Annamária Váradyová
School of Computer Science, Social Media and Web Technology,
Linnaeus University, Sweden
av222xc@student.lnu.se

Albion Morina
Department of Computer Science and Media Technology,
Linnaeus University, Sweden
am224ut@student.lnu.se

Link to our web and mobile application: https://www.youtube.com/watch?v=1z0L-JNmmuI
GitHub Link for our code: https://github.com/Anna8295/web-mobile-app-assig3

Introduction

In this report, we will be discussing a web and mobile application that allows users to login and register if they do not have an account, but, also, the most challenging part was to find and save locations on the web and navigate to them using their phone, also seeing the weather in real time anywhere in the world. The application also provides the ability to take pictures and easily download them on the web. The web and mobile application we are discussing is designed to provide users with a seamless experience when finding locations and navigating to them. Also, users are able to upload a picture in real time in that application after they login in our app. The web application provides users with a map where they can search for specific locations and save them for future reference. The mobile application allows users to navigate to the saved locations and take pictures of the location. The pictures taken on the mobile application can be easily downloaded on the web application. Overall, the web and mobile application we discussed in this report is a versatile and convenient tool that can benefit a wide range of users who need to find, save, and navigate to locations.

Features:

- The web application provides users with the following features:
  - Login and Register page,
  - Map view to search for specific locations,
  - Save locations for future reference,
  - View saved locations on a list,
  - Download pictures taken on the mobile application,
  - Searching for weather at any place,
- The mobile application provides users with the following features:
  - Navigate to saved locations,

- Take pictures of the location,
- Save pictures to the mobile device.

Target Users:

The web and mobile application we are discussing in this report is ideal for individuals who frequently travel, explore new places, or simply enjoy keeping track of their favorite locations. The application can also be useful for individuals who work in industries such as real estate or hospitality, where location-based information is crucial.

Why Target Users Need this App:

Convenience: The application provides users with a simple and easy-to-use platform to find, save, and navigate to locations. With the ability to take pictures on the mobile application and download them on the web application, users can have all their location-related information in one place.

Time-Saving: With the ability to save locations on the web application, users can quickly and easily navigate to their desired location using the mobile application. This feature is especially useful for individuals who are in a hurry or need to visit multiple locations in a short period.

Memory Keeping: The ability to take pictures of locations and save them on the web application provides users with a centralized location to store and manage their memories. This feature is ideal for individuals who enjoy taking pictures and documenting their experiences.

Professional Use: The application can be useful for individuals who work in industries such as real estate or hospitality. For example, real estate agents can use the application to save and manage the locations of their properties, while hotel managers can use it to keep track of local attractions and points of interest for their guests.

Technologies Used:

To develop our web and mobile application, we utilized a variety of technologies, including React, Node.js, and React Native with Expo and also firebase for the storage.

React is a JavaScript library used for building user interfaces, while Node.js is a JavaScript runtime built on the Chrome V8 engine that allows developers to build scalable and high-performance applications. React Native is a framework used to build mobile applications, while Expo is a set of tools and services that helps developers build, deploy, and iterate on native apps. Using React and Node.js for the web application allowed us to create a responsive and dynamic user interface that is scalable and efficient. React's component-based architecture made it easy to develop and maintain the web application, while Node.js provided the necessary backend support for data management and storage.

For the mobile application, we used React Native with Expo to create a cross-platform application that can run on both iOS and Android devices. This allowed us to leverage the power of React Native's native components and Expo's tools and services to create a mobile application that is fast, responsive, and easy to develop. By utilizing these technologies, we were able to create a seamless experience for users across both the web and mobile platforms. The use of React and Node.js for the web application and React Native with Expo for the mobile application allowed us to create a unified system that is efficient, scalable, and easy to maintain.

Existing Solutions in the Market:

There are several existing solutions in the market that provide users with the ability to find and save locations, including Google Maps, TripAdvisor, and Yelp. These applications offer features such as maps, reviews, and ratings, making them useful for users who want to explore new places.

