



## **CS 307 Sprint 3 Planning - Group 6**

### **BoilerPark**

Pranay Nandkeolyar

Utkarsh Majithia

Anthony McCrovitz

Logan Portscheller

George Samra

Neel Vachhani

## Sprint Overview:

This sprint we'll really be working on finishing and polishing up the app to get it ready to demo. We'll be adding some final features and ensuring that our users have a seamless experience from start to finish. Additionally, we're hoping to meet with Purdue Parking and get a camera set up to link with the CV model that we've been building. This will allow our app to be fully functional and responsive.

**Scrum Master:** Neel Vachhani

**Meeting Plan:** M/W/F @ 10:30am

**Risks and Challenges:** Meeting with and setting up the camera will definitely be a challenge this sprint. We're working with a 3rd party and setting up a system that will have to communicate via edge computing to our server. We're confident of this working, but it will still present more of a challenge than a normal user story. Another challenge will be connecting everything at the end of the sprint into a cohesive app. There will be no more time to work, so it's important that we're able to have a connected app.

## Current Sprint Detail

**User Story #1** - As a driver I want a link to a navigation app set to my lot of choice so I can easily see directions to the garage I'm going to

Task #	Task Description	Estimated Time	Owner
1	Add frontend element to the detail garage view	2 hours	George
2	Add frontend element to the garage overview	2 hour	George
3	Create working link to both Apple Maps and Google Maps	3 hour	George
4	Test the frontend elements on the detail view and on the garage view for both Apple and Google Maps	2 hours	George

### Acceptance Criteria:

- Given that the user is logged in, when they open the app I want to see the garage list with an element to take me to a maps app
- Given that I click on a garage from the main view, when it opens the detail view I want to see another element with a link to the maps app
- Given that I click on the element in either the detail screen or the main screen I want the option of opening the link in either Apple or Google maps
- Given that I click the element, when I select the maps option I want to be taken to the correct app
- Given that I have selected a garage, when I am taken to the app I want to have the correct address already entered

**User Story #2** - As a a user, I want to see an average user rating for each lot so that I can factor in the quality of different lots when deciding which lot to park in.

1	Add star rating element to the garage and detail views	3 hours	George
2	Add the element to accept user rating of the garage	2 hours	George
3	Store the accepted rating and use it to recalculate	2 hours	George
4	Persist the user's rating so they don't "re-rate" it every time	2 hours	George
5	Add the ability to filter by average rating	1 hours	George

6	Test the user rating of the garage	2 hours	George
---	------------------------------------	---------	--------

Acceptance Criteria:

- Given that I'm logged into the app, when the main screen loads I want to see an element that shows me the average rating of the garage
- Given that I'm on the main screen, when I click on a garage and am taken to the detail view I want to see an element that displays that average user rating
- Given that I'm on the detail screen when I scroll down I want to see an element that allows me to rate the garage
- Given that I'm on the main screen, when I open the filter menu I want to be able to filter by the average rating of the garage.
- Given that I rate a garage, when I open the main view again I want to see the updated average rating based on my rating of that garage

**User Story #3** - As a user I want to be able to add significant arrival locations to reduce friction and make creating plans easier

Task #	Task Description	Estimated Time	Owner
1	Add an element to a screen that allows the user to save not only home but significant arrival locations	2 hours	George
2	Geocode those locations with a maps API to make them easy to work with	2 hours	George
3	Create Django route to accept the User preferences	1 hour	George
4	Store a list of the locations in the User model	2 hours	George
5	Test the saving and storing of the locations	2 hours	George

Acceptance Criteria:

- Given the user is on the Parking List Screen, when the user types in the name of a parking lot, then the correct lot should appear.
- Given the user searches a given parking lot, when that lot appears, then the information for that lot should be loaded and shown.
- Given that the user is looking for a lot, when the user searches for open lots, then only open lots should be shown.
- Given that the user is looking for a lot, when the user searches for their favorite lots, then only their favorite lots should be shown.

