



Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: Overfind

Final Report

Group Members:

Asena Rana Yozgatlı
Bartu Soykök
Burak Şenel
Mehmet Emre Arıoğlu

Supervisor:

Prof. Dr. Varol Akman

Jury Members:

Prof. Dr. Özgür Ulusoy
Assoc. Prof. Dr. Çiğdem Gündüz Demir

Innovation Expert:

Bora Güngören (Portakal Teknoloji)

May 3, 2018

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

| | |
|---|-----------|
| Abstract | 3 |
| 1. Introduction | 4 |
| 2. Final Architecture and Design | 5 |
| 2.1. Subsystem Decomposition | 6 |
| 2.1.1. Presentation Layer | 7 |
| 2.1.2. Application Layer | 7 |
| 2.1.3. Data Layer | 7 |
| 2.2. Hardware Software Mapping | 8 |
| 2.2.1. Client | 8 |
| 2.2.2. Server | 8 |
| 2.3. Persistent Data Management | 9 |
| 3. Current Status | 10 |
| 4. Impact of Engineering Solutions | 10 |
| 4.1. Global Impact | 10 |
| 4.2. Economic Impact | 10 |
| 4.3. Societal Impact | 10 |
| 5. Contemporary Issues | 11 |
| 6. New Tools and Technologies | 11 |
| 7. Resources | 12 |
| 7.1. SDK's and Libraries | 12 |
| 7.2. Web Sources | 12 |
| 8. Similar Products | 12 |
| 8.1. Eventbrite | 12 |
| 8.2. Event Recommendation Engine | 13 |
| 9. References | 14 |
| Appendix: User Manual | 15 |

Abstract

Most of the people don't have the time to look for events according to their interests or profession and sometimes they miss interesting events due to this intense information flow. The aim of our project is to utilize different event platforms by parsing information about events to make recommendation to our app's users. Overfind users will be asked to fill a survey at the beginning of the application and later they will be recommended events according to this survey and their following feedback. Users will be able to browse the events that are recommended based on their interests and search through all events in our database.

1. Introduction

We are living in an era of information. Every one of us is bombarded with information around us in a sense of intensity. Events are happening in a specific time frame so it is important to notice the events in a timely manner. Most of the people don't have the time to look for events according to their interests or profession and sometimes they miss interesting events due to this intense information flow.

The aim of our project is to utilize different event platforms by parsing information about events to make recommendation to our app's users. Overfind users will be asked to fill a survey at the beginning of the application and later on they will be recommended events according to this survey and their following feedback. User interests will be used as labels and users will be able to search for events and they will be able to look at the recommended events which they can ask for more recommendation.

Overfind is an event recommendation system which will consist of three parts, a mobile application, a server application, and a database. System will recommend events to users based on their personal information. Mobile application will collect the initial user data by making the user fill a small survey, it will collect the rest of user data from user feedback. Server application will collect event information from various websites. Database will store information.

System will also offer some functionalities to improve the quality of usage. User can choose an event that fits his/her interests to see the event details. User can open external links to websites of individual events if available. User can ask for more recommendations. User can search the database for events. User can sort the search results by date or relevance. User can change the language and set a default location.

2. Final Architecture and Design

System consists of three layers: Presentation layer, application layer, and data layer.

Presentation layer represents the client application and it will run on Android OS version 4.4 (KitKat) [1] or later and can be downloaded from the Google Play Store [2]. This layer is composed of 3 parts: User interface for users to interact with the system; data manager to store and manipulate event and user information; controller that interacts with the server and manages data.

Mobile application will collect the initial user data by making the user fill a small survey, it will collect the rest of user data from user feedback and store locally in the device. It will communicate with the server by using https GET method and response will return in JSON format.

