**Project Requirements**

**Project Name: Tutor Agency Web App** *(ESE-Exercise Project)*

**Team: Nr.9**

**Customer: A fake customer, played by J.N**

Revision History

| **Version** | **Date** | **Revision Description** |
| --- | --- | --- |
| .01 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Date: October 15, 2015

# Introduction

1. The Purpose Of This Document

In this document we will specify the requirements our exercise project for the course, “Introduction to Software Engineering” (ESE) should fulfill at the time of the first release. The requirements were collected in agreement between our team, and the stakeholders.

We will attempt to build a system, which provides at least the functionality that is described in this here.

1. Stakeholders

The stakeholders for this project are the assistants of this course, represented by J.N.

1. Definitions

To define some naming convention, we will refer in this project to people, who visit the website as Persons. And to persons with a user account as Users. Further, a user who has a tutor profile, therefore offers tutoring is called a Tutor. Out of simplicity we think of a tutor as either activated or not, meaning he is a paying user or not. Details for this would be specified for an implementation of a payment system.

For now, we also assume that the users are students from the University of Bern (i.e. the available courses for tutors to subscribe to will be the same as available there).

1. System Overview

The system will be a website that serves as an agency between students.

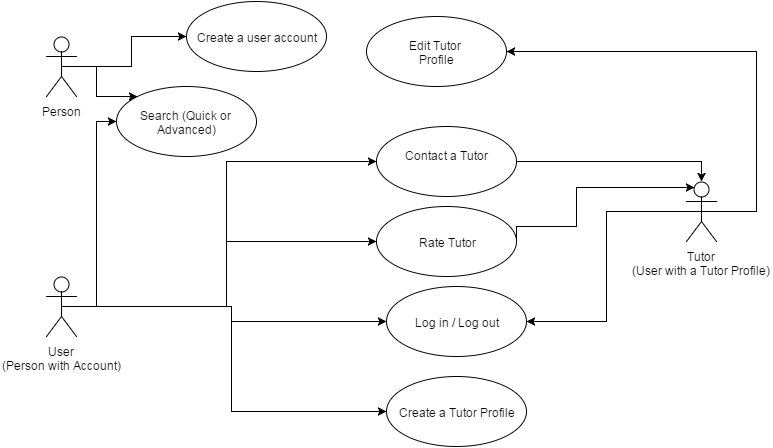
More precisely, it acts as an agency between students who are looking for a tutor, and students, who want to tutor another student.

The system will allow the tutors to create a profile to present themselves. Among other things, this profile will display the courses for which they offer tutoring. And the grade they had in those courses. A function to verify, if the tutor really has passed the course and with which grade is not part of this project.

The system will also have an interface for the students, where they can search from and browse between the available tutors. If they are interested, they can send a requests to the tutor.

Since the service provider wants to earn a commission, and therefore, the student shouldn’t be able to contact the tutor in any other way, the tutor profile will not display specific information (i.e. no email address and not the full name). The tutor will receive the information to contact the student who has requested tutoring, only if he is activated.

# Diagram



# Use cases

1. **Create a user account**
   1. **Actors**

Person

* 1. **Description**

As a person, I want to create a user account, to use the service of the website.

* 1. **Trigger**

Sign up page request to create a user account.

* 1. **Pre-conditions**

The user hasn’t created an account with his e-mail-address yet.

* 1. **Post-conditions**
     1. The system contains a personal account for the user.
     2. If the account couldn’t be created, the system provides enough information for the person to sign up correctly.
  2. **Main Scenario**

1. The person pushes a “Sign Up” button.
2. The person enters the account information.
3. The person pushes a “Create Account” button.
4. The system validates the input.
5. The system creates a user account.
6. A response is displayed, that the account was created successfully.
   1. **Alternative Scenarios**

4a. Invalid input (empty fields or wrong format)

1. Display feedback, for the fields which are empty or not formatted correctly.
2. The person corrects his input and clicks the “Create Account” button again.
3. Resume on step 4.