**User Story #4** - As a user, I want push notifications when a favorite lot drops below X% available so I can leave sooner

Task #	Task Description	Estimated Time	Owner
1	Add UI elements to frontend settings	2 hours	Neel
2	Save notification preferences to User model	2 hours	Neel
3	Test the UI component displays properly	1 hour	Neel
4	Connect saved preferences to Apple APNs	2 hours	Neel
5	Handle changes to push notification preferences (e.g. changes in X%)	1 hour	Neel
6	Implement notification cooldown to avoid overwhelming users	1 hour	Neel
7	Test that push notifications occur when favorited lots drop below X% available	1 hour	Neel

Acceptance Criteria:

- Given a user is on the settings tab, when they access the preferences section, they should see an option to enable push notifications for when favorited lots drop below X%available.
- Given the user is on the settings screen, when they enable push notifications as above, they will be able to enter their percentage threshold X.
- Given that the user has set up push notifications as above, when a favorited lot drops below X% availability, a push notification is sent to the user informing about the drop
- Given that the user received a notification less than 5 minutes ago, when a favorited lot drops below X% availability, a push notification is not sent to the user.

**User Story #5** - As a user, I want to report incorrect lot status (e.g., the app says open but it's actually full) so developers can improve data quality.

Task #	Task Description	Estimated Time	Owner
1	Add report option to every detailed view in frontend	3 hours	Neel
2	Build API route to send report information to backend	3 hours	Neel
3	Save reports made by users in Postgres	1 hour	Neel
4	Design and implement confirmation message	1 hour	Neel

	after a successful report is sent.		
5	Test report send, saving in logs, and completion message	1 hour	Neel

Acceptance Criteria:

- Given a user is on the detailed view of a garage, when they click on the report error button, a popup allowing the user to input a description appears.
- Given that the user has submitted a report using the popup, when the report is successfully sent, a confirmation popup should appear on the user's screen
- Given that the user is prompted with a report popup, when they try to submit the report without a description, an error will appear.

**User Story #6** - As a user I want to see a map with the live counts displayed on top of the garages.

Task #	Task Description	Estimated Time	Owner
1	Add all eligible garages/lots to the map view	2 hours	Neel
2	Link all pins on map view to respective Redis key-value pairs such that availability is accurate	3 hours	Neel
3	Handle automatic updates of availability values from Redis Pub/Sub channel	4 hours	Neel
4	Test the map view and availability updates	2 hours	Neel

Acceptance Criteria:

- Given a user is on the map view, when they click on a pin representing a garage, the description of that pin shows the live availability of that garage.
- Given that the user has the description of a map pin open, when that garage's availability updates, the live availability in the description reflects the change.
- Given that the user has clicked on a map pin and opened its description, when the user presses on the detailed view button, they should be brought to the detailed view of that specific garage.

**User Story #7** - As a developer, I want to sync the camera to the CV model (Modified User Story 1 from Sprint 2)

Task #	Task Description	Estimated Time	Owner
1	Configure the camera to expose a video/data stream and verify local connection over HTTPS	2 hours	Pranay

2	Set up a lightweight server (Flask, FastAPI, or MQTT broker) on the edge device	2 hours	Pranay
3	Stream Video Footage to edge computing resource	2 hours	Pranay
4	Refactor car counter model to utilize these cameras for a subset of parking lots	2 hours	Pranay
5	Test that the updated counts are displayed	1 hour	Pranay

Acceptance Criteria:

- Given that we have the camera set up, when a car moves into the corresponding garage, then the Redis cache decreases the availability for that specific parking lot and the count is updated on the UI
- Given that we have a camera set up, when a car moves out of the corresponding garage, then the Redis cache is accurately updated to increase the availability for that lot and the count is updated on the Garage List UI.
- Given that we have a camera set up, when there is no car moving in or out of the garage, then the Redis cache is not updated and the count is unchanged on the Garage List UI.
- Given that we have a camera set up, when the count is modified, then the correct Redis cache garage value is updated and the correct frontend shows.

