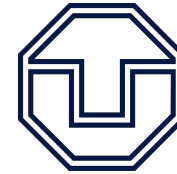


# Application Development for Mobile and Ubiquitous Computing

## Erasmus Web

Maria Sofia Nascimento  
Mariana Aires

## Idea



- Location-based event planner;
  - Centralization of events happening in an area
- Matching users with **fellow students** in the same situation;
  - **Stimulate connections** between users via interactions in the application.



  
erasmus Web

# Features of the application

## ➤ A user can:

- (login/logout)
- see the list of events;
- search and filter those events (search function);
- join a event;
- search for others users;
- make a friend request to another user;
- accept friend requests;
- see a map view of the events;
- see its own profile;
- set up its profile;
- create a event.

# Challenges

## ➤ Usability

- Minimize user input interaction;
- Transmit relevant and opportune information (via notifications);
- Change blindness;
- Difficulty searching for events of interest;

## ➤ Energy

- Minimize the energy consumption caused by GPS usage.

## ➤ Connectivity

- Prefetching of information;

# Usability

- Minimize user interaction

## Physical Context:

- Detect user location with GPS;

```
LocationManager.GPS_PROVIDER && LocationManager.NETWORK_PROVIDER
```

## Adaptation:

- Center the map on user's location;
- Display a (initial) list of events with events whose location is within a certain radius of the users' location, filtering results by event date.



# Usability

- Transmit relevant and opportune information (via notifications);

## Physical Context:

- Detect user location with GPS;

`LocationManager.GPS_PROVIDER` && `LocationManager.NETWORK_PROVIDER`

- Receive updates with events happening nearby;

`Firebase Cloud Functions` && `LocationManager.GPS_PROVIDER`

## Adaptation:

- Send push-notifications when the **user is within 1 km of a event within the next hour**, prompting the user with the event page.



# Usability

- Change blindness

## Physical Context:

- When a user joins an event, the server is triggered and notifies all user's friends (who had joined the event);

`joinEvent(User me, Event e)` && **Firestore** Cloud Functions && **Firestore** RealTime Database

