Parking Spotter

Magnus Bøndergaard Poulsen - 202008368

August Borup Exner - 202005782

Patrick Engelbrecht Sørensen - 202005459

AARHUS
UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

App vision

Our app is a navigation tool designed to help users in Vejle city locate available parking spots with ease. By accessing real-time data from the API Vejle Parking, the app keeps track of parking availability across the city. Users can enter their desired destination, such as "Vejle Rådhus", into a search bar. The app then retrieves parking options, which are displayed in a customizable list that can be sorted by proximity, price, or availability of spots. When a user selects a parking location, the app will launch Google Maps to guide them there. If the chosen lot becomes full or reaches a low availability threshold, the app will automatically redirect them to another nearby lot with available spots, ensuring a seamless parking experience.

User-story

A driver visiting Vejle wants a convenient way to find the closest available parking spot to their destination without having to circle around unfamiliar streets. When the user opens the app, he enters the desired location "Bryggen Vejle" in a search bar. The app then shows a list of available parking spots in the area, automatically sorted by proximity to the driver's destination. Once the driver selects a parking option, the app seamlessly connects to Google Maps, providing GPS navigation to the chosen spot.

Features:

- Firebase,
- Firestore,
- Google Maps API,
- Google Directions API,
- Google GeoCoding API,
- Vejle Parkering API,
- Mobile location,
- Notifications
- Rerouting

Early design overview:

Screen 1: Frontpage with map
Screen 2: Profile page
Screen 3: Settings page
Screen 4: Map with parking overview

Ending in Google maps navigation as seen in figure 1.

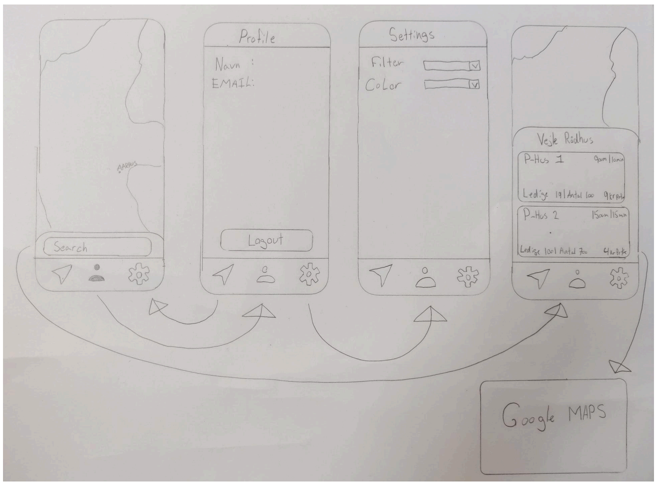
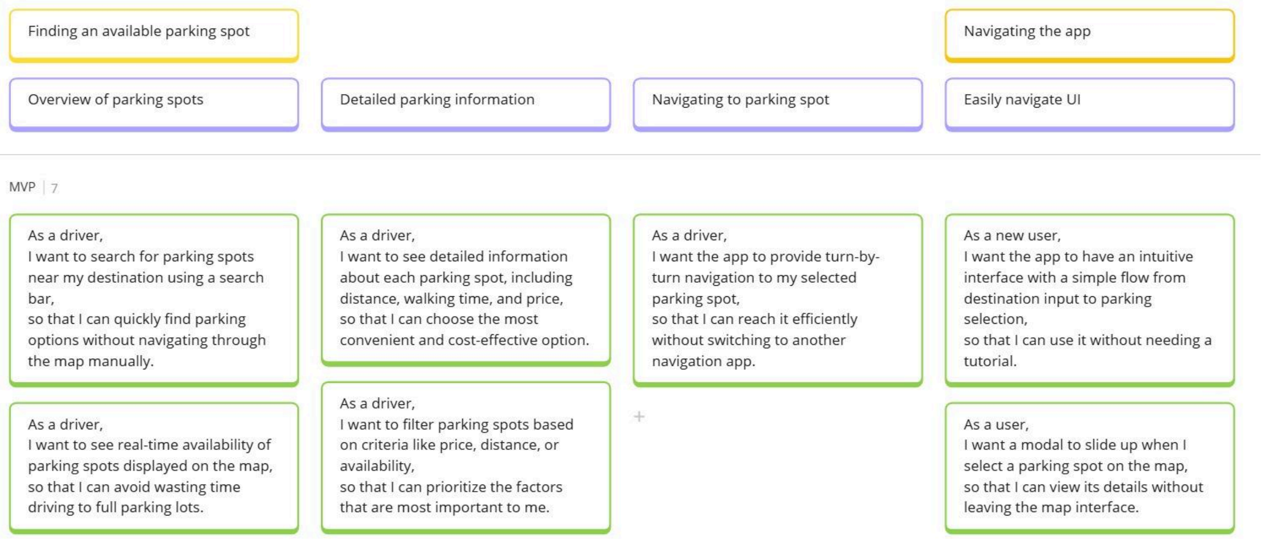


