1 Introduction

The Progressive Web App (PWA) allows users to create, post and search for events. WebRTC was implemented for users to take pictures with front and environment camera. Uploads are made available for photo selection from a folder. Each story can receive likes and comments which is implemented with SocketIO and users are able to follow one another. Attended and interested events are stored and displayed for each user. Web workers are used to allow offline usage with IndexedDB used to store data locally. MongoDB is used to store data but can only be retrieved when user is online. The search function (via location) was implemented with Google API, allowing selected location to be displayed on the map. For security purposes, users can only login with respective Google Account.

2 Diagrams

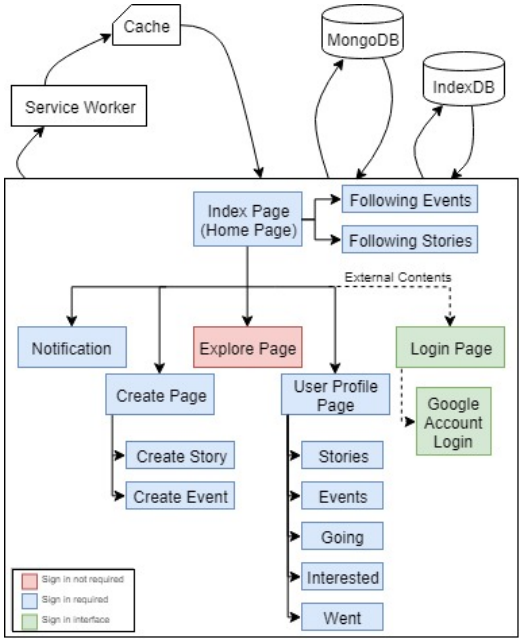

Figure 1: Demonstrates the flow of each web page in this PWA system along with the respective partial pages and external content pages.

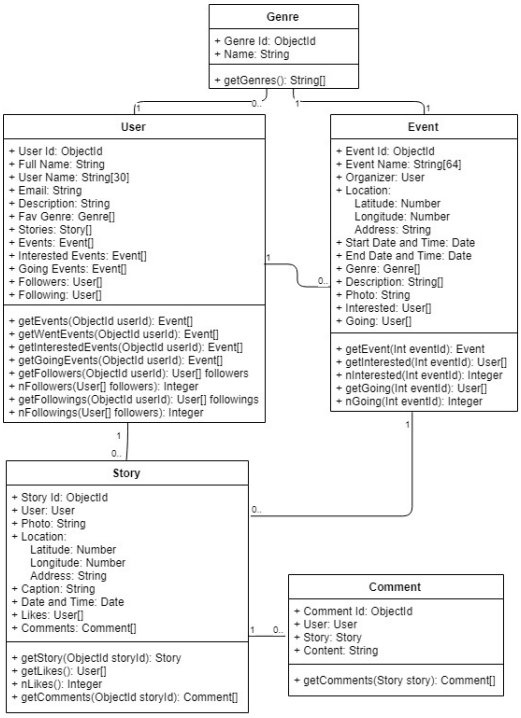

Figure 2: Displays the structure of the database along with the types of content stored.

Figure 1 and figure 2 are the detailed description of the PWA system structure with figure 1 describing the front-end, data storage and data retrieval, while figure 2 describe the types of data stored in the database along with the relationship between the documents.

3 Interface to Insert and Search Data via Forms

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

4 Interface to Search Data via Map

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

5 PWA – Caching of the App Template Using a Web Worker

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

6 PWA: Caching Data Using IndexedDB

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

  Chrome IDB bug: disk space increases for every put https://github.com/google/leveldb/issues/593 https://github.com/google/leveldb/issues/603

7 NodeJS Server Including Non-Blocking Organisation of Multiple Dedicated Servers

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

8 MongoDB

- **Challenges**:
  - Loading of initial data to populate the database.
  - The retrieval and storage of images in the form of bits.
- **Solution**:
  - Initial population of database was resolved with the use of promises.
  - Image retrieval and storage was implemented using *base*64 encoding.
- **Requirements**:
  - Data stored in MongoDB is stored online. Data is not stored locally and cannot be retrieved offline. When user is online, data is retrieved from the database and displayed on the PWA.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

9 Quality of the Web Solution

- **Challenges**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Solution**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Requirements**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.
- **Limitations**: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

10 Conclusions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

11 Division of Work

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus bibendum turpis in sollicitudin molestie.

12 Extra Information

- **Initial population of MongoDB**: run 'npm run initdb' to drop database and repopulate database with default data.