Group 1: Exercise 1

Alessio Petrone 20 127 395
Florian Keta 20 108 908
Kevin Stöckli 20 119 236
Aris Konstantinidis 14 104 350
Pavlos Konstantinidis 14 107 569
Jan Koch 20 109 591

Solutions

Task 1

- User Story 1 (Aris)
    - **As an unregistered user, I want to be able to scroll through all publicly available posts.**
    - Functional requirement: public posts should be fetched from the backend and projected onto the frontend application.
    - Non-functional requirement: private posts should not be able to be accessed by unauthorized (unregistered) users.

- User Story 2 (Pavlos)
    - **As a registered user, I want to be able to post new content (categories tbd).**
    - Functional requirement: on post submission, data is sent to the backend and then returned to the frontend.
    - Non-functional requirement: post content has to conform to given structure and policies.

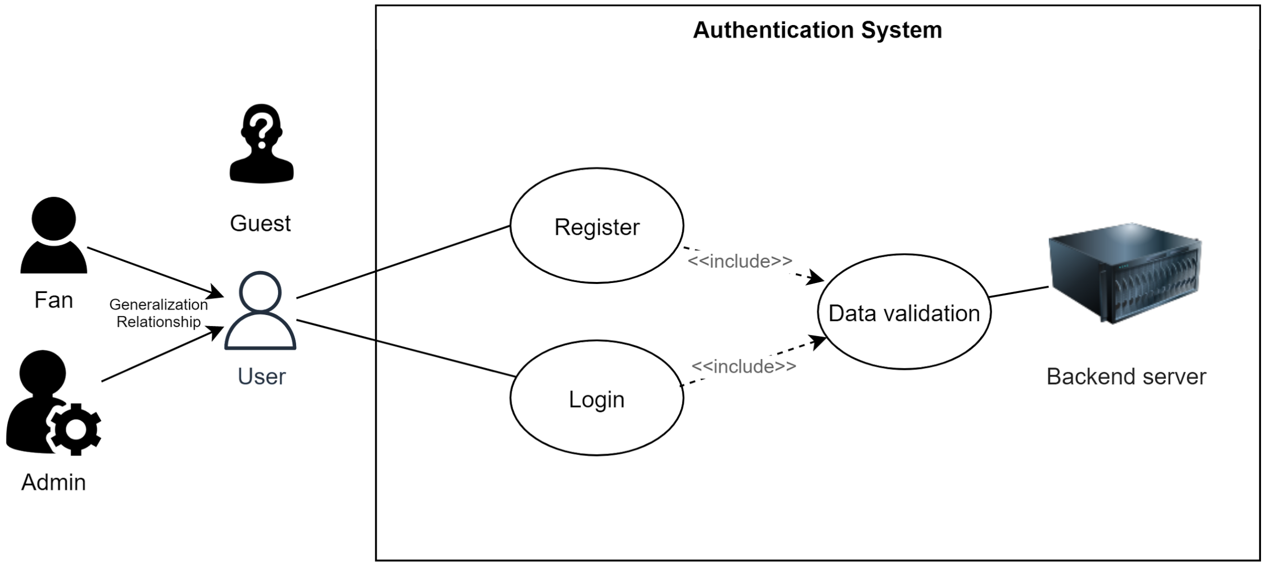
- User Story 3 (Alessio)
    - **As a registered user, I want to be able to personalize my homescreen/loginscreen on the website.**
    - “I” can design the background (choose from personal images) and change the fonts.
    - Functional requirement: having a button on the frontend which can “upload” an image, send it to the backend and return it to the frontend again. Having a drop down menu with different fonts, procedure is the same with the background image.
    - Non-functional requirement: frontend needs given buttons, and specific size requirements for the image.

- User Story 4 (Florian)
    - **As a registered user, I want to be able to report inappropriate or incorrect posts.**
    - Functional requirement: Buttons for reporting posts and giving reasons for the inappropriateness.
    - Non-functional requirement: All (or if they have a certain number of complaints) reports have to be investigated and rated.

- User Story 5 (Kevin)
    - **As a registered user, I want to interact with other people's posts (upvote, downvote, comment, …).**
    - Functional requirement: Have buttons for all interactions, send it to the backend and return it back to the frontend.
    - Non-functional requirement: All interactions should be instantaneous (< 0.5s). User should instantly see what he did (for example highlight upvote button).

- User Story 6 (Jan)
    - **As a registered user I want to be able to delete or edit my posts.**
    - Functional requirement: The backend needs to implement that a user can change his post.
    - Non-functional requirement: The frontend needs specific icons for easy access.

Task 2



*edit diagram here: https://drive.google.com/file/d/1PK6yl_6j3CtWprlPT0v9wIfdEapxZhCx/view?usp=sharing
*include: shows dependency between use cases (see https://www.uml-diagrams.org/use-case-include.html)

Task 3

User Story 1 (Aris): Description

- When a guest user enters the website, all or a subset of all public posts should be automatically fetched from the backend - without much/any delay or user action. The posts should be presented in a smart way, so that they are easy to read and scroll through (maybe vertical scrollable layout).

User Story 2 (Pavlos): Description

- When a user logs in, the GUI should provide strong visual clues to the option of posting content. Without having to reload or change route, the user should be able to post within a couple of clicks. The post must appear instantaneously (no perceivable lag) on the wall (according to the detailed requirements there is only one wall).

User Story 3 (Alessio): Description

- When a user logs in he should have an easily accessible settings page where he can change/upload an image with a simple click on a button and a drop down menu with the available fonts. The changes should only apply to the page when logged into the personal account.

User Story 4 (Florian): Description

- Mechanisms must be in place to prevent inappropriate and false posts. One option to achieve this is to allow the user to report such posts, which will then be investigated. This is necessary because otherwise the platform could be misused.

User Story 5 (Kevin): Description

- When a registered user sees a post he likes, he should be able to identify the upvote button almost instantly. When clicking the button, some feedback should be given to the user that the upvote has gone through successfully. If the user wants to see the comments on the post, a new tab / site with all other comments on the post should appear with the option to create an own comment.

User Story 6 (Jan): Description

- When a user wants to edit/delete his post it should be very intuitive (he doesn't have to look it up). The user needs to confirm if he wants to delete a post so there are less mistakes made.

Task Descriptions

Task 1

1. Write at least 6 user stories (use case) - 1 user story for each team-member
2. Classify which user stories correspond to functional requirements and which to non-functional - each one can classify his own story

Task 2

1. Create a use case diagram for the authentication feature - all together

- At least one actor
- Two use cases

Task 3

1. Write use case descriptions for each identified use case - everyone can describe their own story

Notes

Use case is the specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system

- Login to the timetable system
- Define a new committee
- Schedule a meeting

A **scenario** is a particular trace of action occurrences, starting from a known initial state

- Go to the timetable web site, enter your login, click on "forgot password"..
- Enter "new committee", enter committee name, select committee members from database, indicate committee chair

Functional requirements describe system services or functions

- Enter a new user into the timetable system
- Schedule a meeting
- These requirements must be precise and unambiguous

Non-functional requirements are constraints on the system or the development process

- User data must remain confidential
- Finding a possible meeting date should be instantaneous
- These requirements may be more critical than functional requirements

User Story (definition found on the web)

A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer. Focus on the "who, what, why" and avoid "how". A classic approach to create a user story is to use the following template:

- "As an [user], I want to [action], so I can [result]

Use case Diagrams

The most important components are:

- Actors: who interacts with the system
- Use case: functionality provided by the system
- Relation: between actors and the system