

# robo4you-Manager

Quentin Reisinger\*, Georg Rohrhofer, Lukas Haindl, Marcel Schock  
Technical Secondary College  
Department of Computer Science  
2700 Wiener Neustadt, Austria,

\*Corresponding author Email: reisinger.quentin@student.htlwrn.ac.at

**Abstract— Open Labs are more popular than ever before. They have to provide a good infrastructure to facilitate the working process of the user. To achieve that we created a concept of a software which is able to manage the different devices an Open Lab is allocating. The purpose of this paper is to constitute the concept and functionality of the software.**

**The software is fully functional for different kind of robots and machines and also contributes a lot to Botball and ECER in general as described in this paper.**

## I. INTRODUCTION

In order to optimize and secure the access of robots we designed a solution especially for that. This System was developed out of the need for a good and reliable method to manage an Open Lab and secure the access off the different robots. There exists a comparable management solution of the company "Fabman" [1]. This solution is first of all designed for different machines like laser cutters. To our best knowledge, there is no solution like ours on the market by now.

## II. CONCEPT

### A. Components

The system needs following components:

- our software
- Raspberry Pi [2]
- NFC-reader [3]
- display
- database
- server in a local network

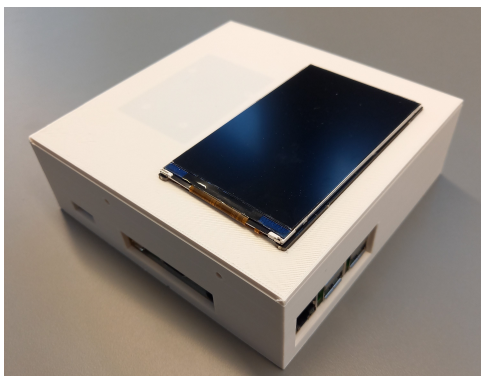


Figure 1. Assembled unlock station consisting of the in II-A mentioned parts

Every Open Lab using this system has an unlock station with the Raspberry Pi and the NFC-reader built in, as shown in Figure 1. It's planned that there are implemented three different tables in a database, one for the users, one for the robots and one for the robot reservations.

### B. General

If the user starts the robot of their choice, the robot sets up a connection to the network. In this state the robot is locked and can't be used. The started robot sends an request to the server just once, to look if the robot is allowed to unlock. As long as the reservation isn't activated the robot won't unlock. To unlock the robot, the user needs a valid reservation and the user has to authorize on the unlock station. If a NFC-Chip gets read by the unlock station it activates the reservation. The server sends an request to the robot that it's allowed to unlock. The request is also sent only once. This system will work if the robot was started first or the other way around that the reservation was activated before the robot was started. The system is designed for robots with network access.

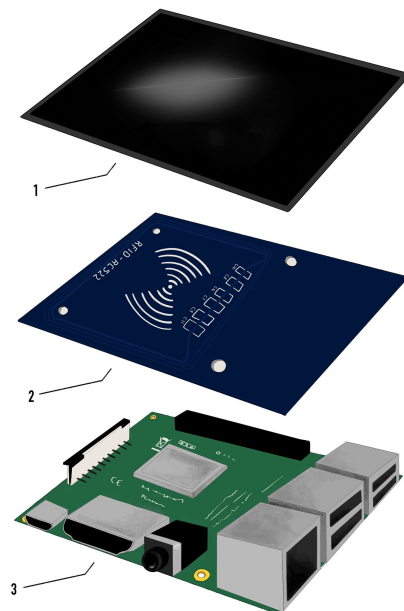


Figure 2. The structure of the unlock station consisting of: 1 Display 2 NFC-reader 3 Raspberry Pi

### C. Software

Every user of the Open Lab gets his own account. This account allows logging in to the reservation software. The software is constructed like a calendar for easy handling and good user experience, as shown in Figure 3.

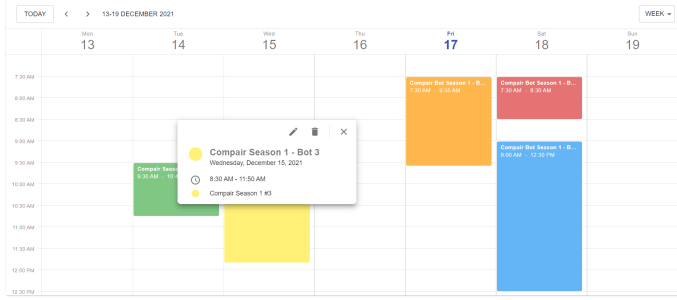


Figure 3. In the calendar you can create a new appointment by clicking on a free time slot. By clicking on your reserved time slot you can edit and delete the appointment.

