



CiteCy Functional Specification

04/12/2020

Bozhidar Sabahov

17703831

Salvijus Mazeikis

17302181

1. Introduction	2
1.1 - Overview	2
1.2 - Business Context	2
1.3 - Glossary	3
2. General Description	3
2.1 - System Functions	3
2.2 - User Characteristics and Objectives	4
2.3 - Operational Scenarios	4
2.4 - Constraints	5
3. Functional Requirements	6
4. System Architecture	9
5. High-Level Design	10
5.1 High-Level Design Diagram	10
5.2 Data Flow Diagrams	11
6. Preliminary Schedule	14
7. Appendices	15

1. Introduction

1.1 - Overview

Nowadays people are surrounded by hundreds of outdoor/indoor activities and places they can visit in a city. Sometimes the number of choices could be overwhelming because each person wants to try everything but there is not enough time.

What if someone else, who knew them well, could decide for them what would be the best and create a plan of activities?

That is what CiteCy was created for. It will gather activities to explore from various resources available online such as Tripadvisor, Yelp etc. These datasets will also help with determining what areas of interests are available to recommend to the user.

The user's preferences will be defined by initially being provided areas of interest by the user and then preferences will be further explored when days will be planned and they give feedback which will be used to further train the recommender system.

The activities will be shown with timestamps, the average time spent there, directions from one activity to another and an option to remove and refresh the activity.

1.2 - Business Context

The most applicable business for this product is to contain advertisements to generate revenue. There would also be an opportunity for promoted restaurants/events which could generate revenue and give user's better prices. Another option to generate revenue is to make certain extra functionalities a paid service. For example, travel instructions.

1.3 - Glossary

Microservices - Arranges multiple smaller loosely coupled services as opposed to a single large monolithic service.

Content recommender - Machine learning solution for building recommender systems.

API - Application programming interface, which allows the software to 'talk' to each other

Firebase - Google provided authentication, database, hosting and storage service.

Ionic - Mobile app framework for progressive web application development

2. General Description

2.1 - System Functions

Make an hourly plan with activities

The main goal of the product is to create a plan of activities that will be suitable for the user. These activities will fit into the user's budget. The plan will have start and end time based on the user's available time. Each activity will have suggested duration but the customer will not be obligated to follow it. In scenarios, when the user overstays at activity more than the suggested time, the plan will be recalculated to ensure accurate time openings and navigation. In addition, travel instructions, for the user's choice of transport, will be provided.

Selecting suitable activities for the user

The activities in the plan will be picked by a recommendation system. Each user will input their favourite types of activities. Then the recommendation algorithm will be able to select activities based on their preferences.

Making updates to the plan

Before or after the user sets out to enjoy their activities, some aspects of the plan can be updated. For example, they can remove an activity or add a new one to the plan. The suggested duration of stay for each activity could be reduced/increased, too.

Let the user rate their activities

In the end, the users will be able to select the activities they liked or did not. Thus, the following recommendations will be better.

2.2 - User Characteristics and Objectives

The users of this system will know how to interact with a website or mobile application.

The user base will be local explorers, they are looking for activities and events to partake in their area.