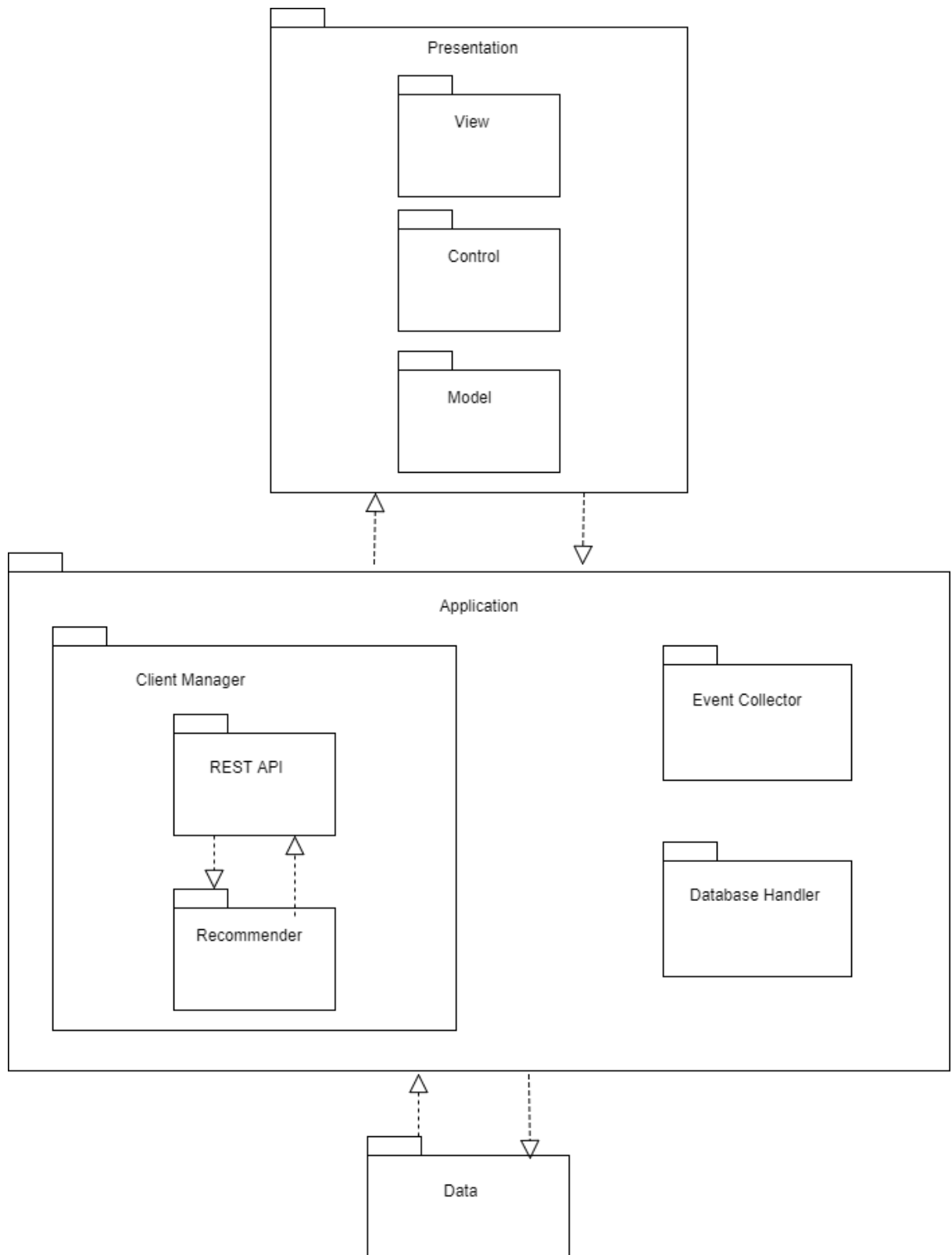
Mobile application will use the device ID to create an initial record when the app launches for the first time. When the mobile application is uninstalled from the device, it will not be possible to restore user data. Mobile application will require storage and network access of the device.

Application layer represents the server application and it will run on a Linux machine. This layer is composed of 3 parts. Client manager that interacts with clients and handles client requests; event manager that interacts with web API's to collect and extract meaning from event information; database handler that interacts with the database server to read or write data.

Server will get user data from client and generate recommendations using the tag relevancy information. Server will not store user data. Client requests will be taken by https GET method and response will sent in JSON format. Server will query database by using neo4j.driver.

Data layer represents the database. Database will run on a Linux machine. We will use a graph database for our storage purpose. We will use Neo4J [3]. Only server can reach the database.

2.1. Subsystem Decomposition



2.1.1. Presentation Layer

Presentation layer will be a mobile application and it will run on Android OS. This application will interact with user and application layer. It will have 3 subsystems.

View is responsible with user interaction. It will display and collect information.

Control is responsible with data manipulation and interaction with application server. It will update user related data based on data collected from view subsystem and it will update event related data based on data collected from application server.

Model is responsible with storing both user related and event related data.

2.1.2. Application Layer

Application layer represents the server application and it will work on a Linux machine. This application will interact with presentation and data layers. It will have 3 subsystems.

Client Manager is responsible with client interactions and performing the client requests. It will have 2 subsystems.

- REST API is responsible with client interactions.
- Recommender is responsible with recommending events based on data given by the client.

