

**Allround Manager**

<b>Project name</b>	<b>Allround Manager</b>	
<b>Project leader</b>	Christian Bachl	
<b>Processing status</b>	<b>X</b>	in process
		submitted
		completed
<b>Project Member</b>	Haris Jusic, David Cvetkovic	

## **1) INTRODUCTION**

The “Allround Manager” basically helps people to manage there events a lot easier and gives you the best possible overview of the event and members of each event. Just by a click, you can also cancel those events and every registered member gets a pop-up message, which shows him the reason why the event is canceled.

## 2) INITIAL SITUATION

Organizing an event involves a lot of organizational steps for the event leader. A non-exhaustive list of tasks could be

- When the event involves some trip a generally accepted destination has to be aligned
- Send invitations to all participants
- Keeping track of registrations or de-registration of participants
- Gathering information about the participants like home addresses, passport numbers, etc.
- Provisioning of information about the event for the participants, like the aim of the event, agenda, other participants, etc.
- bills of outstanding services like a prepayment for an event.

Currently, a wide variety of tools has to be used to accomplish the above-mentioned tasks.

- Most of the communication is done via WhatsApp or similar social media apps.
- Lists of participants, their status, etc. are organized via spreadsheets
- Outstanding bills must be paid with a payment slip or in cash.

The combination of different tools and a very decentralized way of organization makes it hard for the event organizer to stay on top of the things. More often than not these events struggle with late minute change or an announcement. This could be:

- The meeting point of the event/journey.
- A date where the event takes place.

Doing these things may cause some troubles for the event organizer, like:

- when you have a lot of participants you may forget somebody
- When billing, for example, participants want to pay only from the location where they have entered.
- the leader must be every time available like: user wants to de-register or want to know many participants take part in the event.
- getting spammed by registrations in WhatsApp.

### **3) GENERAL CONDITIONS AND CONSTRAINTS**

#### Framework conditions

Unfortunately, there is no money available for our project. Which means we're on our own.

Currently we have good knowledge of C#, C, Java and JavaScript. In the near future we will work with Android to educate ourselves further and it's easier to divide up the work.

We are not in cooperation with any other companies. This is probably also better so that we can concentrate on things more precisely. If it comes to complications, then it only needs to be discussed among the three of us.

A large part of the project will be completed by 17.06.2019. Improvements and enhancements will follow over time. Program is never finished, because there is always something to improve.

#### Technical Constraints

The current hardware infrastructure is as follows:

- Server ...
- Network connection of the server to the internet

This let us only set up a prototypical environment with a max. 100 groups and per Group only 10 events at the same time.

The software is not available on IOS because we do not have the time to convert the android app to an IOS app, so in the beginning, it is only for the Android user.

The Android studio Environment is new for us and that's why the implementation will be slow in the beginning.

#### **4) PROJECT OBJECTIVES AND SYSTEM CONCEPTS**

There are three kinds of user who use our app.

The first kind of user would be the participant, the user generally uses the application for joining events. The benefits for this user would be that every organization or event stuff is clearly done in one place and it is easy callable.

The second kind of user would be a normal user who wants to organize an event but isn't a leader of a typical group like: (Landjugend, Feuerwehr, Jugendschau, Schule).

The third kind of user would be a leader of a larger group like the above-mentioned examples.

The difference between the second and the third kind of user is that the third might handle several events at the same time as the second just manage one event, therefore it is recommended for the third to first launch a group and then create an event. To make a group it makes the organization stuff for the manager clearer and more structured. For just one event the effort to launch a group would be pointless.

## **System Concepts**

### **Register/Login:**

If a person wants to use our software, he first has to register. This is important for our software because otherwise, it doesn't know who you are. In the case when a user wants to leave an event it's important that the user is clearly defined.

### **Set/Launch up an Event:**

If a user launches an event he can use or make his own event-template or he uses some from the template-library who are predefined. To launch an event, it is also necessary to set up some properties/setting like destination, Date, Max-participants etc... There should also be a function where a leader can create an information-link.

#### **Event-Template**

Each event is different, so the user can use the template function to create it according to your imagination. This function can be imagined having a toolbox and a field. The Toolbox contains several options such as a label, a toolbox, etc. To create a text field, the user just searches for text field in the toolbox and drag it into the default field.

### **Set/Launch a Group:**

To set up a Group it is necessary to be a leader of a bigger group. To create a Group some defines must be done. How big should be the group (MaxSizeGroup), permission of participants, available surveys etc...

### **Join an Event**

Depends on the event template. If a user is into a group, he doesn't wait for the confirmation file. If a user joins an event it is important that he see directly all interesting

stuff from the event like, where is the target point, when is the event, destination, what equipment is needed, the list of participants etc....

### **Accept join-requests**

If the leader doesn't have a group, the participants have to wait for a confirmation message. After the user creates an event there should be a function where he sees the open join-requests.

### **Sidebar**

The Sidebar should include fields where a user can see his joined events, joined Groups, his own templates or his notes.



## 5) Opportunities and Risks

- **Opportunities:**

We have two professors who are often involved with organising things and have confronted them with our project. They were very positive and would definitely give our app a chance. In addition, we asked several group organizers how they organize things. The several of them proceed according to the stressful scheme. In any case, our app would simplify their work a lot. With this survey, we can say that 20% of Austrians are planning an event at least once a year. This gives us the opportunity to develop a software who is actually needed at the market. Because of that, our App has a big potential and we may have a chance to make a profit. For example, the first group with a maximum number of 20 users is free. If the customer wants to add more user he pays per user for example 4€.

**Risks:**

When data get lost there is a huge problem. Maybe all the events of a company are gone. In this case, we might get problems with the data-victim. This makes a bad impression for the community and we could lose many futures or existing customers.

## 6) Planning

Major milestones:

- That the Login system works with our database.
- To create an event with a template.
- To join an event or a group
- To list all participants in the group
- That the toolbox and the field work together
- That the user can create his own template with several options
- To create a survey into an event.
- Maybe a massager into a group.
- To get online

Project end: not known yet, hopefully, this school-year

First Prototype for the Interface: not known yet