Creator and Developer
Pedro Marques

06 2022

Room Rental and Roommate Finder System

Keywords

Information Retrieval
Large Scale Data Mining
Recommendation Systems
Mobile Development
REST API

Abstract

This project intends on improving room rentals, with the intention of facilitating this process and promoting the interaction between flatmates. The main goal is that a specific person is able to find habitation near a specific location and following specific characteristics. Furthermore, flatmates should be able to contact each other through this application.

Acknowledgements

Table of Contents

List of Figures

Acronyms

AWP	*Administrative Web Platform*
ARP	*Algorithmic REST API*
PoI	*Points of Interest*

1 Introduction

1.1 Context

This project was developed with no utter intention, solely for the will to work with technology of its creator.

1.2 Motivation

The main objective is to simplify the process of room rental and communication between flatmates and habitation owner, in order to promote house spirit. Furthermore, the main idea of this project came in a University ideology, considering the student's struggles with room rental.

1.3 Document Structure

Besides the introduction, this report has 5 additional chapters. In chapter 2, the current state of the art, or lack thereof, is analyzed. Chapter 3 focuses on the process for defining the project's requirements as well as the requirements themselves, followed by the system architecture. In the next chapter, the implementation of both MUP-RR and R2UA are discussed in detail. Then, chapter 5 presents the results of the final implementation of this project. At last, chapter 6 goes over the final considerations and conclusions derived from this work and the future tasks needed to develop this project further.

2 State of the art

In the first stages of the project we decided to research about current technologies related to our problem. Our project is specific and intended to be used by the university in the future. As such, there are some applications available online for this purpose, however many are not available in Portugal, do not provide much reliability and/or are purely based on the owner's description of the property and not previous habitants.

2.1 Roommates by Roomster

Roommates is a mobile app and web platform whic allows owners to post their listings and users to find rentals. It uses other social networks to find out if other habitants are to your liking. Many filters are available in the website to facilitate search.

Counts with many reports of people complaingn about bad customer service and false advertisers which lead to scams

2.2 Trulia

Trulia is an American online real estate marketplace which is a subsidiary of Zillow. It facilitates buyers and renters to find homes and neighborhoods across the United States through recommendations, local insights, and map overlays that offer details on a commute, reported crime, schools, churches and nearby businesses.

3 Conceptual modelling

This section aims to present the functional and non-functional requirements of KIPA

3.1 System Requirements

To truly accomplish our goals with this project, we needed to establish solid requirements for all of its components: the REST API, the admin web platform, the client web platform, the habitant mobile App and the owner mobile App.

3.1.1 Administrative Web Platform

The Administrative Web Platform should be capable of:

- Visualize traffic per day
- Visualize communication initialized per day
- Access Logs - Can Show logs per client
- Visualize and answer feedback
- Visualize Data Distribution

3.1.2 Algorithmic REST API Functional Requirements

The ARP using Django should primarily be capable of:

- Find listings according to filters, including room size, price, location and proximity to PoI;
- Recommend habitations based on previous research;
- Retrieve Favorited Listings;
- CRUD methods for listings;
- CRUD methods for reviews.
- Send Email Notifications when.
    1. Users want to connect
    2. New review added to listing
    3. Registering/Changing Password

3.1.3 KIPA - Roomer Functional Requirements

The Roomer Mobile App and WebPlatform should primarily be capable of:

- Messaging Area
    1. With homeowners
    2. With other house members
- Search for listings
    1. By Location
    2. By Room Characteristics
    3. By Proximity To Location
- Sort listings
    1. By Proximity to Location
    2. By Date Added
    3. By Price
- Display listings in 2 ways
    1. List Display
    2. Map Display
- Display Personal Page;
    1. Profile Picture
    2. Details like: Name, Age, About Me
    3. Personal Details Like Email and Phone Number should be public or private by choice
    4. Share Social Media
- Display Listing Details
    1. Picture Gallery
    2. Description, House and Room Characteristics
    3. Personal Details Like Email and Phone Number Should Be Private
- Display Listing Reviews with text and/or rating 1/5;

3.1.4 KIPA - For Listers

The ARP using Django should primarily be capable of:

- CRUD methods for a House
    1. Add Pictures
    2. Add Details and House Characteristics
    3. Add Location

4. Add Price
5. CRUD methods for a Room

- Show roomer's proposals
- Show history of proposals
- Retrieve Favorited Listings;
- Accept/Message Proposals
    1. Add Pictures
    2. Add Details and House Characteristics
    3. Add Location
    4. CRUD methods for a Room
- Respond to reviews.
- Display Personal Page;
    1. Profile Picture
    2. Details like: Name, Age, About Me
    3. Personal Details Like Email and Phone Number should be public or private by choice
    4. Other Available Listings
    5. Share Social Media

3.1.5 ARP Non-functional Requirements

The Algorithmic REST API should respect the following non-functional requirements:

- Security: The system will use minimal to none private data. None the less, it is important that security is tight, usage of highly secure cryptographic algorithms is requested. Furthermore, a user can avoid giving us this data by logging in using Google. The system should follow the principles of Privacy by Design and Security by Design, in accordance with the demands of *RGPD*.
- Reliability - In case of failure, there should be mechanisms in play which auto-restart the application, increasing availability.
- Efficiency: The platform should be able to process a high traffic of requests and responses quickly, probably using parallelization techniques.

