Product Backlog - Group 6

Pranay Nandkeolyar, Utkarsh Majithia, Anthony McCrovitz, Logan Portscheller, George Samra, and Neel Vachhani

Project Name: Purdue Parking App

1. Problem Statement

Purdue's campus, with nearly 50,000 students plus faculty, staff, and visitors, lacks a way to view real-time parking availability. Current resources only list parking options but provide no information on open spaces, leaving commuters frustrated and wasting time searching for spots. Our app will provide live availability data to help users make faster, more informed commuting and parking decisions.

2. Background Information

Audience:

The Purdue Parking App serves the campus commuting community: students, faculty, staff, and visitors who need guidance before they arrive on campus and before entering a lot or garage. At Purdue, an estimated 58% of students (30,000 people) commute, not including staff, faculty, and visitors. With roughly 6,000 nearby parking spots, capacity fills quickly during peak hours, creating stress, delays, and even academic or personal issues. Our domain is campus mobility & parking information systems, focused on real-time lot availability for a university setting. Our stakeholders include Purdue Parking Services, Purdue Police (for events/closures), and commuter groups across students and staff.

Similar Platforms:

Several apps list parking locations, capacities, and general rules, but they typically do not provide real-time availability. At Purdue specifically, ParkMobile and other third-party tools offer functionality for street parking and for select cities, yet there is no equivalent service for Purdue's campus lots and garages. Existing campus resources primarily present static information, names of lots, total spaces, and eligibility rules, without live counts to inform time-sensitive decisions.

Limitations:

While current solutions help users discover where parking exists, they fall short on live availability, and thus are often inadequate or inaccurate. They often also omit campus lots/garages at Purdue, leaving a gap precisely where demand is highest. Our app

addresses these limitations by delivering real-time counts of available spaces for Purdue lots/garages, so commuters can plan before they depart and reroute as needed.

3. Requirements/Backlog

3.1. Functional

Authentication and Onboarding:

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

Core Functionality:

- 3.1.5. As a driver I want a screen that shows updated counts for each garage
- 3.1.6. 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)
- 3.1.7. As a user, I want to see a "Last updated <timestamp>" label on each lot so I can trust freshness.
- 3.1.8. As a user, I want to search for lots by name/code so I can find them quickly.
- 3.1.9. As a user I want to have a map with all of the garages/lots listed on them
- 3.1.10. As a user, I want a detailed view (hours, floors, rules, walking time to destination) so I can evaluate options.
- 3.1.11. 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:

- 3.1.12. As a developer I want to set up a camera in a parking lot
- 3.1.13. As a developer I want to use computer vision to detect cars using online training data
- 3.1.14. As a developer I want the computer vision algorithm to not detect other objects or vehicles that are not cars
- 3.1.15. As a user I want an accurate count of total parking spots in all garages
- 3.1.16. As a developer, I want to sync the camera to the CV model
- 3.1.17. 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:

3.1.18. As a user I want to know when to leave my house to make it to events on time using a maps API

3.1.19. 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

3.1.20. As a user I want to be able to add significant arrival locations to reduce friction and make creating plans easier

3.1.21. As a user, I want to set a default "home" or "commute origin" so travel estimates always start from there.

Trip Planning and Calendar:

3.1.22. As a user, I want to link my calendar to the app

3.1.23. As a user I want to see a bar chart that shows how full a given lot is at the given time

3.1.24. As a user I want those logs to be used to see what garages I end up in most frequently by day

3.1.25. As a user I want the garage I end up in and at what time to be logged by the app

3.1.26. As a user I want garages suggested to me based on my logs and where I end up most

3.1.27. As a user, I want the app to suggest the nearest alternative parking structure when my chosen lot is full.

3.1.28. As a user, I want a calendar of known closures/events per lot so I can plan ahead

3.1.29. As a user, I want walking time from lot to event/class added to ETA so plans are realistic.

3.1.30. 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.

3.1.31. 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.

3.1.32. 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)

3.1.33. 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.

3.1.34. 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:

3.1.35. 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

3.1.36. As a driver, I want to filter for ADA/accessible parking spots if the university provides that data.

3.1.37. As a user, I want to favorite lots so they show at the top of my list.

Analytics:

3.1.38. As a user I want to compare the insights from 2 or more garages

3.1.39. 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.

3.1.40. 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

3.1.41. 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

3.1.42. 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:

3.1.43. As a user I want push notifications that tell me when to leave

3.1.44. As a user I want push notifications when lots are closing

3.1.45. As a user I want push notifications when my car is in a lot that is getting towed

3.1.46. As a user I want push notifications when my car is frequently associated with a lot that is being towed

3.1.47. As a user, I want to manage notification preferences (which alerts I get and when).

3.1.48. As a user, I want push notifications when a favorite lot drops below N spaces so I can leave sooner

3.1.49. As a user I want a push notification that tells me when parking passes go on sale

3.1.50. 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:

3.1.51. As a user I want to be able to share a parking schedule with others over text, email e.g.
3.1.52. As a user I want to see a map with the live counts displayed on top of the garages
3.1.53. As a user I want to be able to filter that map by parking pass or favorite lot
3.1.54. As a user, I want to toggle between a list view and a map view depending on my preference.
3.1.55. 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.
3.1.56. As a user, I want to rate accuracy ("was this lot accurate?") so the model improves.

User Interface:

3.1.57. As a user I want an attractive app icon that follows the new liquid glass design language for iOS users
3.1.58. As a user, I want the color scheme of the app to follow the Purdue color scheme as a Purdue-associated app.
3.1.59. As a user I want smooth animations that make using the app more pleasant
3.1.60. As a user, I want to choose dark mode/light mode so the app fits my phone theme

Miscellaneous:

3.1.61. As a student or staff member, I want the app to correctly work on the latest versions or ios and/or android so that I can use it no matter my phone
3.1.62. 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
3.1.63. As a user, I want fallback messaging ("Data temporarily unavailable") instead of blank screens.
3.1.64. 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)
3.1.65. As a developer, I want health checks for cameras/CV services so I'm alerted when a feed drops.
3.1.66. As a developer, I want structured logs + metrics (latency, error rate, queue lag) so I can trace problems.
3.1.67. As a developer, I want to import permit rules from a campus feed so eligibility stays current.
3.1.68. As a developer, I want to ingest event calendars (athletics, concerts) so predicted demand is accurate.

3.2. Non-Functional

3.2.1. 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. PostgresSQL 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% < 1s 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 ≤ 10 s from vehicle passage to user display with 95% < 20s. The system should support ≥ 1,000 concurrent users. The app should have 24hr uptime, apart from scheduled maintenance and outages.

3.2.2. 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.

3.2.3. 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.

3.2.4. 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.