Through the software the user is able to claim a time slot of his choice to use a certain robot. Which robot wants to be used can be selected by drop down menu, as shown in Figure 4. The selected robot will be reserved for that user in the desired time slot.

Figure 4. Creating a new appointment

## III. FIELD OF APPLICATION

The system targets to manage a wide range of different robots. From small competition robots up to industrial robots, every need is covered.

### A. Small competition robots

The best example for managing small competition robots are the robo4you compAir [4] robots. In our software the administrator is able to create different instances of robots which can be selected in the drop down menu to create their own appointment, like shown in Figure 4. The robots need to be connected with the same network that they can communicate with the server. Every robot gets his own number and description to simplify the working process. The robots can only be used by authorised people which is ensured through the process declared in II-B.

### B. Military robots

Military robots usually have a higher security standard. To secure this standard our system can be perfectly used. The system ensures by default a higher security level than without a solution. But military robots may need more security. For that problem the system can be extended so that for the verification process is required a biometrical scan like a fingerprint scan or a face scan. To provide more security it is possible to verify with the need of two independent individuals.

### C. Industrial robots

Industrial robots often don't have a network access. But that's no problem. We have a solution for stationary robots like industrial robots. For this method we need an additional component:

- relay [5]

With the relay the power supply of the robot can be controlled. The power supply is permanently capped until the user verifies himself at unlock station. After this process the relay closes the electrical circuit to allow the robot the working operation. The described solution could also be used for different machines without network access like laser cutters and different kinds of printer.

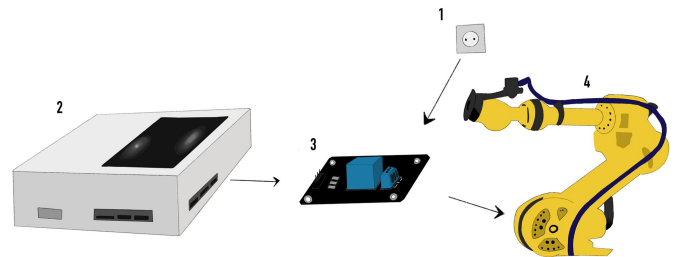


Figure 5. The setup between the unlock station consisting of: 1 power source 2 unlock station 3 relay 4 industrial robot or machine

From the power source the current goes to the relay. The relay blocks the current until the user authorized himself on

the unlock station. After that the unlock station sends a signal to the relay which let the current pass through to the robot.

#### D. Drones

Drones are getting more and more popular this time. Unfortunately they are very expensive and in course of that not accessible for everyone. Our system provides a suitable solution for everyone to use drones. It works like that: an Open Lab provides a few drones and they can be managed by our software to provide security and avoid time collisions when using the calendar.

### IV. VALUE FOR BOTBALL

#### A. Preparation for ECER

For the case that a organisation provides more competitive teams than robots our solution is an essential feature for managing the ECER [6] preparation. Every participating team has to make an appointment and reserve the desired time slot if they want to test their code and strategy for ECER. Otherwise it could come to conflicts between the different teams. The code of the teams is only available for the operating team. Only the code of the logged in team is usable, the other code is hidden and not usable for other teams. That provides the security of a teams code so nobody can spy on their code and steal or copy parts of it. Every robot has different working directories for every team. The code is saved on the robots in the particular directory of the team. In the respective directories are only the assigned team members entitled to make changes on the code. This preserves the code from other teams. In the preparation the teams don't always use the same robots because there may not be enough for everyone. That's why on every robot is installed git [7]. So every team can download and upload their code to GitHub to use every robot with the latest version of the code, no matter on which team they worked previously.

#### B. Botball

1) *Preperation:* It's possible to use the system for different Botball robots. As a user you can have different amount of calendars with different type of robots. You can create your own calendar which only your team can access for Botball to manage your different participating robots. All of your robots are getting an individual identification number and name. In the Botball calendar the robots will show up in the drop down menu like described before in II-C. You only have your Botball robots available in this calendar. If you want to use your robot you still need to create an appointment. That process provides that the robots are always locked if you didn't make an appointment and didn't authorize yourself. You cannot only manage the access and security of your robot. It's also possible that you add a unique description to each one of your robot. That can be used to write down the progress of the robot and possible errors. Due to a desired date a goal can be set where a specific feature should be fully functional on every desired robot. In the calendar can also be made normal appointments without robots like team

building action for example. So this system can be used as an overall management solution for your team and preparation for competitions like Botball. The software is perfect for security but also for planning and time management which is an important feature for preparation of big competitions.