4c. Account under given email exists already

1. Display feedback, that there’s already an existing account under the given address.
2. The person may attempt to enter a different e-mail address, where the use case resumes on step 4.
   1. **Special Requirements**

The “Sign Up”-Button should be visible if and only if a no user is logged in.

* 1. **Notes**

Since person may share the same name, the uniqueness of the accounts has to be validated with the email address,

1. **Login**
   1. **Actors**

User

* 1. **Description**

Asa user I want to log into my account in order to interact with the system and view any information available according to my identity (tutor, student).

* 1. **Trigger**

User pushes “Sign In” button.

* 1. **Pre-conditions**
     1. The user isn’t logged in yet.
     2. The user has an account.
  2. **Post-conditions**
     1. The user is logged into his personal account.
     2. If the log in didn’t work, enough information is displayed, that the user can log in correctly.
  3. **Main Scenario**
     1. The user requests login.
     2. The user enters username and password.
     3. The user sends a request.
     4. System checks if information is valid.
     5. The user is logged in, and can interact with the system, as his identity allows.
  4. **Alternative Scenarios**

4a. No account registered under the given username or the password didn’t match

1. Display feedback message, below the according field.
2. The user fills out the form again.
3. The user sends request to log in.
4. Resume on step 4.
   1. **Special Requirements**

The “Sign In”-Button should be visible if and only if a no user is logged in.

1. **Logout**
   1. **Actors**

User

* 1. **Description**

Asa user I want to log out, so that no other person can use my account.

* 1. **Trigger**

User pushes “Sign In” button.

* 1. **Pre-conditions**
     1. The user is logged in.
  2. **Post-conditions**
     1. The user is logged out, meaning that e.g. no access is provided to any users account settings.
  3. **Main Scenario**
     1. The user requests logout (pushes “Log Out”-Button)
     2. The system ends the session with the designated user.
     3. A message is displayed if the logout was successful or not.

* 1. **Special Requirements**

1. **Tutor Search (Quick)**
   1. **Actors**

Person

* 1. **Description**

Asa person I want to use a search box, where I can enter faculty names, subjects or course names. And get a list with all the tutors related to my search criteria.

* 1. **Trigger**

The person has the search box selected and presses “Enter”-Key or a “Search”-Button.

* 1. **Post-conditions**

The person sees a list of the tutors related to the search criteria.

* 1. **Main Scenario**
     1. The person enters search criteria in the search box.
     2. The person triggers the search as described above.
     3. The search criteria matched to at least one tutor and a list of related tutors is displayed.
  2. **Alternative Scenarios**

3a.The search criteria matched to no tutor.

* 1. A feedback message is displayed, that no match occurred along with a help about which criteria to enter.
  2. **Special Requirements**

Maybe course-, subject-suggestions to choose from when the input is typed.

The search box should always be visible.

1. **Tutor Search (Advanced)**
   1. **Actors**

Person

* 1. **Description**

Asa person I want to use a search box, as for the use case 3, and additionally filter the results for example based on subjects, course name, fee range, rating, and region. Or select search criteria only from available values.

* 1. **Trigger**

Advanced search is requested.

* 1. **Post-conditions**

The person sees of a list of the matching tutors.

* 1. **Main Scenario**
     1. After requesting the search, the person can choose different filters/criteria to search from.
     2. The person pushes a “Search” Button to start the search.
     3. The search criteria matched to at least one tutor and a list of related tutors is displayed
  2. **Alternative Scenarios**

3a.The search criteria matched to no tutor.

1. A feedback message is displayed, that no match occurred along with a help about which criteria to enter.
2. **Contact a Tutor**
   1. **Actors**

User, Tutor

* 1. **Description**

As a user, I want to contact a tutor, i.e. send a tutoring request to him.

As a tutor, I want to get notified when a user requests tutoring from me, and have the possibility to accept this request and contact the user.

* 1. **Trigger**

The user sends a tutoring request for a course, possibly with a button displayed on the tutor profile, next to the course.

* 1. **Post-conditions**

The request wasn’t accepted or the tutor has contact information about the user.