**User Story #8** - As a user I want to compare the insights from 2 or more garages

Task #	Task Description	Estimated Time	Owner
1	Refactor Data and Insights Tab into one	2 hours	Pranay
2	Develop UI picker for comparing applications	2 hours	Pranay
3	Develop a detailed Insights View UI	1 hour	Pranay
4	Develop backend API route to display comparisons of garages	2 hours	Pranay
5	Display key metrics such as availability in numeric and chart form	2 hours	Pranay
6	Test that the data is accurately fetched from the backend	1 hour	Pranay

Acceptance Criteria:

- Given the user accesses the application, when the Data and Insights tabs are refactored into a single unified tab, then all existing functionality from both tabs should remain accessible within the new combined interface.
- Given the user is viewing the unified Data and Insights tab, when the user selects multiple parking lots or applications to compare, then a comparison view should be displayed showing each lot's availability side by side in bar chart form.
- Given the backend API route for parking lot comparisons is functioning correctly, when the UI requests comparison data from the Django API, then the charts should dynamically update with the most recent availability data for each lot.
- Given the comparison charts are displayed, when one or more lots have different availability levels, then the colors of their bars should update based on thresholds (e.g., green for available, yellow for moderate, red for full).

**User Story #9** - As a user I want parking logs of my parking patterns to be used to see what garages I end up in most frequently by day

Task #	Task Description	Estimated Time	Owner
1	Add a button on the detailed view to indicate a user has parked in an area	1 hour	Pranay
2	Create a UserPark PostgreSQL Table to store this user parking information with a foreign key of a username	2 hours	Pranay
3	Develop a backend Django API endpoint to handle a user parking in a lot	3 hours	Pranay
4	Implement a frontend UI section showing "User Insights" showing Users when they park and where	2 hours	Pranay
5	Integrate UI with Insights tab to show them their average parking times compared with the insights for average availability at said time	2 hours	Pranay
6	Test that different user park times throughout a week are correctly displayed and compared with different insights	1 hour	Pranay

Acceptance Criteria:

- Given the user has historical parking logs stored in the system, when they open the Insights tab, then their parking patterns should be analyzed and displayed alongside general garage availability trends for comparison.

- Given the user is viewing the “User Insights” section, when they compare their average parking times with overall garage availability, then the UI should display both datasets together to highlight differences or overlaps in usage.
- Given the backend API for parking data and insights is functioning correctly, when new parking logs are recorded, then the Insights tab should automatically refresh to show updated comparisons between the user’s habits and general availability.
- Given the user selects a different garage within the Insights tab, when they switch comparisons, then the charts should dynamically update to show their parking frequency versus that specific garage’s average occupancy.

**User Story #10** - As a User I want smooth animations, the bottom navigation bar to be polished and guide me to key features, without confusion

Task #	Task Description	Estimated Time	Owner
1	Refactor bottom bar to load the tabs from one config	2 hours	Anthony
2	Add icons the match with purdue branding and light/dark themes	2 hours	Anthony
3	Make the tab highlight slide smoothly when switching tabs	2 hours	Anthony
4	Remember the last tab a user opened and add simple deep links to open each tab	1 hour	Anthony
5	Test the new tab bar behavior and capture screenshots for testing doc and demo	1 hour	Anthony

Acceptance Criteria:

- Given the user toggles between light and dark mode, the tab icons, labels, and background update to stay legible and on-brand.
- Given the user taps a different tab, a smooth highlight animation plays, the correct screen opens on the device.
- Given the user closes and reopens the app, it automatically opens on the tab they last used.
- Given the new tab bar code is in place, the behavior is working in both light and dark mode.

