FINAL PROJECT

Web and Mobile Development

Caroline Carlsson

School of Computer Science, Physics and Mathematics, Linnaeus University, Sweden cc222js@student.lnu.se

Torbjörn Holgersson

School of Computer Science, Physics and Mathematics, Linnaeus University, Sweden th222mw@student.lnu.s

INTRODUCTION

In a critical way we have analyzed Instagrams solution as an social media platform and how the users integrate with this platform. The main focus have been on how the users more frequently seem to focus on the text that follows with the image that are posted and less on the actual image itself.

This way of ignoring the image as the pure visualization of a place, object or person also shines thru when the users are allowed to manipulate the image in different ways thanks to Instagram's own editing tools or others and lets the user to inject the image with different filters or texts.

Our aim in this paper is to present a social media application prototype on which the user are allowed to take picture and publish them on a location that are displayed on a map in the application. The users will be presented with three major parts in the application, naimly the profile page, map page and camera page. In these three cornerstones there will be other smaller services and will be presented further down in this paper.

The stack for this project consist of Express, MongoDB, Node.js and Angular (MEAN) but we are using Ionic instead of pure Angular for the frontend.

IONIC

We chose to work with Ionic mostly because we used it in the previous assignment, but also because it is a good framework for us to use with our knowledge of web programming. Additionally it is easy to produce mobile applications with this framework so we decided to stick with Ionic.

USE CASE DIAGRAM

The use case diagram in figure 1 below shows what ways the user are planned to be able to do in the application. When logged in to the application the user can also be logged out. When the user viewing the profile it can choose to edit the profile, deleting an image and also view likes of the pictures the user have added earlier. The user can also upload an image but only the picture that recently been taken. When viewing the map the user can choose to search for relevant images or places and like an image, the user can as well unlike an already liked image.

Since the user only have one way at this point in the planning to log in, namely by using Google sign in, the user class has no need for any other attributes than the user id and email since password and other wanted data when making a user class have been moved to a Google user account.

The image class however was in need for some more attributes due to that it was needed for the planned services on the application such as marking a place with a image that was taken on that spot.

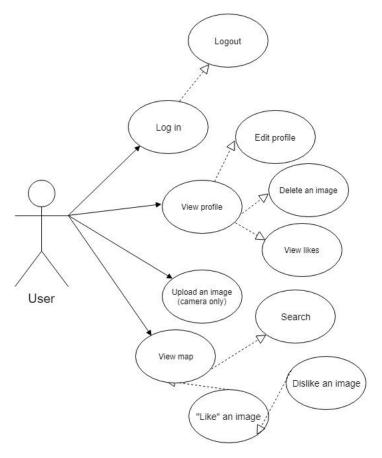


Figure 1. Use case diagram of the application

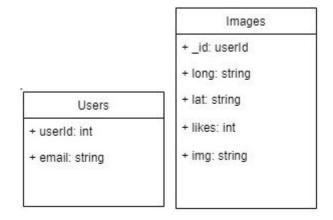


Figure 2. Class diagrams of the user and image collection.

THE MOBILE APPLICATION

The first iteration of the mobile applications design shows three different views on which the user are supposed to encounter. The first image to the left shows a suggestion on how the image flow where the user can browse thru uploaded images could be designed.

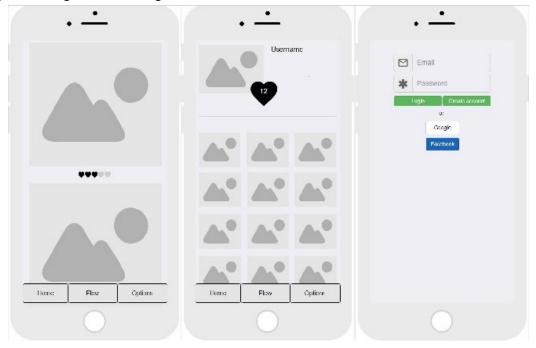


Figure 3. First prototype design of the application.

The image flow service has however been modified to fit more into that the user experience a more concrete relation to the real world and the places the images were taken thanks to the map that has been implemented into the design.

The middle image shows the profile view of the mobile application on which the user can view the images she/he has uploaded to the application, the profile image, the username and how many combined likes the images has.

The last image were the first prototype of how the login page would look like. However we made some changes during the work and decided to only use an external login in order to save time for other parts of the project.

GOOGLE SIGN IN

In order to allow the user to access the mobile application services we have implemented Google Plus Sign. This is a Cordova plugin and it requires a Google plus api key which we retrieved from the Google Developers Console page and is included in the index.html page.

The profile page is placed on tab 1 in the application. There the user should be able to see their email, from logging in with Google, how many images they have uploaded as well as which images this user have liked.

ANGULAR GOOGLE MAPS

To implement the map on tab 2 we have used the Angular module Agm/core. This module uses Google Maps and in order to use this module we also need to retrieve a Google Maps API key.

CAMERA

Tab 3 consists of the camera functionality is implemented with Capacitor. To use the camera we simply installed the Capacitor client to our project and we could then use the client as well as the core in our project.

BACK END

In the back end we aimed to follow the MEAN stack as mentioned before. In the Express server we implemented Express also connects to the mongoDB allowing the server to add images with attached data, deleting images with attached data, adding new users and also to send data back to the application.

With Angular we constructed services in order to be able to implement them into the application. The services consist of one map service and one camera service. The map service allows the application to use Angular Google map component to display a google map for the user using Google API. Furthermore IPAPI is used to obtain a longitude and latitude in order to put a marker on the map. The camera services allows the user to use the camera on a mobile phone in order to take pictures with.

DISCUSSION

We tried using Mongodb Atlas to store our database in the cloud but we encountered some difficulties. We were able to connect and retrieve information about the connections, however we were not able to save any data to this database. We then decided to use a local database for the simplicity. However, Mlab would be an alternative as well if a local storage would not be able but due to that knowledge of accessing the database on a localhost that was the easiest way in order to go on with in the project.

CONCLUSION

We have presented how far we have gotten in this project as well what our plans was, even due we could not fulfill most of them as we did not plan our time right together with other distractions. It is for a great start and alot can come from the spark of this light application.

Node.js version: 8.10.0 Ionic version: 4.10.3

Github: https://github.com/T-bear/4ME305---Web-and-Mobile-Development-Assignment2