Figure 1: Early design overview

Features and Requirements Specification

User Stories

In the figure below we show relevant user stories. This includes the activity of finding an available parking spot, with tasks of overview of parking spots, detailed parking information and navigating to parking spots. The stories describe features such as search bar for navigating to parking spots near final destination, real-time availability of parking spots avoiding waste of time and detailed information and filtering of criterias like, price, distance or availability.
Furthermore, an activity for navigating the app, with stories of an intuitive interface and simple flow, as well as a slide up modal to view parking details.



Finding an available parking spot

Navigating the app

Overview of parking spots

Detailed parking information

Navigating to parking spot

Easily navigate UI

MVP | 7

As a driver,
I want to search for parking spots near my destination using a search bar,
so that I can quickly find parking options without navigating through the map manually.

As a driver,
I want to see real-time availability of parking spots displayed on the map,
so that I can avoid wasting time driving to full parking lots.

As a driver,
I want to see detailed information about each parking spot, including distance, walking time, and price,
so that I can choose the most convenient and cost-effective option.

As a driver,
I want to filter parking spots based on criteria like price, distance, or availability,
so that I can prioritize the factors that are most important to me.

As a driver,
I want the app to provide turn-by-turn navigation to my selected parking spot,
so that I can reach it efficiently without switching to another navigation app.

As a new user,
I want the app to have an intuitive interface with a simple flow from destination input to parking selection,
so that I can use it without needing a tutorial.

As a user,
I want a modal to slide up when I select a parking spot on the map,
so that I can view its details without leaving the map interface.

Figure 2: User stories

UI Diagrams/Sketches

As seen in figure 3 the sign up and sign in screens consists of “Parking Spotter” as title, two text fields, which requires an email and password as input. Then a sign up or sign in button depending on the screen and two text buttons which can switch between the sign up and sign in screen.

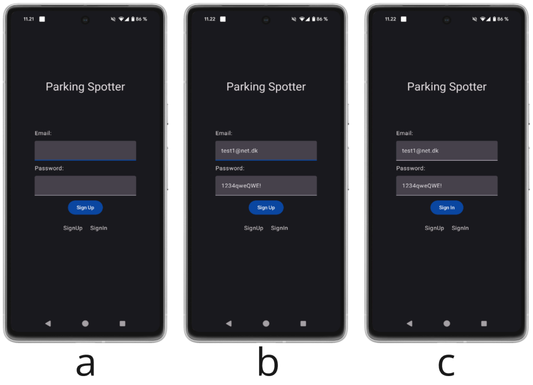


Figure 3: Sign in and sign up screens

The home screen can be seen in figure 4.a with search bar, map and bottom menu for navigation between the screen. 4.b,c shows the destination “Vejle Rådhus” with custom markers on the map showing parking lot information like, availability and price as well as a modal bottom sheet showing further information such as walking distance and time to destination. 4.d is showing a dialog message to let the user confirm if they want to navigate to the selected location. In 4.e Google Maps navigation has been started and a notification is shown to let the user know that navigation is redirected to next best parking lot as the selected one has been occupied.

