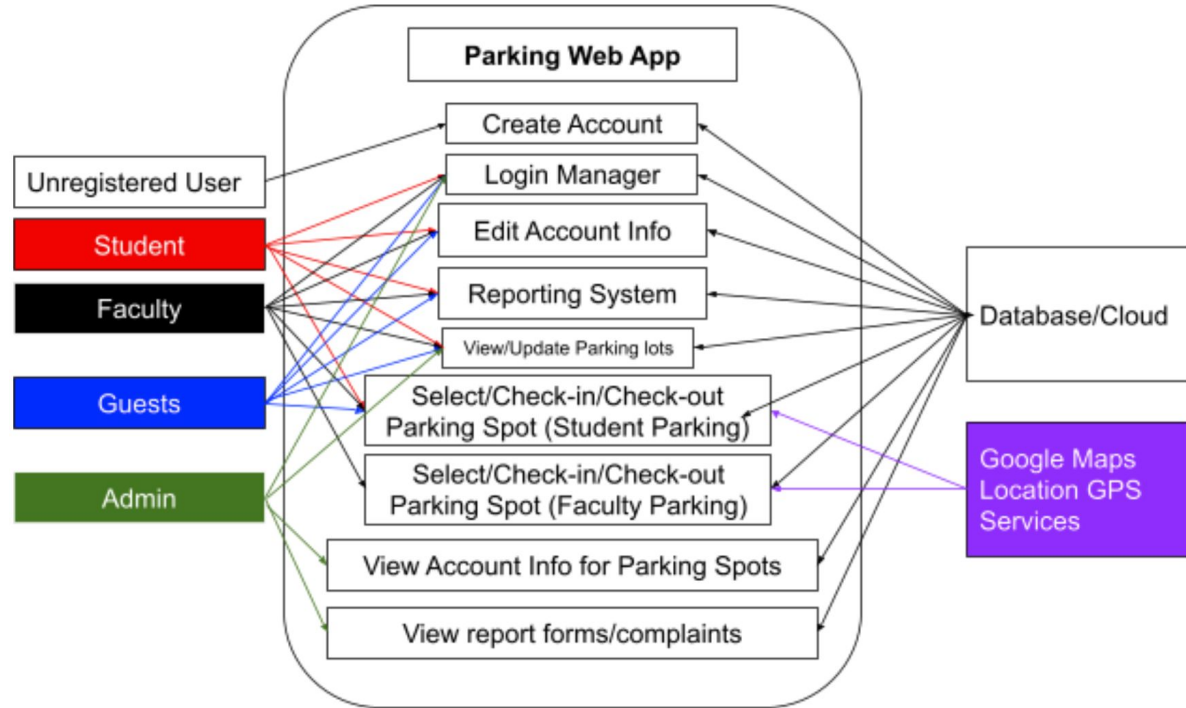


Figure 1 System Diagram

USER STORIES

US-4: Report a Complaint

- All registered users will be able to report a complaint in the case they see an empty parking spot but it is checked-in on the app or if a user is parked in an undesignated spot.

US-5: View Vacant/Occupied Parking Spots

- All registered users will use this feature to see the available parking spots left in certain parking lots by viewing grid maps with lot numbers and spot numbers that are color-coded by availability.

US-6: Check-In/Check Out to Parking Spots According to user type

- Based on user type and location GPS sharing, users can check-in and check-out a parking spot.

US-7: Estimated Check-Out

- Users will have to provide an estimated time in which they will be expected to check out.

USER STORIES Continued...

US-8: View account information of all users parked in parking lot

- Admin users can see account information of all occupied parking spots and have access to the user database

US-9: View report complaints

- Users of the user type admin will be able to view all the complaints other users have reported regarding the webapp and parking spots which will allow admins to resolve any problems that arise.

US-10: GPS location sharing

- Users can only check-in at a parking spot if they are on campus or in the parking lot of the available spot.

US-11: Update parking lot for university closures and events

- User types of the status admin will be able to update parking lots in the case of university events or construction that needs a certain parking lot to be empty.

