

# **Interim Report for a Privacy-preserved Instant Event Sharing System**

Chan Kin Long	(3035184922)
Cheung Tin Long	(3035187833)
Kwok Ching Hei Kevin	(3035185031)

21 January 2018

## **Abstract**

There are many forums and social media in the Internet presently, but no particular website focuses on location and real-time event information sharing. This project was initiated by the insufficiency of functions in the existing applications. The system aims to provide a platform for users to exchange the most updated information of specific locations.

This paper describes the design and implementation of “happens”, the instant event sharing system. It has four important features, including the request-and-response function, photo sharing function, category system and the authentication system. The request-and-respond function let user inquire information based on location and the system connects users in the world by location.

The ultimate product of the system is to build a cross-platform mobile application that can be run on both iOS and Android platforms, as well as a web application. The front-end applications are written in Swift and Kotlin respectively for the mobile versions, while the web version is written in React.js. Moreover, the backend server is based on Google Firebase, which provides authentication, database, file storage and cloud functions.

This paper marks a milestone for the project, which concludes the first half of the development process. Previously, the design phase of the project has been finished and it is in the implementation phase currently. The system will move on to the testing phase in February 2018. With successful tests, the system will be launched to the public in March 2018. During development, feedbacks are collected and improvements should be made in the second half of the development schedule. The first priority in the future is to build the system successfully and maintain a smooth running of the application. New features can be added to the system when it is stable.

Different problems are appearing during the development, such as the choice of the front-end programming language. However, solutions are found, and these problems encountered are dealt with. The system can be improved in future iterations.

For the subsequent development process, more features can be added to the system to enhance the usability. However, there are limitations and restrictions on this final year project, thus, recommendations are provided as a step stone for future development on location-based sharing system.

## **Acknowledgement**

We would like to express our gratitude to our supervisor Dr. TW Chim, who initiated the idea and guides us throughout the project. This project would not be on the right track without Dr. Chim's timely advice. We would like to also show our appreciation to the Computer Science Department of the Faculty of Engineering of the University of Hong Kong, which provides resources for us to develop the system.

# Contents

<b>Abstract .....</b>	<b>2</b>
<b>Acknowledgement .....</b>	<b>3</b>
<b>List of figures.....</b>	<b>5</b>
<b>1     Introduction .....</b>	<b>6</b>
1.1     Background and Motivation .....	6
1.2     Scope of Study .....	7
1.2.1     Request-and-response function .....	7
1.2.2     Photo sharing function .....	7
1.2.3     Category System .....	8
1.2.4     Authentication system.....	8
1.3     Deliverables.....	8
1.4     Structure .....	8
<b>2     Methodology.....</b>	<b>9</b>
2.1     Platform setup.....	9
2.2     User interface design .....	10
2.3     Database Design .....	11
2.4     User Access and Security Design .....	12
<b>3     Current Status .....</b>	<b>13</b>
3.1     Work Accomplished to Date .....	13
3.1.1     Design Phase .....	13
3.1.2     Implementation Phase (Server) .....	13
3.1.3     Implementation Phase (Client) .....	14
3.2     Problems Encountered .....	16
3.2.1     Choice of programming language for the front-end application .....	16
3.2.2     Distinguishing different categories.....	16
3.3     Work Remaining .....	17
3.3.1     Function Implementation .....	17
3.3.2     Testing and official launch .....	18
<b>4     Conclusion.....</b>	<b>19</b>

<b>References .....</b>	<b>21</b>
-------------------------	-----------

## List of figures

Figure 1 – System flow of the request-and-response function. ....	7
Figure 2 – User interface designs prototype of the mobile version of the application .....	10
Figure 3 – Simplified Project Timeline .....	13
Figure 4 – Implemented user interface in iOS version .....	15
Figure 5 – Circular geo-fence set-up .....	17

# **1 Introduction**

This section introduces the project. The overview of the project is discussed, as well as describing the structure of this paper.