How Our Solution is Better/Different:

Our solution stands out from existing solutions in the market by providing users with a more focused and streamlined experience. Unlike other applications, our solution is designed specifically for users who need to find and save locations and provides a more efficient way to manage location-related information.

One of the key improvements we made to existing solutions is the ability to take pictures on the mobile application and easily download them on the web application. This feature allows users to have all their location-related information, including pictures, in one place, making it easier to manage and share their experiences. Another improvement we made is the ability to save locations on the web application and quickly navigate to them using the mobile application. This feature is especially useful for users who need to visit multiple locations in a short period, as it saves time and eliminates the need to search for the location each time.

Overall, our solution provides a more focused and efficient way to manage location-related information, making it a valuable tool for users who need to find, save, and navigate to locations. With the ability to take pictures, save locations, and quickly navigate to them using the mobile application, our solution provides a more complete and streamlined experience for users.

What was missing in the app before and what did you suggest to add?

One possible way to make the app more interactive is to incorporate social features that allow users to share their saved locations and experiences with others. This could include the ability to create and join groups, share location recommendations with friends, and see where friends have been and what they recommend. Another way to make the app more interactive is to gamify the experience by adding challenges, achievements, and rewards. For example, users could earn badges for visiting certain locations or completing specific challenges related to their saved locations. This could incentivize users to explore more and engage with the app on a more regular basis. Another potential way to make the app more interactive is to incorporate augmented reality (AR) features. For example, users could use their mobile devices to scan a location and get more information about it, or even see virtual elements overlaid on the real-world view. This could provide users with a more immersive and engaging experience, as well as enhance their ability to navigate and explore new locations.

Ultimately, the specific features to add to make the app more interactive will depend on the target user demographic and their preferences. It's important to conduct user research and gather feedback to identify the most valuable and engaging features to add to the app.

Architecture Overview:

Our solution for the web and mobile application involves a client-server architecture, with the web application serving as the client and the server as the backend. The web application is built using React, while the backend is built using Node.js. The mobile application is built using React Native with Expo. And for the storage we have used Firebase.

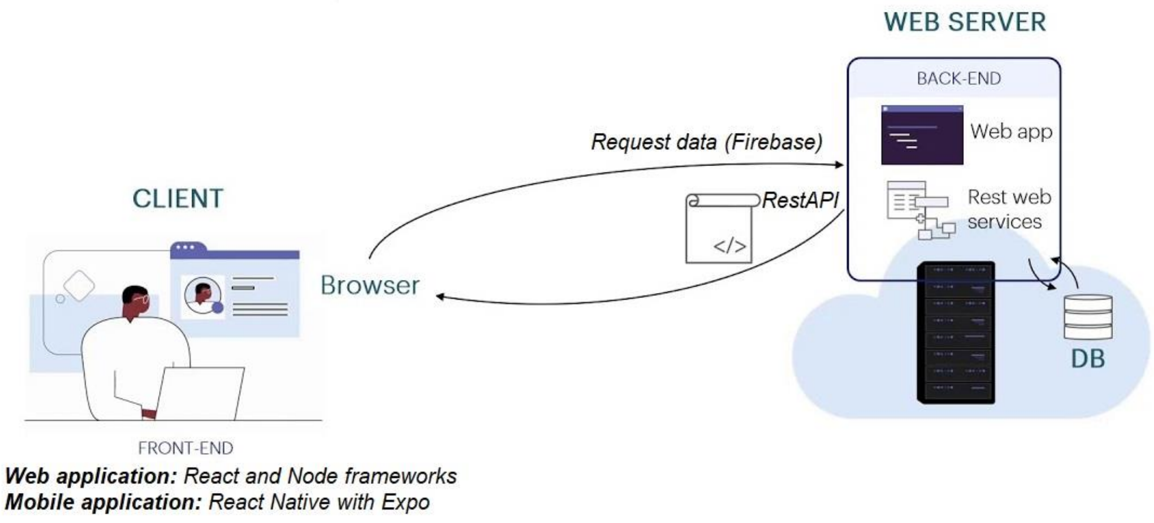

Figure 1 Web and Mobile App Architecture

The web application interacts with the server through a REST API, which allows for the retrieval and storage of location-related information. The mobile application communicates with the server through the same REST API to access and modify location data.

