Final Feature Documentation of the IT-Project Timban

Table of Contents

1. Finished features

a. Dissatisfiers implemented

- Access Web-App
  - The application can be accessed through https://timban.herokuapp.com
  - For testing purposes after deployment localhost:8080 can be used to access the application with some existing dummy data
- Admin Login
  - Locally deployed: “admin@example.com” & “12345678” are the credentials for the admin account
  - Deployed on Service (Heroku): With “woof@fhnw.ch” & “password” the default admin account can be accessed
- Admin Logout
  - Any logged in account can be logged out via the selector button on the top right corner
- User Creation
  - Logged in as an admin user can be created through the admin page (dropdown -> Admin) localhost:8080/admin or https://timban.herokuapp.com/admin
- User Display
  - The user overview is the default landing page once entered the admin view
- User Update
  - From the admin overview user entities can be accessed and modified localhost:8080/overview/id (6 id’s available by default) or https://timban.herokuapp.com/overview/
- User Deletion
  - From the admin overview user can simply be deleted

- User Login
    - Locally deployed: For local deployment predefined users are: Hans, Jakob, Lisa, Jessica, Ruedi. Extending one of those names with “@example.com” and using the generic “password” to enter the application.
    - Deployed on Service (Heroku): There are no users predefined (other than the before mentioned admin user).
- User Logout
    - Same as “Admin Logout”
- Manage Profile
    - Logged in user profile can be modified via the “Settings”-Button on the top right corner
- Change Password
    - Same as preceding feature
- Record User Start Time
    - A logged in user can click the “Check-in”-Button to start record his time and simultaneously change his status to Online.
- Record User Stop Time
    - An “Online” user can click the “Check-out”-Button to stop record his time and simultaneously change his status to Offline.
- View Planned Working Times
    - Automatically displayed when logged in
- View Effective Working Times
    - Same as preceding feature
- View Overtime/Undertime
    - Same as preceding feature

b. Satisfiers implemented

- Edit / Correct Working Times
    - A user can remove a created entry
    - A user can adjust the time of a created entry
        - This feature is currently not running on Firefox but successfully tested with Chrome
    - A user can edit assigned projects
    - An admin can adjust the working hours of a user in the admin overview page

c. Delighters implemented

- Create Project
    - Admin can access/modify/create new projects via the “Projects”-Button from the admin overview localhost:8080/projects or https://timban.herokuapp.com/projects
- Assign Time to Projects
    - Project name and its due date can be modified by clicking on a project from the admin overview page
- View Report in weekly / monthly / yearly views
    - Admin can create a PDF from the admin overview page by clicking the “Print Report”-Button.

d. Added feature implementation through Agile Development

- Admin search bar
    - From the admin overview page, a search bar can be accessed to filter the entries (users, projects & ID's) and the search gets more specific based on the search term
- Validation
    - Entered user data is checked and verified

2. Video Tutorials

If this document is opened in the MS-Team environment or the browser version of Word the videos are accessible directly through the document otherwise the link above the picture needs to be clicked and which redirects the viewer to a browser window.

Deploy Timban Spring Boot Application Locally for Development

Deploy Timban Spring Application on Heroku

TimBan User Experience as Admin User

TimBan User Experience as Employee User

3. Application Deployment

As mentioned in the first two videos, the application can be deployed with two different variants. In this section, the explanation is provided in written and shortened form.

Deploy locally

This section explains how the application can be deployed on a local machine. Thus, this is useful for development but not for productive deployment of the system. Further, the data is stored in-memory. As soon as the application is stopped, the data is lost. This in-memory database is a H2 in-memory database.

The following steps are needed to deploy the system on an Ubuntu OS:

1. sudo apt install maven
2. sudo apt install git
3. git clone https://github.com/MathisHermann/TimBan.git
4. cd TimBan
5. mvn spring-boot:run

Deploy on Heroku

Heroku provides containerized deployment for web-applications. If Timban is deployed on Heroku, the database can be added directly as an add-on. The entities are automatically migrated on the first deployment of the application. Therefore, no further configuration of the database needs to be done.

The following steps explain how the application can be deployed on Heroku:

1. Create new app
2. Add Postgres Add-on
3. Add config Variables:

INSERT_EMAIL	woof@fhnw.ch	Email for the initial admin user
INSERT_PASSWORD	password	Password for the initial admin user
INSERT_USERNAME	Polo Sly	Username for the initial admin user
MAVEN_CUSTOM_OPTS	-pl timban-web	Maven environment variable
PATH_TO_PROJECT	timban-web	Path to the project – so the system knows where the root is
RUN_PROD	prod	Indication if the system is deployed for productive
TOKEN_SECRET	mysecret-...	Cryptographic key that signs the tokens – needed for security

4. Connect to Github (in this case only MathisHermann/TimBan will work)
5. Deploy the application
6. Access the application and enter with the credentials defined by “INSERT_EMAIL” and “INSERT_PASSWORD”.

4. Incomplete features

- Define Holidays
- Change Language
- Add / Change Color of a Project
- Create Project Manager role
    - Currently solved with “Admin”-creation to grant all access and additional features, could be brought back as an epic into a future release version backlog
- View Project Members
- View Project Detail per User
- View Project Detail per Project

Some of these features did not get developed due to a priority shift or resource constraints. Others have become obsolete during the development stage and new ones got implemented instead.

5. Reflection

As with most projects approaching its final stage a realization of not having enough time or resources is getting clearer. More features and ideas could have been implemented if only… The same is true (or even more so) for the documentation part. Reviewing the code, analyzing it, rewriting and cleaning it up would be another topic that would make software development more sustainable and therefore increase its lifecycle. Observing these issues can help allocate more resources (money, time, people) in future projects to increase the quality of the final product and therefore customer satisfaction.
Pair programming has been a very nice experience. It first seems like a very inefficient way of working/producing but looking back many positive aspects can be noted. Especially in a freshly formed team this seems to have a strong impact since it can help create a base understanding of the project and get into the view of other team members. Furthermore, this can be seen as a strong methodology to spread knowledge and in a more business centric aspect, maintain the existing knowledge in the long run.

This documentation can also be accessed with this link.