

# Sistema de seguretat i localització destinat a activitats esportives a l'aire lliure

Kevin Armando Salcedo Hanco

**Resum**– A causa de la necessitat de voler tenir més seguretat i minimitzar les incerteses o perills que es poden generar quan es realitza una activitat a l'aire lliure amb un grup d'amics, ja sigui per si algú s'hagi perdut o, hi hagi un canvi meteorològic inesperat a la ruta. S'ha elaborat aquest projecte per poder satisfer aquestes necessitats.

Per poder dur a terme aquest propòsit, aquest projecte consta de dues aplicacions que proporcionen als usuaris tota la seguretat possible durant la seva activitat esportiva a l'aire lliure. La primera d'aquestes aplicacions consisteix en una aplicació mòbil destinada a l'ús dels usuaris per crear grups i la ruta a seguir. I, la segona aplicació consisteix en una aplicació web que serveix com a controlador per assegurar-se que els clients no tinguin cap inconvenient.

**Paraules clau**– aplicació mòbil, aplicació web, aplicació React Native, VueJS, aplicació de viatges, sistema de seguretat i localització

**Abstract**– Due to the need to want to have more security and minimize the uncertainties or dangers that can be generated when performing an outdoor activity with a group of friends, either in case someone is lost or, there is an unexpected weather change on the route. For these types of reasons this project has been carried out.

In order to carry out this purpose, this project consists of two applications that provide users with all possible safety during their outdoor sports activity. The first of these applications is a mobile application intended for use by users to create groups and the route to follow. And, the second app is a web app that serves as a driver to make sure customers don't have any issues.

**Keywords**– mobile application, web application, React Native application, VueJS, travel application, security and location system



## 1 INTRODUCCIÓ

AQUEST treball final de grau (TFG) va estar motivada pels imprevistos que poden sorgir quan es realitza una activitat a l'aire lliure. Si algun cop vas sortir a fer alguna mena d'activitat lliure, com fer una excursió per la muntanya, t'adonaràs compte que hi ha diversos perills sobretot si vas amb un grup i no es coneix bé la zona. Alguns d'aquests perills són la de patir algun accident que et prohibeixi continuar amb l'activitat i, si es tracta d'un lloc poc transitat, pot provocar un gran perill a l'individu. Un altre problema que comporta és la de què, en cas que es vagi amb grup, algun membre es desviï del camí que es va acordar i, per tant, aquest individu estaria en problemes si no coneix el lloc on han anat o hi hagi un canvi meteorològic i, pot provocar que es perdi.

A continuació es mostraran els diversos apartats en què

- E-mail de contacte: kevinarmando.salcedo@e-campus.uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Marta Prim Sabrià
- Curs 2019/2020

estarà estructurat l'article i una petita descripció del que es mostrarà en aquells apartats. En primer lloc, hi haurà un apartat per mostrar els objectius que es volien aconseguir durant el projecte, més endavant, es mostrarà la planificació que es va dur a terme per poder aconseguir els objectius, també es mostrarà quin és l'estat de l'art del projecte, en comparació d'aplicacions similars. Més endavant, complementari amb els objectius, s'ensenyarà la metodologia que es va aplicar per seguir amb la planificació. Després, es descriuran les tecnologies i recursos que es van aplicar per poder aconseguir els objectius del projecte; es mostrarà tant de l'aplicació mòbil com de l'aplicació web.

Els següents apartats, explicaran tots els components que conformen el projecte. El primer d'ells, és el servidor, on s'explicarà perquè s'utilitza, així com l'explicació de les diferents parts que té i, finalment, els passos per poder realitzar la instal·lació d'aquest servidor. Després, s'explicarà el perquè de l'aplicació mòbil, així com una descripció dels passos a seguir per poder instal·lar el *set up* de l'aplicació mòbil, més endavant es descriurà els diferents *layouts* que ho componen, s'explicarà les funcionalitats que té i, finalment, el funcionament de l'aplicació. Per la part de l'aplicació web, tindrà els mateixos apartats que l'aplicació mòbil,

a diferència que, en lloc d'explicar els *layouts*, s'explicarà els components d'aquest. Per altra banda, hi haurà un apartat on es detallarà la comunicació que hi va haver entre els diferents components del projecte, és a dir, el traspass de dades entre les aplicacions i el servidor.

Finalment, hi haurà un apartat on es comentarà les futures implementacions i millores que es podrien aplicar al projecte. A continuació, s'explicarà les conclusions que es van treure en finalitzar el projecte i, també, hi haurà un apartat pels agraïments. Per últim, està la secció de l'apèndix, on es mostrarà informació addicional que complementa l'article.

## 2 OBJECTIUS

Com es va mencionar a la introducció, aquest projecte va sorgir a partir d'unes necessitats que es demanaven a l'hora de realitzar activitats a l'aire lliure. Per això, l'objectiu d'aquest projecte consisteix a satisfer aquestes necessitats.

El principal objectiu és poder realitzar una activitat a l'aire lliure en un grup, amb la garantia que hi haurà algú que estigui pendent de la seguretat del grup. Per aconseguir aquest objectiu, aquest es va dividir en diversos objectius per poder-lo aconseguir.

Hi ha diferents objectius que estaran localitzats en una o ambdues aplicacions. En primer lloc, per la banda de l'aplicació mòbil es tractarà de recollir la màxima quantitat de dades, i necessàries, per poder veure l'estat del client i, respondre amb la precisió més gran possible.

1. El primer objectiu és la seguretat del grup, donant-los una ruta adequada, juntament amb el mostreig correcte d'aquesta ruta.
2. El segon objectiu és aconseguir la màxima comunicació entre els usuaris.
3. El tercer objectiu tracta de poder donar la posició que té l'usuari dins la ruta, tant dels membres del grup com de l'usuari minimitzant possibles desviacions d'un membre de la ruta.
4. El quart objectiu és donar l'assegurança a l'usuari que hi haurà algú que estigui pendent dels membres del grup.

Per la banda de l'aplicació web, els objectius es basaren a donar seguretat als clients, a partir de les dades que se li envien.

1. El primer objectiu tracta de poder visualitzar correctament les dades que se li envien.
2. El segon objectiu és actuar correctament en cas que sigui necessari.

Per aconseguir tots els objectius plantejats ha fet falta un aprenentatge o consolidació de noves tecnologies com:

1. Aprendre el funcionament d'aplicacions mòbils amb React Native, juntament amb els avantatges i desavantatges.
2. Aprendre sobre l'ús de les APIS de servicis Cloud, en aquest cas de Google, i fer ús d'aquest en aplicacions.

3. Aprendre sobre aplicacions web basades en VueJS, juntament amb els avantatges i desavantatges.
4. Aprendre sobre la comunicació entre aplicacions i el maneig de les dades.

## 3 ESTAT DE L'ART

Actualment trobem diverses aplicacions mòbils que ens poden crear rutes adequades i, fins i tot, qualificades per altres usuaris. Hi ha diverses possibilitats que ofereixen aquests tipus d'aplicacions, en alguns casos les rutes estan fetes per a un usuari únic i, en altres casos, la ruta es pot compartir o es pot crear un grup. Aquests tipus d'aplicacions per realitzar activitats a l'aire lliure o excursions, estan molt enfocades en crear rutes i, no pas al control de la seguretat de l'usuari.