The backend is responsible for processing requests, managing data storage, and ensuring the security and integrity of user data. We chose to use a Firebase database to store the location data, as it is a reliable and widely-used relational database that provides robust data management capabilities.

Argumentation and Motivation of Chosen Framework:

We chose to use React for the web application because it is a widely-used and well-documented framework that allows for the creation of responsive and dynamic user interfaces. Its component-based architecture also makes it easy to develop and maintain the web application, as changes can be made to individual components without affecting the entire application.

For the backend, we chose to use Node.js because of its scalability and ability to handle large amounts of traffic. Node.js also has a vast ecosystem of modules and packages that make it easy to integrate with other technologies and services.

For the mobile application, we chose to use React Native with Expo because of its ability to create cross-platform applications that can run on both iOS and Android devices. This allowed us to develop a single codebase that could be deployed to both platforms, reducing development time and effort. We also chose Expo because of its tools and services that help with building, deploying, and iterating on native apps. Expo provides a variety of features, such as push notifications and in-app purchases, that can be easily integrated into the application.

In summary, our chosen frameworks were motivated by their popularity, ease of use, scalability, and ability to integrate with other technologies. By using these frameworks, we were able to create a robust and efficient system that provides users with a seamless experience across both the web and mobile platforms.

Use-case diagram for application features/functionality (defined in task 2 and 3)?

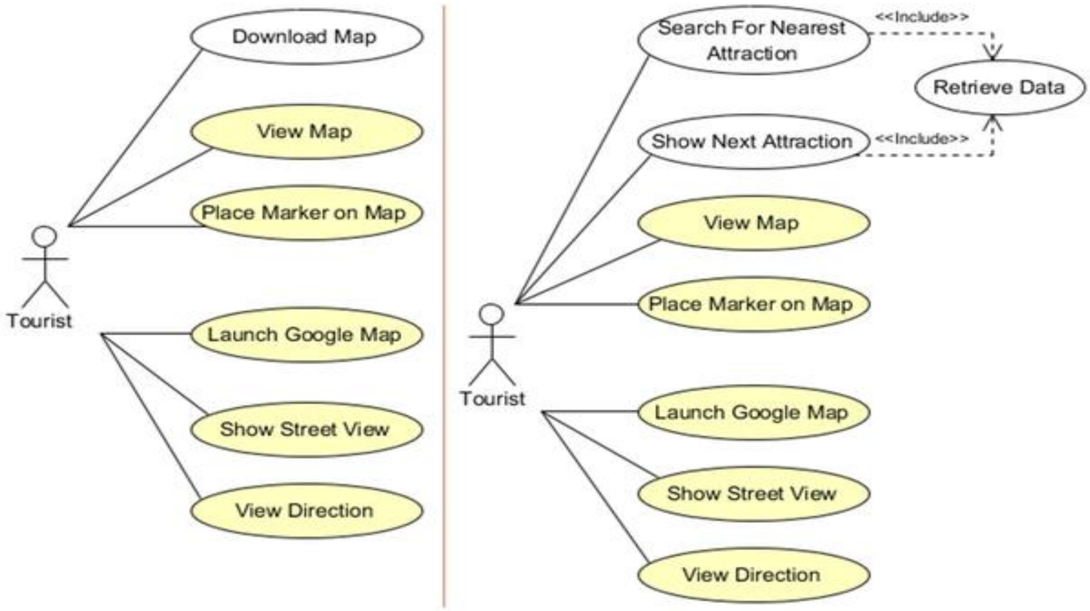

Figure 2 The use-case diagram. We got tourist as an example.

