|  |  |
| --- | --- |
| **final report** | **Abstract**  An event-sharing application that allows user to find, create, and like any events that located nearby or based on specific locations  **GROUP MEMBERS**  Nguyen Minh Toan – ID: 300282895  Truong Thanh Thao – ID: 300284134 |

Table of Contents

[Overview 1](#_Toc531550136)

[Features and Technical challenges: 1](#_Toc531550137)

[Features 1](#_Toc531550138)

[*Front-end* 1](#_Toc531550139)

[*Back-end* 1](#_Toc531550140)

[*Others* 1](#_Toc531550141)

[Technical challenges 1](#_Toc531550142)

[*The real-time database doesn’t provide querying feature.* 1](#_Toc531550143)

[*Passing data between fragments and activities is quite hard.* 2](#_Toc531550144)

[*Using Firebase Storage to store uploaded images* 2](#_Toc531550145)

[Installation guide 2](#_Toc531550146)

[Requirements to run the app: 2](#_Toc531550147)

[Functions and user’s manual 3](#_Toc531550148)

[Functions 3](#_Toc531550149)

[User’s manual 5](#_Toc531550150)

[Incomplete functions 6](#_Toc531550151)

[*Events are not categorized* 6](#_Toc531550152)

[*Administration features* 6](#_Toc531550153)

[*Advanced searching* 6](#_Toc531550154)

[*Sharing via social network* 6](#_Toc531550155)

[Disclaimer 6](#_Toc531550156)

[Reference 6](#_Toc531550157)

# Overview

The purpose of our group project is to create a useful application that is as close as possible to the real-world practice. Hence, we try to implement the latest techniques in both front and back-end to produce EventMe.

EventMe is an Android application locating user’s device current location and recommend any nearby events around the area. Users can view details information about a particular event, including event’s name, date and time, location, description and numbers of people interested in that event. In addition, users can like or unlike any events and create their own events. This application allows multi-users using with real-time experience.

# Features and Technical challenges:

## **Features**

### *Front-end*

* Dynamic fragments: All the main three pages, which are the home page, create-event page, and user-profile page are fragments. Nested fragments are included in user’s profile page to load both lists of created events and liked events.
* Bottom navigation bar with icon
* Toolbar: toolbar is more flexible than Action bar as we can design our custom search toolbar with edit text.
* Tab Layout: is implemented in the user’s profile page in order to make
* Card view: for display the details information of an event
* Floating Action Button: for creating like/unlike button
* Picasso library: for loading images

### *Back-end*

* Firebase (Authentication, FireStore, Storage)
* Google Map and Geocoding

### *Others*

* ViewModel and LiveData

## 

## **Technical challenges**

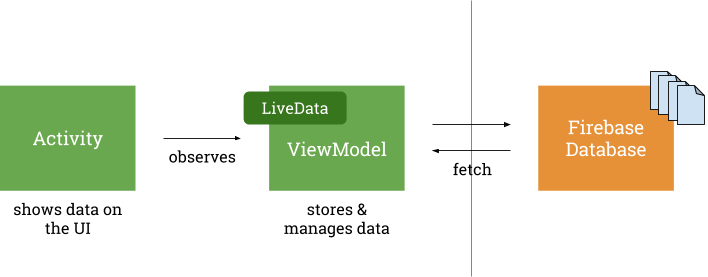
### *The real-time database doesn’t provide querying feature.*

Because of its simple approach by design, it did not support query feature, which can only return the whole data of the document. Therefore, we cannot search events based on location or time. Our solution is switching to Firebase Cloud Firestore which provided some simple compose query such as greater than, less than, equal, contains.

But Cloud Firestore was still not enough because it did not provide the full-text searching, that we cannot search events based on location fragment (such as just city name), it can only search the whole location (full address). Our final decision was pulling all events and searched them with java code, which works really well.