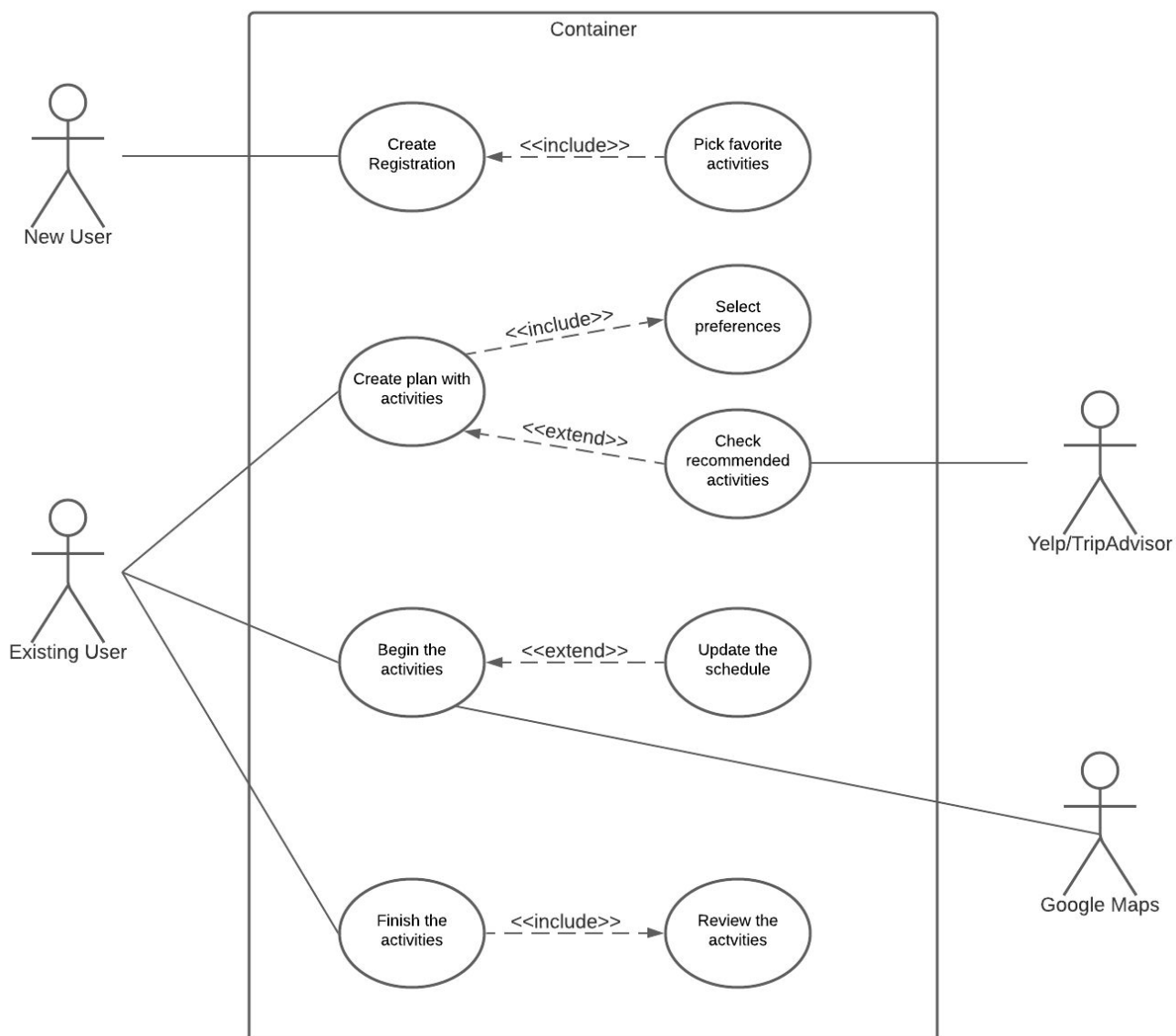
The user should expect a simple and intuitive operation, nothing much should be requested and the simple steps can be broken down into registering, selecting from a varied selection of interests, selecting preferences and receiving a curated day.

Additional features would aim to only enhance the simple user experience, therefore care would be taken into consideration of not changing the simple process and instead be an alternative for others.

Other such features could include, sharing of planned days, reviews and comments of places, user profiles and gamification.

2.3 - Operational Scenarios

The operational scenarios shown below include the applicable fundamental operational scenarios.



2.4 - Constraints

The following are a few notable constraints that have been identified, they will be minimised as much as possible but nevertheless, they are as follows:

- **Location:** It will be difficult to provide an amazing experience to all users as some data may just not be available in their region. If this were being built to a tech giant level then data would be gathered to ensure each region has every available option.
- **Data:** There may be constraints by the fact that there are not many datasets available to be used and recommendations will be constrained to only show online data.
- **Time:** Although there is nothing concrete being constrained in this section, this project will not be able to have every additional feature as that would require sacrificing or rushing the main features. Therefore time will constrain additional feature implementation.
- **API calls:** Since this will be a university project there is going to be no use of paid services. Therefore there will be a maximum amount of times the targeted APIs can be queried. There will therefore be a need to save data we've retrieved and try to optimise its use.

These constraints will be later re-assessed to see if they do actually limit the scope of the project or if they were able to be overcome.

