

# Product Backlog

Group 6: Ade Aiho, Heta Hartzell, Mika Laakkonen, Jonne Roponen

## Priority 1:

### User Story:

As an EV driver, I need a map with pins to know where the charging stations are, so that I can easily find a way to charge my electric car.

### Implementation:

- Set up map component
- Add map pan and zoom controls
- Fetch charging station data
- Display charging stations as pins
- Make pins interactive

### Acceptance criteria:

- There is a map view
- The map displays charging stations as pins on a map
- The map should have pan ability (drag)
- The user can zoom in/out on the map
- Each pin is clickable to show further details (landing not implemented on this level)

Story points: 5

–

### User Story:

As an EV driver, I want the map of charging stations to have a clear distribution of pins, so I can easily find stations near my location.

### Implementation:

- Decide on a clustering solution for pins and add the clustering functionality (a ready library? function for logic?), prevent pin overlap

### Acceptance criteria:

- Pins should group into clusters when many stations are in the same area, and break apart when user zooms in
- Nearby pins should not overlap
- Each pin should represent an individual charging station when zoomed in

Story points: 3

–

User Story:

As an EV driver, I want to use a search engine to quickly locate specific charging stations or areas on the map.

Implementation:

- Create search engine UI (bar component and features)
- Search functionality (process of queries, keyword filtering)
- Display of relevant results

Acceptance criteria:

- A visible search engine
- User can enter keywords such as a station name, city, postal code...
- Search results should display relevant stations on the map (not all!)
- Users should be able to filter search results (provider, connector type)

Story points: 5

–

User Story:

As an EV driver, I need to know the charging speeds of different charging stations, so that I can choose the best one for my needs.

Implementation:

- Fetching and displaying data for charging speeds
- Design display
- Implement filtering by speed

Acceptance criteria:

- Each station shows its maximum charging speed in kW
- Charging speeds are clearly visible when user clicks a pin
- Charging stations can be filtered by speed

Story points: 3

–

User Story:

As a user, I want a visual indicator and filtering option to see whether a charging station is available or occupied, so I can decide which station to go to.

Implementation:

- Fetch real-time data
- Add visual indicator
- Implement updating
- Add filtering option for only available ones

Acceptance criteria:

- Each station displays a visual indicator (green/red)
- Users can filter stations by availability status
- Indicator updates in real-time

Story points: 3

–

User Story:

As a new user, I need to register to the app, so that I can view my profile and enter my personal details.

Implementation:

- Create a form for registration (UI)
- Add validation (email&passwd) and error handling
- Add instant access to profile after validation
- Profile updating options
- Security?

Acceptance criteria:

- Users can register by providing a username, email, and password, with a confirmation step (email) in a new window
- After successful registration, users can access a profile where they can enter or update personal details (e.g., name, preferred charging station type, saved stations)
- An error message appears if registration fails (if the email is already in use)

Story points: 3

–

User Story:

As a registered user, I need the app to remember my login details, so that I don't have to login every time I open the app.

Implementation:

- Add remember me checkbox to login screen
- Automatic login logic?
- Setup automatic logout
- Add manual logout option

Acceptance criteria:

- Users who choose to stay logged in should not need to re-enter login details each time
- A secure option for remembering login details should be available on the login screen ("Remember Me" checkbox).
- Users should be automatically logged out if they haven't used the app for a specified period (e.g., 30 days) or if they manually want to log out

Story points: 3

–

User Story:

As an EV driver, I need to know the prices of different charging stations, so that I can choose the best one according to my needs.

Implementation:

- Fetch pricing data
- Display pricing data on station details
- Real-time updates
- Add filtering option for pricing? low to high?

Acceptance criteria:

- Each charging station displays its pricing details
- Pricing information is visible when the user taps on a charging station pin or views station details
- Prices update regularly to reflect any changes from the provider, ideally in real-time or with periodic refreshes (changes every 15min)
- The user can filter or sort stations based on price to quickly find the most affordable options

Story points: 3

–

## Priority 2:

### User Story:

As an EV driver, I want to sort stations by trusted providers so I can prioritize those that offer my preferred payment options and features.

### Implementation:

- Fetch provider data
- Display provider label or logo on station details
- Create filtering option

### Acceptance criteria:

- Users can view a list of trusted providers and select one OR more to filter results
- Stations display a provider label or logo to make identification visually easy
- The sorting option stays as the user moves across the map or adjusts the zoom level

Story points: 3

–

### User Story:

As a user, the app needs to know my location, so that it's easier for me to locate the nearest station.

### Implementation:

- Make a prompt for location permissions (handle location denial)
- Implement map centering
- Fetch and display nearest stations based on location
- Add a “locate me” button for refreshing

### Acceptance criteria:

- The app prompts the user for location permission on initial use
- Once location access is granted, the app automatically centers the map on the user's current location and displays the nearest stations
- If location permissions are denied, the app still allows the user to manually enter a location to find nearby stations
- The app refreshes the user's location on request (“locate me” arrow button)

Story points: 3

–

#### User Story:

As an EV driver, I want to see each station's payment options so I know whether I have the appropriate payment method.

#### Implementation:

- Fetch payment options data (from provider)
- Display payment options in station details
- Payment methods labels for UI?
- Implement filtering by payment method
- Free stations filter? If there are many

#### Acceptance criteria:

- Each charging station shows its available payment methods, such as credit/debit card, mobile payment apps etc. in the station details
- Payment options are clearly labeled and consistent across stations, making them easy to understand
- If no payment is required at a station (e.g., free charging for 15min), this is informed
- (Users can filter or sort stations by preferred payment methods(?))