### On server side (when user joins a event):

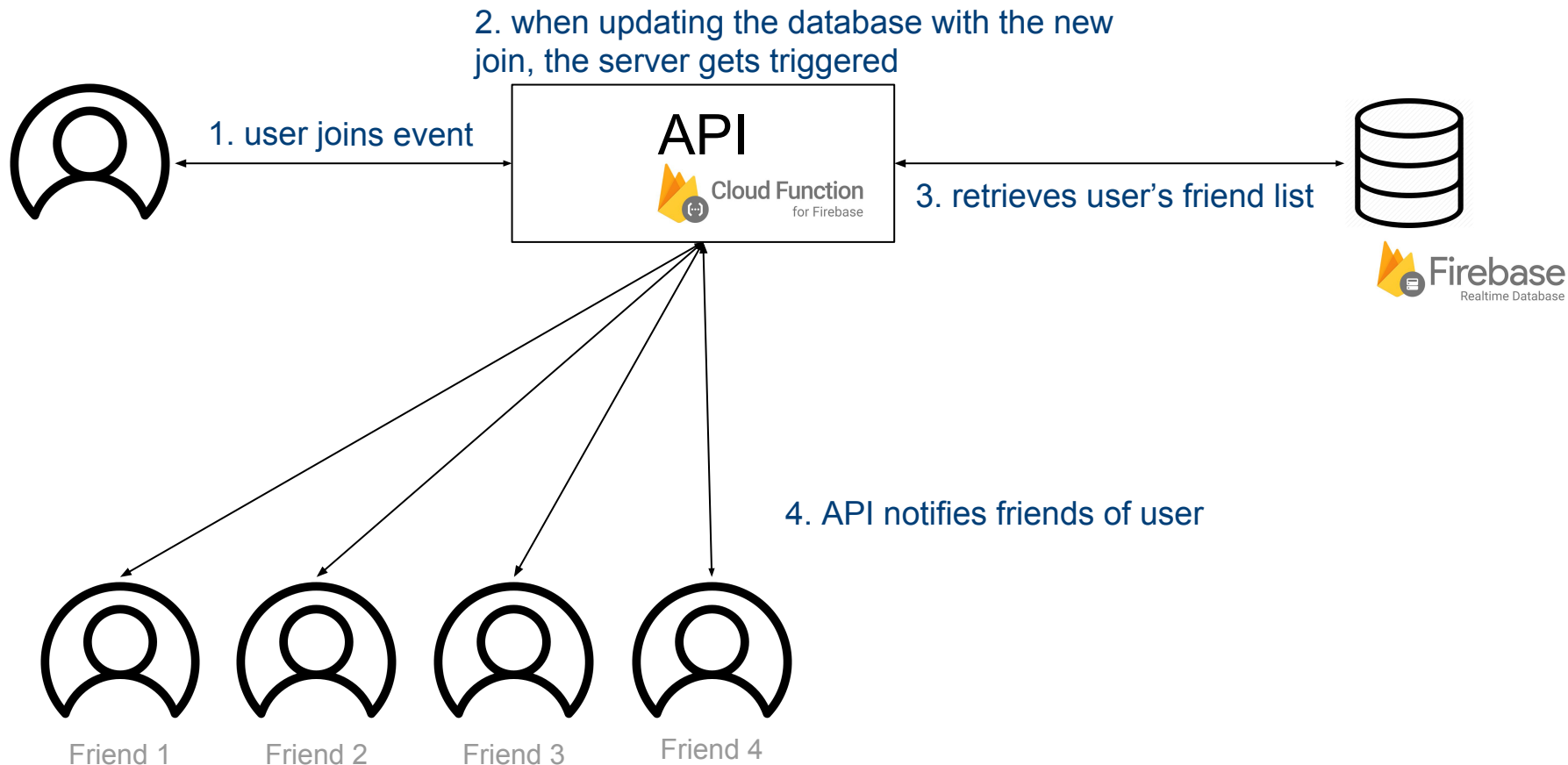
```
if(request.Type == userJoinedEvent){  
    Event event = request.Event;  
    List<User> usersInEvent = eventsDb.get(event.Id).Participants;  
    List<User> friendsInEvent = user.Friends.intersect(usersInEvent);  
    notifyAll(user, event, friendsInEvent);  
}
```

## Adaptation:



- **Send push-notification** when a friend creates or joins an event (e.g. Dinner Gathering @ChristmasFair).

# Usability - How notifications of friends joining events work





# Usability

- Difficulty searching for events of interest


## Physical Context:



- Filter by date and, furthermore, user preference if necessary;

Firestore RealTime Database

## Adaptation:

- Allow the user to search for events (input word and search the location, category and title of the events); 
- Allow the user to filter the results more: either by type, date or (limited to the map) zooming in showing a smaller range of events;

# Energy


- Minimize the energy consumption caused by GPS usage.

## Physical Context:

- Detect the GPS location on a chosen interval of time (10 min, since humans can walk at 12 min per km);

```
Thread.sleep(1000 * 60 * 10);  
update = locationManager.GPS_PROVIDER;  
if(!update.isEqual(deviceLocation))  
    activateGPSTracking();
```

## Adaptation:

- Enable GPS tracking only when the device is detected to be moving  (comparing the discrete GPS coordinates to the previous reading: if it moved farther than 500m, then GPS is turned on, disabling itself in a 30 seconds if the user doesn't move more than 500m in that time), or when using the map view of events.

# Connectivity

- Prefetching information (automatic user profile);

## Physical Context:


- Detecting devices connection type;

`android.net.ConnectivityManager` && `android.net.NetworkInfo`

- Detecting devices location (automatically associating a location to the application users profile information);