3. Functional Requirements

Requirement ID	1
Description	An algorithm which creates an hourly plan with activities by taking into consideration the user's budget, location, type of transportation and time availability. The plan consists of travel instructions (directions and the time it takes to reach the destination), one or more activities, including a suggested duration of time to spend on an activity.
Criticality	Since it stands as the core of the product, the functionality is vital to the system. Without it, the product does not achieve its sole function.
Dependencies	The algorithm requires input from the user and the recommendation system. The latter will provide a list of types of activities which are suitable for the person.
Technical Issues	The main issue will be to make the algorithm fast and accurate. The plan should have diverse activities and they should not repeat in following executions. The algorithm should not take a long time to create a plan. Another issue will be to find/estimate the cost of each activity.

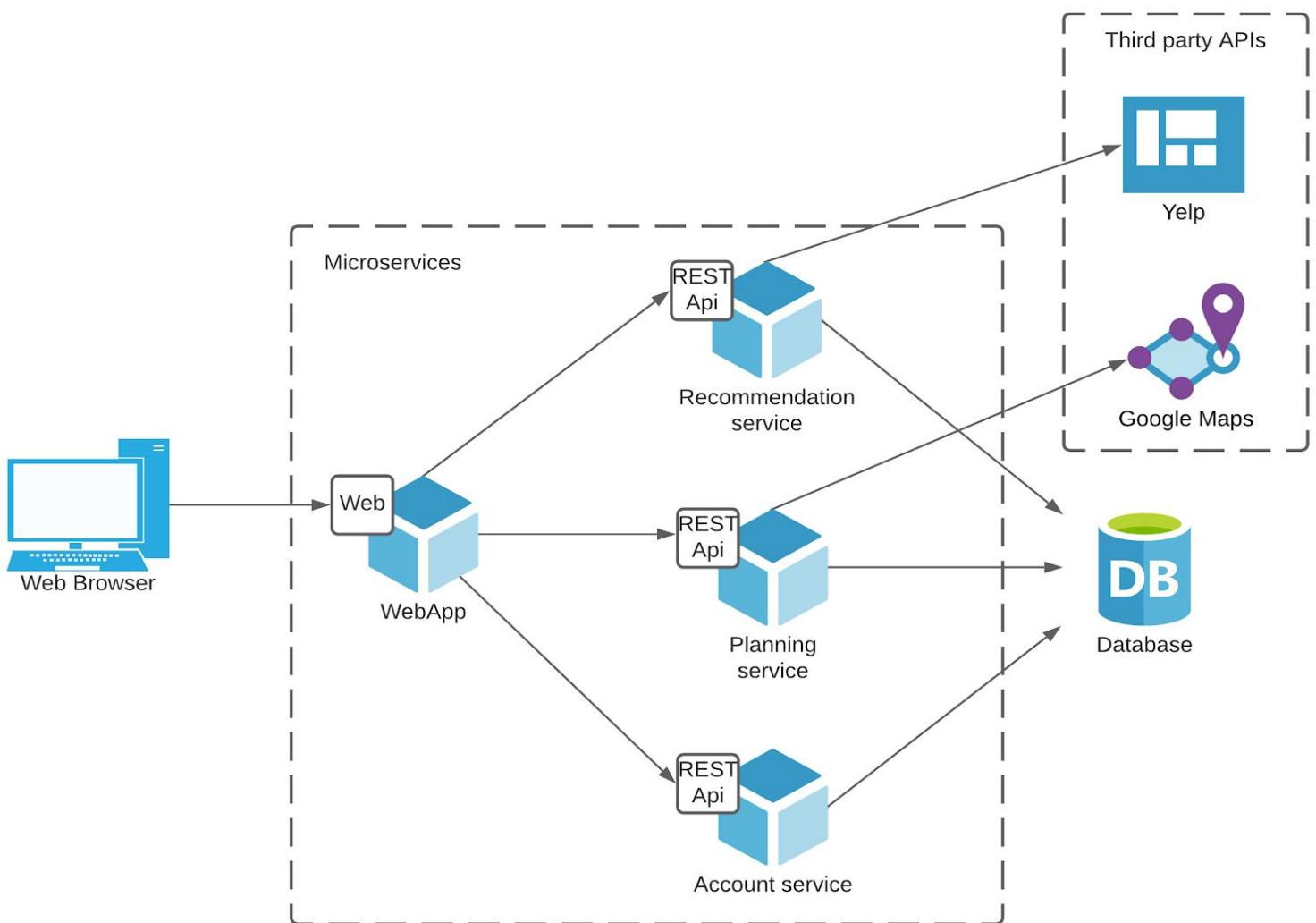
Requirement ID	2
Description	Recommendation system for activities. It is a content-based recommendation algorithm which takes the data given by the user and it makes a user profile. Then, it is used to suggest activities to the user. As the user rates the activities, the algorithm improves.
Criticality	It is a vital function to the product because the main goal is to provide the user with custom-tailored activities.
Dependencies	It requires feedback from the user to be able to improve the recommendations.
Technical Issues	The main issue is that a recommendation algorithm needs a large amount of data to function optimally.