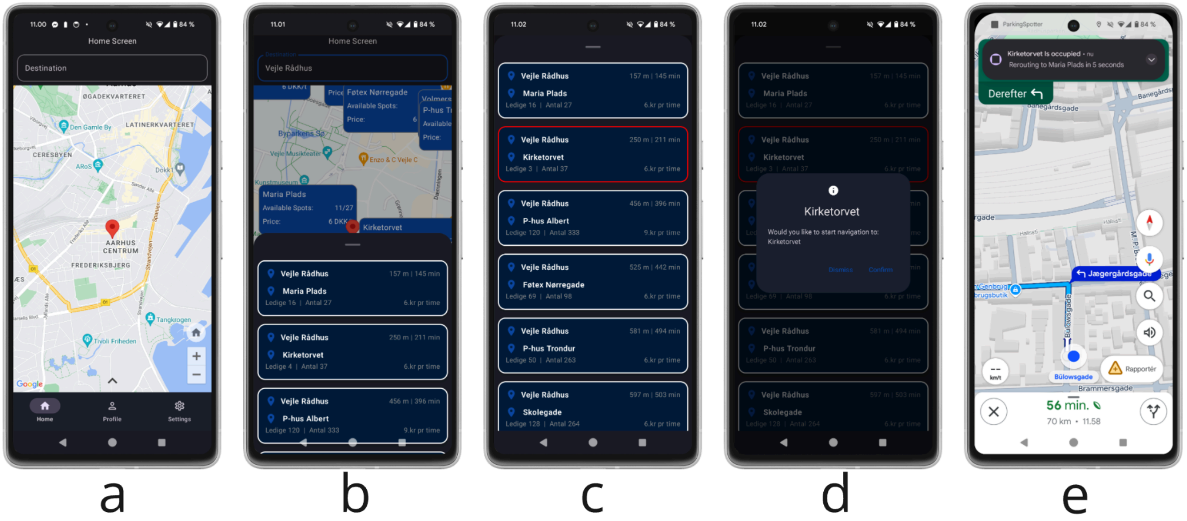


Figure 4: Home screen with search and navigation to destination

The profile screen with the user email and two empty text fields of name and home address is seen in figure 5.a. 5.b shows a user inputting name and home address into the text fields. 5.c shows the saved state of the name and home address. In 5.d the user has navigated to the home screen and pushed the “Home” button icon to navigate home which gives a dialog message for the user to confirm. 5.e shows the navigation to home in Google Maps.

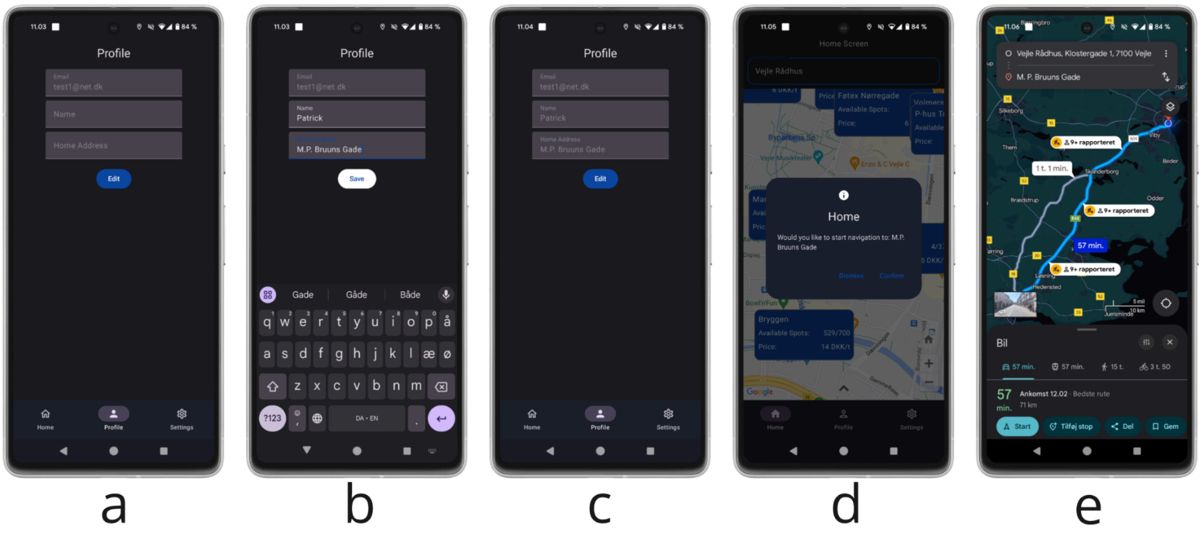


Figure 5: Profile screen with navigation to home address

In figure 6.a the settings screen shows the filtering options, available spots, price and distance, in a drop down list, which can be used for parking spot selection. In 6.b, further settings are shown including selection of themes. 6. c,d and e show the parking selection with different filtering options highlighting the order difference.