FUNCTIONAL REQUIREMENTS

- Add user account
- Verify User Account
- Edit Account Info
- Admin capability to view user account info
- Add complaint for Reporting System
- Admin view all complaints for reporting system
- Parking Check In/Out
- Admin Parking Closure feature
- Estimated Check-out Feature



NON-FUNCTIONAL REQUIREMENTS

Performance

- At least 10 simultaneous users
- System's response time of 400ms

Logical Database

- Capability of storing all user information for at least 10 users
- Receive data and update contents in database based on user input
- Send information to web app to display update grid map view and user information
- Capable of allowing admin users to access all information on other users.

Security

- HTTPS used for all web requests
- Password protection for all users - Encryption
- Security questions required in account registration (for account verification)
- Database only accessible to admin users with authentication
- Data integrity and confidentiality with user account information

Reliability

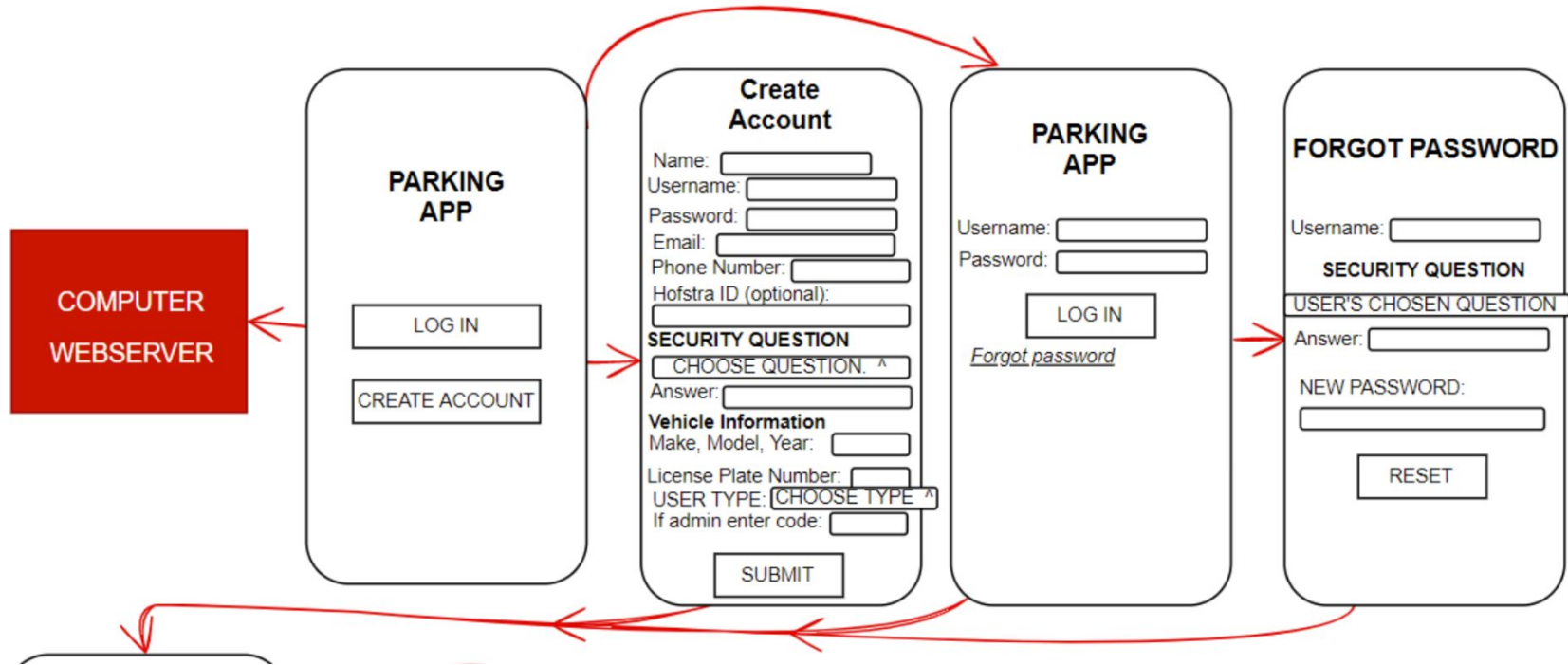
- All user data backed up regularly
- 99% uptime for all end users facing feature

