

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

# TeamCalendar

*Team ProgExTRAORDINAIRE:  
Klejdi Galushi, Marc Roemer, Felix Schneider*

Supervisor:  
Salvatore Sabba

June 10, 2024

# Contents

<b>1</b>	<b>Target determination</b>	<b>3</b>
1.1	Mandatory criteria . . . . .	3
1.2	Target criteria . . . . .	3
1.3	Optional criteria . . . . .	3
1.4	Exclusion criteria . . . . .	3
<b>2</b>	<b>Product deployment</b>	<b>3</b>
2.1	Application area . . . . .	3
2.2	Target groups . . . . .	3
2.3	Operating conditions . . . . .	3
2.4	Development setup . . . . .	3
<b>3</b>	<b>Product overview</b>	<b>3</b>
<b>4</b>	<b>Product functions</b>	<b>3</b>
<b>5</b>	<b>Product data</b>	<b>3</b>
<b>6</b>	<b>Product services</b>	<b>4</b>
<b>7</b>	<b>Quality requirements</b>	<b>4</b>
<b>8</b>	<b>User interface</b>	<b>4</b>
<b>9</b>	<b>Non-functional requirements</b>	<b>4</b>
<b>10</b>	<b>Technical product environment</b>	<b>4</b>
10.1	Software . . . . .	4
10.1.1	Client . . . . .	4
10.1.2	Server . . . . .	4
10.2	Hardware . . . . .	4
10.2.1	Server . . . . .	4
10.2.2	Client . . . . .	5
10.3	Orgware . . . . .	5
10.4	Product interfaces . . . . .	5
<b>11</b>	<b>Special requirements for the development environment</b>	<b>6</b>
11.1	Software . . . . .	6
11.2	Hardware . . . . .	7
11.3	Orgware . . . . .	7
11.4	Development interfaces . . . . .	7
<b>12</b>	<b>Division into sub-products</b>	<b>7</b>
<b>13</b>	<b>Additions</b>	<b>7</b>

14 Glossary	7
15 Sources	7

# 1 Target determination

## 1.1 Mandatory criteria

- Big readable text

## 1.2 Target criteria

## 1.3 Optional criteria

## 1.4 Exclusion criteria

# 2 Product deployment

## 2.1 Application area

## 2.2 Target groups

The target groups for this tool are:

- Students
- Companies
- Sport clubs
- Institutions

Based on this it is assumed that the frontend needs to be readable by people of all age groups.

## 2.3 Operating conditions

## 2.4 Development setup

1. Create a venv in `./Backend` using `requirements.txt` with `pip install --upgrade -r requirements.txt`
2. Activate venv
3. Navigate to `mysite` and run development server with `python manage.py runserver`
4. Execute `npm install` in `./Frontend`
5. Navigate to `my-app` and run development server with `npm start`

### **3 Product overview**

### **4 Product functions**

### **5 Product data**

To be saved Data will include:

- Username
- Calendar events consisting of:
  - Title
  - Start Time
  - End Time
  - Reminder Time
- Users with access to calendar

### **6 Product services**

### **7 Quality requirements**

### **8 User interface**

### **9 Non-functional requirements**

### **10 Technical product environment**

#### **10.1 Software**

##### **10.1.1 Client**

- Any browser that supports react 18.3.0 web apps or newer

##### **10.1.2 Server**

- python (3.12.2)
- Django (5.0.4)
- django-filter (24.2)
- djangorestframework (3.15.1)
- SQLAlchemy (2.0.30)
- MySQL

## **10.2 Hardware**

### **10.2.1 Server**

Linux or Windows server with internet access.  
Mac not tested.

### **10.2.2 Client**

Any device capable of running a browser that supports react 18.3.0 web apps or newer with internet access.

## **10.3 Orgware**

The deployed code needs to be documented to make servicing it possible. Intended documentation is the GitHub project where developers made comments for their decisions, Mask Draft and an Entity-relationship diagram for the database.

## **10.4 Product interfaces**

The frontend and backend need to be connected via their ip addresses in a network.

The backend and database need to be connected via their ip addresses in a network.

External interfaces are not needed.

## 11 Special requirements for the development environment

### 11.1 Software

- Git for downloading the project
- GitHub for accessing the project
- PyCharm for editing the backend and database
- Visual Studio Code for editing the frontend and backend
- python (3.12.2)
- nodejs (20.12.2)
- @fullcalendar/core (6.1.11 or newer)
- @fullcalendar/daygrid (6.1.11 or newer)
- @fullcalendar/interaction (6.1.11 or newer)
- @fullcalendar/react (6.1.11 or newer)
- @testing-library/jest-dom (5.17.0 or newer)
- @testing-library/react (13.4.0 or newer)
- @testing-library/user-event (13.5.0 or newer)
- axios (1.6.8 or newer)
- react (18.3.0 or newer)
- react-dom (18.3.0 or newer)
- react-scripts (5.0.1 or newer)
- web-vitals (2.1.4 or newer)
- asgiref (3.8.1)
- Django (5.0.4)
- django-filter (24.2)
- djangorestframework (3.15.1)
- Markdown (3.6)
- sqlparse (0.5.0)
- SQLAlchemy (2.0.30)
- MySQL

## 11.2 Hardware

Windows or Linux computer capable of running all software outlined in 11.1 Software.

Mac not tested.

## 11.3 Orgware

The agile software development method will be used. GitHub will provide all tools necessary for agile development. Sprints last a week and end on Thursdays in the lecture.

GitHub is used for:

- Storing code
- Managing code
- Sharing code
- Version control
- Continuous Integration/Testing (GitHub actions)
- Access control
- Bug tracking (Issues)
- Task management (GitHub projects)

## 11.4 Development interfaces

The frontend and backend need to be connected via their ip addresses in a development environment.

The backend and database need to be connected via their ip addresses in a development environment.

External development interfaces are not needed.

## 12 Division into sub-products

## 13 Additions

## 14 Glossary

## 15 Sources

- <https://en.wikipedia.org/wiki/GitHub> — viewed: 10.06.2024