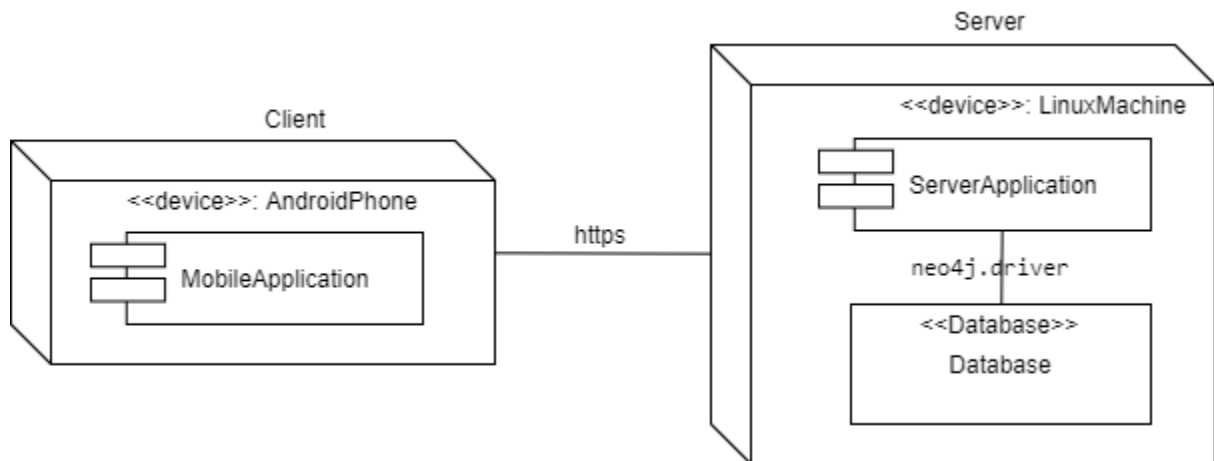
Event Collector is responsible with collecting event information and extracting information from the data collected. It will communicate with web API's to collect and process data.

Database Handler is responsible with interaction with the database server. It will collect information from database server when requested by client manager and will store data to database server when requested by event manager.

2.1.3. Data Layer

Data layer represents the database and it will work on a Linux machine and it will be a graph database. It will interact with application layer.