Requirement ID	3
Description	Allow user edits to the plan at any time. They can remove activities or add a new one from the recommended list. In addition, the suggested time to spend on the activities can be edited, too.
Criticality	It is not as important as the previous two requirements, so it will not be given top priority until the core functions are ready.
Dependencies	-
Technical Issues	Reloading the activities will have to be very efficient and it will be a challenging UI problem to solve as to ensure it appears simple.

Requirement ID	4
Description	Evaluation of their recommendations will be done to improve the user recommendations, data from their experience is to be collected. They will be able to rate the activities after they have attended them.
Criticality	It is important to implement this feature because otherwise, the recommendations will not improve.
Dependencies	User has to provide feedback for the function to complete its purpose.
Technical Issues	Methods of requesting feedback will have to be made appealing and intuitive as possible from the UI perspective.

4. System Architecture

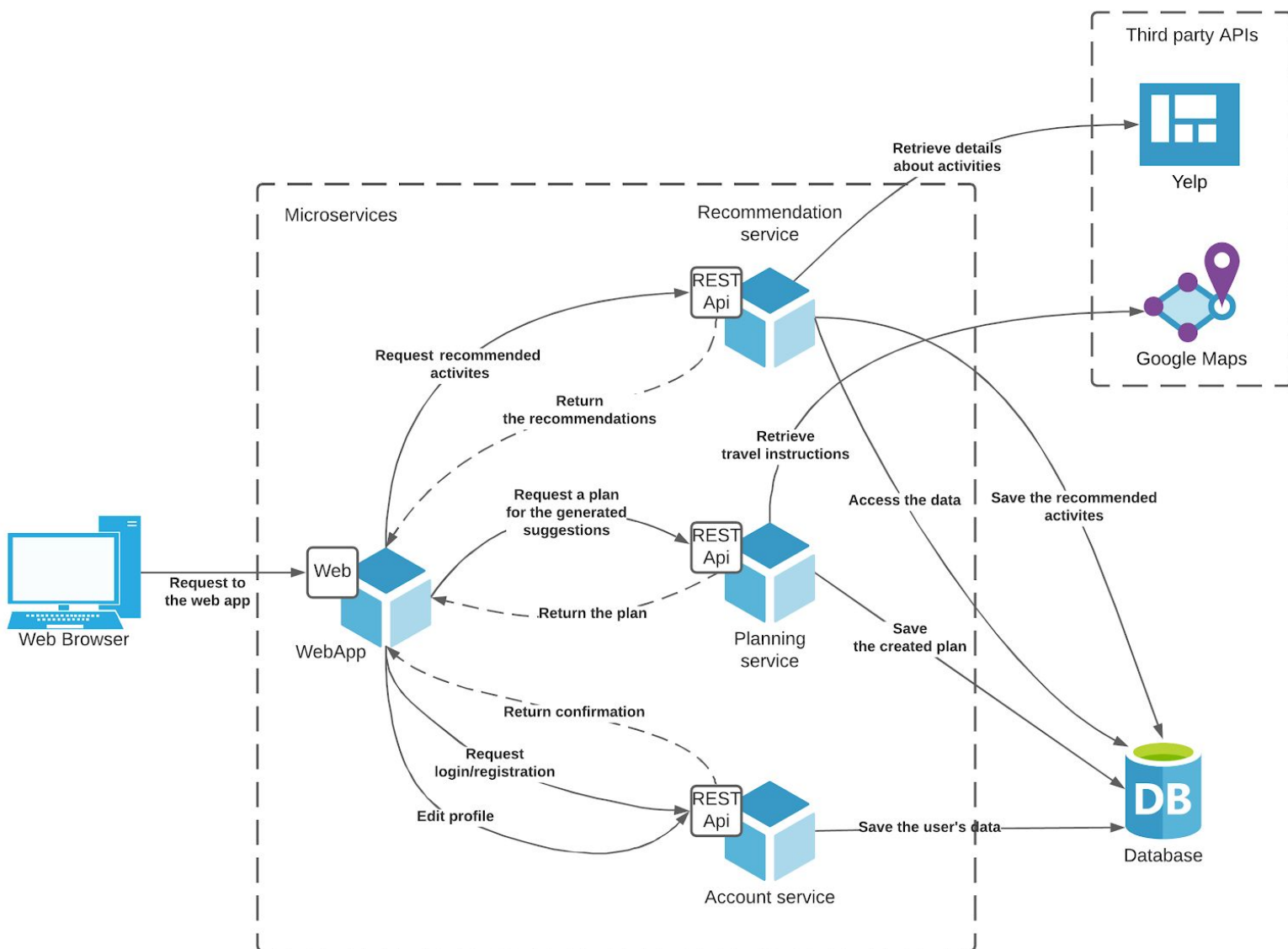
The system is divided into 4 services - WebApp, Recommendation service, Planning service and Account service. The WebApp redirects the requests to the corresponding service. Each service is independent and has a database scheme which is private to that service. This allows us to deploy a service independently and test it safer and faster. Moreover, in the long term plan, it will be faster and easier to upgrade the services.