A més, algunes aplicacions d'aquest estil tenen la limitació de ser exclusives d'un Sistema Operatiu, sigui Android, iOS o un altre. Aquests tipus d'aplicacions natives tenen un cost molt alt, en comparació a les alternatives de desenvolupament i, més encara si es vol desenvolupar per altres Sistemes Operatius. En els últims anys, han sortit noves formes de desenvolupament mòbil i, aquestes tenen els seus avantatges i desavantatges en comparació al desenvolupament híbrid. Tanmateix, en aquest cas, un dels objectius és la comunicació entre els usuaris i, no tots els grups tenen un mateix tipus de Sistema Operatiu mòbil.

En el present article es van abordar aquestes dues característiques, on hi ha poc o res d'aplicacions que ofereixen control de la seguretat de l'usuari i, aplicacions mòbils que es puguin utilitzar per diversos tipus d'usuaris mòbils.

## 4 PLANIFICACIÓ

Per tal poder entregar el projecte en la data d'entrega, s'ha hagut de realitzar una planificació. Com es veu a la Fig. 1, aquesta planificació va estar dividida en diverses fases, on cada fase es corresponia a una dada d'entrega d'informe de seguiment. Aquestes fases estan orientades al desenvolupament de diferents apartats del projecte:

1. **Informe Inicial:** Aquesta fase està dedicada a la planificació del projecte.
2. **Fase I:** Aquesta fase està dedicada al desenvolupament de l'esquelet de les dues aplicacions. A més, també de la creació de la base de dades i la seva connexió amb el servidor.
3. **Fase II:** Aquesta fase està dedicada al desenvolupament de les funcionalitats de les aplicacions.
4. **Propostes lliuraments:** En aquesta fase es fa el desenvolupament de les propostes dels lliuraments finals.
5. **Lliuraments Finals:** Aquesta fase està dedicada a l'entrega i presentació del projecte.

Nombre de tarea	Duración	Comienzo	Fin
▲ Informe Inicial	19,88 días	mar 11/02/20	dom 08/03/20
Reunió Inicial	0,88 días	mar 11/02/20	mar 11/02/20
Informe Inicial	18 días	mié 12/02/20	dom 08/03/20
▲ Fase I (bases projecte)	31 días	lun 09/03/20	lun 20/04/20
Creació diagrames projectes	3 días	lun 09/03/20	mié 11/03/20
Creació base web app	11 días	jue 12/03/20	vie 27/03/20
Creació base app mòbil	12 días	jue 12/03/20	vie 27/03/20
Connexió entre apps	7 días	vie 27/03/20	mar 07/04/20
Implementar DB	2 días	jue 09/04/20	dom 12/04/20
Realització nou informe	6 días	lun 13/04/20	lun 20/04/20
▲ Fase II (funcionalitats projecte)	25 días	lun 20/04/20	lun 25/05/20
Implementar funcionalitats web app	20 días	lun 20/04/20	dom 17/05/20
Implementar funcionalitats app mòbil	20 días	lun 20/04/20	dom 17/05/20
Realització informe 2	1 día	vie 22/05/20	dom 24/05/20
▲ Propostes lliuraments	25,88 días	lun 25/05/20	sáb 27/06/20
Proposta de presentació	13 días	vie 22/05/20	mié 10/06/20
Proposta d'informe	12 días	jue 11/06/20	sáb 27/06/20
▲ Lluraments Finals	9,88 días	dom 28/06/20	jue 09/07/20
Llurament Dossier	1 día	dom 28/06/20	dom 28/06/20
Llurament Poster	6 días	dom 28/06/20	dom 05/07/20
Defensa TFG	1 día	mié 08/07/20	jue 09/07/20

Fig. 1: Planificació del projecte

## 5 METODOLOGIA

Per tal d'aconseguir els objectius marcats anteriorment, es va aplicar una metodologia que ens permet aconseguir les tasques proposades a la planificació. La metodologia que es va aplicar en aquest projecte va ser la de Kanban [1], una metodologia àgil que ens permet visualitzar totes les tasques que es realitzaran al llarg del projecte.

Aquesta metodologia va ser escollida, degut a les diferents tasques que es tenia i, en ser el projecte d'una sola persona, no es podria realitzar les reunions que es calien. Kanban va proporcionar visualització de les tasques a realitzar, acabades i les que s'estan fent en aquell moment. En tenir les tasques bastant diferenciades, les dependències no eren abundants i, en cas que alguna tasca es compliqui, es podia començar una altra tasca i després retornar amb aquella.

Per últim, es va utilitzar un tauler *online* per tal visualitzar les tasques a realitzar en aquella fase. Aquest tauler es veu a la Fig. 2.

## 6 TECNOLOGIES I RECURSOS

En aquest apartat es mostrarà quines tecnologies i recursos es van utilitzar al llarg del projecte. A més, es parlarà de quin és el motiu pel qual van ser escollits.

En les dues aplicacions que es van desenvolupar, el llenguatge escollit per programar-los va ser JavaScript [2], ja que aquest és el principal llenguatge pel desenvolupament de pàgines web i ens permet realitzar activitats complexes en aquestes pàgines.

Per la banda de la base de dades, es va escollir el llenguatge SQL [3] per la seva facilitat per crear bases de dades, a més que treballa amb base de dades relacionals. Com a sistema de gestió de base de dades, es va escollir MySQL [4], a causa de la facilitat d'instal·lar i utilitzar el programa. En

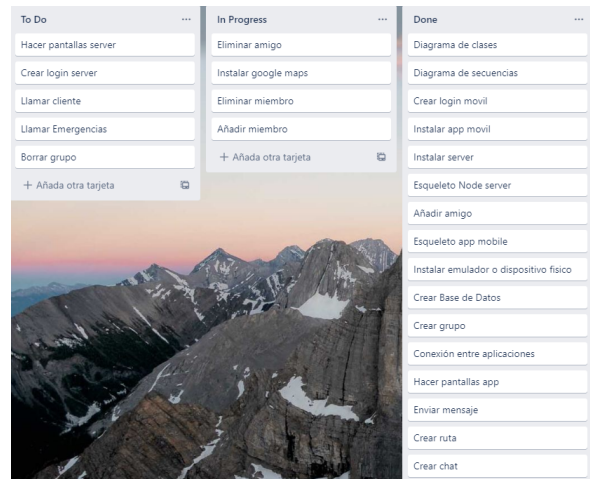


Fig. 2: Tauler Kanban amb les tasques a realitzar

aquest projecte es va escollir utilitzar la interfície d'usuari de MySQL Workbench, ja que aquest és l'entorn oficial de MySQL i, a més, es pot treballar en local i dependre d'estar connectat a Internet com pot ser phpMyAdmin; aquest entorn va ser utilitzat per crear la base de dades i crear les taules. Finalment, també es va utilitzar el MySQL Shell per visualitzar les dades de la base de dades, ja que aquest és més ràpid i no consumeix tants recursos com el *Workbench*.