2.2. Hardware Software Mapping



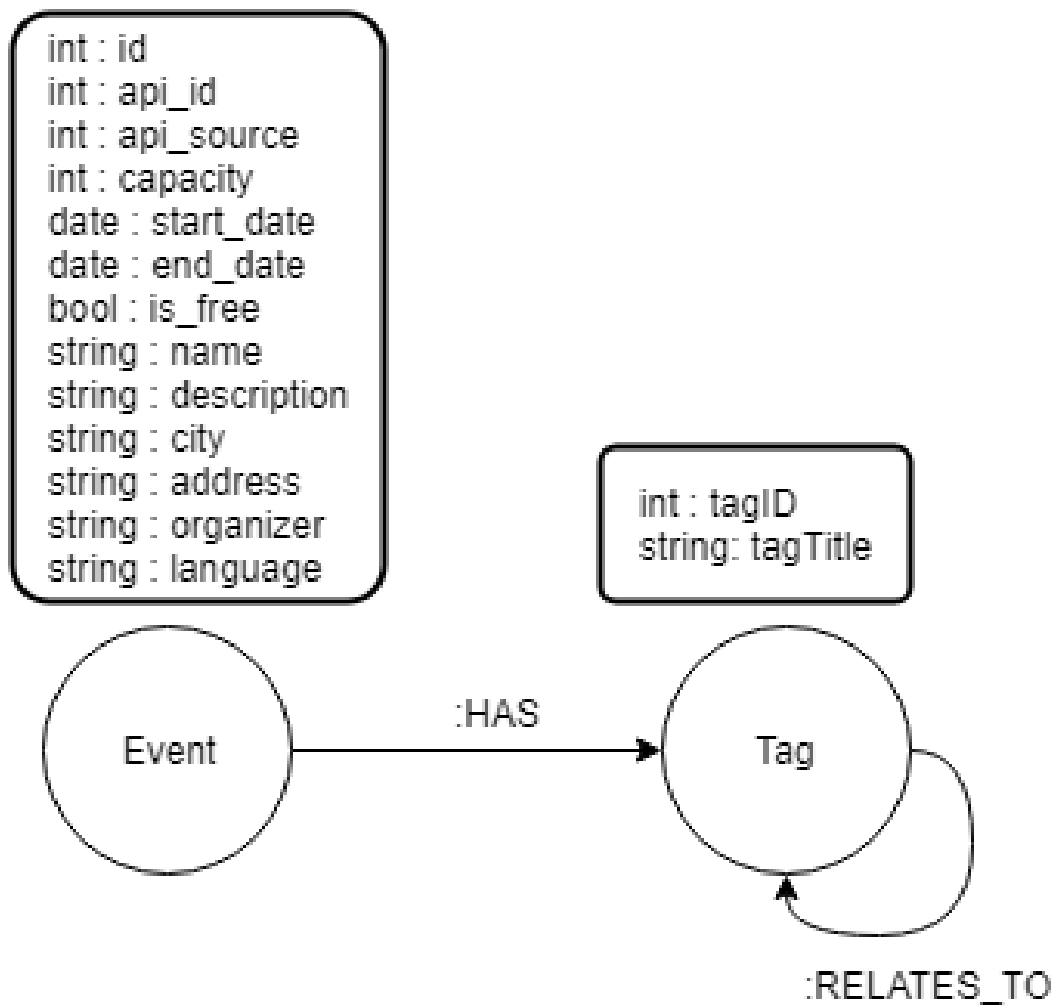
2.2.1. Client

Users have to supply their own devices to use our app. Client side of the application will take place on user's android phone. User can use recommendation and search services by communicating with our server by using https GET method and response will return in JSON format. User's device will store his/her likes locally in the device.

2.2.2. Server

Server will be run on a Linux machine. Server is responsible for listening client requests at any time and collect new events periodically. Client requests will be taken by https GET method and response will sent in JSON format. Server will query database by using neo4j.driver. Database will run in the same machine as server. Only server can reach the database.

2.3. Persistent Data Management



3. Current Status

Currently the backend of our android client is not fully finished. Storing user data in the client is not done. User survey to collect initial data is not implemented. In the server dates of events are not handled properly yet. We will make improvements in our recommendation algorithm. Rest of our requirements are implemented and functioning properly.

4. Impact of Engineering Solutions

Engineering solutions implemented by this project have global societal and economic impacts that are explained below

4.1. Global Impact

With the information overload each of us faces in current world it is difficult to find and attend to all of the different events that might interest us. This application makes it possible to find and keep track of such events.

Since this is an interest based event recommendation application, it can be used for events all over the world. Events recommended will be based on user interest and location in order to minimize irrelevant recommendations. As such accurate recommendations will increase the amount of people attending to these events.

4.2. Economic Impact

All users will be able to use this application free of charge. Some events recommended by this application could require an entrance fee of sorts. Since accurate recommendations will mean an increase in the amount of people attending an event it will increase the revenue for these paid events. As with all recommendation platforms this application can also function similarly to an advertisement for the events.

4.3. Societal Impact

With the increase in the amount of events both hobby related like concerts, sports, cinema etc. and educational related like conferences, workshops etc. its becoming harder for people to find an event they would like to attend. For the organizers of these sort of events it is also a challenge to be able to reach people who will be interested in the event and attend. These application will help solve both of these problems. Having all of the information about the events as well as a schedule will also encourage people to actually attend more events than they would otherwise.

5. Contemporary Issues

- Search history of users will not be stored in the server to preserve user privacy.
- Internet connection is necessary in order to use the application. So, any user with internet connection and speed problems will have a hard time using the application.
- Application will be able to support any Android version higher than 4.4 KitKat. Devices running lower versions of Android or other operating systems will not be able to use the application.
- As amount of events increase data amount in the server will increase quickly, in the future we might transition into a cloud system.
- Currently the language of the application is English. In the future we might look into supporting other languages.

6. New Tools and Technologies

- **Neo4j:** A NoSQL graph database [4]. We saved all our data including users, venues, trips, preferences to this database. Since our data has a very complex structure with many interconnected relations, representing it as a graph allowed us to visualize the data better. Also the query speed provided by Neo4j database allowed us to have a fast server.
- **Android Studio:** An IDE for Android Development [7]. We used Android Studio in the client side of our project, to implement the Android application. We used the emulators Android Studio provides to test our code. With its XML, Gradle, and Maven supports, Android Studio eased the coding process.
- **Glassfish:** GlassFish is the Open Source Java EE Reference Implementation
- **RESTful Web Services:** RESTful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are lightweight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.
- **Amazon Web Services:** Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.
- **Maven:** is a software project management and comprehension tool [8]. We used Maven with Android studio and it made the build process easier. Furthermore, we used Maven for the library dependencies.
- **Gradle:** is another build tool used by Android Studio [9]. We used Gradle to ease project build process. Also we included all third-party libraries we use and our server models easily to our system using Gradle.