### *Passing data between fragments and activities is quite hard.*

Because the implementation of multiple fragments within the same activity, many issues passing data between them occurred obviously. The UI has to be updated in many places when the data was changed, which caused complicated and inconsistent source code. After hours of researching the best practices recommended by many senior Android developers, we decided to implement ViewModel & LiveData architecture in our project. By applying this method, we can handle all data changed and updated UI in only one listener method, which is significantly handy for our app.



*View model & live data structure (*[*source*](https://medium.com/@lgvalle/firebase-viewmodels-livedata-cb64c5ee4f95)*)*

### *Using Firebase Storage to store uploaded images*

Our app allows users to upload event images and profile photo in order to provide a better experience for the app. Our solution is using Firebase Storage to store the image uploaded by users. One of the challenges, when we implemented this feature, is that we have to upload the photo first, then get the uploaded URL to insert into event information before saving it to database

# Installation guide

## **Requirements to run the app:**

* The phone/emulator must be API level 26 and above.
* Permission to access Google Map API. Because of security, Google Map API key requires the SHA-1 fingerprint to view the map on the device (this is for development environment only). To acquire the fingerprint, please read the instruction [here](https://developers.google.com/maps/documentation/android-sdk/signup#fingerprint).
* When logging in to the phone/ emulator, click allow ‘EventMe’ application to know your location

# Functions and user’s manual

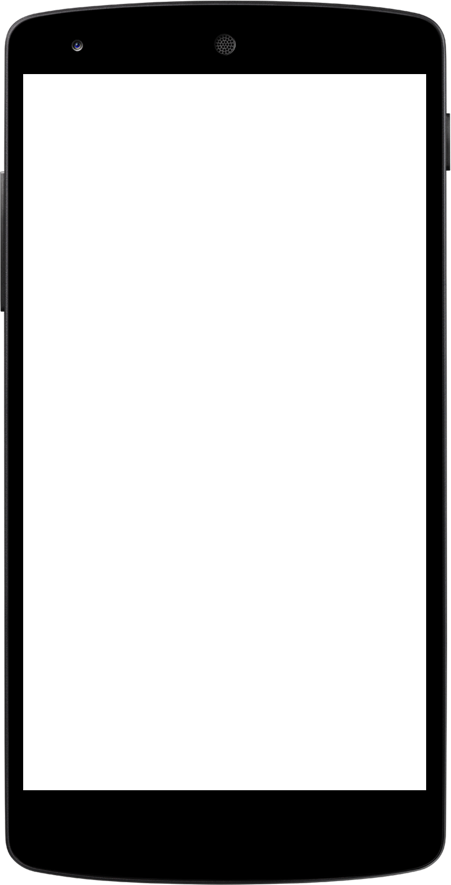
## **User’s manual**

[Application usage work-through](http://recordit.co/OvgL4dxPpl?fbclid=IwAR0AG4LlzZ0PkgtrWKBAeYzGXaL5NDvU3zU2xQnykj4bNOtZMWr0-lGgyDQ)