## **1.1 Background and Motivation**

In the era of instant information, people are curious about the events happening in the world: A driver would like to know the traffic condition of his destination; a girl would like to know the queuing situation outside a shop selling limited edition bags. In the current development of information technology, end-to-end information sharing is a major focus. There are different blogs, forums and social networking sites providing a platform for users to exchange information. However, popular communication applications, such as Facebook and Twitter, are not location-oriented, thus it is hard to get the instant information of a certain place. Moreover, the information shared in the above application cannot be verified and this leads to a low accuracy in information exchange.

For solving the above problems, an instant event sharing system for requesting and uploading instant information by location is developed in this project. The system is a real-time application which provides a platform for users to request for the immediate situations of a certain place and retrieve the expecting information. The information is provided by the users in that certain place. It is foreseeable that the system will soon be widely used and lots of information can be exchanged through it. It aims to satisfy all kinds of curiosity and we believe that hands make light work. Therefore, the world would be connected if every user is willing to share events happening near them.

## 1.2 Scope of Study

This project is to develop an instant event sharing system. Different events can be shared through the system, such as a traffic jam in a specific district or the queuing information of the Book Fair. The system is a client-server application. The client side has iOS, Android and web versions; while there is a back-end server storing data and doing cloud functions. It has four important features, including the request-and-response function, photo sharing function, category system and the authentication system.

### 1.2.1 Request-and-response function

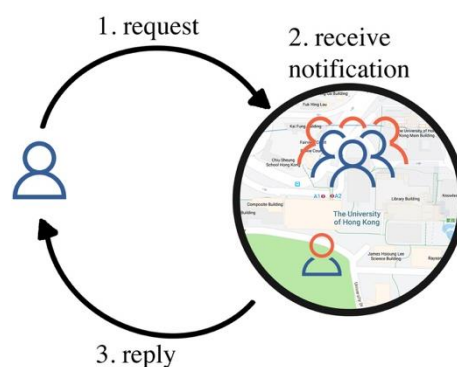


Figure 1 – System flow of the request-and-response function.

First, the request-and-response function is the main feature of the system. The system flow is shown in Figure 1. User can request for information of a certain place, and other users around that area will receive notification. They can then reply to the request by uploading images, showing the actual situation of the events happening at that location.

### 1.2.2 Photo sharing function

Second, it is the photo sharing function. Similar to other social media, users are allowed to upload photos and write description on the post. Moreover, user can categorise their post to specify what the post is about. The post can be publicly accessed by other users, and they can give comments as well as up-vote or down-vote the post. However, the system limits the photo uploaded must be directly captured using the app. This prevents old photos from being uploaded to the system because the system focuses on the instant event happening at the time, so only new image can be uploaded. Furthermore, voting and comment system are provided alongside with the photo sharing function. User can vote (similarly to like/dislike) and comment on each post.

### **1.2.3 Category System**

Third, it is the category function. As our system are similar to a forum and it does not limit what user should post, a category system is developed to categorize the post. User can subscribe to any categories and the can find the post they would like to read easily.

### **1.2.4 Authentication system**

Finally, the authentication system register user for using the system. Although normal user without registration can use the system viewing the posts, the upload function is only limited to verified registered users. In addition, the system can block the user from using it for posting explicit images or giving inaccurate information. This prevents user from repeatedly misuse the system.

## **1.3 Deliverables**

Our project is to develop a mobile application for iOS and Android devices, as well as a web application. It is named “happens” and it can be downloaded from the Apple App Store and also Google Play Store once it is completed and deployed to the public. A website has also been set up, which runs the web version of the system. The web address to the website is <https://happens.hk/>.

## **1.4 Structure**

This paper mainly contains two parts; First, it introduces the system. The development process, which includes the steps, methods and also the problems encountered during the study, is discussed in the first part of the paper.

Second, the remainder of the paper introduces the future development of the project. The schedule and alternatives of the future plan is discussed in the latter part of the paper. As we are in the middle of the development process, comments have been received in the previous months and some alternative functions and features could be added to our system. Recommendation is discussed in this paper and the system can have improvements afterwards.



## **2 Methodology**

This section is about the methodology of the project. Since this is a developmental project, the analysis of the set-up and design is discussed.