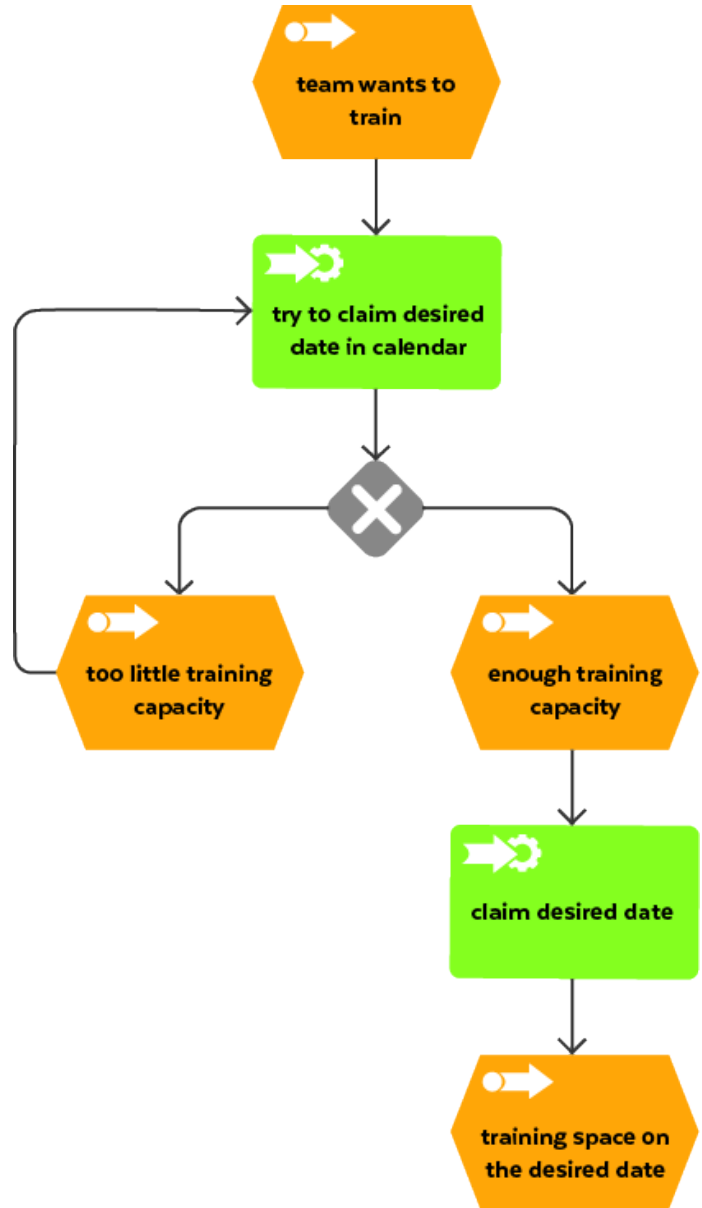


Figure 6. Modeled Process of a team making an appointment to train

2) *Competition:* The system is not only advantageous for Botball preparation, it can also be used in the competition itself. It's a big improvement while participating on the competition because you can manage and update your robots more easily while competing. The robots are connected with the system. On the software you can monitor the status of each robot that enhances the user friendliness and provides easier handling. The robots can retrieve status information from the

server of each other that helps the robots to coordinate on the game table.

### C. Course of training

We modeled the process of the training for the teams divided into the process of making an appointment and the process of the actual training itself.

1) *Making an appointment as shown in Figure 6:* When a team wants to make an appointment on specific date they would like to train, they need to use the calendar and try to claim their desired date. If the of the organisation preset maximum training capacity is reached, which means that no more team can train on this date because there are already training as much teams as possible, the team has to try to claim a new date. There is no priority which team can make an appointment first. The team who makes the appointment first gets the appointment. If the date is free and there is enough training capacity, the software will allow the team to claim this date and that provides the team their secure training space on their desired date.