## **Functions**

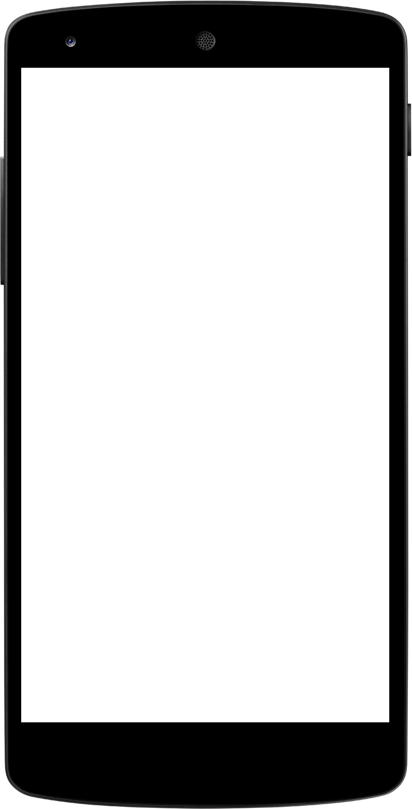
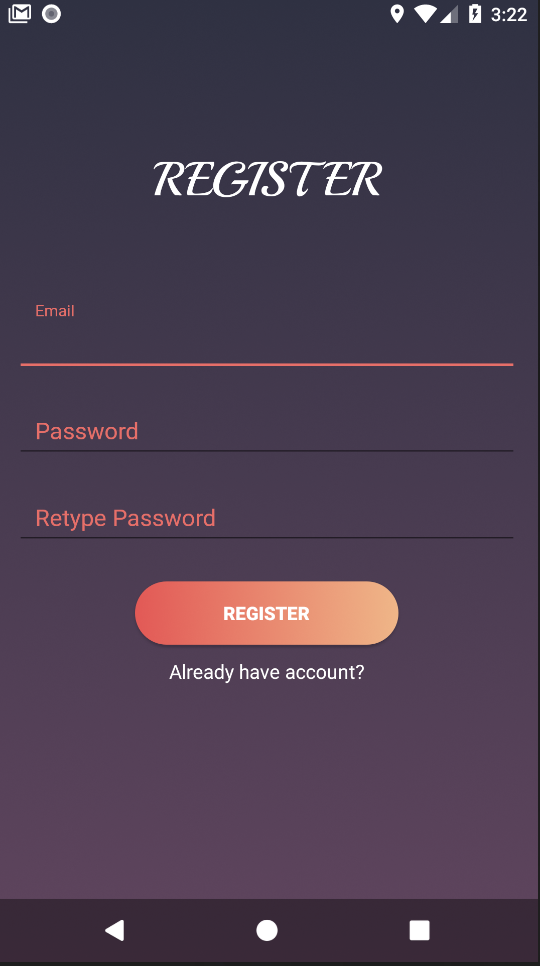
Authentication page:

* Users can sign up for a new account by inputting email and passwords
* Users log in with their registered account by inputting email and passwords

A screenshot of a cell phone

Description automatically generated

Input your account information

Register for new account



Search events by location

(default: device’s location)