7. Resources

7.1. SDK's and Libraries

- The Android Support Library
- GSON Library
- Jersey Bundle
- Neo4j Java Driver
- JSR311 API
- JDK 1.8

7.2. Web Sources

- <https://developer.android.com/guide/>
- <https://javaee.github.io/glassfish/documentation>
- <https://aws.amazon.com/mobile/resources/>
- <https://neo4j.com/docs/>

8. Similar Products

8.1. Eventbrite

Eventbrite is an online event organizing service [4]. Users have two roles in eventbrite system: organisers and attendants.

Organisers can create events and sell the tickets of these events if they wish to. Organisers can provide event details such as name, description, location, date, time, price, seating plan, etc. Organisers are encouraged to choose a topic and a

subtopic for their events. These fields are treated as labels and are predefined.

Organisers can promote their events using Eventbrite services.

Attendants can purchase tickets of the events found on eventbrite's database. They can see their purchased events. They are also given some recommendations based on the events they attended.

Recommendation service of Eventbrite considers both the event taste of attendants and the promotional need of organisers.

Our system, Overfind, will not allow the creation or promotion of events. It will search the web and display events that are publicly available. Overfind will create and assign its own labels to events based on the information provided. Overfind will use this self-generated information as labels of events. Overfind will address only the event taste of attendants (users of our system), in order to provide more personalised recommendations.

8.2. Event Recommendation Engine

This software[5] is designed as a model for the event recommendation engine challenge on Kaggle[6].

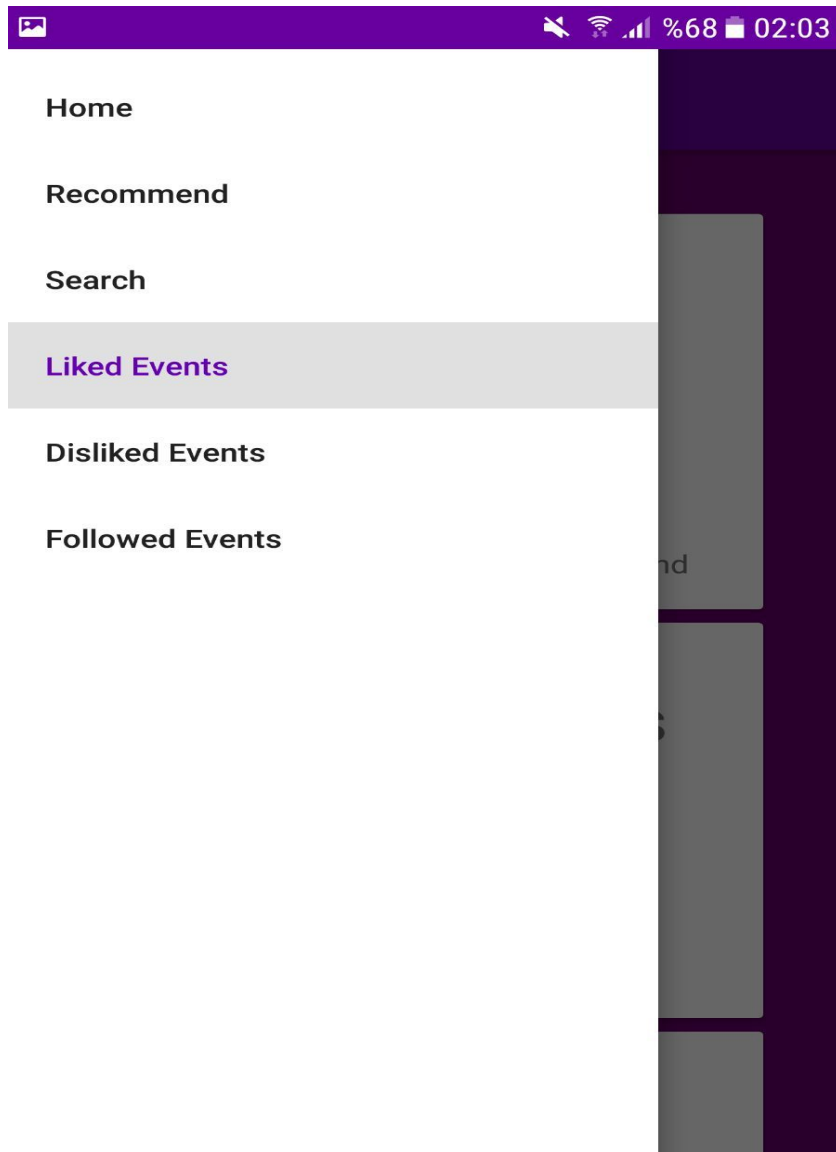
The software uses datasets including users' friends, friends' likes, their interests, etc. This includes rating of the neighborhood that consists of users with common interests such as information of who is attending which events, location similarities of events that are attended by users in that neighbourhood, and event similarities. This is an example of collaborative filtering model with user based approach.