Finalment, parlarem de les APIs de Google Cloud que fan servir les dues aplicacions i, és la base de les dues aplicacions, com que sense aquestes APIs, no s'hauria pogut realitzar el projecte. Hi ha 2 APIs de Google Cloud que hem fet servir:

- **Maps SDK for Android:** Aquesta API és la base de tot, ja que ens permet visualitzar un mapa amb vista de satèl·lit i utilitzar el GPS. En primer lloc, l'API ens permet visualitzar el mapa i poder configurar-lo de diverses formes, per exemple, es pot escollir quin tipus de visualització tindrà el mapa (satèl·lit, topogràfic, híbrid, etc.). Es pot configurar molt més, però en aquest projecte no és necessari.

En segon lloc, aquesta API ens permet utilitzar el GPS del dispositiu i, poder seguir la posició a temps real de l'usuari que utilitza l'aplicació.

Aquesta API és de lliure accés i, per tant, no hi ha costos per utilitzar-la.

- **Routes:** Aquesta segona API, ens permet crear la ruta que vol seguir el grup. Permet escollir dues posicions dins del mapa creat abans i, a partir d'aquí, crear la ruta. L'API ens ofereix diferents paràmetres per construir la ruta, un d'aquests és en quin mitjà de transport es realitzarà el trajecte i, a partir dels camins que té aquest mitjà de transport, es pot crear la ruta. En el cas d'aquest projecte, al ser dirigit a activitats a l'aire lliure, l'opció que es va escollir va ser la de "anar a peu".

L'API de Routes, a diferència de l'API de Maps, no és de lliure accés, és a dir, consumeix crèdits ja sigui pel nombre de consultes o pel temps utilitzat. Llavors, si es vol utilitzar aquesta API, hi ha dues opcions, la primera és ingressar diners dins del Cloud o utilitzar

una opció de Google Cloud, la qual ofereix una certa quantitat de crèdits per poder fer ús de diferents APIs que són de pagament.

Ara parlarem estrictament de les tecnologies i recursos que es van utilitzar, exclusivament, en les dues aplicacions.

## 6.1 Aplicació Mòbil

En l'aplicació mòbil, el llenguatge de programació utilitzat ha sigut JavaScript [2] i es va utilitzar React Native [5] per poder desenvolupar aplicacions tant per Android com per iOS, el motiu pel qual es va escollir React Native va ser que permet desenvolupar aplicacions en diferents sistemes operatius, ja que en un grup de persones, no tots porten un telèfon mòbil amb sistema Android o tots porten un amb iOS, normalment hi ha algun membre del grup que porta un mòbil d'un altre sistema operatiu. Si s'hagués desenvolupat en el llenguatge natiu del sistema, Java o Objective C, les persones que portarien un altre sistema operatiu no podrien connectar-se amb la resta del grup i, per tant, el projecte no tindria molt sentit.

El servidor que s'utilitza està basat en NodeJS, un entorn que treballa per la capa del servidor i utilitza el llenguatge de programació JavaScript; aquest servidor és compartit amb el de la pàgina web. Hi ha diferents motius pel qual va ser escollit, el primer motiu va ser que utilitza JavaScript, ja que aquest llenguatge és molt comú d'utilitzar i, per tant, no hi haurà grans complicacions en desenvolupar les aplicacions, a més, tot el projecte està fet amb JavaScript i no hi haurà problemes en manejar diferents tipus de llenguatges. El segon motiu és la facilitat de llibreries que ofereix NodeJS i els diferents *frameworks* que hi ha i que té compatibilitat, com React Native, React o VueJS.

Per últim, els recursos que utilitza l'aplicació mòbil per desplegar l'aplicació són la d'un dispositiu mòbil físic i un emulador Android, que ve instal·lat en el paquet d'Android Studio [6], que ens permetrà desplegar diferents dispositius mòbils que correran l'aplicació.

## 6.2 Aplicació Web

Per la banda de l'aplicació web, el llenguatge de programació utilitzat, com s'ha mencionat abans, també és JavaScript i utilitza el *framework* de VueJS, ja que l'aplicació web serà basada en una *single page* i que mostrarà les dades activament. El motiu pel qual s'ha escollit VueJS és per la compatibilitat amb NodeJS i NPM, a més, de l'ús de JavaScript com a principal llenguatge en totes les aplicacions. Per últim, en l'aplicació només hi haurà una sola pàgina que mostrarà totes les dades i, aquesta s'estarà actualitzant amb les dades que s'hagi seleccionat.

Com s'ha indicat abans, el servidor és compartit amb l'aplicació mòbil, i el motiu és que aquestes dues aplicacions han d'estar connectades perquè hi hagi comunicació i, sobretot, que la pàgina web mostri les dades que arriben de l'aplicació mòbil.

## 7 SERVIDOR

Tota aplicació que necessita mostrar diferents dades dependent de l'usuari, utilitza una base de dades i un servidor, el

primer per emmagatzemar les dades i la segona per accedir a ell. Per això, tant l'aplicació mòbil com l'aplicació web comparteixen i s'intercanvien dades que necessiten mostrar, per aquest motiu aquestes dues aplicacions hauran de compartir el servidor i aquest respondrà diferent dependent de l'aplicació.

Aquest servidor utilitzarà el llenguatge JavaScript i, utilitzarà NodeJS com a llenguatge del *back-end*. Aquest llenguatge és asíncron i ens permetrà realitzar funcions paral·lelament en lloc d'esperar que cada funció acabi perquè comenci una altra. La base de dades serà MySQL i utilitzarà el llenguatge SQL.

## 7.1 Instal·lació

En primer lloc s'haurà d'instal·lar el paquet de NodeJS [8]. Aquest paquet contindrà NodeJS i el seu gestor de paquets NPM. Aquest primer és un entorn de servidor que permet executar JavaScript per la banda del servidor. El segon, NPM, ens dona l'eina que ens permet instal·lar, actualitzar i desinstal·lar els diferents paquets que utilitzarem al llarg del projecte.

## 8 APLICACIÓ MÒBIL

L'aplicació mòbil que s'ha desenvolupat està destinada a l'ús d'un usuari qualsevol, és a dir, que té poc o cap coneixement tècnic. L'usuari que utilitza aquesta aplicació ho fa amb l'objectiu de crear un grup per poder realitzar una activitat a l'aire lliure, amb la seguretat que cap membre d'aquell grup tingui cap inconvenient o problema mentre es realitza la trajectòria.

## 8.1 Instal·lació

El primer pas per realitzar la instal·lació de React Native [7], és instal·lar l'entorn de NodeJS, aquest entorn proporciona tant NodeJS com NPM, requisits necessaris per a la instal·lació de React Native; si s'ha seguit la instal·lació del servidor, totes les dependències ja estaran instal·lades. Tots els paquets i llibreries que s'utilitza en React Native s'instal·la amb NPM.

En primer lloc, s'instal·la la llibreria *create-react-native-app*, una llibreria que ens permet la instal·lació de totes les carpetes, arxius i dependències que necessita una aplicació React Native. Un cop instal·lat tot correctament és el torn d'executar l'aplicació en un dispositiu.

Hi ha dues formes d'executar l'aplicació en un dispositiu. La primera és tenir un dispositiu mòbil físic, la segona és tenir un emulador que permet simular un dispositiu físic. En qualsevol dels casos, és necessari tenir AVD Manager, una interfície que ens permet administrar els Android Virtual Device (AVD). Aquesta interfície es pot instal·lar per si mateixa, però és més recomanable que s'instal·li amb l'Android Studio.