**User Story #11** - As a User, I want helpful messages when filters show no garages so I know what to do next.

Task #	Task Description	Estimated Time	Owner
1	Build a reusable EmptyState React component with icon slot, title, and action button	2.5 hours	Anthony
2	Add themed styles Purdue, light, dark, and	2 hours	Anthony

	animations		
3	Render the empty message n the garage list when the filtered results are empty and setup clear filters action	2 hours	Anthony
4	Render the empty message in the detail/favorites view when they have zero items	1.5	Anthony
5	Write the tests to ensure the empty state message renders on the frontend appropriately	2	Anthony

Acceptance Criteria:

- Given search/filter results return zero garages, the list view shows the empty-state component with message and clear button.
- Given the user taps the clear button, search text and selected passes reset and garages reappear.
- Given any favorites or detail list becomes empty, the same empty-state component is shown.
- Given the device switches between light and dark mode, the empty-state styling adapts while staying readable and on-brand.

**User Story #12** - As a User I want a floating “back to top” button on the garage list so I can jump without scrolling forever.

Task #	Task Description	Estimated Time	Owner
1	Add a floating action button styled with Purdue colors and an up arrow icon	2 hours	Anthony
2	Track scroll positions on the garage list and show the button after the user scrolls	1 hours	Anthony
3	Smoothly animate the button and hook it up the scroll the garage list	1 hour	Anthony
4	Ensure the button adapts to the light/dark mode and Purdue Color Theme	1 hour	Anthony
5	Ensure testing on the IOS simulator and work across iPhone screen sizes	1 hour	Anthony

Acceptance Criteria:

- Given the user scrolls the garage list past the threshold, the floating “Back to top” button fades in near the bottom-right corner.
- Given the user taps the button, the list scrolls smoothly back to the top.
- Given the user scrolls back near the top, the button fades out.
- Given light or dark mode is active, the button styling stays readable and on-brand.

**User Story #13** - As a user, I want the walking time from lot to event/class added to ETA so plans are realistic.

Task #	Task Description	Estimated Time	Owner
1	Identify how to get walking distance/time (e.g., Google Maps API).	2 hours	Utkarsh
2	Update backend to fetch walking duration from lot to destination.	4 hours	Utkarsh
3	Add walking time to total ETA calculation.	2 hours	Utkarsh
4	Update frontend to show total ETA (driving + walking).	1 hours	Utkarsh
5	Test ETA calculations and UI for accuracy.	2 hours	Utkarsh

Acceptance Criteria:

- Given that the user has selected a parking lot and a destination (event or class), when the system calculates the ETA, then the total ETA should include both driving and walking time.
- Given that the system has fetched both driving and walking durations, when the ETA is displayed, then the user should see the total ETA along with a clear breakdown (e.g., “5 min drive + 7 min walk”).
- Given that the user changes their parking lot or destination, when the screen updates, then the total ETA should automatically recalculate and display the new combined time.
- Given that walking distance data is unavailable or cannot be fetched, when the ETA is displayed, then the user should still see the driving ETA with a note that walking time is unavailable..
- Given that valid walking and driving data exist, when the ETA is shown on the event/class screen, then the total time should be accurate within approximately one minute of real travel time.

**User Story #14** - As a user, I want a popup to appear when I’m inside a parking garage so I can report how full it looks, helping improve data accuracy.

Task #	Task Description	Estimated Time	Owner
1	Detect when user’s GPS location is within the boundary of a known garage.	2 hours	Utkarsh
2	Design a popup with three options and trigger when inside the garage (only once per visit).	4 hours	Utkarsh
3	Create or update backend endpoint to store occupancy reports.	2 hours	Utkarsh

4	Add a confirmation message after successful submission (“Thanks for your report!”).	1 hours	Utkarsh
5	Test for correct popup triggering and smooth UX.	2 hours	Utkarsh

Acceptance Criteria:

- Given that the user’s location is detected inside a known garage, when the app confirms the position, then a popup should appear asking the user to report how full the garage is.
- Given that the popup is shown, when the user selects one of the three fullness options and taps “Submit”, then the app should record and send this report to the backend with the correct metadata (garage ID, timestamp, and choice).
- Given that the report submission is successful, when the server confirms receipt, then the user should see a confirmation message thanking them for their feedback.
- Given that the user has already submitted a report recently for the same garage, when they remain inside the same location, then the popup should not reappear
- Given that the user leaves the garage, when their GPS exits the boundary, then the popup should close automatically if still open.