Overfind will use content based filtering which is focused on using existing user profiles and getting information about users by initial surveys. In this process we compare events that user rated positively with the events that aren't yet rated by users.

9. References

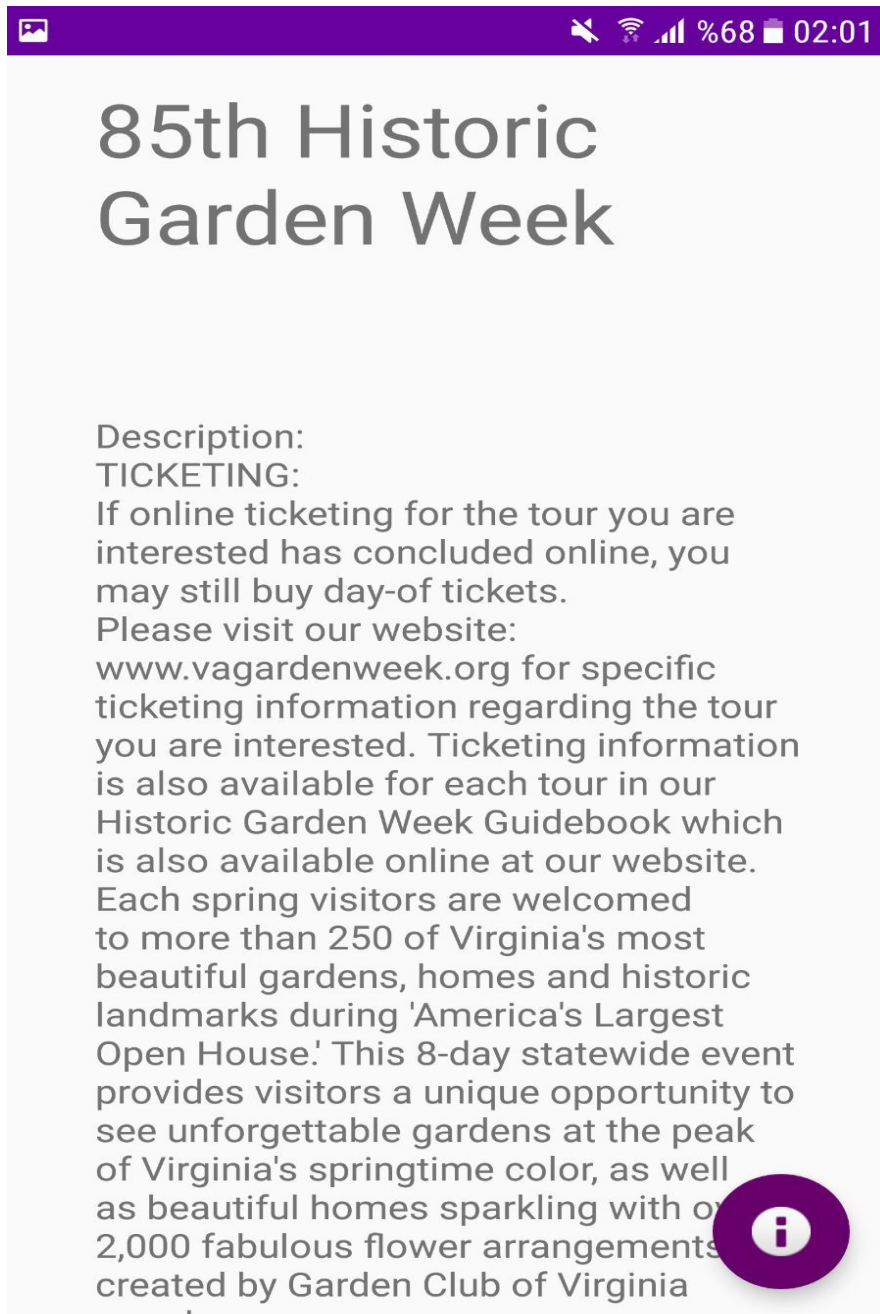
- [1] "Android - 4.4 KitKat", Android, (n.d.). [Online]. Available: https://www.android.com/intl/tr_versions/kit-kat-4-4/. [Accessed:03-05-2017].
- [2] "Android - Play", Android, (n.d.). [Online]. Available: <https://www.android.com/play/>. [Accessed:03-05-2017].
- [3] "The Neo4j Graph Platform – The #1 Platform for Connected Data", Neo4j Graph Database Platform, 2017. [Online]. Available: <https://neo4j.com/>. [Accessed:03-05-2017].
- [4] "Eventbrite - About Us", Eventbrite, 2017. [Online]. Available: <https://www.eventbrite.com/about/>. [Accessed:03-05-2017].
- [5] "Model for the Event Recommendation Engine Challenge on Kaggle.com", GitHub, 2017. [Online]. Available: <https://github.com/andreiolariu/kaggle-event-recommendation>. [Accessed:03-05-2017].
- [6] "Kaggle", Kaggle, 2017. [Online]. Available: <https://www.kaggle.com/kaggle>. [Accessed:03-05-2017].
- [7] M. Studio, "Meet Android Studio | Android Studio", Developer.android.com, 2017. [Online]. Available: <https://developer.android.com/studio/intro/index.html>.
- [8] Maven – Introduction", Maven.apache.org, 2017. [Online]. Available: <https://maven.apache.org/what-is-maven.html>.
- [9] "Gradle Build Tool", Gradle.org, 2017. [Online]. Available: <https://gradle.org>.

