Functional requirements for OSER website

Florimond Manca

September 3, 2017

This document lists use cases that describe functional requirements for the OSER website project.

Contents

Use case 1. User logs in.

1.1 Prerequisites

- User is not logged in.

1.2 Main flow

1. User accesses to login page.
2. User is prompted with login details (email and password).
3. User enters login details and submits them.
4. System requires the user's login details from the database.
5. System validates that entered details match those from the database.
   Alternative flow: incorrect login details
   (a) System tells user login has failed.
   (b) Return to main flow step 2.
6. System displays the dashboard.

1.3 Outputs

- User is logged in.

Use case 2. User logs out.

2.1 Prerequisites

- User logged in.

2.2 Main flow

1. User clicks navigation link to logout.
2. System logs out user.
3. System tells user they have logged out.
4. System shows login page.

2.3 Outputs

- User not logged in anymore (becomes visitor).

Use case 3. User registers as student.

3.1 Prerequisites

- User is not logged in.

3.2 Main flow

1. User accesses to the login page.
2. User does not have an account, they click on a "register" link and system shows register page.
3. User is prompted with register details (first name, last name, email, password, role).
4. User enters register details as student and submits them.
   Alternative flow: email already used
   (a) System tells user email already used.
   (b) Return to main flow step 3.
5. User is taken to student complementary register page.
6. User enters complementary register information and submits them.
7. System creates new user account and associated student profile.
8. User is notified that account has been created.
9. System shows login page to login with new account.

3.3 Outputs

- User account and student profile exist in the database.

Use case 4. User looks for upcoming meetings.

4.1 Prerequisites

- User logged in as student or tutor.

4.2 Main flow

1. User accesses to dashboard.
2. User sees on page the list of 5 nearest upcoming meetings, if any.

Use case 5. User looks for *all* upcoming meetings through dashboard.

5.1 Prerequisites

- User logged in as student or tutor.

5.2 Main flow

1. User accesses to dashboard.
2. User clicks on link to see all upcoming meetings.
3. System shows page with list of all upcoming meetings.

Use case 6. User looks for *all* upcoming meetings through nav link.

6.1 Prerequisites

- User logged in as student or tutor.

6.2 Main flow

1. User clicks on nav link to see all upcoming meetings.
2. System shows page with list of all upcoming meetings.

Use case 7. User looks for upcoming visits.

7.1 Prerequisites

- User logged in as student or tutor.

7.2 Main flow

1. User accesses to dashboard.
2. User sees on page the list of 5 nearest upcoming visits, if any.

Use case 8. User looks for *all* upcoming visits through dashboard.

8.1 Prerequisites

- User logged in as student or tutor.

8.2 Main flow

1. User accesses to dashboard.
2. User clicks on link to see all upcoming visits.
3. System shows page with list of all upcoming visits.

Use case 9. User looks for *all* upcoming visits through nav link.

9.1 Prerequisites

- User logged in as student or tutor.

9.2 Main flow

1. User clicks on nav link to see all upcoming visits.
2. System shows page with list of all upcoming visits.

Use case 10. User looks for visit details through dashboard.

10.1 Prerequisites

- User logged in as student or tutor.

10.2 Main flow

1. User accesses to dashboard.
2. User clicks on details button for visit of interest.
   Alternative flow: visit of interest not on dashboard
   (a) User accesses all visits [UC8, UC9].
   (b) Return to main flow step 2
3. System shows visit details page showing: title, place, date, meeting time, estimated end time, remaining seats, description.

Use case 11. User signs up for visit.

11.1 Prerequisites

- User logged in as student.

11.2 Main flow

1. User accesses to details for visit of interest [UC10].
2. User clicks on button to participate to the visit.
3. System adds user to list of participants to visit.
4. System tells user they have signed up for the visit.

11.3 Outputs

- User is signed up for visit.
- User receives email with visit detail.

Use case 12. User signs off from visit.

12.1 Prerequisites

- User logged in as student.
- User signed up for visit of interest [UC11].

12.2 Main flow

1. User accesses to visit list [UC8 or UC9].
2. User clicks on button to sign off from visit.
3. Pop-up asks user if sure to sign off.
4. User confirms.
   Alternative flow: user does not confirm
   (a) Pop-up disappears.
   (b) Return to main flow step 1.
5. System removes user from list of participants to visit.
6. System tells user they were signed off from the visit.

12.3 Outputs

- User is not signed up for visit anymore.