5. High-Level Design

5.1 High-Level Design Diagram

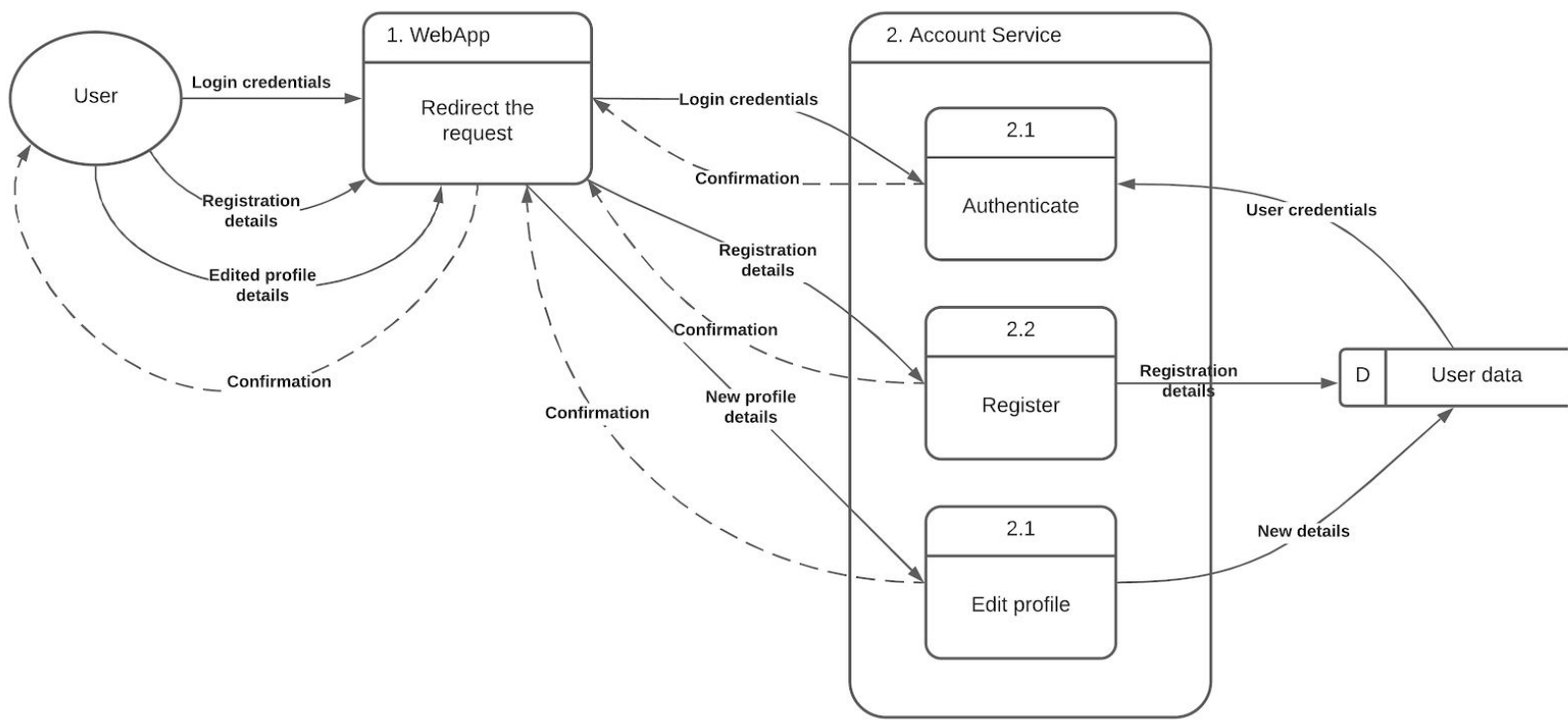
The diagram displays the basic processes occurring in the system with a high-level overview of them. It displays the basic functionality processes of the system.



