

Project Plan – Rario

K428219 – Jaakko Rajala

TIEVA31 – Principles of Programming Graphical User Interfaces, project work, 5cu

Date: 4.6.2021.

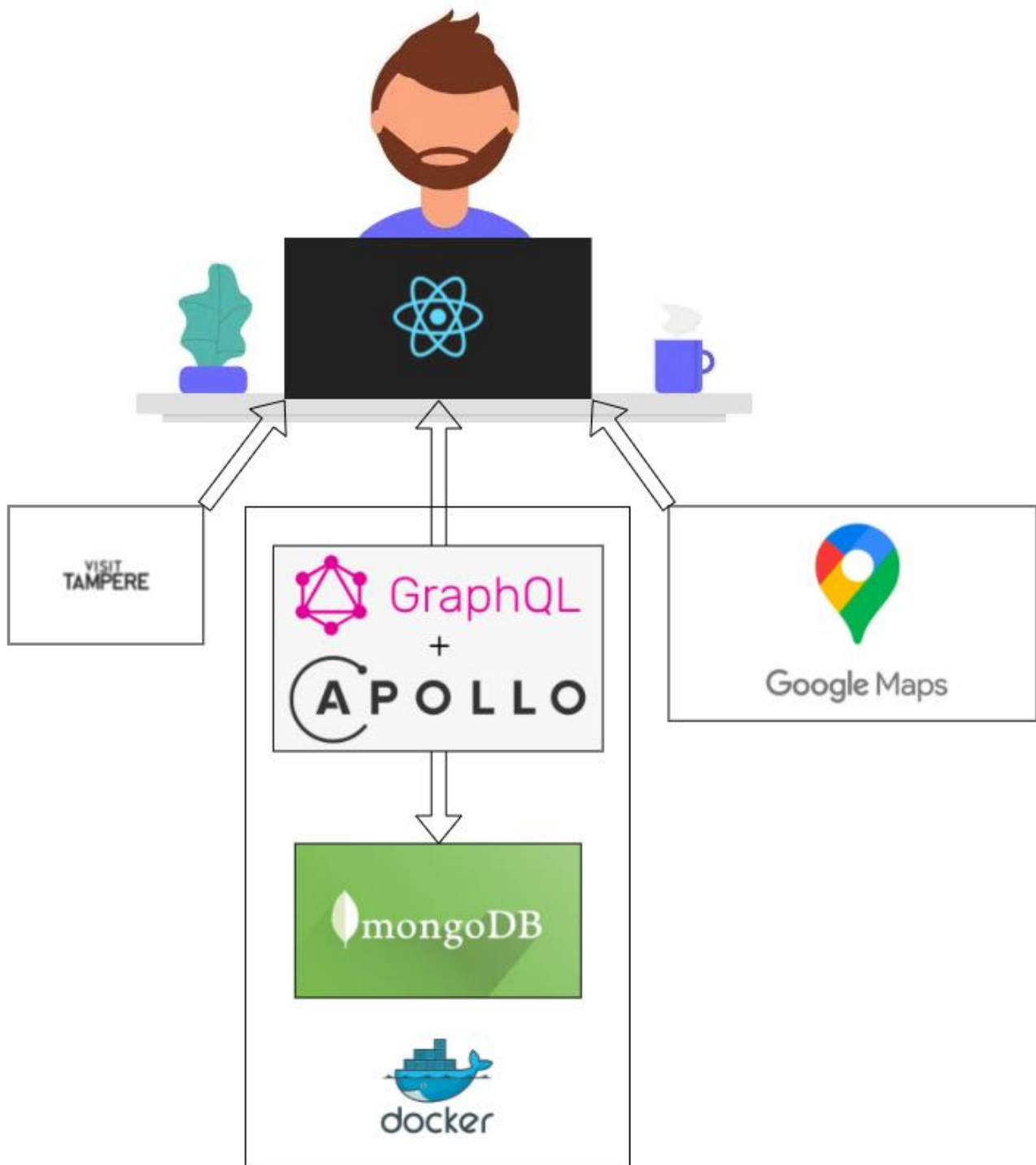


Figure 1 High-level view of the system architecture

Introduction

A Bulletin Board that lets users view local events/happenings in real time- in Tampere. Users log in and receive latest events in the city. It's a full-stack application developed using modern conventions such as microservice architecture, containerization, Reactjs library aswell as material-ui.

