Frankfurt University of Applied Sciences

TeamCalendar

Team ProgExTRAORDINAIRE:
Klejdi Galushi, Marc Roemer, Felix Schneider

Supervisor:
Salvatore Sabba

June 11, 2024

Contents

1 Target determination

1.1 Mandatory criteria

- Program consists of front end, back end and database
- Users can be changed once chosen
- Calendars can be shared
- Calendars have an owner
- Calendar ownership can be transferred
- Calendar events can be added, edited and removed
- Calendar events have a start and end time
- User will get notified at a custom time and at the start of the event

1.2 Target criteria

- Readable text
- Multiple custom notifications

1.3 Optional criteria

- Exclusive calendar events, that reserve a certain time slot so that calendar events can not overlap
- Warn user if creating a calendar event will result in overlapping calendar events
- Show next available time slot when creating a new calendar event
- Add users in the front end
- Delete users in the front end
- Email notifications
- Change calendar event color

1.4 Exclusion criteria

Viewing the calendar of other users will not be implemented as changing a user is possible.

2 Product deployment

2.1 Application area

This tool should be used by teams or individuals to organize themselves by providing a timetable. For this purpose team calendars can be shared and each user will get shown an overview of all calendars they and their team is responsible for.

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

The Program should be deployed on a single server that is running 24/7 in the cloud or on premises for each company. The program should not need be to administrated once deployed.

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

Upon first opening the website a user selection will be shown where a user can be selected. Upon choosing an user a calendar will be generated with all of the calendar events that the user has access to. User can be changed at the top right. There also exists a group windows for managing groups.

4 Product functions

In the calendar view:

- Calendar events can be moved by dragging the events with the mouse.
- Calendar events can be created by clicking on an empty part of a day grid.
- Calendar events can be deleted
- Reminder times can be modified after clicking on an calendar event

5 Product data

To be saved Data will include:

- Users
  Name
  Email
- Group name
- Calendar events consisting of:
  Title
  Description
  Start Time
  End Time
  Reminder Times
- Users with access to calendar

6 Product services

The program should generate the calendar view in a reasonable time and all calendar events have to be correct.

7 Quality requirements

It is required that the software implements all Mandatory criteria, and responds to the user in a reasonable time. The software should be easy to understand, maneuverable and text should be big enough to be read. Once deployed it should work without needed intervention from a software developer. If needed the server should be able to be scaled up vertically to accommodate more traffic and data. The client side should be accessible for all users with an up to date web browser that is able to run react 18.3.0 web apps or newer.

8 User interface

9 Non-functional requirements

The saved data shall not be accessible for 3rd parties. The data shall only be used for the intended purposes of providing a timetable and reminders. It shall not be used by anyone that has access to them for any purpose other than what the user consented to. The data shall not be sold or be used for personalized advertisements. The data shall be persistent and encrypted.

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. Once deployed the tool should not need to be administrated.

10.4 Product interfaces

The frontend and backend need to be connected via their ip adresses in a network.
The backend and database need to be connected via their ip adresses 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 with internet access.
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 front end and back end need to be connected via their ip addresses in a development environment.
The back end and database need to be connected via their ip addresses in a development environment.
The database is hosted externally at `http://descus.de`.

12 Division into sub-products

The product is layered and divided into 3 parts according to a layered architecture. The top layer is the front end providing an interface to the end user. The second layer is a back end that provides an endpoint for the front end to access data. The third and last layer is the database where the data is persistently saved. It is connected to the back end via ip address and password.
The front end is written in react with nodejs and javascript.
The back end is written in django using python3. The database is written using SQLAlchemy and MySQL.

13 Additions

The Functional Specification Document shall be delivered by 12.06.2024
The Mask draft shall be delivered by 19.06.2024 at the latest
The database concept diagram shall be delivered by 26.06.2024 at the latest.
The final product shall be delivered by 10.07.2024 at the latest.

14 Glossary

15 Sources

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