3.1.6 KIPA Apps Non-functional Requirements

The web platform and mobile apps should respect the following non-functional requirements:

- Usability: Should be intuitive and easy to use.
- Compatibility: Should work with most web browsers, including a responsive designs and the mobile apps should work on any operating system, namely Android OS and iOS.
- Efficiency: Given the vast amount of potential users, performance is of high importance and the platform should be able to process a high traffic of requests and responses quickly.

3.2 System Architecture

3.2.1 Domain Model

3.2.2 Technological Model

The technological model allows us to have an overview of the technologies used by the system. In a top-down view, in the backend we have the Spring Boot ARP that communicates with both the AWP and KIPA. The ARP connects directly to the Firebase Database to store data. The mobile apps are developed with Flutter where as the web platforms are developed with .NET Core.

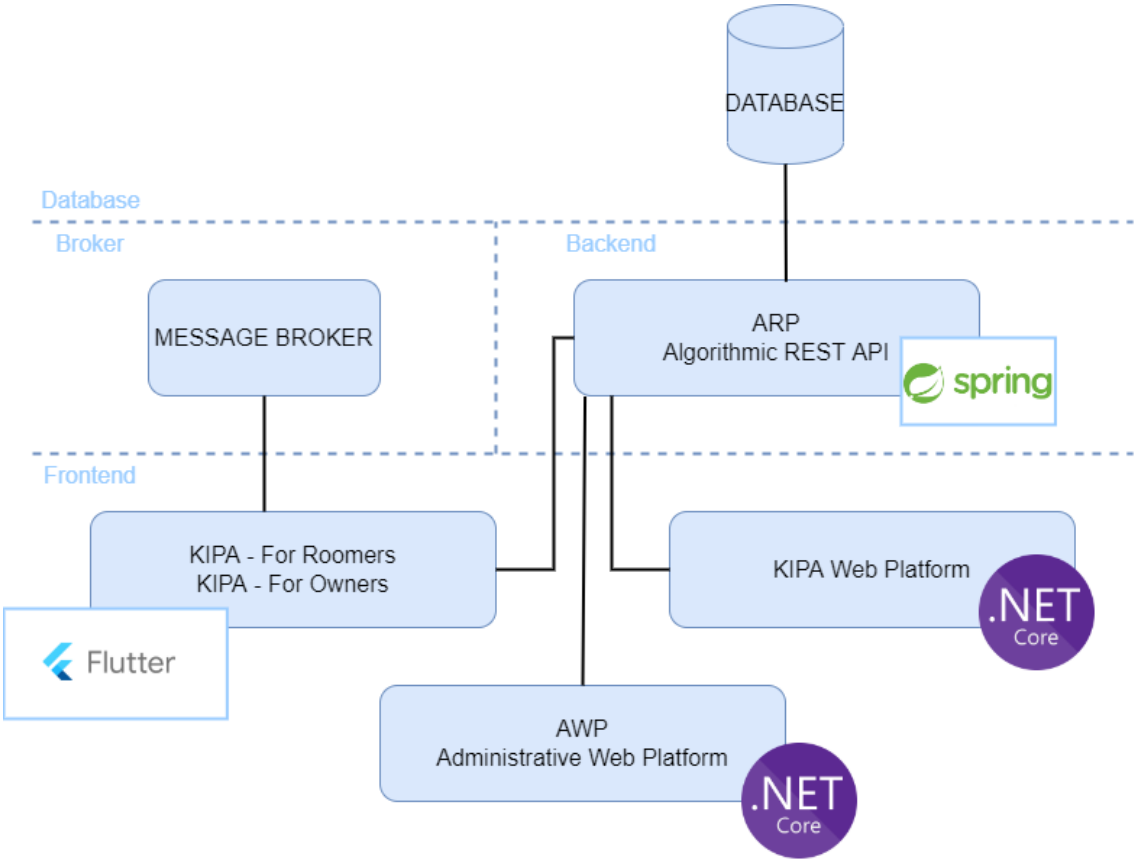

Figure 1: Technological Model

4 Implementation

4.1 ARP - Algorithmic REST API

4.2 AWP - Administrative Web Platform

4.3 KIPA - For Roomers

4.4 KIPA - For Listers

5 Discussion of Results

6 Conclusions

6.1 Future Work

References

[1] BEST. *BULLET SOLUTIONS*. https://bullet-frontend.dev.ua.pt/.

[2] BESTLegacy. *BESTLegacy Api*. https://bullet-api.dev.ua.pt/swagger-documentation/index.html.

[3] *Flutter Awesome*. https://flutterawesome.com.

[4] Microsoft. *.NET Tutorials*. https://docs.microsoft.com/en-us/dotnet/core/tutorials/.