Functionality

The application will have the following main functionality:

- Client events:
 - **REQUIRED** - Drawers: Menu
 - Player can navigate to the settings through a drawer-menu.
 - **REQUIRED** - Drag and drop: friends to events.
 - Players can select friends from their list and drop them over the events in the map to invite them to the event.
 - **REQUIRED** - Snackbar: Notifications
 - Player is notified of updates and changes
- Client dynamic manipulation (user):
 - **REQUIRED** - "Login/logout" (Buttons/Form)
 - Login details: **Undo** and **Redo**?
 - **REQUIRED** - "Search" (search-bar)
 - **Search** for events
 - **Display** current events by *Type*
 - **REQUIRED** - "Display event info" (event-icon-button on map)
 - **Display** info that's available on the event
 - **REQUIRED** - "Locate me" (GPS-icon-button)
 - **Find** the user on the map
 - **OPTIONAL** - "Locate friend" (GPS)
 - **Send** current location to friend.
 - **OPTIONAL**- Chat with friends
 - **Send and receive** messages from friends through wb-protocol.
- Database dynamic manipulation:
 - **REQUIRED** - User Account:
 - Add new user
 - remove user
 - **REQUIRED** - Events:
 - ADD a rating (max 5 stars) to an event
 - ADD a new event
 - UPDATE a new event (If owner)
 - REMOVE a new event (if owner)
 - **OPTIONAL** - Friends
 - ADD new friend
 - GET friends
 - REMOVE friend
- Settings:
 - **Dark-mode**: on/off
 - Kieli/language settings, **internationalization**: Finnish/English/Swedish?

- (Table-component) Profile: change
 - Avatar (profile Picture)
 - Password
- keyboard shortcuts and support for keyboard navigation
 - View Friend list (ctrl + y)
 - Open settings (ctrl + s)
- custom component with custom painting (or possibly custom html visuals, e.g., in React)
 - Logo (roof ridges of tampere + näsineula)

User Interface

I've included a couple of wireframes to display how the interface will work.