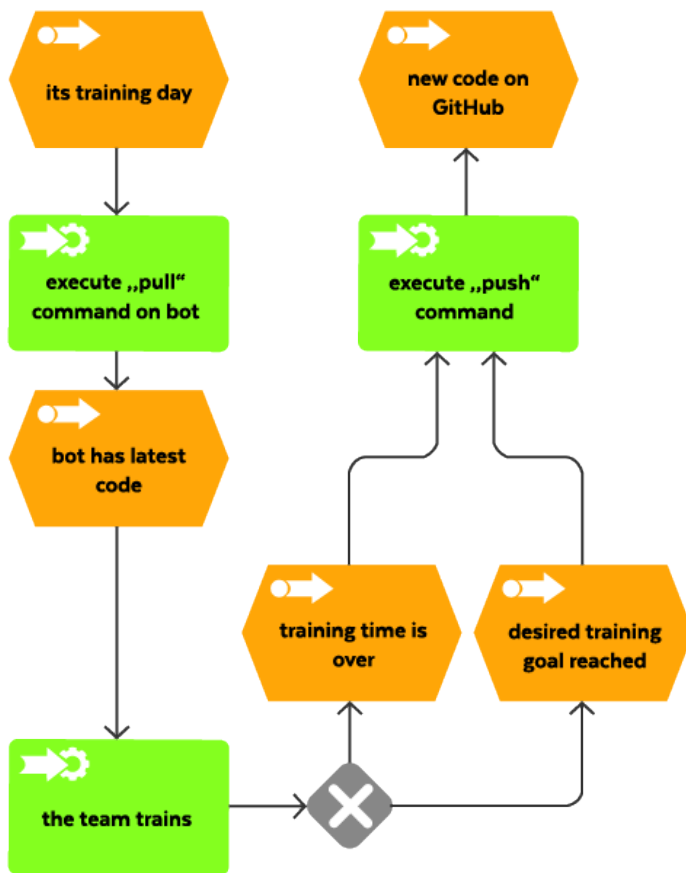


Figure 7. Modeled Process of a training session

2) *Process of training as shown in Figure 7:* When the date of the training for a team arrives that means it's training day. First action that has to be performed is to execute the "pull" command on the robot, that achieves that the latest code from GitHub is on the robot. When the robot is on the last stand, the

team can start to train. The team can train until their training time is over. They can also quit their training session earlier if they reach their desired training goal for example. In this case the left time is free again and can be used by other teams. After the training it's very important to don't forget to execute the "push" command. After this command your latest code from the robot is on GitHub and you can use it next time also on a completely different robot. That means your code is always safe on GitHub and you can use it from everywhere on any robot.

### V. CONCLUSION

In our laboratory we often have the need for a system like this. Because in our laboratory the "AIRlab" there often come together a big amount of people who want to work. That's why we often had working space problems. In an Open Lab you often don't know the other teams because they always work on other times. Of course you want to avoid that someone can use your code and maybe work on your code by accident. When using our system you don't have any problems with that because your code is secured and not accessible for others. For the organisers it's a big improvement because they don't need that many robots and equipment because anything can be shared like it should be in an Open Lab. Of course this saves a lot of money which can be invested in newer equipment and improvements on the infrastructure. At the moment the described system is just a concept and nothing was elaborated from us. This paper could be a good foundation for working on a system like this.

### REFERENCES

- [1] Fabman GmbH, "Everything you need to run your space effectively." <https://fabman.io/product/>, 2022
- [2] A. Nayyar and V. Puri, "Raspberry Pi-A Small , Powerful, Cost Effective and Efficient Form Factor Computer: A Review," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* 5(12), vol. 5, no. 12, pp. 720–737, 2015
- [3] H. A. Al-Ofeishat and M. A. Al Rababah, "Near field communication(NFC)," *Int. J of Comput. Sci. and Netw. Security (IICSNS)*, vol. 12, no. 2, pp. 93–99, 2012
- [4] Verein zur Förderung von Wissenschaft und Technik an Schulen (F-WuTS), "Competitive Robotics Reimagined," <https://comp-air.at>, 2022
- [5] V. Thonti, "Basic Working Principle of Relay - Construction and Types," <https://circuitdigest.com/article/relay-working-types-operation-applications>, 2017
- [6] PRIA, "ECER 2022," <https://ecer.pria.at>, 2022
- [7] GitHub, "Learn about the version control system, Git, and how it works with GitHub." <https://docs.github.com/en/get-started/using-git/about-git>, 2022