MyMons

App project 10

Navn	Studienummer
Lasse Borring Petersen	202208165
Benjamin Harboe Strunge	202209864
Esben Inglev	202210050

App Vision

The app allows users to collect and view rare Pokémon, which you can boast about and show off to your friends. Users can catch random Pokémon through the app, store them in their collection, and see where each Pokémon was caught on a map. The app combines fun with technology by integrating Firebase Authentication for secure user accounts, Firestore for storing user and Pokémon data, and PokéAPI for fetching Pokémon information such as names, images, and types.

Each user has a personal profile where they can view account details and manage their collection. The app aims to create a light, gamified experience where users can feel a sense of progression and ownership over their Pokémon while exploring real-world locations through GPS.

User-stories

- Auth:
  - As a user, I want to be able to navigate to a sign-up page, and create an account.
  - As a user, I would like to be able to navigate to a login page, and enter my email and password to login.
- Catch:
  - As a user, I want to navigate to the "Catch pokémon" page and obtain a randomly generated pokémon
- Collection:
  - As a user, I want to view all pokémon that I have caught in an overview.
  - As a user, I want to be able to click on a specific pokémon, to view details about it.
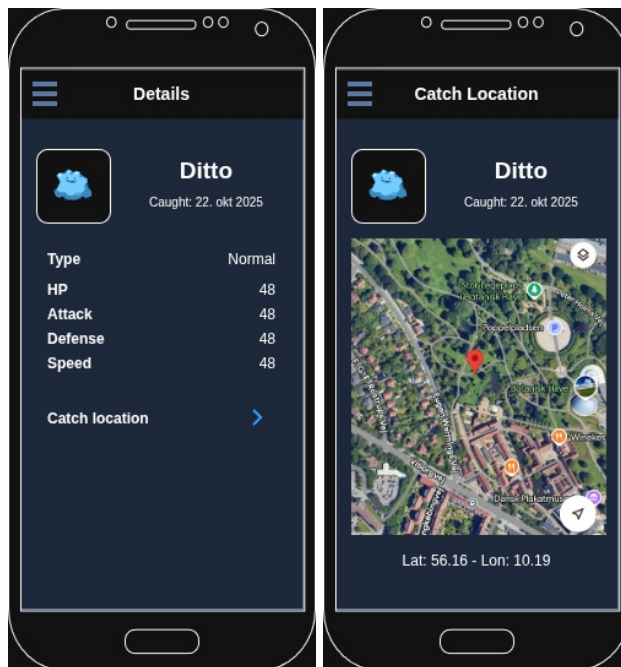 - As a user, I want to view on a map where a specific pokémon was obtained.

- Profile:
  - As a user I want to navigate to the "My profile" page where I can view all the details about my account
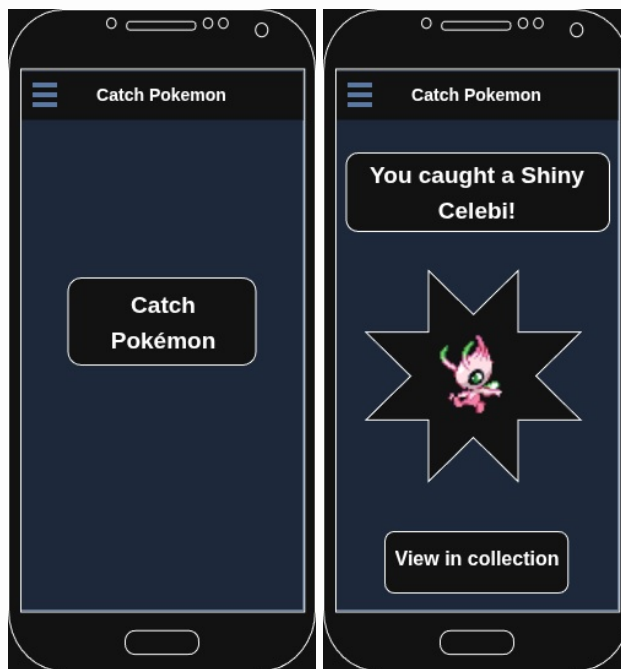
Mockups

Profile page and collection page
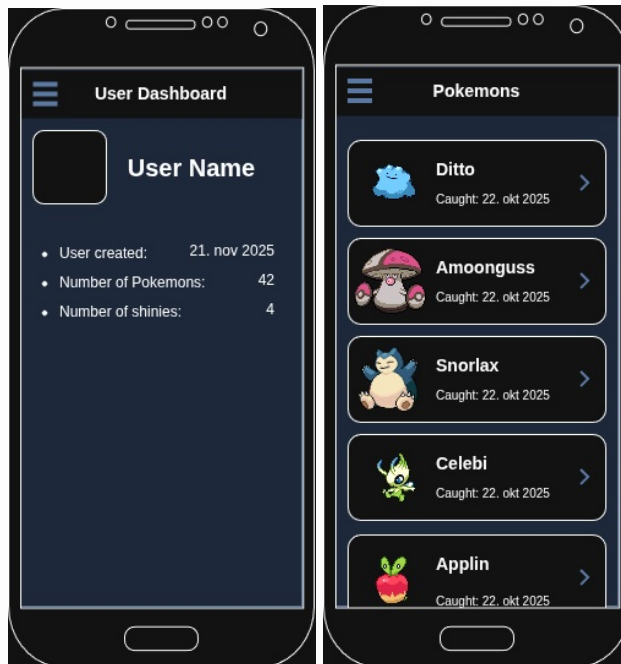
Pokemon details and catch location

Pokemon catch page

List of Technologies

For the database, we will be using Google's firestore database. This database is easy to work with, since its all in the cloud, and there is native support for using it with android apps. It supports multiple ways of querying for data, like shallow queries to only receive data at

the document level, as well as sorting, filtering and limits to your queries. You can also use realtime listeners to continuously update the data that your clients have fetched.

This database choice works well with the firebase 'ecosystem', from which we will also be using firebase auth for our sign-in flow. It features multiple different ways of signing in, with both normal email and password login supported, as well as other identity providers, such as Google- and Facebook-login.

For the pokémon data, we will fetch it from PokéAPI, which is a free restful api that contains all the data about every pokémon.

Each pokémon will have a catch location, this requires a gps service, and to view the location a map service. We will use the Google Maps Platform Api. Google Maps Platform is a collection of all the publicly available information of the global infrastructure. It's a powerful tool with many capabilities.

Main Risks

Any app that uses authentication and authorization, always comes with security risks. This can for example be due to faulty implementation or weak password protection.

We are heavily dependent on an external API for our data. So if the api structure changes, goes down, or we are hit with rate limits, our app could be severely impacted. But looking at the documentation for the API found here, this seems very unlikely to happen.

The app saves the coordinates of where each pokemón was caught. This introduces some privacy risks, as user's approximate locations are collected and stored.

And as we are working with pokémon, history tells us that there is a chance that Nintendo might sue us if it was ever published, so that is also worth remembering. Source