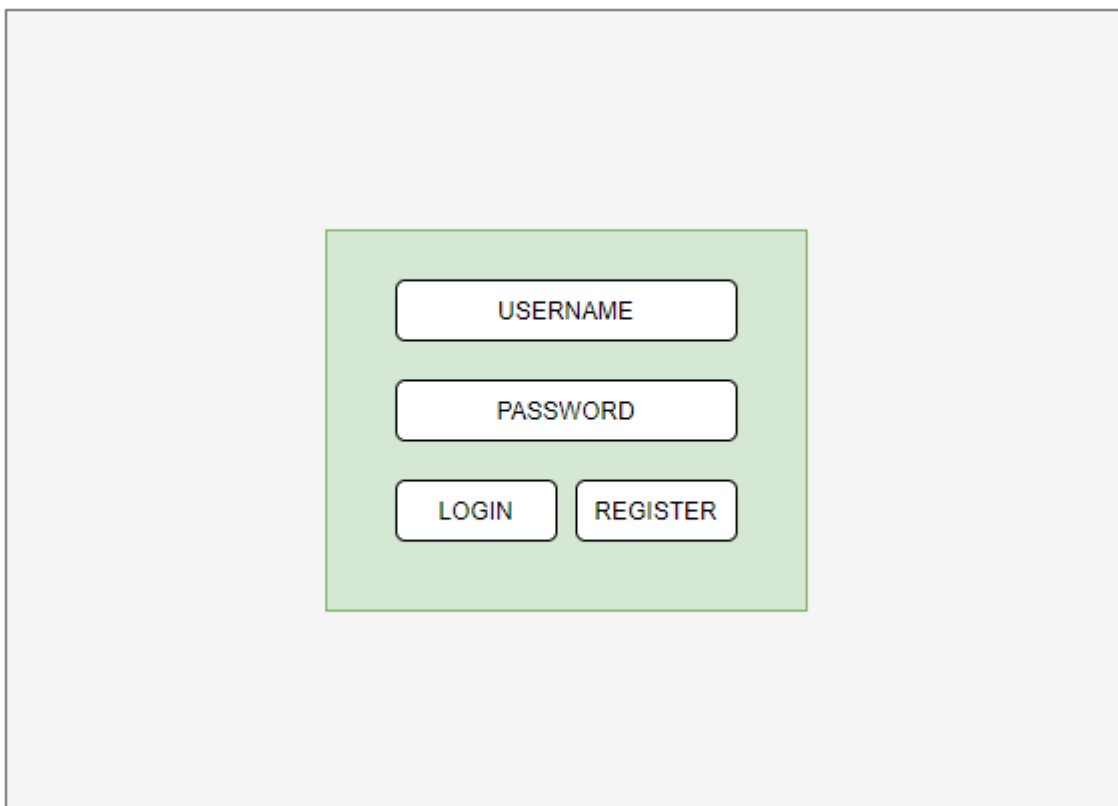


Figure 2 Login screen

The first example is of the login screen. This includes two text-fields and two buttons (register and login). Register takes the user to a page where they fill out their information. Possible OAuth through facebook/google account could be used.

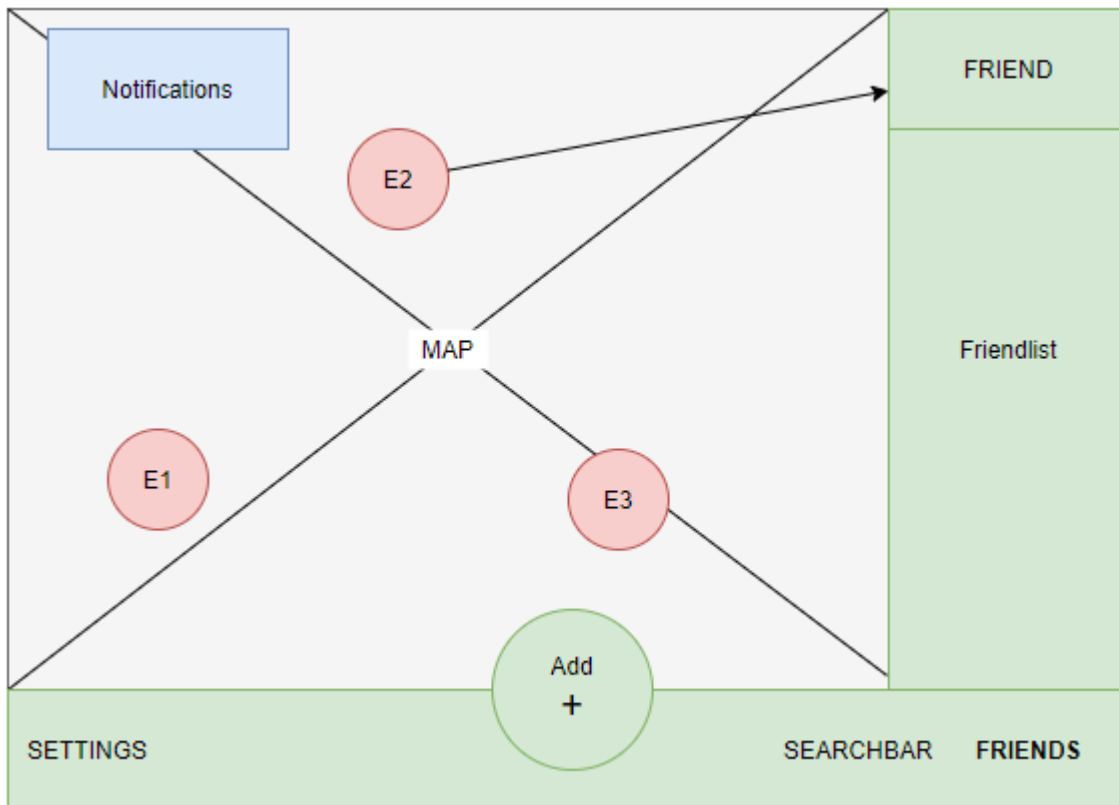


Figure 3 Main screen with Friends-drawer open

Next is the main screen/view in a situation where the user has their friends-drawer open. They can use this opportunity to **drag** one of the events displayed on the map on top of one of their friends. When done so, the friend receives an invitation to take part in the event.