**User Story #15** - As a user, I want the app to suggest the nearest alternative parking structure when my chosen lot is full.

Task #	Task Description	Estimated Time	Owner
1	Create UI element to tell users when lot is full and the nearest lot that isn’t full	2 hours	Logan
2	Create logic to calculate the distances to the nearest garage	2 hours	Logan
3	Trigger the garage is full pop-up when the user clicks on a full garage	2 hours	Logan
4	Check that the availability nearby garages to only show garages that are not near to full	2 hours	Logan
5	Tell the user when all lots are full with a pop-up	2 hours	Logan
6	When the user clicks on the garage recommended on the pop-up, the detail view of the garage should be displayed	2 hours	Logan

Acceptance Criteria:

- Given a user presses on a garage that is full (on the garage list), a new page or pop-up should be displayed alerting them that the garage is full (or near full)

- Given a user presses on a garage that is full and the new page pops up, the nearest not-full parking garage should be displayed
- Given a user presses on the recommended garage on the pop-up, that garage's detailed view should be displayed

**User Story #16** - As a user, I want a calendar screen that incorporates all events. *(modified user story 17 from sprint 2)*

Task #	Task Description	Estimated Time	Owner
2	Create or update the backend endpoint to fetch all events with relevant details.	3 hours	Utkarsh

Acceptance Criteria:

- Given that the user opens the calendar list screen, when the screen loads, then the user should see all upcoming events displayed in an organized list.
- Given that multiple events occur on the same day, when the events are displayed, then they should be grouped together by date.
- Given that the user views an event from the list, when the event details are shown, then the user should see its name, date, time, and location clearly.

**User Story #17** - As a user, I want a confidence level on each lot's count (High/Medium/Low) so I can better assess how much I can trust the data for certain lots. *(modified user story 12 from sprint 2)*

Task #	Task Description	Estimated Time	Owner
2	Update backend or data model to include a confidence field for each lot's count.	2 hours	Utkarsh
3	Implement backend logic to calculate or assign confidence levels based on data freshness or camera reliability.	3 hours	Utkarsh

Acceptance Criteria:

- Given that I am on the parking list page, when I view the parking lots, I should see a confidence label (High, Medium, or Low) next to each lot's count.
- Given that the data is reliable, when the user views the reliability of the count, it should show "High" confidence with a clear color or icon.
- Given that the data is moderately reliable, when the user views the reliability of the count, it should show "Medium" confidence with a distinct but neutral color or icon.
- Given that the user opens a detailed view, when they check to see the parking lot count, then they should see the confidence level displayed along with its count.

**User Story #18** - As a user, I want to rate accuracy (“was this lot accurate?”) so the model improves.

Task #	Task Description	Estimated Time	Owner
1	Display a button on each garage’s detailed view	2 hours	Logan
2	Create a UI page to pop up when the user pressed the rate accuracy button	3 hours	Logan
3	Create a backend API route to send the user’s accuracy rating to the backend	4 hours	Logan
4	Add average rating statics to the garage object in the backend	2 hours	Logan

Acceptance Criteria:

- Given a user navigates the detailed view for a garage, there should be a button displayed for the user to rate the accuracy
- Given a user presses on the rate accuracy button on the detailed view, a new page should pop up prompting the user for their accuracy rating
- Display the average accuracy rating of each garage on the garage’s detail view

**User Story #19** - As a User I want the settings screen organized into clear sections so I can find the right controls easily