### **2.1 Platform setup**

As our system is deployed in both Android and iOS platform in mobile version, there are some hardware and software requirements to ensure the system provides the same user experience in both versions. Although two different platforms are set up for the development, the functionalities and the design of the user interface (UI) of the two versions will be made unanimous. This is because it is important to give the same user experience to any user, regardless they are using Android or iOS. The Android version is written in Kotlin using Android Studio while the iOS version is written in Swift using XCode. They are both official integrated development environment (IDE) for Android and iOS application.

For the web platform, React – a JavaScript library, is used to create a single page application (SPA). With this library, developers can easily make reusable components and handle a large amount of data. Also, a single page application can enhance the user experience since the client browser only needs to make a partial refresh on the webpage to retrieve updated information. This reduces the webpage loading time and make the application more interactive. There is no particular choice of the development platform but the web application will be deployed to a Linux server upon the completion.

## 2.2 User interface design

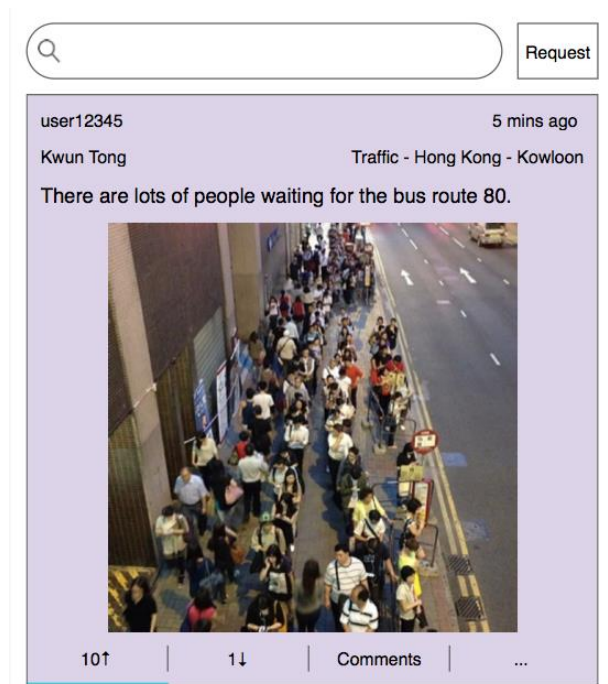


Figure 2a

*Post view of the application*

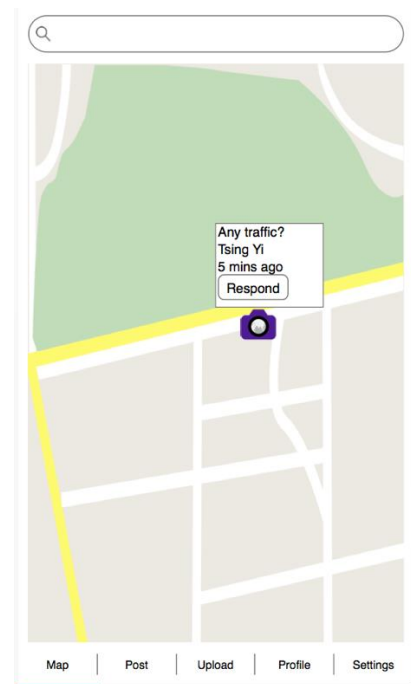


Figure 2b

*Map view of the application*

Figure 2 – User interface designs prototype of the mobile version of the application

The user interface (UI) design of a system plays a vital role in attracting user to use the application. A friendly interface is a must as it provides enhanced user experience to the users. Therefore, the front-end application prototype has a simple UI design containing only a main panel and a bottom tab-selection bar in each page. This design is consistent throughout every page in the system.

The post view and map view of the system are selected to be shown as an example.

For the post view, the posts are sorted one by one in a card form (see Figure 2a). In each post, information such as location, description and category of the post are shown along with the image. There are buttons below each post for users to vote, comment, or share the post. This is a likewise design to other communication application. Hence, users could smoothly adapt to this new system. Furthermore, users can efficiently explore posts they would like to read by entering searching criteria on the top search bar. The posts below will then be filtered correspondingly.