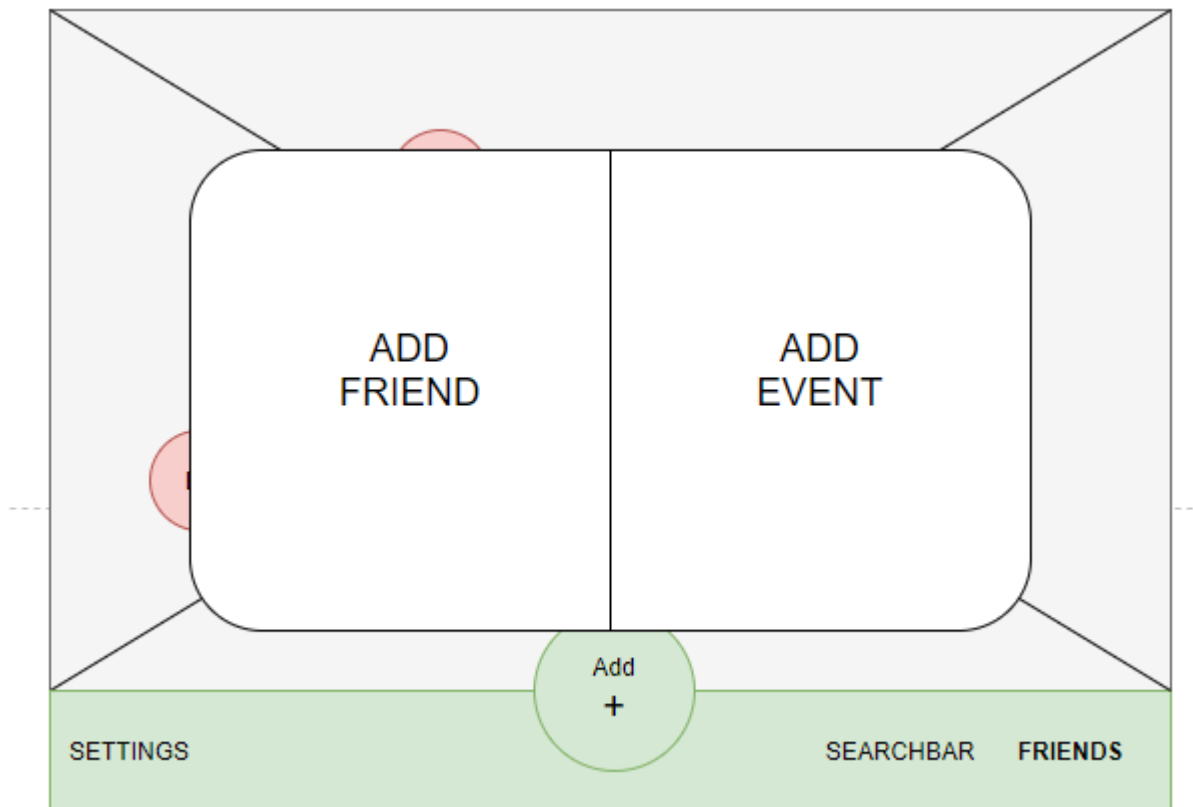


Figure 4 Dialog to add a new event or friend

Next we have a dialog which opens from the add button in the middle. This lets the user add a friend or an event. These options open their own separate components that initiate the desired adding process.

