# Ubiquitous Computing

Ruben Teimas, m47753

15 July 2021

## 1 Introduction

For the final assignment of *Ubiquitous Computing* we were asked to do the create a mobile application using the technologies we learn in class.

This would go from structuring the application using *Android* elements to choose and implement the database as well as consuming data over *http* or apply *machine learning* models on the mobile device.

## 2 Application Description

The application I've chosen to implement is the ***GBL*** (*Glorified Book List*). It answers the need to keep track of our reading/whish-list by aggregating books on custom made lists.

## 3 Choosing the right components

The application's login will be made exclusively trough *Google Authentication*, mainly because they have everything implemented and there is no need for me to store each user's credentials.

In order to add a more fluid experience to the user the books will be served using a *book api* instead of *hardcoded books*. After knowing that *GoodReads api* was dead I searched for other *api*'s but I could not find any better thatn *Google Books API*. To consume the data *Retrofit* would be used.

To store the lists and the book info I thought about using the *Firebase*'s *Realtime Database*. I would use the user authentication unique identifier as the parent node and save the user's lists under it. There would be a node called *listsNames* which would have the lists created by the user, and another node called *lists* which would have the lists content.

The application will be compose by multiple activities and fragments. A *ViewModel* will be used alongside *livedata* to observe the response from the *API*.

This last choice is mainly due the fact that when, for instance, the phone is rotated the activity will restart, therefore, making another request to the *API*, which can be extremely bad in environments with a bad internet connection.

# 4 What would go wrong...

Unfortunately I was not able to implement the applications full list of features.

The user can create a list and get a view with the name of it's list, however, he cannot do anything with them...
He can also search for a book, get the responses and read the book details but he cannot add the book to the specified lists.

I ended up not using the *ViewModel* and *livedata* as I initially planned because I was having difficulty implementing the logic.

# 5 Conclusion

The development of the final assignment could have gone better. I knew what I wanted to do and what I had to do, but unfortunately was not able to implement it due to various motives.

These variance between what I had planned and what I ended up implementing eventually added extra complexity to the application, sometimes making it harder to work on it.

Nonetheless, I learned a lot from this assignment, I learned how to structure, authenticate read and write from *Firebase*. I learned how to consume data over *http* requests on a mobile device and how to use various *Android*'s components (expect the livedata).