Task #	Task Description	Estimated Time	Owner
1	Refactor the settings screen layout into Purdue themed sections (Account, Notifications, About, etc.)	3 hours	Anthony
2	Make each section collapsible and add a quick summary row at the top	1.5 hours	Anthony
3	Ensure the new layout works in both light/dark mode and supports screen readers with section headers labeled	1 hour	Anthony
4	Test the layout still allows the settings to be enabled and disabled	1 hour	Anthony

Acceptance Criteria:

- Given the user opens Settings, the controls are grouped under clearly labeled sections with Purdue styling.
- Given a section header is tapped, it expands/collapses while preserving state.

- Given light or dark mode is active, section headers and rows stay readable and on-brand.

**User Story #20** - As a user I want to be able to filter the map by parking pass or favorite lot

Task #	Task Description	Estimated Time	Owner
1	Add logic to filter by parking pass on the map page	1 hours	Logan
2	Add logic to filter by favorite lot on the map page	1 hours	Logan
3	Add buttons on the map page to filter by parking pass and filter by favorite garages	1 hour	Logan
4	Call backend APIs to fetch the user's favorite garages	2 hours	Logan
5	Call backend APIs to fetch the user's parking pass	2 hours	Logan
6	Test the filtering on parking pass	1 hour	Logan
7	Test the filtering on favorite lot	1 hour	Logan

Acceptance Criteria:

- Given a user navigates to the map page, they should see two buttons: one for filtering by favorite lot and one for filtering by parking pass
- Given a user has favorited garages on the garage page, when they navigate to the map page and click the filter by favorite button, only their favorite garage's pindrops should be shown
- Given a user has saved a parking pass, when they navigate to the map page and click the filter by favorite button, only garages with their parking pass should be displayed with a pindrop

**Hour Summary for each person:**

Logan: 32 hours

George: 30 hours

Anthony: 30.5 hours

Utkarsh: 30 hours

Pranay: 30 hours

Neel: 30 hours

[illegible]

[illegible]

[illegible]

# Remaining Backlog

## 1. Product Backlog

### 1.1. Functional

#### Authentication and Onboarding:

- ~~1.1.1. As a user, I want sign in with Apple to make creating a user account easier~~
- ~~1.1.2. As a user I want sign in with Google to make creating a user account easier~~
- ~~1.1.3. As a user I want sign in with email and password to allow me to make an account and save my preferences~~
- 1.1.4. As a new user, I want a simple onboarding flow (tutorial/walkthrough) so I understand the app quickly.

#### Core Functionality:

- ~~1.1.5. As a driver I want a screen that shows updated counts for each garage~~
- ~~1.1.6. As a user I want to be able to see accurate availability of lots (Whether the lot is open or not e.g. football games)~~
- ~~1.1.7. As a user, I want to see a “Last updated <timestamp>” label on each lot so I can trust freshness.~~
- ~~1.1.8. As a user, I want to search for lots by name/code so I can find them quickly.~~
- ~~1.1.9. As a user I want to have a map with all of the garages/lots listed on them~~
- ~~1.1.10. As a user, I want a detailed view (hours, floors, rules, walking time to destination) so I can evaluate options.~~
- ~~1.1.11. As a user, I want a color gradient on each parking structure (from green to red) that signifies how full a given parking structure is (e.g. green for empty and red for full).~~

#### Computer Vision:

- ~~1.1.12. As a developer I want to set up a camera in a parking lot~~
- ~~1.1.13. As a developer I want to use computer vision to detect cars using online training data~~
- ~~1.1.14. As a developer I want the computer vision algorithm to not detect other objects or vehicles that are not cars~~
- ~~1.1.15. As a user I want an accurate count of total parking spots in all garages~~
- 1.1.16. As a developer, I want to sync the camera to the CV model
- ~~1.1.17. As a developer I want to be able to count the automobiles accurately and store the data in a database and thus find the available spots~~