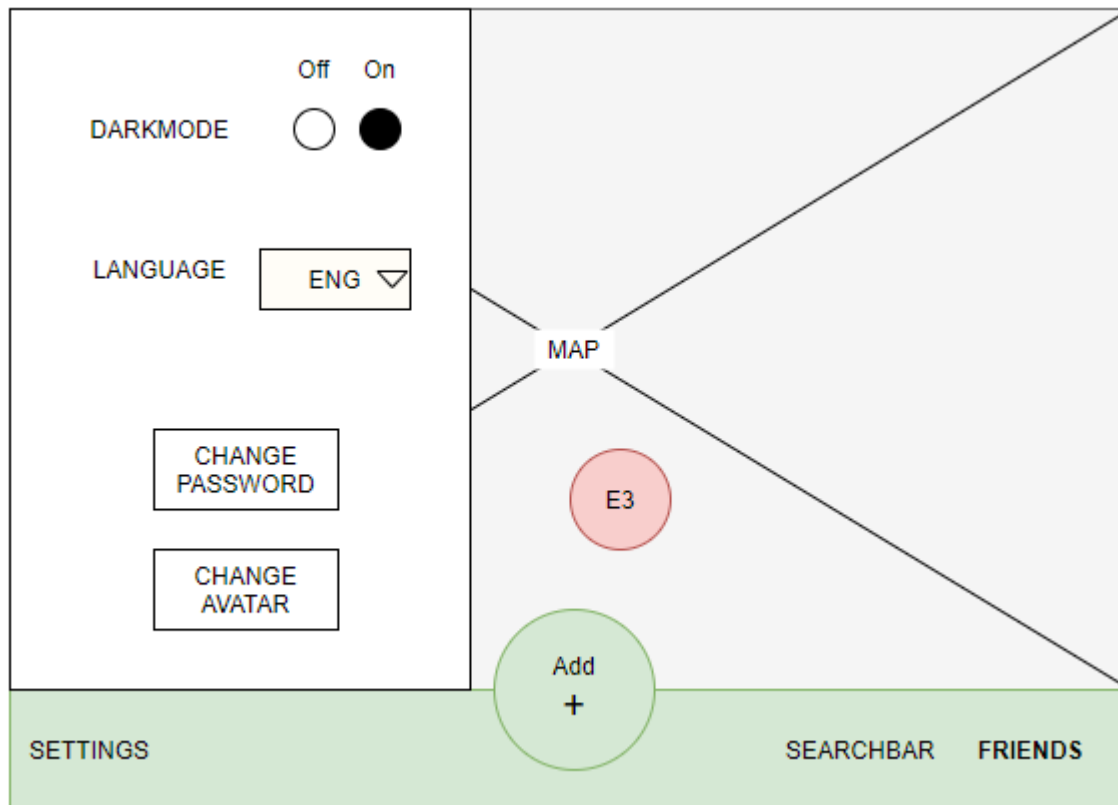


Figure 5 Main view with settings-drawer open

Lastly, we have the settings drawer open. Here the user can pick whichever option they want: darkmode on/off, as well as language. They can also change their password, or avatar.

Used Tools

The idea is to use the Reactjs library with TypeScript on the frontend, with a touch of Redux to handle state-manipulation. The majority of the frontend's components will be built using Material-ui. GraphQL will be used to converse with our own backend that deals with user data. The Backend will likely have a NoSQL database like MongoDB, since it's more suitable for dynamic data-manipulation where SQL would be best suited for static data. We'll also containerize the backend using docker. We'll also use googles map API to locate the events on the map, and we'll be fetching the basic event data from VisitTampere API (<https://visittampere.fi/api-docs/>).

Project Timeline

The project will be developed during the summer months, and **will be submitted to moodle hopefully at the end of July (FINAL DEADLINE)**. We already have a solid framework of the frontend, with certain functionalities like menu, settings, darkmode and so forth to be added in June.

When will the development work happen including possible intermediate steps etc.

Clearly indicate the final deadline when the final project will be submitted to Moodle.

Relation to Other Projects/Distribution of Work

As said in the “Used Tools”-section, we’ll be using Googles own map-api on the frontend, as well as the VisitTampere API. Material-ui will also be used as GUI framework.