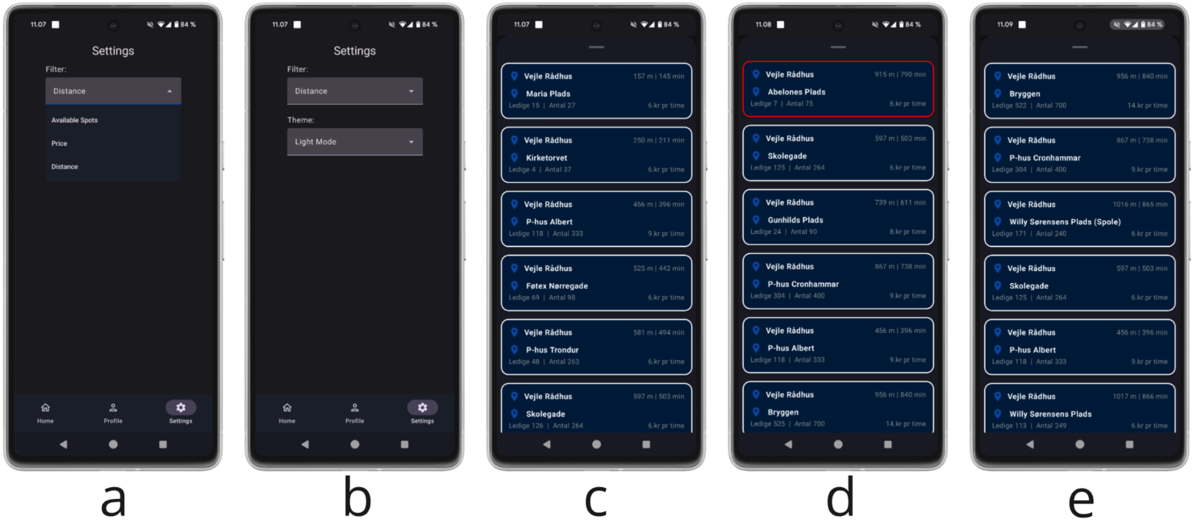


Figure 6: Setting screen showing filtering options

The user flow seen in figure 7 starts with a sign up and sign in screen, after login the user is directed to the home screen. From the home screen three options are available. The user can input a destination to the search bar, which directs the through to destination, from destination the user decides upon a parking spot and is then directed to selection where the user has to confirm the selection of parking spot and finally directs to Google Maps for navigation. Secondly, the user can navigate to the profile screen that displays user information or lastly, the user can navigate to the settings screen which shows filtering options for the parking options.