`LocationManager`.GPS\_PROVIDER

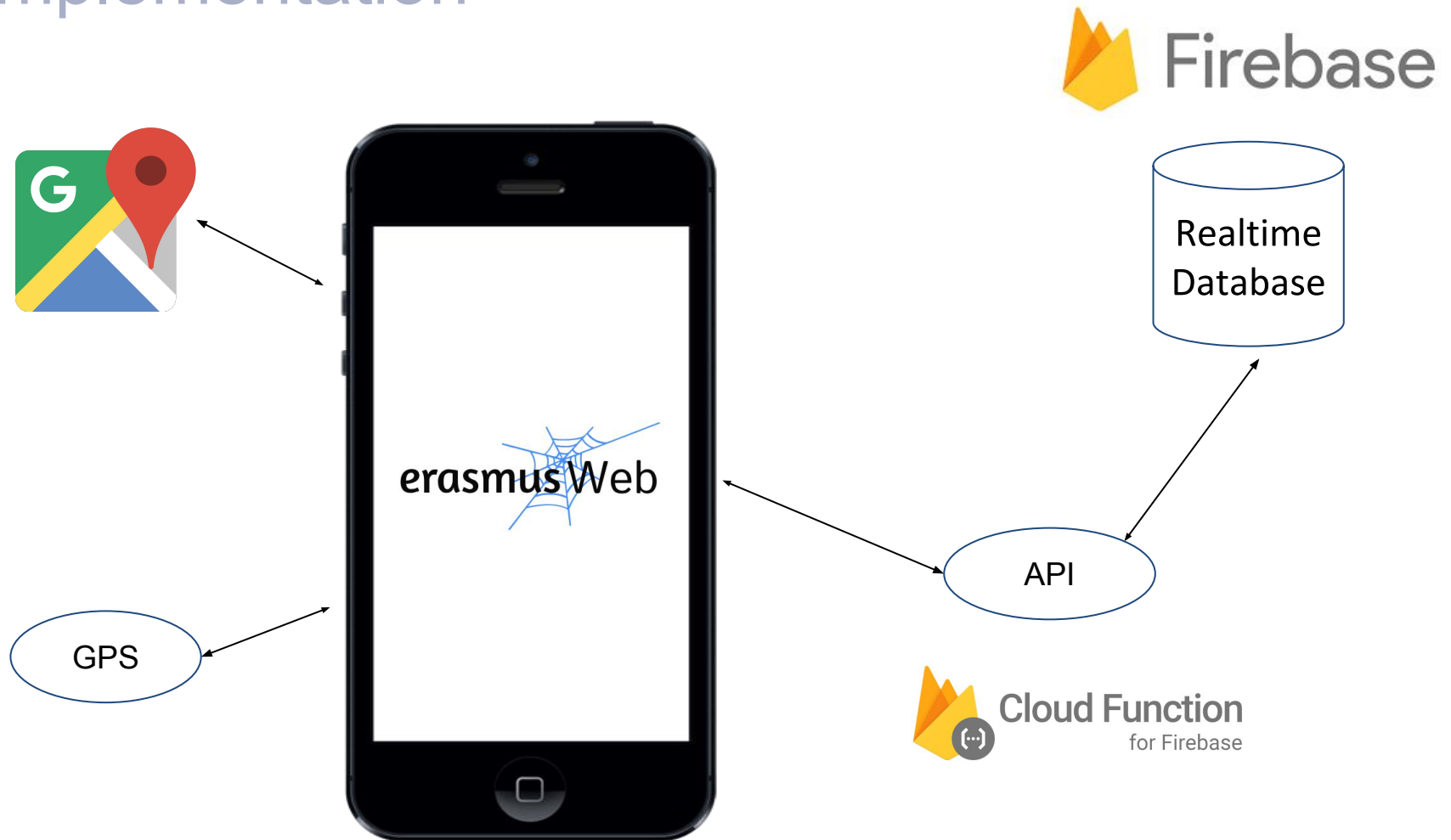
## Adaptation:

- **Prefetch the list of events near user's location**, prioritizing  joined events, events that are happening sooner and based on user preference; if wifi is available, images of such events (otherwise they will only be downloaded when the user requests to see them).

# Technologies

- Android OS
- Android Studio
- Firebase Realtime Database
- Firebase Cloud Functions
- Google Maps API
- GPS

# Implementation



# Work Plan

- **01.11.2018:** ~~First presentation~~
- ~~November~~
  - ~~Begin of implementation~~
  - ~~Back end development~~
- December
  - ~~First prototype~~
  - Front end development
- **14.12.2018:** Adaptation Concepts Presentation
- January
  - UI design
  - Bugfixing
  - Testing
- **01.02.2019:** Final Presentation

END