ESE UniBe

Requirements Specification

Version 1

Team 6
6.10.2015

Revision history

Version	Date	Revision description
0.1	05.10.2015	Crude document and first specification
0.2	06.10.2015	Added use cases and diagram
1.0	06.10.2015	Initial version

Contents

1. Introduction

1.1. Purpose

This document provides the specifications for the website/application project for the ESE course at the University of Bern.

1.2. Stakeholders

Primary clients: Mario Kaufmann (ESE TA), Andrea Caracciolo (ESE Chief-TA)

1.3. Definitions

People	
User	Either a student or a tutor, who uses the website to provide/take lessons
Student	A type of user, who searches a tutor for a specific subject and pays his fee
Tutor	A type of user who gives lessons in a subject he passed for an hourly wage
Owner	The organization who owns and provides the service
Product	
System	The server machine which executes algorithms to provide the webpage with its contents and accounts
Site	The website, which is accessible by a web browser which displays all content related to the users (tutor profile etc.) on a front-end and provides basic administrative settings for the owner on a back-end

1.4. System overview

The platform is targeted at university students who are preparing for a specific exam. The goal is to focus on very specific matching criteria and to connect students with tutors that have sufficient knowledge of the subject. A tutor ideally has taken the exam more or less recently and has passed it with a good grade so that he can provide assistance in the required subject. The tutor on the other hand creates a profile page and provides the necessary information for a student to find him. He is able to ask an hourly wage of which the websites owner then receives a certain percentage as a commission.

1.5. References

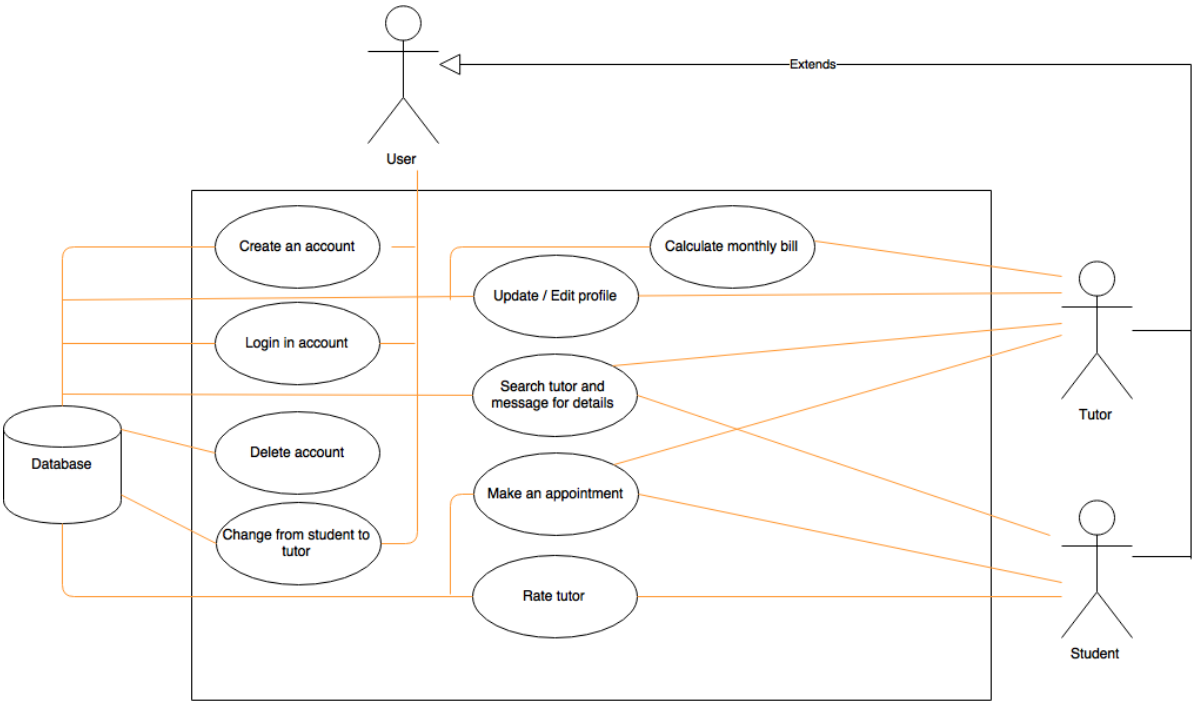
http://ese.unibe.ch

https://github.com/unibe-ese/ese2015-team6

2. Overall description

2.1. Use cases

A. Diagram



B. Use cases

1. Create an account

1.1. Actors

- Potential Users

1.2. Description

- As a student I want to register so I can browse for tutors.
- As a tutor I want to register so I can provide lessons for students

1.3. Trigger

- User lands on page without having an account.