For the map view of the system (see Figure 2b), the main panel is a map with markers. The markers cluster the events happening in adjacent locations and user can click on the markers to see what is happening around. Moreover, user can see the requests on the map. User

can click on the marker and choose to respond to the requests.

The system uses common UI designs appearing on most of the existing applications presently, therefore, users can learn to use this new system in a short period of time. Moreover, the same UI design is applied to both iOS and Android version of the system. This gives identical user experience to both iOS and Android users. Even if users change their phone, they do not need to adapt to different versions of the application.

## **2.3 Database Design**

In this project, the database is conjointly important as the front-end application. Personal information for the authentication system, as well as the image and information of each post are stored in the database. It is important to have an organised database for handling substantially large amount of data.

Firebase is used for developing the back-end server. Firebase is a mobile and web application development platform running by Google. It is specifically designed for real-time and scalable application, since it can accommodate a large application that has over a million users. The server in this system is a centralised database that can be updated by every user. Firebase is capable of providing real-time data update between devices, even with a huge amount of data [1]. More importantly, Firebase organises the update of the data automatically. As a result, the data concurrency issue during update is dealt with it appropriately. It is unnecessary for us to develop an algorithm for safe data update. Firebase Realtime Database is a NoSQL Database which formulates the data in a key-and-value format. By designing the data structure carefully, same piece of data can be duplicated and stored in multiple places. This technique is especially useful because the data can be stored in the form of exactly how they will be retrieved and used by the client applications. Therefore, instead of using SQL to join multiple tables which is relatively slow, the server can directly return the requested data.

## **2.4 User Access and Security Design**

As mentioned in section 1.2.4, an authentication system is needed in our project for ensuring no user uses our system to do illegal events or posting inappropriate images. Firebase provides an easy set-up for the authentication system. Users can register accounts by providing their email and password. Firebase can verify the email address by sending a verification email to the user. After the link in the verification email is clicked, the account is verified, and the upload function can be used by the user. This authentication process is essential to verify the identity of the users.

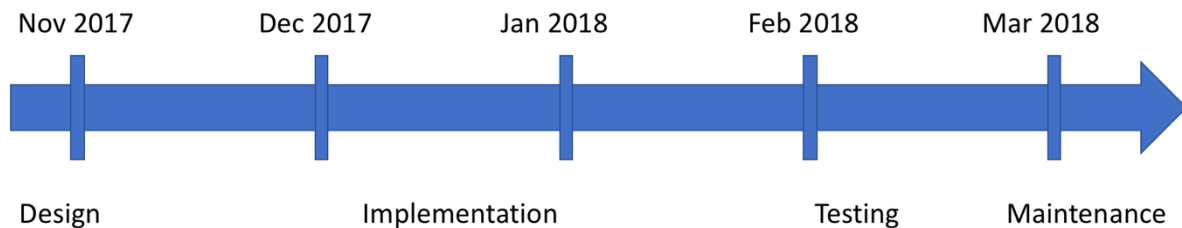
Security is important in the system as well. As the system would get sensitive location data from the user, security measure is done accordingly to protect personal data. The location data of each user is not saved in the database to minimise the opportunity of data breach. The system requires user permission to get the user location. Users can deny the permission at any time. Safety measure is done to prevent disclosure of personal data and the privacy of the user is preserved.

In fact, Firebase in more than just database, authentication and storage systems and they are integrated to work with each other. For example, the database and storage can be configured to be accessed (read and write) by authorized users only. Authorization used to be a complicated issue for secure system because the server needs to store sensitive data including passwords or token. Transmitting these data between client and server also needs to be considered. With Firebase Authentication, it has already taken care of the issues of storing these data and connecting to Firebase is automatically under a secure channel with HTTPS.

### 3 Current Status

This section is about the current status of the project. The work accomplished to date, and the future plan is discussed. Moreover, the schedule of the project is illustrated, to show the progress of the project.

#### 3.1 Work Accomplished to Date



*Figure 3 – Simplified Project Timeline*

The project is divided into four phases, including design phase, implementation phase, testing phase, and maintenance phase. As shown in Figure 3, the design phase is done previously, and the project is on-going the implementation phase.