Una vegada tenim el dispositiu on córrer l'aplicació, s'haurà d'executar l'aplicació i comprovar que tot funciona correctament. A mesura que es vagi avançant en un projecte de React Native, les dependències s'hauran d'implementar mitjançant NPM.

## 8.2 Layout

Els diferents *layouts* que conté l'aplicació mòbil són les següents:

**Login** Aquest *layout* proporciona la funcionalitat de fer un *login* dins de l'aplicació. Has d'estar registrat dins de l'aplicació.

**Register** Aquest *layout* proporciona l'opció de registrar-se dins l'aplicació; per fer-ho s'han de posar les dades que es demanen. demanen.

**Home** Aquest és el *layout* principal una vegada s'ha registrat. Dona les funcionalitats principals que té un usuari. Les principals funcionalitats són la de FriendList i la de Group; el Chat hauria d'estar dins de FriendList.

**Group** Aquest *layout* proporciona les opcions que tens en crear un grup. Si no tens un grup creat, no hauries de poder veure les opcions. Les principals funcionalitats són la de CreateRoute o veure el Route, l'altra és la de veure els detalls de la llista de membres.

**Route** Aquest és el *layout* on es mostra la ruta creada pel grup.

**MapOrigin** Aquest és part de Route i té la funció d'escollir el punt d'origen de la ruta.

**MapDest** Aquest és part de Route i té la funció d'escollir el punt de destí de la ruta.

**Chat** Aquest *layout* proporciona l'opció de parlar amb un usuari de la llista d'amics.

## 8.3 Funcionalitats

L'aplicació ofereix diferents funcionalitats depenent del *layout* on ens trobem:

**Sign in** Aquesta funcionalitat pertany al *layout* Login i, dona la funció d'iniciar sessió en l'aplicació.

**Sign up** Aquesta funcionalitat pertany al *layout* Login i, dona la funció d'enregistrar-se en l'aplicació.

**Register** Aquesta funcionalitat pertany al *layout* Register i, dona la funció d'enviar les dades per crear l'usuari.

**Group** Aquesta funcionalitat pertany al *layout* Home i, dona la funció d'anar al *layout* de Group i accedir a totes les seves funcionalitats.

**FList** Aquesta funcionalitat pertany al *layout* Home i, dona la funció de veure tota la llista d'amics que tens en aquell moment.

**Route** Aquesta funcionalitat pertany al *layout* Group i, dona la funció de veure la ruta creada pel grup. Si no hi ha cap ruta creada, et deixarà crear una nova ruta.

**GMembers** Aquesta funcionalitat pertany al *layout* Group i, dona la funció de veure tots els membres d'aquell grup.

**Origin** Aquesta funcionalitat pertany al *layout* Route i, dona la funció de seleccionar el punt d'origen de la ruta.

**Dest** Aquesta funcionalitat pertany al *layout* Route i, dona la funció de seleccionar el punt de destí de la ruta.

**CSelect** Aquesta funcionalitat pertany al *layout* Route i, dona la funció de veure i confirmar la ruta creada a partir de l'origen i destí.

**Chat** Aquesta funcionalitat pertany al *layout* Chat i, dona la funció de realitzar una comunicació amb un altre usuari.

## 8.4 Funcionament

En aquest apartat s'explicarà com un usuari nou hauria d'utilitzar l'aplicació.

A l'inici de l'aplicació, surt un *layout* on es demana que es faci un *login*, tanmateix, en aquest punt l'usuari no hauria de tenir un compte, per això, primer s'ha d'enregistrar i, un cop es cregui un compte ja es podria fer el *login* i es podria entrar a l'aplicació.

Ja dins l'aplicació es trobarà el *layout* principal, el Home, on es podrà veure dues opcions, la de FriendList i la de Group. Aquesta primera opció permet entrar en el *layout* de FriendList i veure la llista d'amics que té l'usuari, en aquest *layout* també es podrà afegir i eliminar un amic. Per tal d'afegir un nou amic, s'ha de saber el seu *nickname* en l'aplicació, també, aquest amic afegit ha d'estar connectat per poder respondre la invitació; aquesta invitació sortirà a la part de baix del *layout* de l'altre usuari i, aquest decidirà si acceptar o no la invitació. Per últim, dins d'aquest *layout* es pot realitzar un chat amb un amic si se selecciona la seva casella.

Després, dins del *layout* de Group, si no hi ha cap grup, el *layout* donarà l'opció de crear-ne un o veure la sol·licitud d'entrar a un grup. El *layout* de Group es veu a la Fig. 3. En el cas d'esperar una sol·licitud, només cal acceptar la invitació i, ja s'actualitzarà amb les dades de l'usuari amb aquest grup i amb totes les seves característiques. Per l'altra banda, si es vol crear un grup, es demanarà que es cregui una ruta, en la Fig. 4 es veu el *layout* de confirmació de ruta. En acabar de crear la ruta ja et deixarà accedir als camps del *layout* Group.

Un cop dins d'aquest actualitzat *layout* de Group, sortirà unes noves opcions, tal com es veu a la Fig. 5. Aquestes són 4 opcions i són les següents: Route, Members, Members Location i Delete Group. L'opció de Route permet veure la ruta que es va crear per aquell viatge, a més, mostra la ubicació on es troba l'usuari en aquell moment. L'opció de Members permet veure tots els membres del grup, inclòs l'usuari. Té un funcionament semblant al de FriendList, des d'aquest *layout* es pot afegir nous membres i eliminar-los, també es pot realitzar un xat amb un altre membre del grup; per tal d'afegir un nou membre, s'ha de saber el seu *nickname* i, que aquest estigui connectat.

Després, la següent opció del *layout* és el Members Location, on es podrà veure quina és la posició del membre en aquell moment dins la ruta. Per últim, l'altra opció és el Delete Group, on es podrà eliminar el grup i totes les relacions amb els membres del grup. Un cop s'hagi recorregut la ruta, aquesta opció permetrà acabar-la i poder crear-ne una de nou.