#### Mapping and Navigation:

- ~~1.1.18. As a user I want to know when to leave my house to make it to events on time using a maps API~~
- ~~1.1.19. As a driver I want a link to a navigation app set to my lot of choice so I can easily see directions to the garage I'm going to~~
- ~~1.1.20. As a user I want to be able to add significant arrival locations to reduce friction and make creating plans easier~~
- ~~1.1.21. As a user, I want to set a default "home" or "commute origin" so travel estimates always start from there~~

### **Trip Planning and Calendar:**

- ~~1.1.22. As a user, I want to link my calendar to the app~~
- ~~1.1.23. As a user I want to see a bar chart that shows how full a given lot is at the given time~~
- 1.1.24. As a user I want those logs to be used to see what garages I end up in most frequently by day
- 1.1.25. As a user I want the garage I end up in and at what time to be logged by the app
- ~~1.1.26. As a user I want garages suggested to me based on my logs and where I end up most~~
- 1.1.27. As a user, I want the app to suggest the nearest alternative parking structure when my chosen lot is full.
- ~~1.1.28. As a user, I want a calendar of known closures/events per lot so I can plan ahead~~
- ~~1.1.29. As a user, I want walking time from lot to event/class added to ETA so plans are realistic.~~
- 1.1.30. As a user, I want to see a 'Time-to-Full' estimate on the lot detail screen so I can determine whether the lot could be filled by the time I get there or pick an alternative garage.
- ~~1.1.31. As a user, I want a confidence level on each lot's count (High/Medium/Low) so I can better assess how much I can trust the data for certain lots.~~
- ~~1.1.32. As a user, I want to see the price of parking in different lots so that I can pick the best garage considering my parking budget. (Grant St and Harrison St Garages)~~
- ~~1.1.33. As a a user, I want to see an average user rating for each lot so that I can factor in the quality of different lots when deciding which lot to park in.~~
- ~~1.1.34. As a user, I want to see which lots have covered spots for shade during the hot summer months or protection from the elements like rain and snow.~~

### **Eligibility, Accessibility and Personalization:**

- ~~1.1.35. As a student or employee, I want to filter lots by my parking pass so I can know what garages I'm allowed to park in~~
- ~~1.1.36. As a driver, I want to filter for ADA/accessible parking spots if the university provides that data.~~
- ~~1.1.37. As a user, I want to favorite lots so they show at the top of my list.~~

#### **Analytics:**

- 1.1.38. As a user I want to compare the insights from 2 or more garages
- 1.1.39. As a user, I want a bar chart displaying historical average hourly insights to see when the best time to park in a given garage is.
- 1.1.40. As a user, I want seasonal/weekly based historical insights to see which days it may be easier to park in a week and how seasonal weather affects parking
- 1.1.41. As a developer, I want to train a model to predict parking spots based on historical hourly, weekly, and seasonal trends as well as current data and active users
- 1.1.42. As a user, I want the app to prompt me with alternative transportation methods if most parking lots are expected to be full by a given departure time.

#### **Notifications and Preferences:**

- 1.1.43. As a user I want push notifications that tell me when to leave
- ~~1.1.44. As a user I want push notifications when lots are closing~~
- 1.1.45. As a user I want push notifications when my car is in a lot that is getting towed
- 1.1.46. As a user I want push notifications when my car is frequently associated with a lot that is being towed
- ~~1.1.47. As a user, I want to manage notification preferences (which alerts I get and when).~~
- ~~1.1.48. As a user, I want push notifications when a favorite lot drops below N spaces so I can leave sooner~~
- ~~1.1.49. As a user I want a push notification that tells me when parking passes go on sale.~~
- 1.1.50. As a user I want a push notification after I enter a garage that tells me to park safely between the lines

#### **User Feedback and Sharing:**

- 1.1.51. As a user I want to be able to share a parking schedule with others over text, email e.g.