##### 3.1.1 Design Phase

The first phase of the project is design phase. It is about the design of the architecture of the system. This project was initiated in late August 2017. The original idea was to develop a road condition sharing system. Nevertheless, the idea was too narrow and it was estimable that the target user group is small. After further discussion, the idea of information sharing is maintained, while the project is broadened to develop an instant event sharing system.

The detailed project plan is given to Dr. Chim, our supervisor, in late September 2017. After consultation, the scope of this project stated in section 1.2 is established. The approved design is concluded and implementation is based on the discussion results. Furthermore, the set-up and designs discussed in section 2 is determined in this stage. This stage is completed in November.

##### 3.1.2 Implementation Phase (Server)

The second phase of the project is implementation phase. This phase started immediately after the design phase.

The server set-up is implemented first. This is because it is the centre of the

system and it provides all the functions for the front-end applications. Currently, the structure of the database in Firebase has been set up, as a result, data can be stored in an organised manner. Data upload together with the data retrieval methods are carried out. Therefore, the photo sharing function is completed. Images, along with the post information, can be shared through the system. In addition, with Firebase prepared, the authentication system is finished. Users can be registered with verification through the system.

On the other hands, an Amazon Elastic Compute Cloud (Amazon EC2) instance has been set up. Its primary use is hosting the web application available at <https://happens.hk> (the website is under maintenance at the time of writing this paper). The above domain has been purchased and managed in <https://www.hkdnr.hk/>.

### **3.1.3 Implementation Phase (Client)**

The implementation for the client application in both mobile and web versions has been initiated as well.

First, the search engine (index by location, category, user), which can retrieve related posts corresponding to different filter options, is completed. For searching by location index, users enter the location name in the search box and then a Google maps view will be shown with the entered location being the center point of the map. Posts that are uploaded near this center point will be shown. The order of the posts will be shown from the greatest to the smallest timestamp (i.e. from the most recent to the oldest). For searching by category, since our application are general that users can post whatever things or events they want. It is supposed that there will be a huge amount of the types of content uploaded by users. Ranging from sports, music to education, entertainment and so on, the content will be in a terrible mess if all types of content are put together. Therefore, users can search by category. After entering the category that they are interested in, only posts within the category will be shown. For searching by user, if there is an active user who always post interesting things, then other users probably want to keep track of the status of that user. In this regard, user can search for a user name and subscribe the user. After the subscription, user will be alerted when the subscribed user makes a new post. Other than subscribing the searched user, the user profile can be viewed upon searching.

Second, the photo sharing function is completed. User can upload photo to the

system using the mobile application. Moreover, the voting and comment function alongside with the post has also been completed. Each user can either up-vote or down-vote on a post once. This is implemented as a statistic reflecting the quality of a post. The greater the number of upvoters, the more interesting or the more trustworthy the content is.

