ESE UniBe

Requirements Specification

Version 1.1

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
1.1	13.10.2015	Corrections from last meeting
2.0	08.12.2015	Final 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

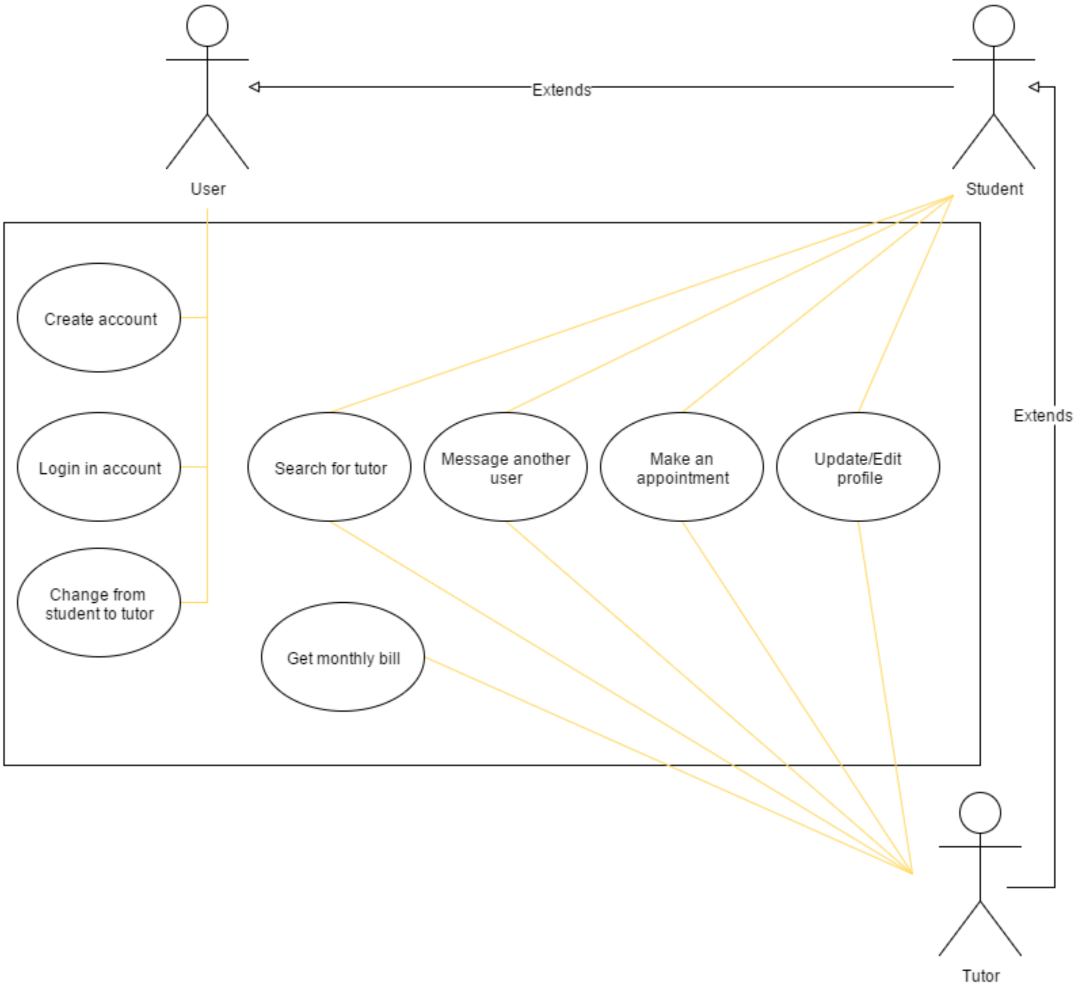
http://ese.unibe.ch

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

2. Overall description

2.1. Use cases

A. Diagram



B. Use cases

1. Create an account

1. Actors

- Potential Users

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

3. Trigger

- User lands on page without having an account.

4. Pre-conditions

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

5. Post-conditions

- User has an account

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. Registration is confirmed

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

8. Notes

2. Login in account

1. Actors

- User

2. Description

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

3. Trigger

- User lands on page having an account.

4. Pre-conditions

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

5. Post-conditions

- User is logged in

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

7. Alternative Scenarios

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

8. Special requirements

9. Notes

3. Update/edit profile

1. Actors

- User

2. Description

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

3. Trigger

- User enters profile site

4. Pre-conditions

- User is logged in

5. Post-conditions

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

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

7. Alternative Scenarios

1a. A 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. Use case resume in step 4

8. Special requirements

- Should not take too much time
- Email address can't be changed

9. Notes

- Appointment can't be denied afterwards

4. Search a tutor and message for details

1. Actors

- Student
- Tutor

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

3. Trigger

- Student enters main page.

4. Pre-conditions

- Student is logged in

5. Post-conditions

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

6. Main Scenario

1. The student enters keywords and chooses sort order
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

7. Alternative Scenarios

8. Special requirements

9. Notes

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

5. Make an appointment

1. Actors

- Student
- Tutor

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

3. Trigger

- Student chooses to make an appointment with the tutor

4. Pre-conditions

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

5. Post-conditions

- Student and tutor have an appointment

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 system marks the chosen slot as reserved
4. The tutor confirms the appointment
5. The system marks the slot as taken

7. Alternative Scenarios

4a. The tutor denies the appointment

5. The system resets the slot status to open

8. Special requirements

9. Notes

6. Calculate monthly bill

1. Actors

- Owner
- Tutor

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

3. Trigger

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

4. Pre-conditions

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

5. Post-conditions

- Tutor can view his bill in bill overview page

6. Main Scenario

1. The system calculates a percentage of the monthly wages of the tutor
2. The system makes an overview of the total wage earned and the billing amount
3. The bill is now displayed on the bill overview page
4. The tutor pay the bill
5. The system marks the bill as paid and saves it in an archive

7. Alternative Scenarios

1a. The tutor browses to the bill page

1. The system calculates a percentage of the current time period wages of the tutor
2. The system generates an overview a calculation
3. The site will show the temporary billing amount to the tutor

8. Special Requirements

9. Notes

7. Rate the tutor

1. Actors

- Students

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

3. Trigger

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

4. Pre-conditions

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

5. Post-conditions

- Student has rated the tutor

6. Main scenario

1. The student opens the appointment overview page
2. The students rates the appointment with a score between 1 and 5
3. The student publishes the rating

7. Alternative scenario

8. Special requirements

9. Notes

8. Change student to tutor profile

1. Actors

- Student

2. Description

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

3. Trigger

- Student has enough knowledge and chooses to tutor himself

4. Pre-conditions

- Student has an account

5. Post-conditions

- Student account has switched role to tutor

6. Main scenario

1. The user clicks on the upgrade account to tutor button on the edit profile page
2. System changes account to tutor
3. Site redirects (former) student to edit profile page

7. Alternative scenario

8. Special requirements

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
- In order to register a user needs to specify their name and email address, enter a password
- 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 at least the name of the tutor and can contain the university they attend, their hourly fee, 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 calendar, they can enter a message which then is sent to the tutor
- A User can delete their profile

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
- Tests should cover at least 70% of the source code
- The code has to be sufficiently documented