1.4. Pre-conditions

- User is not logged in
- User has not yet created an account

1.5. Post-conditions

- User has an account
- User has confirmed his e-mail address
- User is logged in

1.6. Main Scenario

1. User browses to landing page
2. User chooses to create new account
3. User chooses his role (student or tutor)
4. User fills in and commits registration form
5. System sends confirmation e-mail with activation link

6. User activates account by browsing to send link
7. User is logged in automatically
8. Site redirects User to main page

1.7. Alternative Scenarios

4a. User does not fill all required fields or enters invalid values
1. Site displays error message
2. User corrects and recommits form
3. Use Case resumes on step 5

1.8. Special Requirements

- Confirmation e-mail should be sent in less than 1 min.

1.9. Notes

- What information are needed for both, students and tutors, to create an account?
- What information will be public or private?
- General terms and conditions of trade (GTCT)?

2. Login in account

2.1. Actors

- User

2.2. Description

- As a User I want to login to check my account

2.3. Trigger

- User lands on page having an account.

2.4. Pre-conditions

- User is not logged in
- User has created an account

2.5. Post-conditions

- User is logged in

2.6. Main Scenario

1. User browses to landing page
2. User chooses to log in
3. User fills in and commits login form
4. User is logged in.
5. Site redirects User to main page

2.7. Alternative Scenarios

2a. User has forgotten his password
1. Site displays wrong password message
2. User chooses to reset his password
3. User enters his account e-mail address

4. System sends password reset e-mail
5. User uses link to get to reset password form
6. User enters new password
7. User is redirected to login form
8. Use Case resumes in step 3

3a. User does not fill all required fields or enters invalid values
1. Site displays error message
2. User corrects and recommits form
3. Use Case resumes on step 4

2.8. Special requirements

- Password reset E-mail should be sent within a minute

2.9. Notes

3. Update/edit profile

3.1. Actors

- User

3.2. Description

- As a user I want to update/edit my existing profile information

3.3. Trigger

- User enters profile site

3.4. Pre-conditions

- User is logged in

3.5. Post-conditions

- Database refreshed the users profile
- Updated profile is directly online

3.6. Main Scenario

1. User select to edit the profile
2. User add/change existing information
3. User saves the changing
4. Database refreshes the profile and the updated profile is directly online

3.7. Alternative Scenarios

1a. The tutor select to change the existing time-schedule
1. The tutor enters all new dates for the schedule
2. The tutor sets a date for with this schedule should be valid
3. The tutor saves the change
4. Some reserved or confirmed appointments exist for this period
5. The tutor is asked what to do with the reserved appointments

6. The tutor is noted again of the existing confirmed appointments, thus they can't be denied afterwards
7. Use case resume in step 4

3.8. Special requirements

- Control mechanism for the existing appointments with the new time-schedule (step5) Should not take too much time, max 10 seconds
- Decisions about existing appointments will be noted to the students

3.9. Notes

- Should it be possible to deny an appointment afterwards?
- What information can the user select to be private or public?

4. Search a tutor and message for details

4.1. Actors

- Student
- Tutor

4.2. Description

- As a student I want to search / find a tutor who suits my needs
- As a student I want to communicate with said tutor
- As a tutor I want Students to find my profile
- As a tutor I want to communicate with potential students

4.3. Trigger

- Student enters main page.

4.4. Pre-conditions

- Student is logged in

4.5. Post-conditions

- Student has sent a message
- Tutor has received said message

4.6. Main Scenario

1. The student enters keywords and chooses filters
2. The student selects a tutor to view profile
3. The student chooses to message the tutor to discuss details
4. The student enters and sends the message
5. The tutor receives the message
6. The tutor answers the message

4.7. Alternative Scenarios

5a. The tutor is offline

1. The system sends an e-mail notification to the tutor
2. The tutor receives the e-mail and clicks the notification link

3. The tutor clicks the notification link and is taken to the message site
4. Use Case continues on step 6

4.8. Special requirements

- Notification e-mail should take no longer than a minute to send

4.9. Notes

- Information shown: name, region, short summary about achieved diploma, time-schedule, rating of other students, achieved notes of lectures

5. Make an appointment

5.1. Actors

- Student
- Tutor

5.2. Description

- As a student I want to make an appointment
- As a tutor I want to specify time slots for appointments
- As a tutor I want to make appointments according to my schedule

5.3. Trigger

- Student chooses to make an appointment with the tutor

5.4. Pre-conditions

- Student is logged in
- Student has found a suitable tutor

5.5. Post-conditions

- Student and tutor have an appointment

5.6. Main Scenario

1. The student goes to the schedule of the tutors profile
2. He chooses an appropriate open time-slot to make an appointment for
3. The tutor is notified of a student having made a request
4. The system marks the chosen slot as reserved
5. The tutor confirms the appointment
6. The student is notified of the confirmation and the system marks the slot as taken

5.7. Alternative Scenarios

5a. The tutor denies the appointment

1. The system notifies the student of the denial
2. The system resets the slot status to open

5.8. Special requirements

- Notification e-mail should take no longer than a minute to send

5.9. Notes

- Should it be possible to disable e-mail notifications?