PROJECT DESIGN: SYSTEM DESIGN DIAGRAM Database Flow

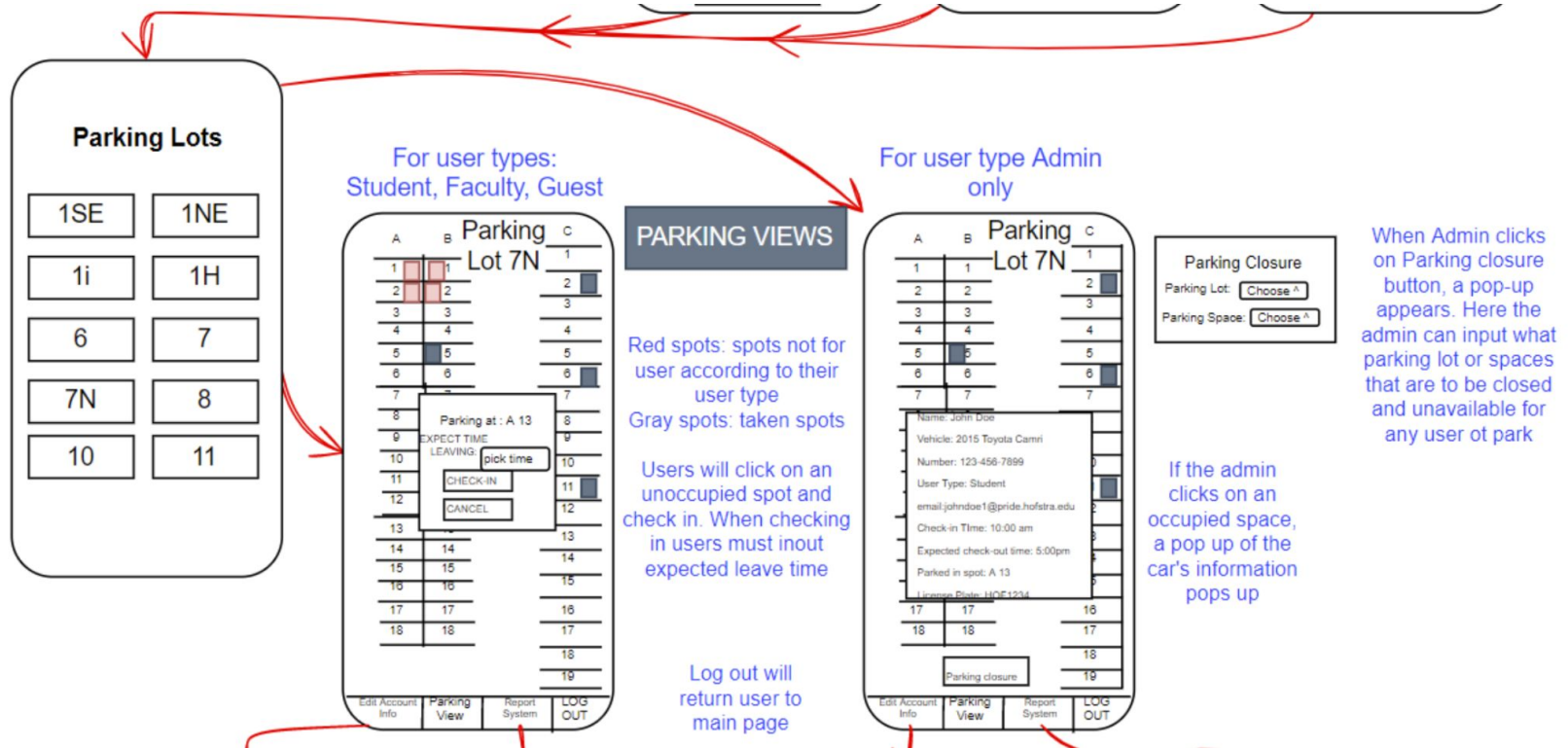


Figure 2 System Database Overview

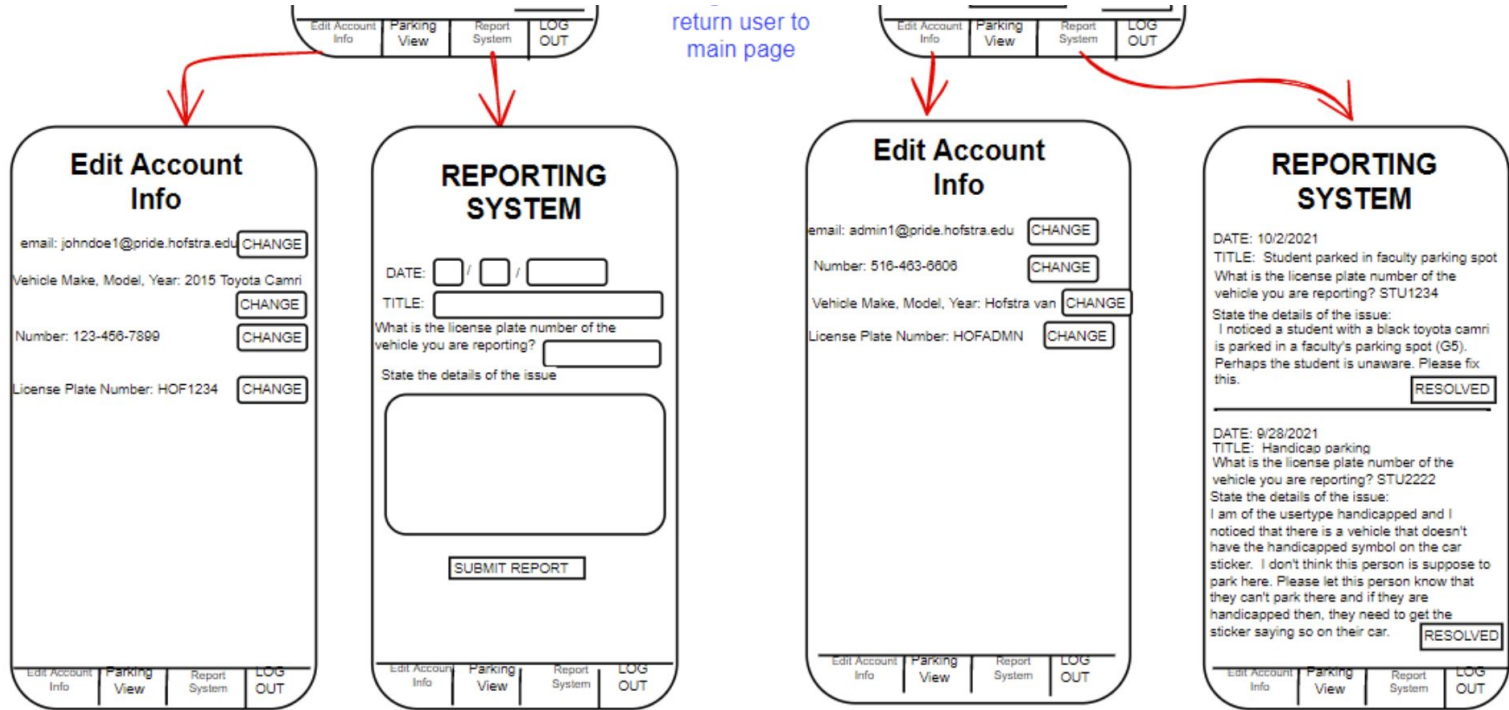
PROJECT DESIGN: SYSTEM DESIGN DIAGRAM Part 1



PROJECT DESIGN: SYSTEM DESIGN DIAGRAM Part 2



PROJECT DESIGN: SYSTEM DESIGN DIAGRAM Part 3



TOOLS + METHODS



Tools

- Visual Studio
- Firebase Realtime Database
- Google's Geolocation API

Methods

- Password based authentication
- Firebase authentication



Test Plan

Test strategy for this project

- Unit Testing done at each end of the sprint cycles
- Integration (and Regression) testing done at the end of Sprints 2, 3, and 4
 - Test the working product as a whole
 - This will be done by manual testing
- After all unit tests, integration tests, and regression tests have been completed, a complete system test will be conducted to ensure that the whole website is functional and works successfully.

PROJECT PLAN: SPRINT 1

Scrum Master: Kavleen Kaur
Period: January 2022

Goals:

- Secure connection with the database
- Base webpages/UIs
- Account management (New user)

Table 6.1.1: Backlog list for a Sprint 1

Task #	Task description	Hardware/Software resources	People(s)	Estimated hours	Points difficulty	Cross-reference
1	Create a project and all webpages (interface layout) except parking grid	Visual Studio Code,	Jumana and Kavleen	20	5	US-1...US-11
2	Research how to create & link Database to website		Jumana and Kavleen	12	8	C-1..C-2, US-1...US-2
2	Create Database and organize data entities in database	Firebase, Microsoft SQLServer	Kavleen	5	7	C-1..C-8, US-1...US-11
3	Link and make a connection between project and database	Visual Studio Code, Microsoft SQLserver, Firebase	Kavleen	5	8	C-1..C-8, US-1...US-11
4	Test the connection by making new user		Kavleen	5	8	C-1..C-8, US-1...US-11