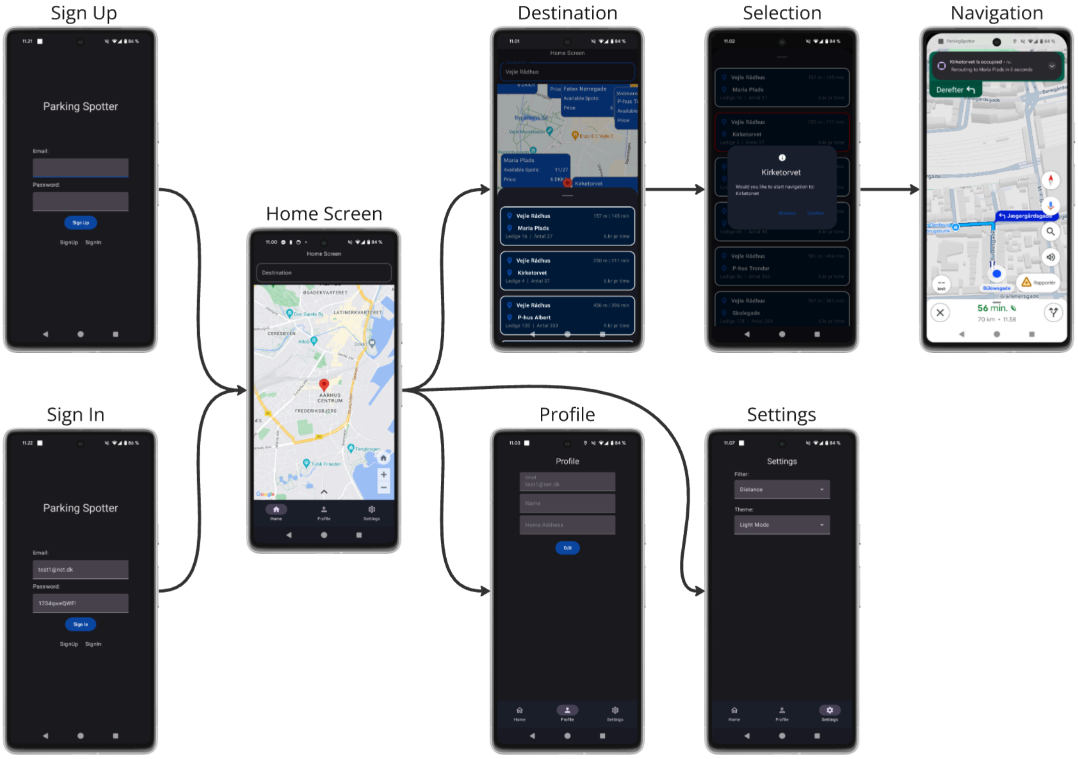


Figure 7: User flow

Component Diagram

The diagram in figure 8 shows the core concepts of how the app works. From the Navigation screen the user is either choosing between login or signup screen. When the firebase authentications service validates the input and account the navigation creates a user viewmodel and changes its content to house the home screen as well as a tab bar for the bottom menu. The home screen fetches the Vejle parking API, shows Google maps, and gets the user location. When searching from the home screen the Vejle parking API results is shown as a list of cards in a drawer, which also utilizes the Google directions API to calculate the remaining distance and time from the individual parking spot to the end destination.

When navigating to the profile screen the user has the ability to input their home address and name which then updates the firebase database accordingly. The home address can be used on the home screen home button for a quick navigation to home.

The last menu item is the settings screen, from there the user can change the filtering which is how they want to sort the parking spots shown in the bottom drawer. This is also updated in the user viewmodel and firebase database.