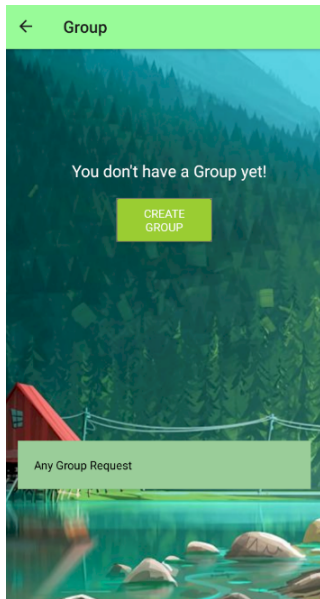


Fig. 3: *Layout* on es mostra el grup abans de crear-lo o d'acceptar-ne algun

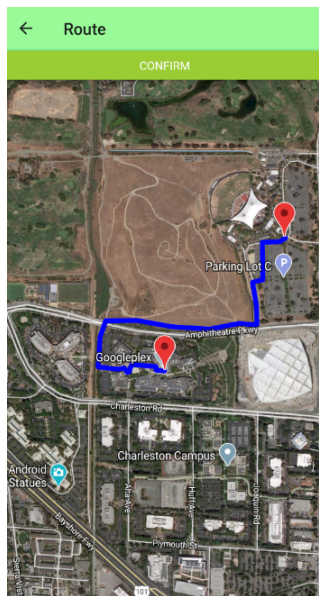


Fig. 4: *layout* on es confirma la ruta seleccionada

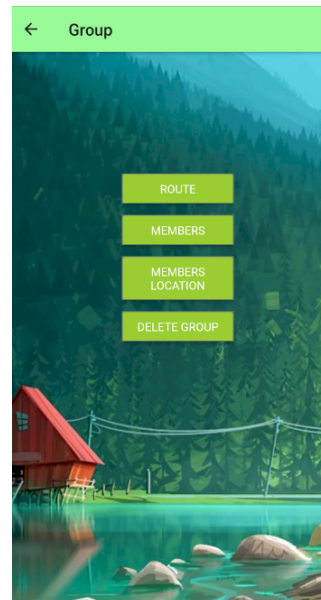


Fig. 5: *Layout* on es mostra totes les opcions del grup creat

## 9 APLICACIÓ WEB

Aquest component va ser dissenyat amb l'objectiu de poder controlar i informar els usuaris de l'aplicació mòbil i actuar, en cas que sigui necessari, en conseqüència sobre les accions que realitzen. Aquesta aplicació només serà accessible per a usuaris de confiança, que actuaran com a Administradors, on la seva funció és comprovar que els clients no pateixin cap inconvenient.

L'aplicació web utilitza per a la banda del servidor l'entorn de NodeJS. El servidor ha sigut descrita amb detall a la secció 7. Per la banda del client utilitza VueJS, els recursos que utilitza l'aplicació web són a la secció 6.2.

### 9.1 Instal·lació

La instal·lació de l'aplicació web cal, en primer lloc, connectar el servidor amb l'aplicació web, fent que només quan s'executi el servidor l'aplicació funcioni. Aquesta part ho fem amb el *framework* Express, que ens permet fer l'encaminament de la nostra aplicació.

Un cop tenim connectada la nostra aplicació, cal tenir instal·lat VueJS per poder utilitzar-la en la part del *frontend*. Aquesta part ho podem fer tant, tenint Vue en el nostre projecte o fent referència al seu URL. Fins a aquest punt, ja s'hauria de poder utilitzar VueJS en la nostra aplicació.

Per últim, hem d'utilitzar un *plugin* de VueJS que ens permeti utilitzar Google Maps en l'aplicació.

### 9.2 Components

Com hem explicat abans, VueJS és una aplicació d'una sola pàgina, per tant, només hi haurà un *layout* compost per diferents components que permet realitzar diferents funcionalitats. Aquest *layout* proporciona totes les accions que pot realitzar l'Administrador.

**Login** Aquest component s'encarrega de comprovar si s'està enregistrat a la base de dades i, veure si és un usuari vàlid (Administrador).

**GroupsList** Aquest component mostra una llista dels diferents grups creats i connectats.

**GroupRoute** Aquest component mostra la ruta del grup seleccionat.

**GroupMembers** Aquest component mostra els membres del grup seleccionat.

### 9.3 Funcionalitats

Aquestes són les diferents accions que es poden fer dins de l'aplicació web:

**Sign in** L'aplicació permet comprovar si s'està enregistrat a la base de dades i, si es tracta d'un usuari vàlid.

**Select Group** Un cop mostrat la llista de grups, es pot seleccionar qualsevol d'ells i veure les seves característiques.

**Send alert** Un cop seleccionat el grup, l'Administrador pot enviar una alerta a tots els membres del grup.

**Send message** Un cop seleccionat el grup, l'Administrador pot enviar un missatge a un membre del grup.

**Member Details** Es pot veure els detalls de qualsevol membre del grup.

### 9.4 Funcionament

En aquest apart es mostrarà com un usuari ha d'utilitzar l'aplicació web.

El funcionament de l'aplicació web és bastant més senzill que el de l'aplicació mòbil, com que només s'ha de fer de controlador. En aquesta aplicació només es veurà un *layout* que canvia depenent de les nostres accions i mostrarà els components adequats.

En primer lloc, es veurà el component del Login, on l'aplicació web s'assegurarà que l'email i la contrasenya són les correctes i, a més, que siguis un usuari vàlid (Administrador). No hi haurà un component per registrar-se com que només hi haurà persones de confiança i, l'única forma d'estar en la base de dades és registrar-te manualment.

Un cop s'hagi validat les teves dades, es mostrarà el component amb la llista de tots els grups que estan connectats. L'Administrador pot seleccionar qualsevol grup i, es mostraran diferents components amb la data del grup. Aquests components són els següents: GroupList, GroupRoute i GroupMembers.

Ja tenint desplecats els diferents components, com es veu a la Fig. 6, es podrà veure les diferents opcions que es poden fer. Si es vol fer un xat amb algun membre del grup seleccionat, només cal fer *click* en el botó que està al costat del *nickname* del membre del grup. Si es vol enviar una alerta a tots els membres del grup, cal fer *click* en el botó d'Alert al costat del grup. Per últim, per veure la posició dels membres del grup, només cal fer *click* en el seu *nickname*.

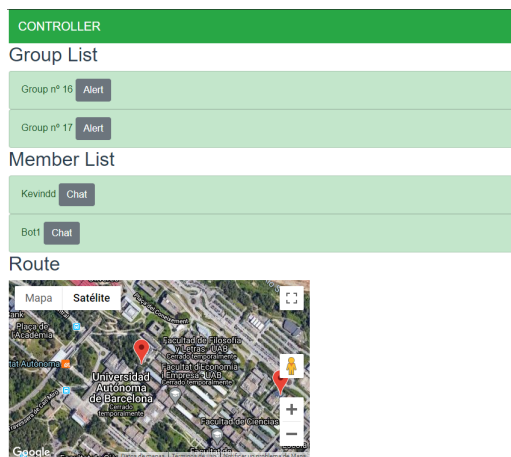


Fig. 6: Aplicació web amb els components del grup seleccionat

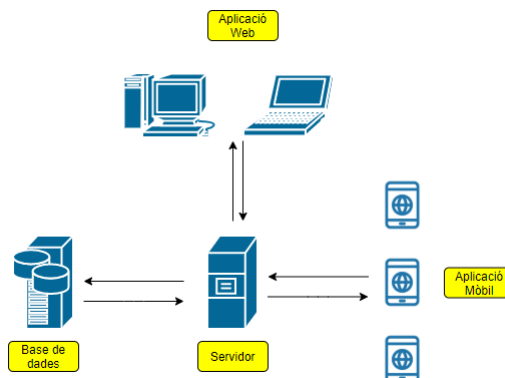


Fig. 7: Diagrama de xarxa

## 10 COMUNICACIÓ

En aquest apartat es tractarà de com les diferents aplicacions es comuniquen i quins *frameworks* o biblioteques es fan servir per fer-ho. El diagrama de la xarxa es veu a la Fig. 7.