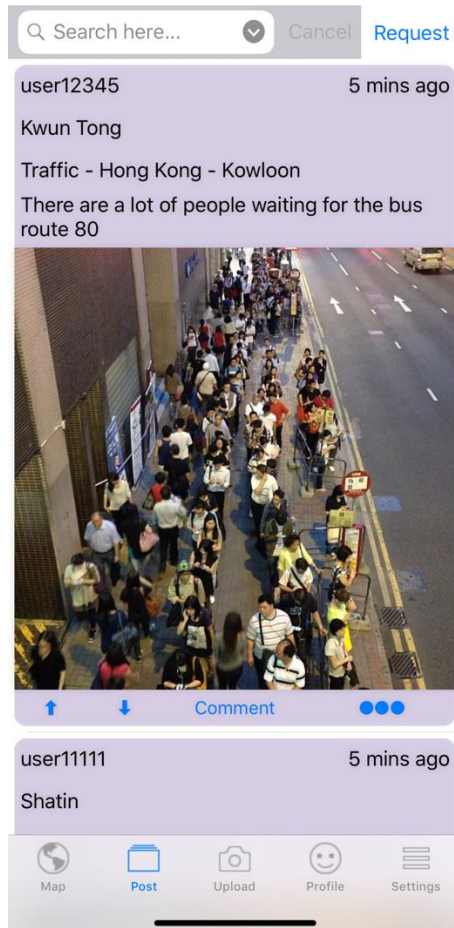


Figure 4a – Post view of the system

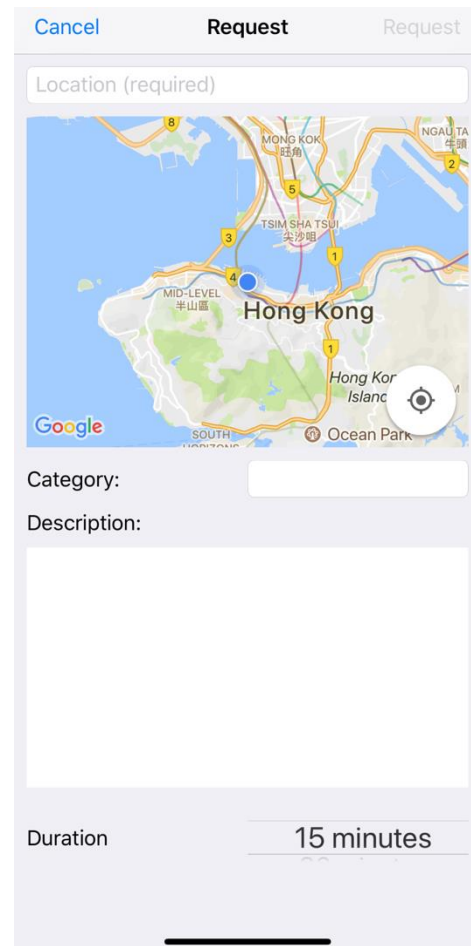


Figure 4b – Request view of the system

Figure 4 – Implemented user interface in iOS version

Moreover, the user interface is completed for the iOS version (see Figure 4). It is implemented according to the design prototype in section 2.2.

## **3.2 Problems Encountered**

There are two major problems discovered during the development. The first problem is about the choice of programming language for the front-end application, while the second problem is distinguishing different categories.

### **3.2.1 Choice of programming language for the front-end application**

In the initial plan of the project, React Native was planned to be used for the implementation of the front-end application. React Native is a framework using JavaScript language developed by Facebook, Inc. announced in 2015. The major feature of React Native is that it can generate a cross-platform application, i.e. running on both iOS and Android platform, in just one single implementation [2]. For instance, Facebook and Airbnb are using React Native.

However, during research and implementation, the problem of lacking official support from Google or Apple is found. For example, there is no official support from Google Maps API in React Native. Google Maps in React Native is only supported by third party and some main features such as location markers clustering cannot be implemented. The map is a major component in the system and it cannot be developed without the map. After discussing with the supervisor, two native versions of the system using Kotlin for Android and Swift for iOS are built separately. The platform setup is documented in section 2.1.

### **3.2.2 Distinguishing different categories**

Category system as mentioned in section 1.2.3 is a major function that help the users to categorize their information and for the other users to search for it. In the early stage of the project, we planned to make each category in the form of type of the information plus the location information. The location information may further consist of three levels (country, state, city) to indicate the detailed location of the information. For example, a piece of information may be categorized to “Traffic in Mongkok, Kowloon, Hong Kong”.

Nevertheless, during the implementation, we found that it is hard to design and develop a system for the user to input this information. The users are required to input 4 fields and it is not user-friendly. Hence, we has updated the design such that the category are in the form of one type plus one location. We will also make use of Google