General information of the events

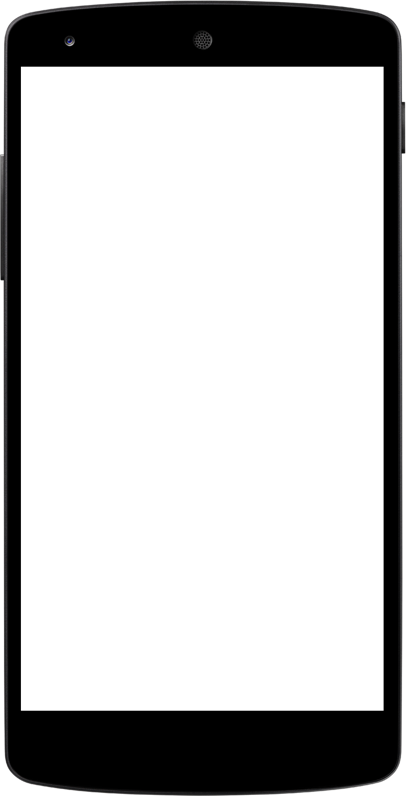
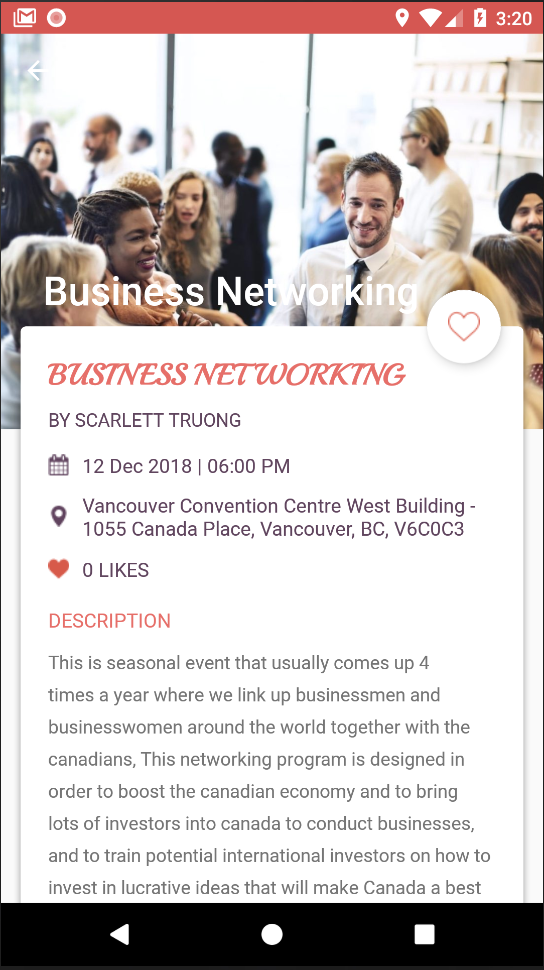
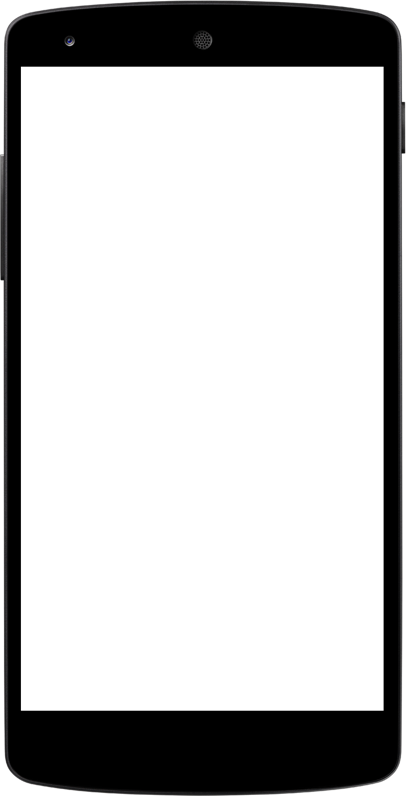
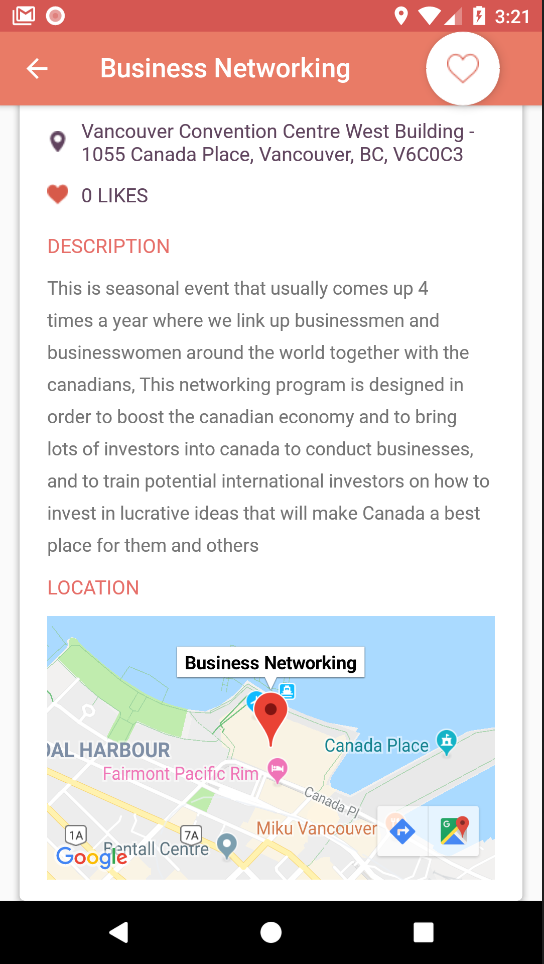
Only fetch events in the future

Homepage:

* View events nearby user’s device location
* Search events based on location
* View details of an event include date & time, location, description, creator.
* Pull the list down to refresh events list.