En primer lloc parlarem de la comunicació Servidor-AppMòbil. Hi ha dues formes de realitzar aquesta comunicació.

La primera es realitza amb l'aplicació mòbil fent un *request* i el servidor actua en resposta a aquest *request*. Aquesta comunicació fa servir el *framework* d'Express i utilitza el protocol HTTP on realitza els diferents mètodes de sol·licitud HTTP (GET, POST, etc.). Aquest respon únicament a l'aplicació que ha utilitzat aquests mètodes de sol·licitud.

La segona utilitza la biblioteca Socket.io [9], una biblioteca que permet l'enviament de dades tant per la banda del servidor, com per la banda de l'aplicació; ha d'estar instal·lat tant al servidor com en l'aplicació. Aquesta biblioteca utilitza els esdeveniments per fer ús de les seves funcions. S'utilitza aquesta biblioteca, ja que aquest ens permet l'enviament de dades a un o diversos dispositius; normalment s'utilitza aquesta biblioteca per crear aplicacions de missatgeria.

Després, està la comunicació servidor-base de dades, i aquesta s'utilitza per rebre, inserir, esborrar o modificar les dades de la base de dades. La comunicació es fa mitjançant el mòdul *mysql*, aquest ens permet crear una connexió amb la base de dades mitjançant les credencials d'aquesta. Una vegada connectada, aquest mòdul ens permet fer diferents *queries* a la base de dades.

Per últim, està la comunicació que hi ha entre l'aplicació web amb el servidor i l'aplicació mòbil. En el primer cas, la comunicació funciona per mitjà dels *requests* del *fetch* i amb les respostes amb GET i POST per part del servidor. Tanmateix, la comunicació amb un dispositiu mòbil no va ser possible.

## 11 TESTS

En aquest apartat es mostrarà els diferents tests que s'han aplicat al llarg del projecte, per comprovar que s'han realitzat correctament les funcionalitats.

### 11.1 Aplicació mòbil

En l'aplicació mòbil es van realitzar els següents tests per comprovar la correctesa del funcionament:

1. En iniciar sessió, comprovar que l'usuari estigui registrat en la base de dades.
2. En iniciar sessió, comprovar que la contrasenya no es mostri per pantalla.
3. En iniciar sessió, comprovar que l'usuari no pugui accedir mitjançant codi maliciós.
4. En registrar un usuari, comprovar que posi les dades amb el format correcte.
5. En registrar un usuari, fer que la contrasenya es xifri a la base de dades.

6. En registrar un usuari, comprovar que no s'injecti codi maliciós.
7. En estar a Home, comprovar que els botons redirigeixen correctament.
8. En estar a FriendList, comprovar que s'estigui visualitzant correctament la llista d'amics.
9. En estar a FriendList, comprovar que es pugui eliminar un amic i, que s'elimini en els dos usuaris.
10. En estar a FriendList, comprovar que es pugui afegir un amic que sigui un usuari vàlid per a l'aplicació.
11. En estar a FriendList, comprovar que la sol·licitud arribi correctament a l'usuari que ha fet la proposta.
12. En estar a FriendList, comprovar que la resposta de la sol·licitud arribi tant a l'emissor com al receptor.
13. En estar a FriendList, fer que la llista d'amics s'actualitzi, o no depenent de la resposta d'amistat d'un altre usuari.
14. En estar a Group, comprovar que es mostri un *layout* diferent, depenent si hi ha un grup creat o no.
15. En estar a Group, si no hi ha grup creat, comprovar que es pugui crear un grup correctament.
16. En estar a Group, si no hi ha grup creat, comprovar que la ruta creada estigui formada per punts vàlids i, que la ruta, sigui creada correctament.
17. En estar a Group, si no hi ha grup creat, comprovar que arribin correctament les propostes de sol·licitud de grup.
18. En estar a Group, si n'hi ha grup, comprovar que la ruta creada sigui la correcta.
19. En estar a Group, si n'hi ha grup, comprovar que es veu la posició de l'usuari dins la ruta.
20. En estar a Group, si n'hi ha grup, comprovar que tots els membres siguin els correctes.
21. En estar a Group, si n'hi ha grup, comprovar que es pot agregar un nou membre a partir d'un usuari vàlid dins l'aplicació.
22. En estar a Group, si n'hi ha grup, comprovar que es pot veure la posició de tots els membres del grup.
23. En estar a Group, si n'hi ha grup, comprovar que es pugui esborrar el grup i, que aquest es reflecteixi en tots els membres del grup.

### 11.2 Aplicació Web

Aquests són els diferents tests que havia de passar les diferents funcionalitats de l'aplicació web:

1. Comprovar que l'usuari estigui enregistrat a la base de dades.
2. Comprovar que l'usuari sigui un usuari de tipus Admin.

3. Comprovar que la contrasenya estigui xifrada.
4. Comprovar que la contrasenya no es mostri per pantalla.
5. Comprovar que la llista de grups sigui les que estiguin en actiu en aquell moment.
6. Comprovar que només es mostri per pantalla la llista de grups sense abans fer *click* en un grup.
7. Comprovar que es mostrin correctament els components, una vegada s'hagi seleccionat un grup.
8. Comprovar que els membres del grup i la ruta són les correctes.
9. Comprovar que es veu correctament la ruta del grup.
10. Comprovar que canvia la informació si se selecciona un altre grup.

## 12 LÍNIES DE FUTUR

En aquest apartat es veurà quines noves implementacions es podrien o s'haurien d'implementar. En el primer lloc, es parlarà dels apartats que no s'han pogut acabar, en aquest cas seria la part d'enviament de dades de l'aplicació web a un dispositiu mòbil en específic, ja que permet actuar dependent de l'activitat del client enviant-li missatges o alertes. També, un apartat a afegir seria la de poder realitzar trucades als usuaris, en cas que la connexió a Internet caigui, aquest és punt important per acabar de complir amb els objectius de seguretat dels usuaris.

Ara bé, per la part de l'aplicació mòbil, les noves implementacions vindrien per la part de professionalitzar més l'aplicació. En primer lloc, pujar-lo a un servidor de Cloud i així no dependre d'un ordinador personal que té moltes limitacions. També s'hauria de pujar-lo a les diferents tendes d'aplicacions, com pot ser la Play Store o l'Apple Store.

Per últim, s'hauria de fer una millora en la part de *frontend*, on no hi ha hagut un aprofundiment en l'estudi del disseny ni l'adaptabilitat del disseny a diferents dispositius mòbils. Crear eines per millorar l'experiència entre usuari i sistema.

## 13 CONCLUSIONS

Com a conclusions finals es pot veure que s'han arribat a concretar la majoria dels objectius, només en la part de l'aplicació web han hagut problemes a l'hora de concretar-los. Per això, a pesar de realitzar una planificació, s'ha de tenir en compte els retards i els complicaments. En aquest cas, el problema va ser la inexperiència a l'hora de crear un projecte i la mesura dels seus temps.