One/two possible scenarios of the usage the app (use one/two of the use-cases presented in the use-case diagram)?

Adding a new location:

Albion is a travel blogger who wants to add a new location to his list of travel destinations. He opens the web application on his laptop and navigates to the "Add Location" page.

On the form, Albion inputs the name, description, and address of the new location. He also uploads a photo he took of the location during his travels. Albion submits the form, and the web application sends a POST request to the server to add the new location to the database. Later, when Albion is out exploring a new city, he opens the mobile application on his phone and navigates to the "Add Location" screen.

He inputs the details of the location he is currently at and takes a photo with his phone. He submits the form, and the mobile application sends a POST request to the server to add the new location to the database. Albion can now view the new location he added on both the web and mobile applications, along with the photo he uploaded.

Finding the way to a saved location:

Annamaria is a tourist who saved a list of locations he wants to visit on the web application. She opens the mobile application on his phone while on the go and navigates to the "Search button" screen. She selects a location from his list and taps on the "Directions" button, which opens the mobile device's default maps application with the location pre-filled. Annamaria can now follow the directions to the location using the maps application, without having to manually enter the address or search for it. Once she arrives at the location, Annamaria takes a photo with her phone using the mobile application and uploads it to the server. Later, when Annamaria is back at her hotel, she opens the web application on her laptop and navigates to the location she just visited. She can now view the photo she took on the location's page and easily download it for her personal use or to share with friends and family.

Implementations details:

Web Application:

We used React to build the web application's user interface. The web application interacts with the backend through a REST API, which is implemented using Node.js framework. We used Axios to handle HTTP requests in the web application. The web application includes a form for users to add new locations, which sends a POST request to the server to add the location to the database. The web application also includes a page to display a list of saved locations, which sends a GET request to the server to retrieve the list of locations from the database. We used the Google Maps API to display a map and location markers on the web application's UI.

Backend:

We used Node.js and Firebase to implement the REST API for the backend. The backend includes routes for handling requests related to location data, such as adding a new location or retrieving a list of saved locations. The backend includes middleware for handling authentication and authorization, to ensure the security and integrity of user data.

Mobile Application:

We used React Native with Expo to build the mobile application. The mobile application interacts with the backend through the same REST API used by the web application. We used Axios to handle HTTP requests in the mobile application. The mobile application includes screens for displaying a list of saved locations and adding a new location, which send HTTP requests to the server to interact with the database. We used the React Native Maps library to display a map and location markers on the mobile application's UI. The mobile application includes functionality for taking and uploading photos, which are stored in the database and can be downloaded from the web application.

These are just some of the implementation details for our solution in our web and mobile application.

References

Aggarwal, S. (2018). Modern Web-Development using ReactJS. *International Journal of Recent Research Aspects*, *Vol 5*(Issue 1), pp133-137.

Kumar, A., & Singh, R. K. (2016). COMPARATIVE ANALYSIS OF ANGULARJS AND REACTJS. *Vol 7*(Issue 8), pp225-227. https://doi.org/http://dx.doi.org/10.21172/1.74.030

Yang, D., Ortuno, M. A., Bernales, V., Cramer, C. J., Gagliardi, L., & Gates, B. C. (2017). Structure and Dynamics of Zr6O8 Metal−Organic Framework Node Surfaces Probed with Ethanol Dehydration as a Catalytic Test Reaction. *Journal of the American Chemical Society*.

Yin, Z., Wan, S., Yang, J., Kurmoo, M., & Zeng, M. H. (2019). Recent advances in post-synthetic modification of metal–organic frameworks: new types and tandem reactions. *Coordination Chemistry Reviews*, *Volume 378*, Pages500-512. https://doi.org/10.1016/j.ccr.2017.11.015

Wu, W. (2018). *React Native vs Flutter, cross-platform mobile application frameworks* (p. 28pages). Metropolia University of Applied Sciences.