6. Calculate monthly bill

6.1. Actors

- Owner
- Tutor

6.2. Description

- As an owner I want the server to generate automatically a monthly bill
- As a tutor I will look the current bill at any time

6.3. Trigger

- Tutor get to the page bill
- A month is over and the owner/tutor need the bill

6.4. Pre-conditions

- User is logged in
- Month/time period is over

6.5. Post-conditions

- Tutor is noted of the current bill(email)
- Owner gets the bill of each Tutor

6.6. Main Scenario

1. The system takes a percent of the monthly wages of the tutor
2. The system makes an overview of all given lectures with a detailed calculation
3. The system notes the owner and user with the bill-information
4. The tutor pay the bill
5. The owner controls it and inform the system about the payed bill
6. The system marks the bill as paid and saves it in an archive

6.7. Alternative Scenarios

4a. The tutor do not pay the bill

1. The system remembers the tutor/owner about the not payed bill after 1 month with an admonition
2. Use case starts at the step 4

1a. The tutor calls the page bill

1. The system takes a percent of the current time period wages of the tutor
2. The system makes an overview of all given lectures with a detailed calculation
3. The site will show the overview to the tutor

6.8. Special Requirements

- Email signature must be done

6.9. Notes

- How big is the percent of the owner?
- Who have to check if the bill is payed?
- Are every month a new reference number need for the bill or can they be reused?

7. Rate the tutor

7.1. Actors

- Students

7.2. Description

- As a student I want to rate the tutor after a private lesson
- As a tutor I want students to rate me, so that others can see my rating

7.3. Trigger

- The time of a private lesson is over
- The student, who has get a private lesson logged in

7.4. Pre-conditions

- The time of the private lesson is over
- Student is logged in

7.5. Post-conditions

- Student has rate the tutor
- Rating link is afterwards not available

7.6. Main scenario

1. The system notes the student, with a link, to rate the tutor after the private lesson
2. The student follows the link to the rating site
3. The students fills out the questions (slots 1(good)-5(bad))
4. The students can add a comment for other students
5. The student publish the rating
6. The system makes the rating link unavailable

7.7. Alternative scenario

1a. The student does not follow the link

1. After every log in the student will be noted of the ongoing rating

7.8. Special requirements

7.9. Notes

- Notification only if the student is only or else via email?
- What questions for the rating?
- Anonym rating or not?
- How long should a rating be available?
- Is a rating need after every lesson or for every subject?
    - Is there a subject selection under the time-schedule of the tutor?
- How many times can a student rate a tutor for different lessons?

8. Delete account

8.1. Actors

- User

8.2. Description

- As a User I want to be able to delete my account

8.3. Trigger

- User doesn't need account anymore

8.4. Pre-conditions

- User has an account

8.5. Post-conditions

- User information has been removed from database

8.6. Main scenario

1. The user clicks on the delete button on the edit account page
2. System flags the account to be deleted
3. System sends deletion cancellation E-Mail to User
4. Every day at 00:00 the deletion counter for the account is increased by 1
5. When the account deletion reaches 7 (1 Week) the account is deleted and the cancellation link is invalidated

8.7. Alternative scenario

5a. The user follows the link for deletion cancellation

1. The user confirms the cancellation on the website
2. System removes deletion flag and resets deletion counter

8.8. Special requirements

- Deletion confirmation e-mail is sent within a minute

8.9. Notes

- Should the deletion cancellation be resent a day before the account is deleted?

9. Change student to tutor profile

9.1. Actors

- Student

9.2. Description

- As a student I want to be able to change my account to a tutor

9.3. Trigger

- Student has enough knowledge and chooses to tutor himself

9.4. Pre-conditions

- Student has an account

9.5. Post-conditions

- Student account has switched role to tutor

9.6. Main scenario

1. The user clicks on the upgrade account to tutor button on the edit account page
2. Student confirms tutor service agreement
3. System changes account to tutor
4. Site redirects (former) student to create/edit profile page

9.7. Alternative scenario

9.8. Special requirements

9.9. Notes

2.2. Actor characteristics

Student	A University Student which requires aid in a specific subject. Technical knowledge varies quite much.
Tutor	A more experienced Student, TA or even professor who is able to help a student understand his subject. Technical knowledge usually higher but this is not a must.

3. Specific requirements

3.1. Functional requirements

- It's possible to register as a student or as a tutor
- Tutor and students can reset their password
- A tutor can register all the courses they have successfully completed, specifying the mark they got on the exam
- A tutor can customize the content of their profile
- A profile must contain the name of the tutor, the university they attend, their hourly rate, the languages they are familiar with and the timeslots they are free to tutor
- A profile can in addition contain a picture and a description
- A tutor's profile can only be viewed by registered users
- A student can use a keyword search to find profiles of potential tutors
- A clickable Calendar is displayed on the tutors profile, where a student can click if they are interested in a tutoring session
- If a student clicks on a viable field of the field, they can enter a message which then is sent to the tutor

3.2. Non-functional requirements

- Users can access the web portal with current browsers from different platforms
- User passwords are encrypted before being stored in the database
- Notification emails should be sent within one minute
- Tests should cover at least 70% of the source code
- The code has to be sufficiently documented