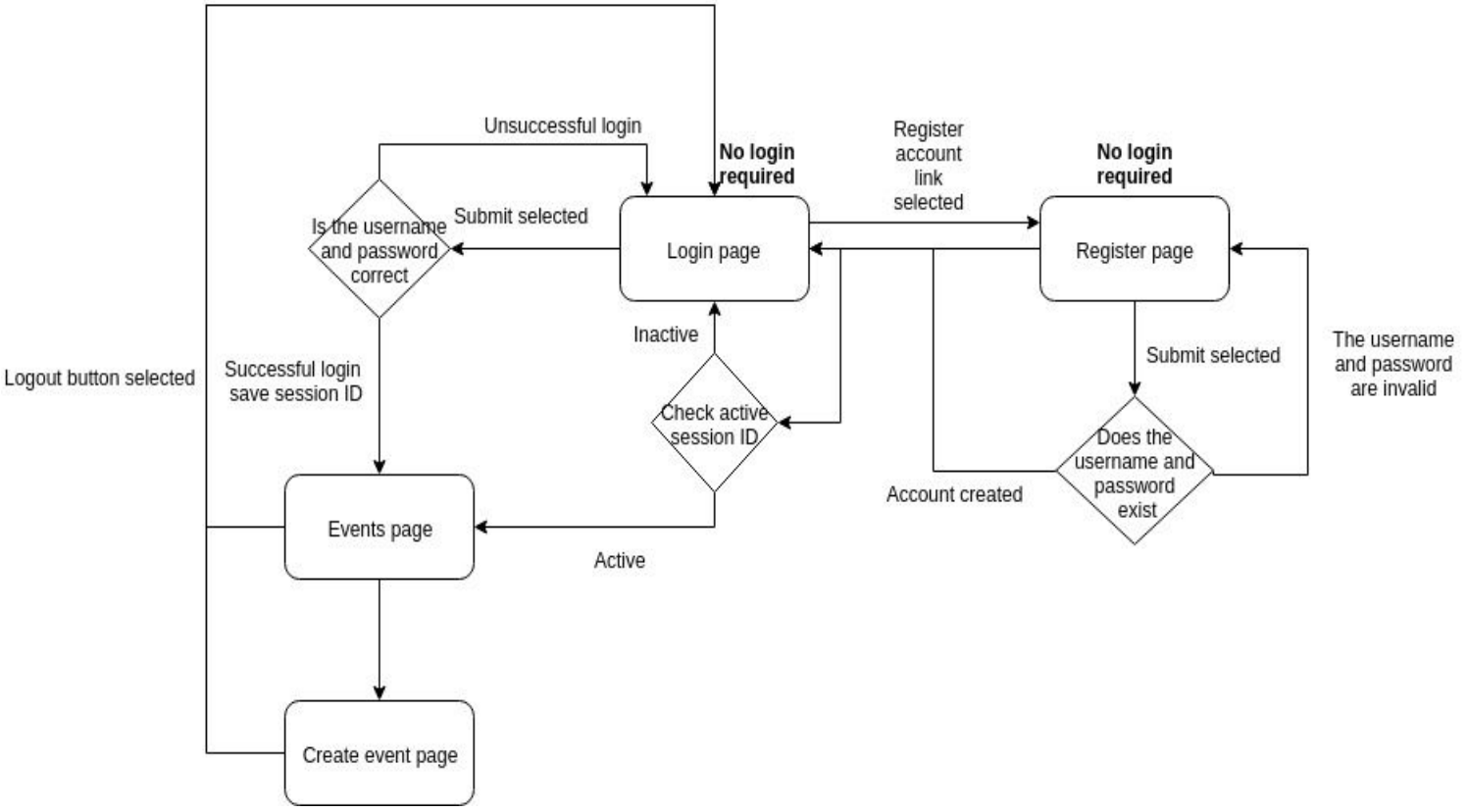
Team Z Assignment Report

Introduction

This report covers work done for the first part of this assignment, namely the task of building a fully-fledged PWA, associated remote nodejs server, and the subtasks that this involves. What follows is a solution diagram for this first part of the assignment, before a breakdown of the individual tasks that have been completed for this iteration of the assignment. We finally conclude this first part of the assignment and discuss how this work will aid us in the final submission, followed by the work allocation within the team and any additional information.

Solution Diagram



Tasks - Challenges, Solutions, Requirements and Limitations

Interface as a whole
As we decided to focus on functionality for this first part of the assignment there was not much effort put into the design of the interface. We have integrated bootstrap and tested its functionality so that when we get to the point of styling the web app things should run smoothly and quickly.

Interface to insert and search data via forms

The challenges faced during this task were to take into account where the parameters were being passed and how they were handled, whether or not ajax should be used or not. Our solution so far allows users to register an account, login to the application as well as create events. All forms requiring different data insertion. This solution meets the main requirements of this task, however there are current limitations as to what information can be searched for on the forms and is still undergoing development.

Interface to search data via map

This task is currently under development and is not in the current iteration of the application.

PWA: caching of the app template using a web worker

The challenges faced during this task are to identify the files needed for the application to run smoothly while offline. The solution developed for this was to make use of a web worker in which the files required in order the application to work are cached, allowing it to work when offline. This meets the requirements as it allows data to be stored and queried while offline. There are potential limitations as we don't yet know the extent of this implementation in terms of how it will apply to the final application.

PWA: caching data using IndexedDB

The challenges faced during this task are to store data locally into an IndexedDB structure to allow for searching locally. Our solution initialises the database with 3 object stored: Users, Logged in users and events. This is an IDB promise that if active the data can be read and saved both online and offline. The limitations of the current implementation are we are using 2 similar object stores which could be merged into 1 with a better implementation.

NodeJS server including non-blocking organisation of multiple dedicated servers

This task is currently under development and is not in the current iteration of the application.

Conclusion

In conclusion for this iteration of the project, there are a number of functionality elements that need to be implemented, as well as better implementation of functions that are currently in the application. Furthermore the styling of the web pages will need to be improved upon, however this was not a main focus of this iteration of the project as we mainly wanted to focus on functionality which has mostly been achieved.

Division of work

David Dixon = Functionality, style and report writing.

Simon Drake = Functionality, style and report writing.

Haider Rasool = Functionality, style and report writing.

Extra information

To run the application:

- Run NPM install on package.json
- Run ‘google-chrome --ignore-certificate-errors’ in terminal for Linux (or a windows equivalent)
- Visit ‘https://localhost:3001/’