5.2 Data Flow Diagrams

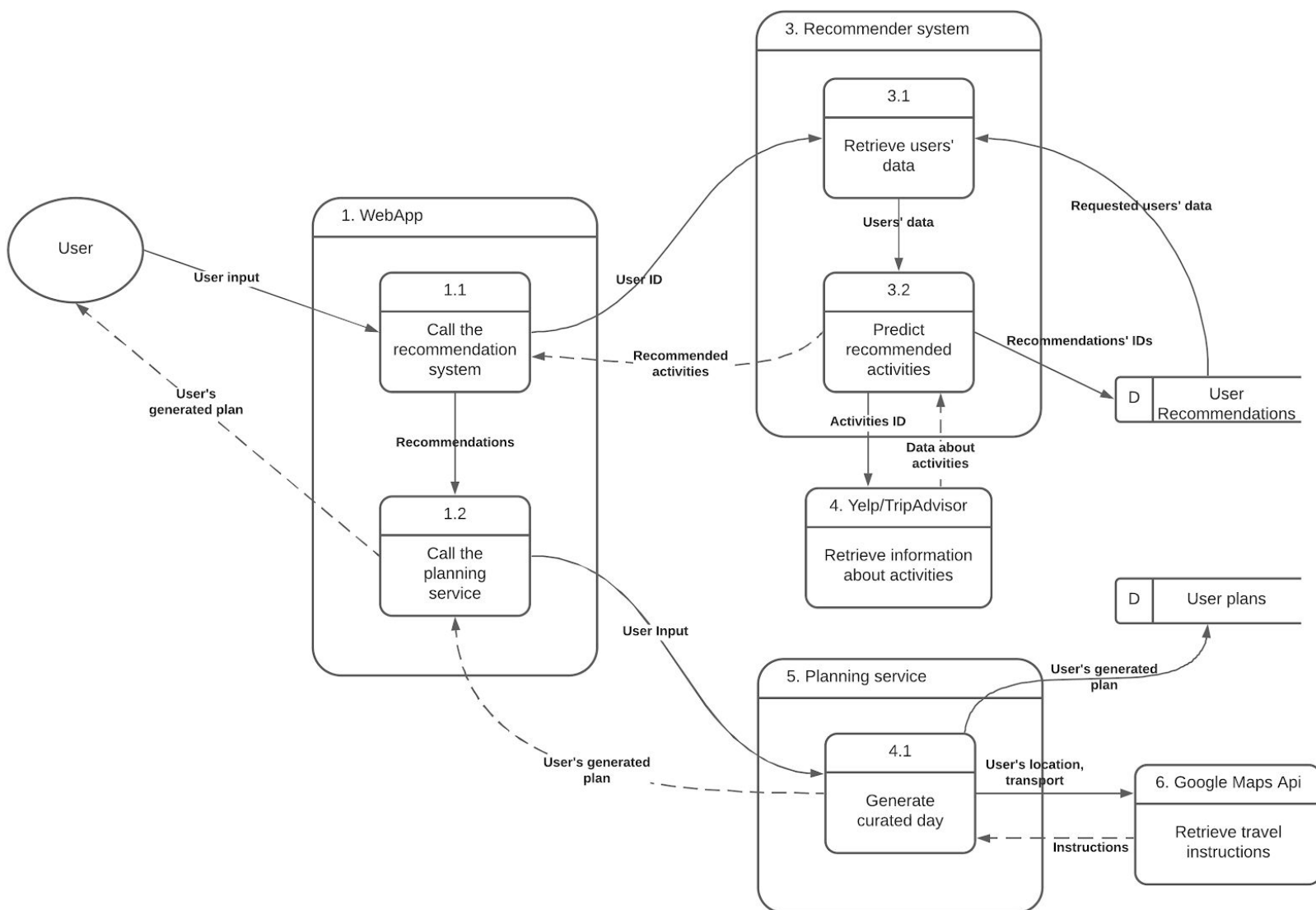
5.2.1 User accessing the system

Displays the data flow between the application's web app and account services, authenticating and granting access to the user.



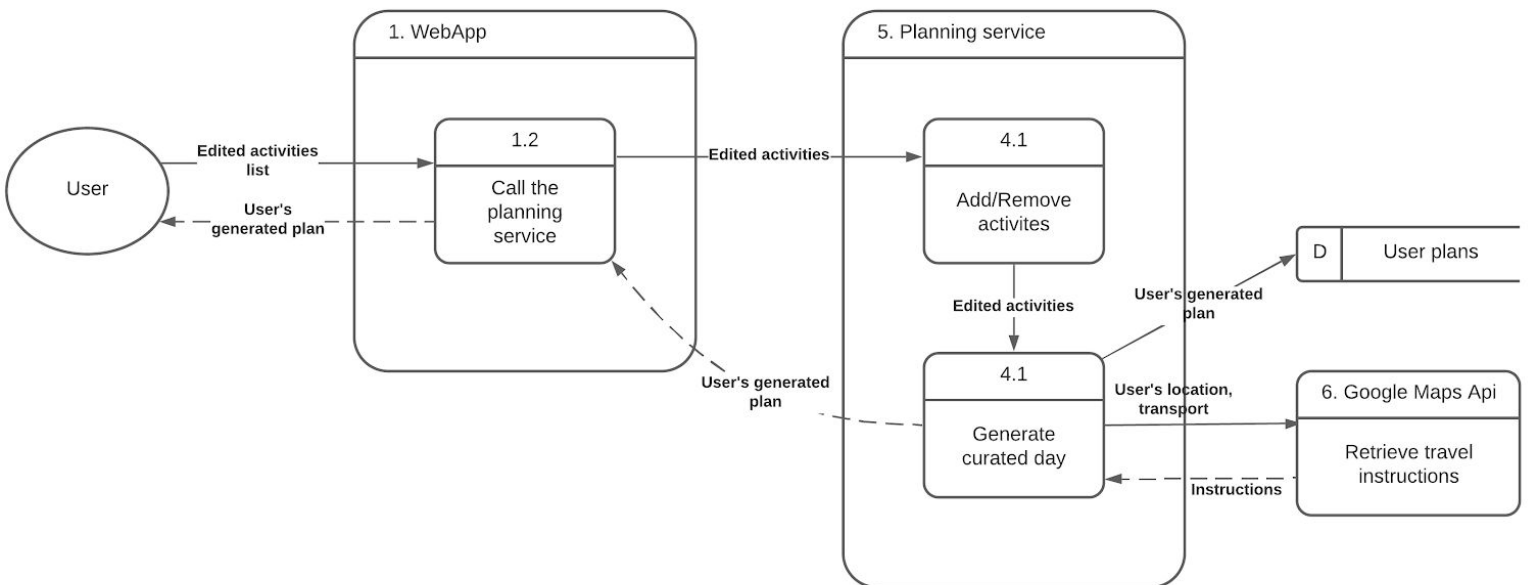
5.2.2 User requesting for a planned day generation

Displays the main functionality of the system, this data will be managed by the application and therefore great care will be taken to ensure that the data is handled appropriately. As well as that, minimal data will be sent to ensure the efficiency of the system.



5.2.3 User editing day plan by adding or removing activities

The following diagram displays a smaller function of the system, but there is still data of the user being sent and security needs to be of the highest priority. The following logic displays a simple real-time switch of activities initiated by the user.



6. Preliminary Schedule

The project will be completed in sprints and iterations of the application will be made to ensure that there are layers of functionality being built and everything is not being joined together at the end.

[illegible]

7. Appendices

- <https://ionicframework.com/>
- <https://firebase.google.com/>
- <https://www.yelp.com/developers>
- <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>