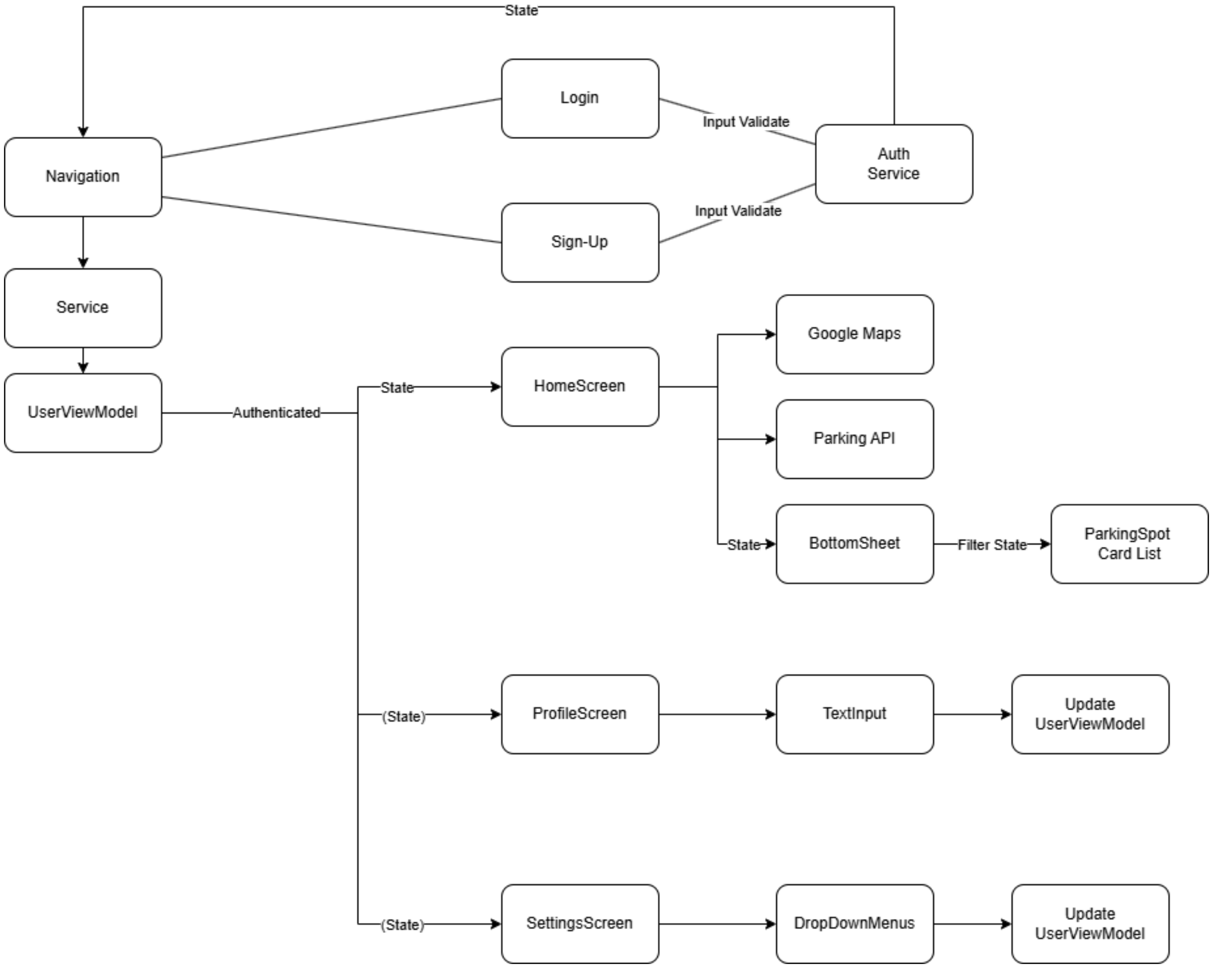


Figure 8: UML diagram - Showing component

Use Case Diagram

As seen in the diagram below there are multiple notable use cases. After signing up and in, the user can add name and home address to enable the "navigation to home" button on the map, inside the home screen. These variables as well as the change of settings are then stored to firestore database and state is saved to the app. An additional use case is to enter the final destination into the search bar, which shows the surrounding parking lots. Furthermore, a filtering feature can be used to filter the order of the parking lots when choosing in the modal bottom sheet. Finally, clicking one of these parking spots on the modal bottom sheet or the parking markers on the map activates a prompt to open Google Maps and navigate to the specific parking spot.