El projecte va tindre moltes dificultats en tot el seu desenvolupament, tanmateix, es comentaran els més destacats. En primer lloc, va ser a l'hora d'utilitzar React Native i la seva instal·lació, on es va manejar per primer cop aquest llenguatge. Després, va ser implementar una API d'un servei Cloud, en aquest cas Google Maps. Però, la dificultat més gran que va tenir el projecte, va ser en el moment que es volia enviar dades d'un dispositiu mòbil a un altre, per exemple, a l'hora d'enviar-li un missatge. Per últim,

l'última complicació va ser en haver de reestructurar l'aplicació web, com que en la primera versió no es podia afegir *plugins* i, per tant, no es podia utilitzar Google Maps.

Tanmateix, com ja es va mencionar, es van complir la majoria dels objectius i, es va aprendre molt de les noves tecnologies, com són React Native, VueJS i NodeJS. A més, s'ha après que una bona planificació pot portar a un bon resultat, també s'ha vist que un error a la part inicial del projecte pot créixer molt a mesura que s'avança en el projecte. Un cop es té una estructura i una planificació del codi, tornar enrere és molt complicat i tardà, per això s'ha de fer una anàlisi més exhaustiu del llenguatge a programar juntament amb els *plugins* i eines que es vulguin utilitzar.

Per la part del desenvolupament mòbil, es van trobar algunes dificultats a l'hora de crear l'aplicació mòbil en React Native. En ser una tecnologia relativament nova, va haver-hi documentació escassa o, en alguns casos mal documentats. Aquest tipus de desenvolupament encara té moltes limitacions en comparació del desenvolupament natiu i, a pesar que en alguns casos pugui ser una millor opció, en aplicacions més complexes no és una bona opció per aquests tipus de limitants. Tanmateix, per desenvolupament que necessitin poc ús d'eines molt específiques d'algun Sistema Operatiu i, que no siguin molt complexes, aquest tipus de desenvolupament és una bona opció, ja que el cost de desenvolupament és més baix.

Com a conclusió de les dues aplicacions, l'aplicació mòbil ofereix moltes característiques que ens permet realitzar una activitat a l'aire lliure amb tota la confiança. Tanmateix, encara falten més implementacions i sobretot que tingui una interfície d'usuari més refinada. Per la banda de l'aplicació web, s'aconsegueix visualitzar les dades que ens interessa per tal de comprovar la seguretat dels nostres usuaris.

## AGRAÏMENTS

En primer lloc, cal agrair a la meva tutora, Marta Prim, ja que sempre va estar disponible cada cop que vaig tenir un dubte, a més, de tenir molta comprensió a l'hora de realitzar alguna reunió i, a l'hora de donar *feed-back* en les meves entregues. També agrair al coordinador Jordi Pons, per tenir tot ben documentat i, en cas d'un dubte, respondre ràpidament. Per últim, als professors de STW que em van donar molta documentació pel llenguatge de NodeJS.

Finalment, agrair a la Universitat Autònoma de Barcelona (UAB), que a pesar de les complicacions amb el confinament, van donar solucions per poder entregar amb tranquil·litat el projecte.

## REFERÈNCIES

- [1] kanbanize, "Kanban: explicación", [En línia], Disponible a: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>, Accedit: 05 de Març de 2020.
- [2] developer.mozilla, "JavaScript", [En línia], Disponible a: <https://developer.mozilla.org/es/docs/Web/JavaScript>, Accedit: 05 de Març de 2020.

- [3] Carlos Eduardo Plasencia Prado, “¿Qué es y por qué aprender SQL?”, DevCode, [En línia], Disponible a: <https://devcode.la/blog/que-es-sql/>, Accedit: 06 de març de 2020
- [4] MySQL.com, “Why MySQL?”, [En línia], Disponible a: <https://www.mysql.com/why-mysql/>, Accedit: 06 de març de 2020
- [5] ReactNative.dev, React Native, [En línia], Disponible a: <https://reactnative.dev/>, Accedit: 07 de març de 2020
- [6] Jamal Eason, “Android Emulator - AMD”, Android Developers, Nov 2018, [En línia], Disponible a: <https://android-developers.googleblog.com/2018/07/android-emulator-amd-processor-hyper-v.html>, Accedit: 14 d'abril de 2020
- [7] TutorialsPoint.com, “Environment Setup”, [En línia], Disponible a: [https://www.tutorialspoint.com/react\\\_native/react\\\_native\\\_environment\\\_setup.htm](https://www.tutorialspoint.com/react/_native/react_native_environment_setup.htm), Accedit: 15 d'abril de 2020
- [8] Elivar Largo, “Cómo instalar nodeJS y NPM”, ecodeup, Maig 2018, [En línia], Disponible a: <https://www.ecodeup.com/como-instalar-node-js-y-su-gestor-de-paquetes-npm-en-windows/>, Accedit: 15 d'abril de 2020
- [9] npmjs.com, “Socket.io”, [En línia], Disponible a: <https://www.npmjs.com/package/socket.io>, Accedit: 02 de maig de 2020

## APÈNDIX

### A.1 Base de Dades

En aquest apartat de l'apèndix podràs veure el diagrama de classes que té la base de dades del projecte.

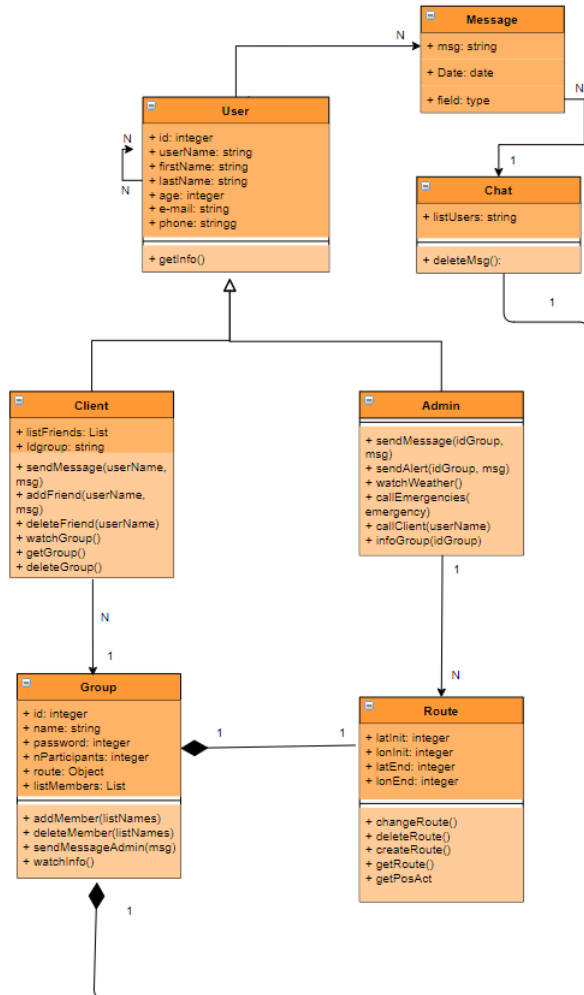


Fig. 8: Base de dades del projecte

### A.2 Aplicació mòbil layouts

A continuació es mostrarà tots els layouts que hi ha en l'aplicació mòbil.