Appendix: User Manual



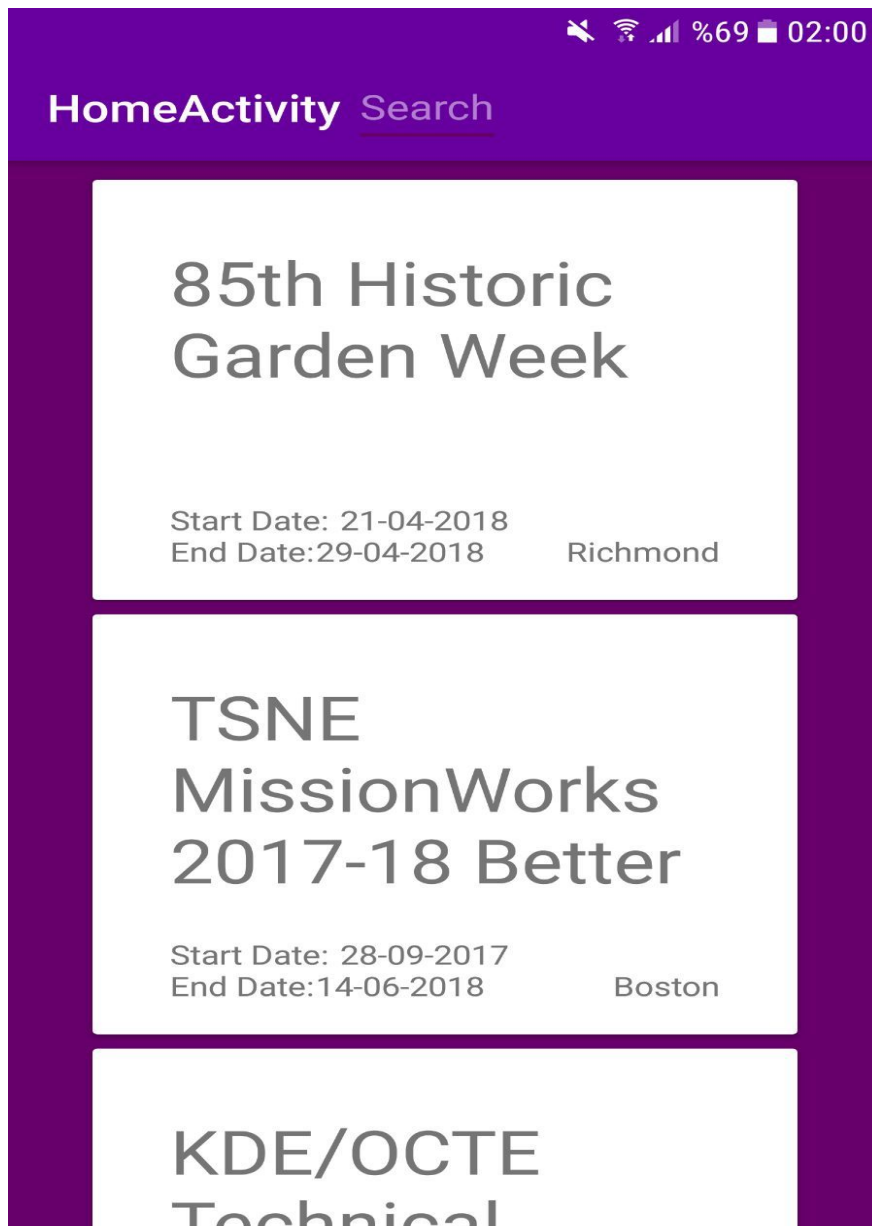
Picture 1: Sidebar

From the first screen user can swipe the screen in order to access the sidebar. From the sidebar user can go to their recommendations page, search a specific event and check their liked and disliked events as well as the events they are currently following.



