Smart Personal Organizer

Johann Haag, Tarik Hosic und Simon Blaha

October 20, 2018, Leonding

Contents

1 Introduction

There are a lot of personal organizer providers. These organizers are very similar in their functionality, but they have one thing in common, their functionalities are very limited. The personal organizers are kept very simple. It is nearly not possible to set complex conditions, which are checked automatically.
Some organizers have the functionality to share an appointment but it is not possible to let the personal organizer find automatically an good appointment for you and your friends.
That's why it's often hard and time-consuming to find a suitable date for yourself and your friends, where everyone has time.
Often appointments have conditions to be able to take place at all.
For example, a nice weather is needed to have a barbecue with his friends.
There are also often cases where appointments are interdependent, so it may be that an appointment is not taking place because an appointment with which it was related could not take place.

2 General Conditions and Constraints

For our project we are going to need 2 APIs to be able to provide complex conditions. The first API is needed to get a the weather forecast. The API openweathermap is providing this functionality but for free use it has some restrictions, which can be found here.

The second API is needed to calculate the time of the way, which is between 2 dates. Google is providing an API for that. One of the major factors of our project is the accessability with the internet. The personal organizer will be available when the user's offline, but for example some functions like the group function want be available without internet connection. Due to the limited amount of time we will have to work on our project we most likely will only be focusing on a mobile version for Android.

3 Project Objectives and System Concepts

The personal organizers that exist today are just boring and do not offer people much. Our goal is to create a new personal organizer that will attract the attention of many people with additional features. The new personal organizer should include features such as weather and route, friend function, grouping function, profile and chat, rights system, dependency between appointments, lists and comments, showing birthdays of friends, editing appointments, notifying the group, matching the appointments over the whole group included. Nevertheless, we are aware that for the first time we have to realize the basic functions that an appointment book contains. Appointments have basic functions such as the weekly, monthly and yearly view, add appointments, schedule conflict check, buffer times delete appointment, push notifications, backup and synchronization of appointments (cloud) and local storage. After we have finished the basic functions of a personal organizer, we want to incorporate our additional features, which we have already mentioned, and thus have the new personal organizer realized.

4 Opportunities and Risks

During the development of our idea we will have to take some risks. For example, one risk would be that the APIs used are unavailable or unrecognized. In addition, they may be unreliable. Another risk is the usability for the user. Due to the multitude of functions, there is a potential risk that the user-friendliness of the project will be lost. With our idea we want to get involved in clubs, groups of friends and private persons. fix. In addition, we want to reduce communication problems.

5 Planning

The project will start on the October 30, 2018.

5.1 Members

- Johann Haag
- Tarik Hosic
- Simon Blaha (Project leader)

5.2 Milestones of the Project

- Prototype Client
- Prototype Server
- Client Core
- Server Core
- Advanced Client
- Advanced Server
- Client GUI
- Additional Client Functions
- Additional Server Functions
- End of Project

Prototype Client

Description:

A rudimentary prototype will be created. It uses the frameworks and APIs that will be used later in the application.

Goal:

The purpose of the prototype is to determine how the APIs and frameworks used work. As a result, the further course of action in the project can be better defined.

Date: 10.12.2018

Prototype Server

Description:

A rudimentary, little server is created with basic functionality.

Goal:

The purpose of the server is to determine which technology is suitable for programming the server. Furthermore, there is an overview of how to implement the server.

Date: 15.12.2018

Client Core

Description:

The core of the client application has the basic functionality.

Goal:

The core should provide a good foundation upon which the rest of the application can build.

Date: 30.1.2019

Server Core

Description:

The core of the server should have basic functionality.

Goal:

It should provide a good foundation upon which the rest of the server can build.

Date: 15.2.2019

Advanced Client

Description:
The client core with additional functions.
Date: 15.3.2019

Advanced Server

Description:
The server with additional functions.
Date: 30.3.2019

Client GUI

Description:
A GUI for the Client.
Date: 20.4.2019

Additional Client Functions

Description:
Additional functions for the client.
Date: 20.5.2019

Additional Server Functions

Description:
Additional functions for the server.
Date: 19.6.2019

End of Project

Description:
It is the end of the project.
Date: 20.6.2019