### 3.3 Work Remaining

### 3.3.1 Function Implementation

A map of the NW 21st St area in Miami, Florida. A large black circle highlights a specific area bounded by NW 21st St to the north, NW 17th St to the south, NW 6th Ave to the west, and NW 10th Ave to the east. Within this circle, there is a red pin icon located near the intersection of NW 17th St and NW 8th Ave. The map shows various streets including NW 25th St, NW 23rd St, NW 21st St, NW 19th St, NW 17th St, NW 15th St, NW 14th St, NW 12th St, NW 10th St, NW 8th St, NW 6th St, NW 4th St, NW 2nd St, NW 1st St, NW 25th Ave, NW 23rd Ave, NW 21st Ave, NW 19th Ave, NW 17th Ave, NW 15th Ave, NW 14th Ave, NW 12th Ave, NW 10th Ave, NW 8th Ave, NW 6th Ave, NW 4th Ave, NW 2nd Ave, NW 1st Ave, NW 25th Pl, NW 23rd Pl, NW 21st Pl, NW 19th Pl, NW 17th Pl, NW 15th Pl, NW 14th Pl, NW 12th Pl, NW 10th Pl, NW 8th Pl, NW 6th Pl, NW 4th Pl, NW 2nd Pl, NW 1st Pl. Major roads like E West Expy (I-95) and Dolphin Expy (SR-112) are also visible. The map includes a scale bar indicating distances from 0 to 0.5 miles.

First, it is the request function. The request function is challenging because of the geo-fence set-up to send notification to nearby user. It is now designed to implement a fixed radius around the request location to find nearby user (see Figure 5). Later in the development, the request function may limit to a certain region or building to send to specific users.

Third, customized map markers should also be implemented. Since there will be many types of post, it would be better to distinguish between each type if different markers are used. In this regard, we will use colors and shapes to classify each type of

post. Moreover, there would be a number on top of the marker to cluster many posts nearby.

### **3.3.2 Testing and official launch**

Then, the project can be moved on to the testing phase. This phase focuses of different testing over the system. The testing phase is scheduled to be completed by March 2018. After all different tests, the system will be launched to the market and be publicly available for users. Then, the development is moved on to the final maintenance phase. This ensures the system is maintained working. Subsequently, there may be feedbacks coming from the users after the official launch. The system will be updated with new features suggested by the users.

## 4 Conclusion

This report has shown the first-half development progress of the final year project. This project was initiated by the insufficiency of existing functions in the current social media applications. A new instant event sharing system is decided to be built and it is a platform that can encourage people to exchange real-time information on different places. It takes references from the strength of existing applications, as well as providing new features to cover the deficiency.

This paper described the design and implementation of “happens”, the instant event sharing system. It has four important features, including the request-and-response function, photo sharing function, category system and the authentication system. The request-and-respond function let user inquire information based on location and the system connects users in the world by location.

Moreover, this project is to develop the system in both Android and iOS platform, as well as a web application. Android and iOS platforms almost cover all smartphone users in the world. Therefore, this system can be reached by a significant amount of smartphone users. Both applications would be native, i.e. written in officially supported language of Kotlin and Swift for Android and iOS respectively. Both versions are identical, to provide the same user experience to all users. On the other hand, the web version is written in React.js, but it has no photo sharing function, as we limit the system only accept image instantly captured from the camera. In addition, the back-end server is organised using Google Firebase. It is a software development platform, which provides all-in-one features to the development, such as database hosting and authentication system. It is convenient to use Firebase to manage the server side of the system.

Presently, the design phase of the project is done, and it is now in the implementation phase. Some function such as request-and-response system is not yet completed, while the UI design have been implemented. This phase is scheduled to be finished by February 2018 following by the testing phase. After several successful tests, the system will be launched to the market in March 2018.

Despite there are different problems appearing during the development, solutions are found to tackle them. The rationale to write two native applications, rather than using React Native to construct a cross-platform application, is that React Native is not officially supported by Google and Apple. This increases the difficulty of writing complicated functions, such as implementing the map and geo-fence.

During development, feedbacks are collected, and improvements should be made in the second half of the development schedule. Our future work is to finish implementation and testing firstly, following by maintaining the smooth running of the application as well as adding more features to it.

## References

- [1] Google Developers, “Firebase,” Google, 3 Oct 2017. [Online]. Available: <https://firebase.google.com/docs/database/usage/limits>. [Accessed 23 Dec 2017].
- [2] A. Narayan, “React Native vs Native iOS/Android,” Course Report, 15 11 2017. [Online]. Available: <https://www.coursereport.com/blog/so-you-want-to-build-a-mobile-app-react-native-vs-native-mobile>. [Accessed 30 Dec 2017].