Event details:

* View event’s information
* Like any events that user are interested
* View Google Map of the event’s location

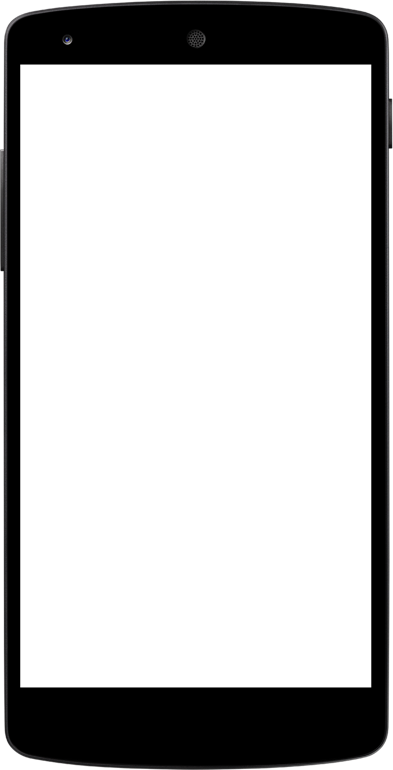
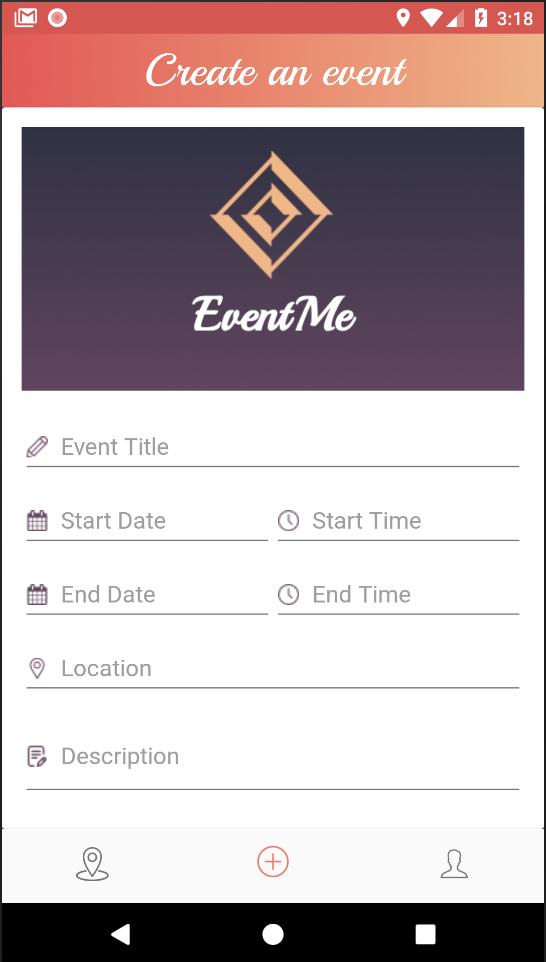


Details information of an event:

* Event name
* Creator
* Date & Time
* Location
* Number of people liked
* Description