* 1. **Main Scenario**
     1. The user sends a request (5.3).
     2. The tutor gets a notification about the request.
     3. The tutor accepts the request.
     4. The user get notified that the tutor should contact him soon.
     5. The tutor receives the contact information.
  2. **Alternative Scenarios**

3a.The tutors account is not activated (permitted from accepting the request).

1. The tutor is told to activate his account to accept the request.

3b.The tutors declines the request.

1. The user gets informed about the decline.
   1. **Special Requirements**
2. **Create Tutor Profile**
   1. **Actors**

User

* 1. **Description**

As a user who wants to tutor, I want to create a profile which displays the courses for which I offer tutoring. And that gives other users the possibility to request tutoring from me.

* 1. **Trigger**

The trigger would be to find as an option in the account settings.

* 1. **Post-conditions**

The user is now a tutor with his own (not yet activated) profile.

* 1. **Main Scenario**
     1. The user selects “Create Tutor Profile”-Option in his account settings.
     2. He is asked to fill out a form with the information for the profile (e.g. age, semester, faculty)
     3. He is asked to select the courses for which he wants to tutor, and to enter the grades.
     4. He is asked to confirm the entered information.
     5. The profile is created.
     6. The user gets to see the profile.
  2. **Alternative Scenarios**

4a.The user does not confirm.

1. If he leaves the form, the information get discarded.
2. **Edit Tutor Profile**
   1. **Actors**

Tutor

* 1. **Description**

As a tutor, I want to edit my profile to keep it up to date.

* 1. **Trigger**

The trigger would be to find as an option in the profile settings.

* 1. **Post-conditions**

The tutor has edited his profile. Changes are saved or discarded.

* 1. **Main Scenario**
     1. The tutor triggers the editing mode (7.3).
     2. The tutor adds or deletes courses to his profile or changes personal informations.
     3. The tutor pushes “Save” Button.
     4. The changes are saved and the profile is updated.
  2. **Alternative Scenarios**

3a.The tutor discards changes.

1. The profile is not be changed.
2. **Rate a tutor**
   1. **Actors**

Tutor, User

* 1. **Description**

As a user who got connected to a tutor and therefore received tutoring, I want to rate the tutor. So I can help other users to find good tutors.

* 1. **Trigger**

The tutor accepts the request from the user.

* 1. **Pre-Conditions**
     1. The tutor has received a tutoring request from the user.
  2. **Post-conditions**
     1. The rating is stored persistently.
     2. The rating is visible on the tutor’s profile.
  3. **Main Scenario**
     1. The user receives along with the notification he receives when the tutor accepts his request a message where he can rate the tutor once he had same tutoring lessons.
     2. The user finds in his personal account options the possibility to rate the tutor.
     3. The user rates the tutor and confirms.
     4. The option to rate this tutor is discarded.
     5. The rating appears on the tutor’s profile.
     6. The tutor gets notified about the rating.
  4. **Notes**

For the rating we will use a five-star-system.

# Specific Requirements

1. **Functional Requirements**
   1. **FR1 Tutor**

A user can create a tutor profile. The profile contains of the following information:

1. Username
2. Gender
3. Subjects, Courses, Grades
4. Semester
5. University
6. Fee Range
7. Rating
   1. **FR2 User**

A user account consists of the following information:

1. Full Name
2. Username (generated from full name)
3. Email Address
4. Password
5. Tutor Profile
   1. **FR3 Search (Quick)**

A quick search should find tutors with the following keywords options:

1. Subjects
2. Course Names
3. Usernames
   1. **FR4 Search (Advanced)**

An advanced search should provide options for all the tutor profile information (FR1) except the username.

* 1. **FR5 Rating**

The user can rate his tutor after accepted tutoring request.

* 1. **FR6 Initial Contact**

The initial contact between user and tutor personally happens once the (activated) tutor accepts a request. The tutor will then receive the contact information (email) of the user.

* 1. **FR7 Account activation**

For now, a tutor must have an activated account in order to accept requests. Depending on the payment system this might change into a commission he has to pay in order to accept the request.

1. **Non-Functional Requirements**
   1. **NFR1 Privacy**

Privacy of the tutor and the user must be ensured. The full name and the email address should only be revealed after the request is accepted.