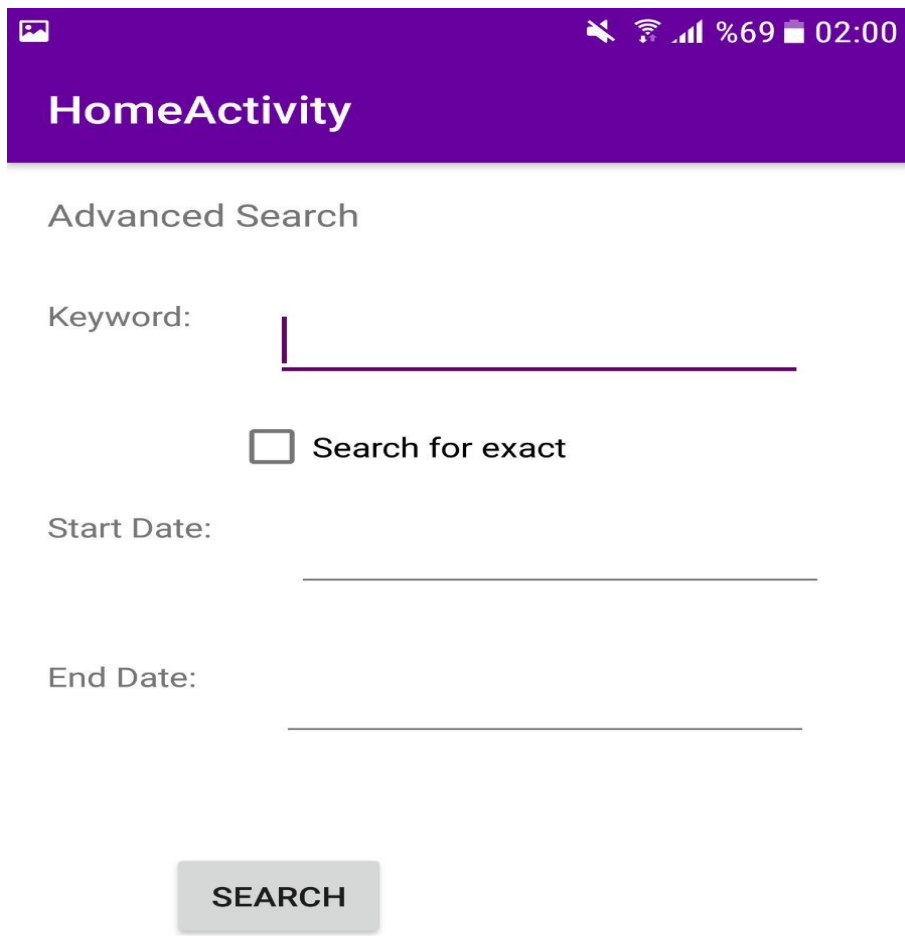
Picture 2: Event Description Page

Upon selecting any event from home, recommendation or other pages user will be redirected to the event description page in which the user can read the event description as well as address and date information. From this page user can press the info button to visit the events website, this screen also has the follow, like and dislike buttons.



Picture 3: Recommendation Page

In the recommendation page user can see the events name, date and location. From this page pressing any of the events will send user to the event description page of the respective event.



The screenshot shows a mobile application interface with a purple header bar. The header bar contains a small icon on the left, status icons (signal, Wi-Fi, battery) and the time '02:00' on the right, and the title 'HomeActivity' in white text. Below the header, the text 'Advanced Search' is displayed. There are three input fields: 'Keyword:' with a purple underline, 'Start Date:' with a grey underline, and 'End Date:' with a grey underline. A checkbox labeled 'Search for exact' is positioned between the 'Keyword' and 'Start Date' fields. At the bottom, there is a grey button with the text 'SEARCH' in black capital letters.

HomeActivity

Advanced Search

Keyword:

☐ Search for exact

Start Date:

End Date:

SEARCH

Picture 4: Search Page

Using search page user can search a specific event by their name or start and end dates.