#### Story points: 3

–

#### User Story:

As an app user, I want the app to load quickly and handle data efficiently so I have a seamless experience when browsing for stations.

#### Implementation:

- Optimize app load
- Enable data caching (store frequently searched stations? local storage or in-memory?)
- Dynamically load stations data based on map view
- Reduce API call frequency with smart refresh intervals

#### Acceptance criteria:

- The app loads reloads periodically
- Station data is cached to reduce load times for frequently accessed locations and details
- The app dynamically loads stations as users zoom or pan on the map, reducing the load on click
- Data handling is optimized for real-time updates, ensuring minimal delay in station availability and status changes

Story points: 3-5

–

User Story:

As a user, I want to leave reviews, ratings, and comments on the stations I use to inform other drivers.

Implementation:

- Design rating and review window
- Implement review submissions and storage
- Display submitted reviews and average rating
- Allow users to edit and delete reviews
- Add “report” option for moderation
- 

Acceptance criteria:

- Users can leave a star rating (1-5 stars) for each station
- Users can write a short text review or comment, with a character limit to avoid novels
- Submitted reviews should display the username and date of submission
- Users can edit or delete their own reviews, and can report inappropriate comments or reviews for moderation
- The latest reviews and an average rating should be displayed in each station’s details

Story points: 3

–

User Story:

As an EV driver, I need help navigating to the charging station, so that I don't get lost.

Implementation:

- Add a “navigate” button
- Integrate with some external navigation provider
- Display (distance? est. time?)
- Location access!

Acceptance criteria:

- The app provides a “Navigate” button on each charging station’s details page
- Tapping “Navigate” relocates user to navigation app (google maps, apple maps) with the station’s location preloaded

- Users can view distance and estimated travel time to the selected station

Story points: 3

–

User Story:

As a user, I want to report station issues (blocked by cars, malfunction etc.) directly via the app, and see these issues displayed as warnings so other users can be aware.

Implementation:

- Issue reportin UI
- Issue submission and storage logic
- Display warning with automatic expiration
- Apply user confirmation for issues

Acceptance criteria:

- Users can report issues for specific stations, selecting from provided categories (“blocked by car,” “station malfunction”) or adding a custom comment with “else” option
- Once an issue is reported, a warning icon or banner appears on the station’s details, visible to all users
- Issue reports include the time of reporting and display prominently until resolved or automatically expire after a set period (24 hours?)
- Users can view details of reported issues and, confirm that the issue still exists → 24h more added

Story points: 3-5

–

User Story:

As a user, I need personalized recommendations from the app based on my charging history, so that I don't need to think about which charging station to go to every time.

Implementation:

- Track users charging station history
- Develop an algorithm that recommends stations, times etc.
- Display recs when app is launched
- User can update preferences?

Acceptance criteria:



- The app tracks the user's charging history, including frequently used stations, time of use per visit, and preferred charging speeds
- Personalized recommendations appear as suggested stations when opening the site(?)
- The app factors in availability, proximity, and charging preferences (e.g., preferred station provider, connector type) to generate recommendations

Story points: 3

## Optional:

User Story: As an EV driver, I want to be notified by email if electricity prices rise or drop significantly, so I can plan my charging accordingly.

Implementation:

- Add notification settings in user profile view
- Backend functionality for price tracking and alerts
- Configure email notifications
- Implement adjustable threshold
- Monitor notification frequency to prevent spam mail (prevent double mails?)

Acceptance criteria:

- Users can tick a choice box to email notifications for price changes in settings
- Notifications are triggered when electricity prices rise or fall by a predefined threshold
- Emails include the current price and the percentage change
- Users can adjust their notification threshold (5%, 10%, 20%)(?)

Story points: 3-5

–

User Story:

As a user, I want to see predictions of station occupancy based on historical data so I can plan when the best time to charge is, avoiding peak times.

Implementation:

- Analyze historical occupancy data
- Design an algorithm for predictions
- Display prediction for occupancy in station details
- Provide a few days predictions in advance

Acceptance criteria:

- The app displays predicted occupancy levels (e.g., “low,” “moderate,” “high”) for each station based on historical data trends
- Predictions are visible in each station’s details, indicating likely occupancy at specific times of day (e.g., hourly or in time blocks like “morning,” “afternoon”)
- Users can view occupancy trends for upcoming days to help with long-term planning
- Predictions refresh periodically to remain truthful

Story points: 3-5

–

User story:

As an EV driver, I want to see user-uploaded pictures to easily identify the exact charger location when I arrive.

Implementation:

- Enable photo uploads
- Create a photo gallery on **station details** (only visible if there are uploads)
- Allow users to add captions
- Implement reporting system for inappropriate photos, moderator review and removal of flagged photos

Acceptance criteria:

- Users can upload photos for each charging station, with a limit on file size and format
- Photos appear in a gallery on the station's details page, ordered by upload date with the most recent displayed first
- Users can add captions
- A reporting mechanism allows users to flag inappropriate images, which are then reviewed and removed if necessary

Story points: 3

–

User story:

As an EV driver, I want to see a Google Maps street view -image of the charger location, so that I can find it easily.

Implementation:

- Embed Google maps street view
- Enable pan and zoom in street view
- Handle unavailability of street view locations (hide map?)
- Toggle for map view and street view?

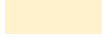




Acceptance criteria:

- Each charging station's detail page includes the map image, that can be changed into an embedded Google Maps Street View image showing the station's location
- Users can pan and zoom within the Street View viewfinder to explore nearby areas

- If Google Street View is unavailable for a location, a message indicates that Street View is not available

Story points: 3

## Elements:

-  = tasks having to do with map
-  = tasks that need fetching req logic
-  = tasks about elements for the station page
-  =
-  =