PROJECT PLAN: SPRINT 2

Scrum Master: Jumana Rahman
Period: February 2022

Goals:

- Account verification + password manager
- Security
- All UIs + buttons working
- Edit Profile

Table 6.1.2: Backlog list for a Sprint 2

Task #	Task description	Hardware/Software resources	People(s)	Estimated hours	Points difficulty	Cross-reference
1	Ensure the all the components in create account page works and is stored in the database	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen	5		C-1, C-2, C-3
2	Apply encryption methods to secure user information in the database		Kavleen	10		I2; O2, US-2, UI-2, C-2
3	Use the database to check information to allow secure login from user.	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	5	5	C-2
4	Successful logout	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	5	4	C-2
5	Functionality of edit profile page. Make sure the contents changed on this page also changed in the database.	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	10	6	C-6, I4, O4, US-2, US-3
6	Functionality of forgot password page. Verify users with security questions. Login with the new password to check if it is successful.	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen	10	8	I2; O2, US-2, UI-2, C-2

PROJECT PLAN: SPRINT 3

Scrum Master: Kavleen Kaur
Period: March 2022

Goals:

- User View + Parking View of parking lots
- Functionality for user types
- Admin parking closure + pop ups
- Check-in/check-out

Table 6.1.3: Backlog list for a Sprint 3

Task #	Task description	Hardware/Software resources	People(s)	Estimated hours	Points difficulty	Cross-reference
1	Research and create user interface for parking lot views	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen and Jumana	10	10	C-4, C-5
2	Different parking spots available depending on usertype	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen and Jumana	25	10	C-4, C-5
3	Working Check-in/Check-out for all user types.	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	8	6	I2, O2, US-2, UI-2, US-8
4	GPS	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	5	5	I2, O2, US-2, UI-2, US-8
5	Automatic check-out for users using estimated check-out time	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	3	4	US-2, UI-2, US-8
6	Admin view of parking lot	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen	10	9	C-5
7	Fetch data from database and display when admin clicks on a parking spot to display all info on vehicle and car owner.	Firebase, Microsoft SQLServer, Visual Studio Code, Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	8	5	C-4, C-5, I2, O2, US-2, UI-2, US-8
8	Functionality of admin parking closures pop up.	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen	15	10	US-2, UI-2, US-8
9	Verify if admin parking lot closure works by logging into a user account and seeing if the parking lot closed is available to park in	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen	10	9	C-4, C-5, I2, O2, US-2, US-8

PROJECT PLAN: SPRINT 4

Scrum Master: Jumana Rahman
Period: April 2022

Goals:

- Reporting System
- Admin + User Views
- Touch ups

Table 6.1.4: Backlog list for a Sprint 4

Task #	Task description	Hardware/Software resources	People(s)	Estimated hours	Points difficulty	Cross-reference user stories, require-elements, design
1	Ensure all UI components of the Reporting System page works	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana and Kavleen	8	5	C-7, I5, O5, US-9, UI-5
2	Create New Report	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	5	4	C-7, I5, O5, US-9, UI-5,
3	Reports link to database; send + fetch	Firebase, Microsoft SQLServer, Visual Studio Code	Jumana	5	5	C-8, UI-6, I5, US-9, UI-5, UI-6
4	Admin sees new reports	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen	5	5	C-8, UI-6, I5, US-9, UI-5, UI-6
5	Resolve reports & increase count (if a vehicle was reported)	Firebase, Microsoft SQLServer, Visual Studio Code	Kavleen	5	5	C-8, UI-6, I5, US-9, UI-5, UI-6

RISK PLAN

RP-1: Task List/Time

Likelihood: 5/10

Impact: 4

Actions to alleviate:

- Use January to get a head start to research and plan.
- Implement and stick to a strict time schedule per week and month.

RP-2: Database Connection

Likelihood: 7/10

Impact: 5

Actions to alleviate:

- Use January to fully configure the database and web application.
- Research and plan before creating a database and web application.

THANK YOU

Can't Park??



Choose CamPark