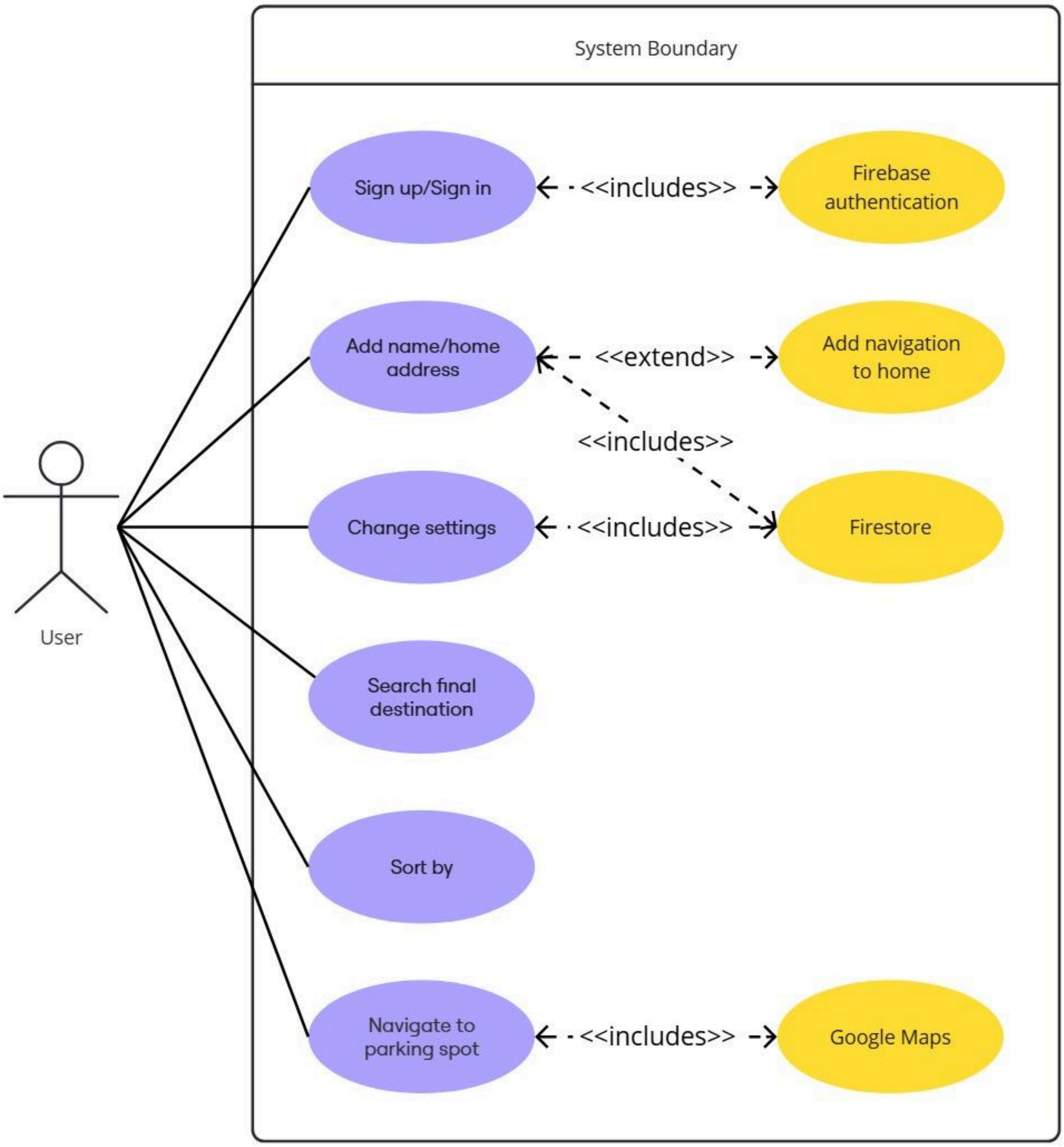


Figure 9: Use case diagram

Conclusion

In conclusion, Parking Spotter is possible to help with the complexity of handling parking in larger cities, specifically to gain knowledge about parking lots in real-time such as, availability, prices and driving/walking distances as well as reroute the user once the chosen parking spot becomes unavailable. Some of the challenges that still remain is how to make the navigation as seamless as possible. Now that we have integrated real-time information, it should be possible to navigate the driver directly to a free parking lot without having to change apps. However, the current technology stack needs changing to incorporate navigation directly into Parking Spotter to provide the most effective and seamless experience. Furthermore, as real-time availability of data for parking lots is scarce, further development of the sensor network is needed to provide enough data to give Parking Spotter the opportunity to show its full potential.

Known Bugs and Issues

- Slow loading time generally, map and modal bottom sheet.
- Reloading home screen/map after every screen switch
- Size of custom markers (parking information) should scale differently
- If editing profile name and home address both get automatically reset instead of letting the user make changes on the existing.
- Setting only works for the filter variable and not the theme yet.
- Modal bottom sheet sometimes loads incorrectly and needs reloading to show correct information.

Work Plan

All of the group members have generally contributed to most parts of the app. However, there have been specific focus areas for each. The ones not specified have mainly been contributed equally.

August

- Home screen
- Modal bottom sheet
- Google Maps/directions integration

Magnus

- Settings screen
- Rerouting
- Google Geocoding

Patrick

- Profile screen
- Smaller UI tweaks
- Google Geocoding