User can like an event

View with Google Map

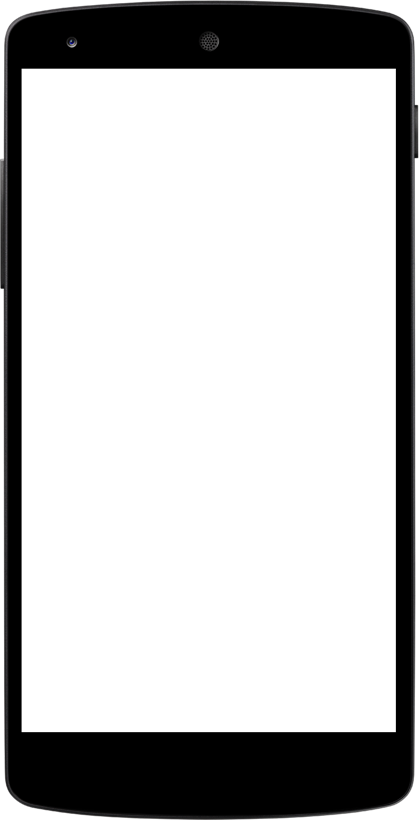
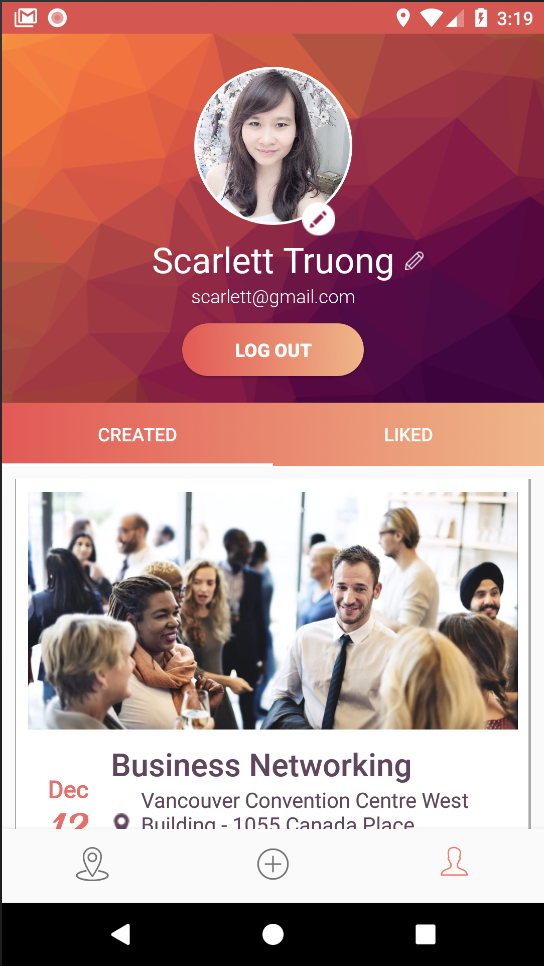


Upload your own event’s image

Input your event’s information

Event create page:

* Create own event, other users can see it as well
* Upload your own event’s image and it stores in Firebase Storage



Upload your own profile’s image

Edit your name

View all your created events

View all your liked events

User profile

* Display user information (first name, last name, avatar)
* Button: Log out
* Buttons: Edit user’s avatar and edit user’s name
* List events created by the logged user
* List events liked by the logged user

# 

# Incomplete functions

There are no incomplete functions based on the proposal. However, there are some functions that our group want to implement for better improvements:

### *Events are not categorized*

The app would be better if the events are categorized in various sections, such as concert, movies, technology, art, holiday, etc. Since the number of events can grow significantly, categories can help users find their favourite events. Besides, users can follow a specific category to get notifications about new events created under that category.

### *Administration features*

The app’s content is based on user contribution. Therefore, the reliability of the created events is not high. Because of that, a management system is quite necessary. The administrators can manage all the events and verify whether the events’ information is correct. Besides, they can manage the users as well.

### *Advanced searching*

Currently, the application supports finding event based on location. It is more comfortable for users if they have more abilities to filter events when the events amount is large enough. Event name, category are the good criteria.

### *Sharing via social network*

Sharing is one of the powerful tools to spread information to the world. In order to increase the number of users, sharing events via social networks is the best manner, since people who use social networks are enormous.

# Disclaimer

EventMe application is our first original product that is created for course CSIS 4175 - mobile app development 2 at Douglas College.

# Reference

* ViewModel & LiveData documentation:  
  <https://developer.android.com/topic/libraries/architecture/viewmodel>
* ViewModel architecture image source:  
  <https://medium.com/@lgvalle/firebase-viewmodels-livedata-cb64c5ee4f95>
* Google Map implementation documentation:  
  <https://developers.google.com/maps/documentation/android-sdk/map>
* Firestore implementation documentation:  
  <https://firebase.google.com/docs/firestore/quickstart>