- ~~1.1.52. As a user I want to see a map with the live counts displayed on top of the garages~~
- 1.1.53. As a user I want to be able to filter that map by parking pass or favorite lot
- ~~1.1.54. As a user, I want to toggle between a list view and a map view depending on my preference.~~
- ~~1.1.55. As a user, I want to report incorrect lot status (e.g., the app says open but it's actually full) so developers can improve data quality.~~
- 1.1.56. As a user, I want to rate accuracy ("was this lot accurate?") so the model improves.

#### **User Interface:**

- ~~1.1.57. As a user I want an attractive app icon that follows the new liquid glass design language for iOS users.~~
- ~~1.1.58. As a user, I want the color scheme of the app to follow the Purdue color scheme as a Purdue-associated app.~~
- 1.1.59. As a user I want smooth animations that make using the app more pleasant

#### **Miscellaneous:**

- 1.1.60. As a student or staff member, I want the app to correctly work on the latest versions of iOS and/or Android so that I can use it no matter my phone
- ~~1.1.61. As a developer I want to set up a Redis cache and a Postgres SQL database that stores the data and sends it to the frontend~~
- 1.1.62. As a user, I want fallback messaging ("Data temporarily unavailable") instead of blank screens.
- 1.1.63. As a user, I want the app to integrate with my car's infotainment system (CarPlay/Android Auto) so I don't have to look at my phone while driving. (If time permits)
- 1.1.64. As a developer, I want health checks for cameras/CV services so I'm alerted when a feed drops.
- 1.1.65. As a developer, I want structured logs + metrics (latency, error rate, queue lag) so I can trace problems.
- 1.1.66. As a developer, I want to import permit rules from a campus feed so eligibility stays current.
- 1.1.67. As a developer, I want to ingest event calendars (athletics, concerts) so predicted demand is accurate.

## **1.2. Non-Functional**

### **1.2.1. Architecture and Performance:**

Our Project's Frontend will be written in React Native for cross platform delivery in iOS and Android. The backend will be implemented in

Java/Python and will be deployed as containerized microservices. Redis will be present to provide low latency caching for live counts. PostgreSQL will be used as the main database to store daily counts, analytics and metadata. A lightweight computer-vision service will process camera streams and publish enter/exit events to the backend. We will target an end-to-end median response time of  $< 500$  ms for API reads with  $95\% < 1$  s and a mobile app cold start time of  $< 2$  s on mid-range devices. The live availability counter will reflect events with a freshness target of  $\leq 10$  s from vehicle passage to user display with  $95\% < 20$  s. The system should support  $\geq 1,000$  concurrent users. The app should have 24hr uptime, apart from scheduled maintenance and outages.

#### **1.2.2. Security and Privacy:**

All network communication will use TLS 1.2+. Authentication will support Sign in with Apple/Google and email/password and tokens will be short-lived with secure refresh. At rest, Redis and PostgreSQL volumes will be encrypted; secrets will be managed via environment variables with no secrets in source control. Camera processing will not store raw video frames unless explicitly authorized for test/debug, the default pipeline emits counting events only (no faces/plates). Any temporary debug captures will be time-boxed and access-controlled. Access to admin tools will require role-based access control.

#### **1.2.3. Usability and Design:**

The interface will be designed for one-handed use, with primary actions reachable within 2 taps from the home screen. We will support dark/light modes, large-text settings, and ensure high contrast for key screens. Map and list views will be interchangeable, with persistent preferences. Critical information such as lot status and “Last updated <timestamp>” will be visible at a glance. We will implement a functional UI with smooth transitions that will look pleasing on both Android and iOS devices.

#### **1.2.4. Compliance and Ethics:**

We will respect campus policies on camera placement and data handling, and commit to privacy-preserving CV (no biometric identification, no license plate retention). Location and calendar access will be opt-in with clear purposes stated. User-generated feedback will be moderated to remove sensitive data or misuse. If required by campus policy, we will provide a Data Protection Impact Summary for review.