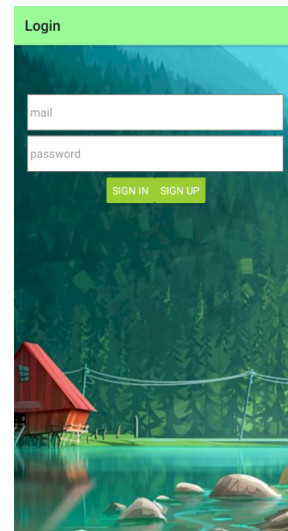


Fig. 9: Login

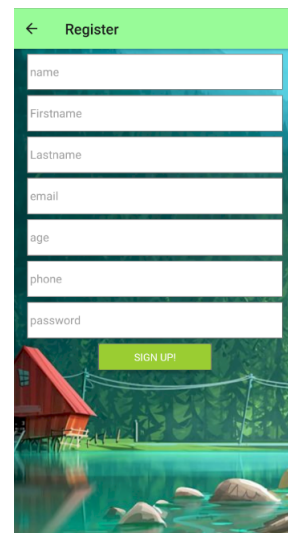


Fig. 10: Register

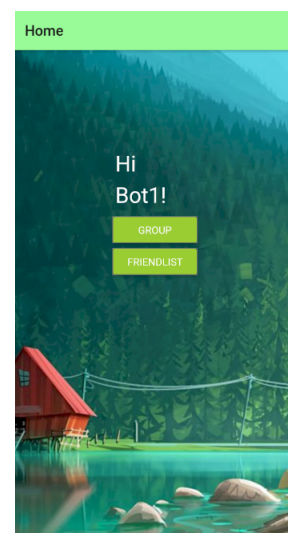


Fig. 11: Home

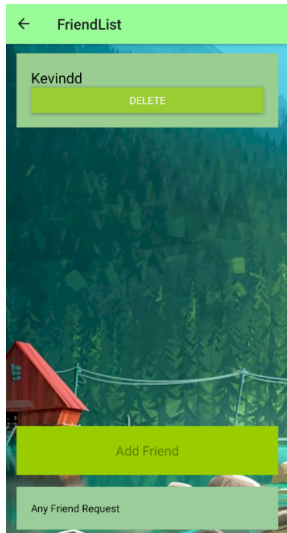


Fig. 12: Friend List

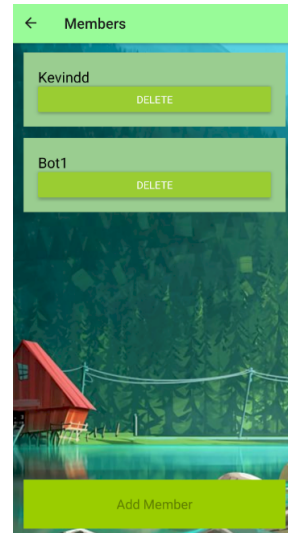


Fig. 15: Member List

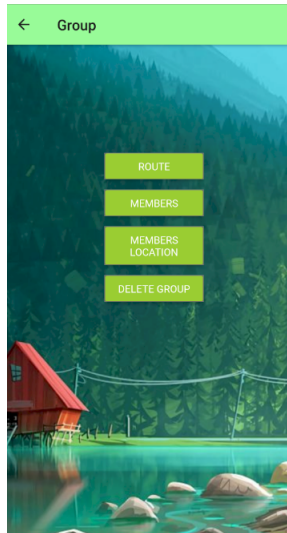


Fig. 13: Group

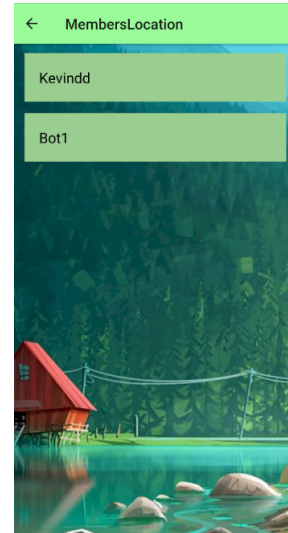


Fig. 16: Member Location

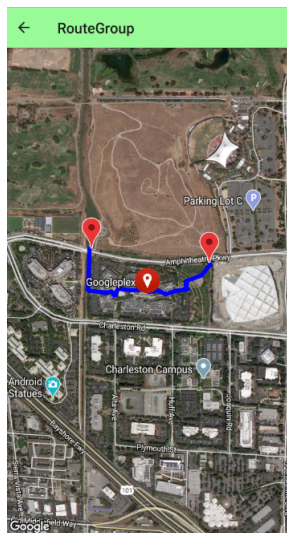


Fig. 14: Route

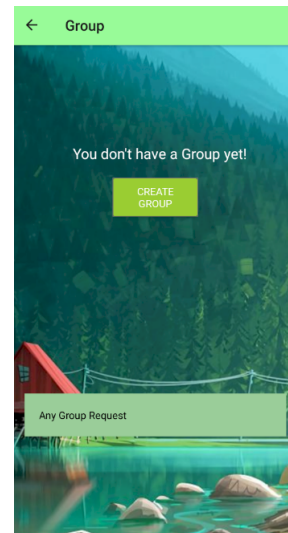


Fig. 17: Create Group

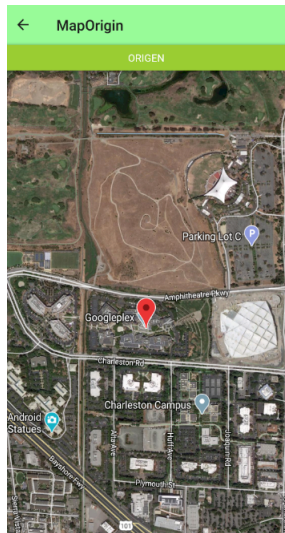


Fig. 18: Origin

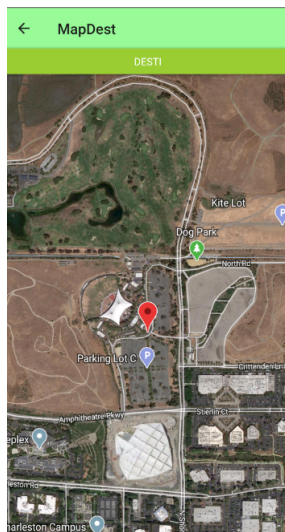


Fig. 19: Destination

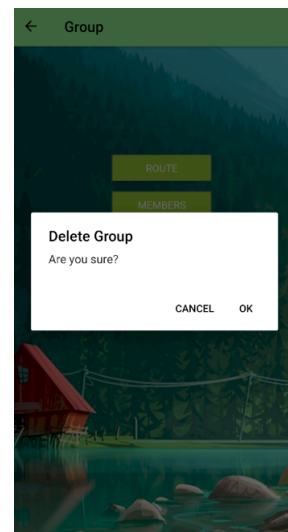


Fig. 21: Delete Group

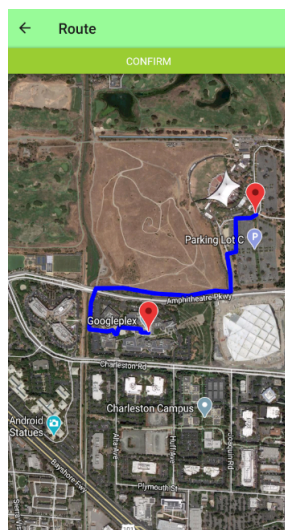


Fig. 20: Confirm Route