**Project Requirements**

**Project Name: Tutorium** *(ESE-Exercise Project)*

**Team: Nr.9**

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

**Revision History**

| **Version** | **Date** | **Revision Description** |
| --- | --- | --- |
| .01 | 18.10.15 | Initial SRS |
| 1.0 | 09.11.15 | Release version 1.0 |
| 2.0 | 05.12.15 | Release version 2.0 |
|  |  |  |
|  |  |  |
|  |  |  |

Date: December 5, 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 end project phase. 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 document.

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 an account as Students. Further, a Student who has an upgraded account is called a Tutor. A tutor can offer tutoring by creating an Offer, which contains a Course. A course is a lecture/activity of a university.

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.

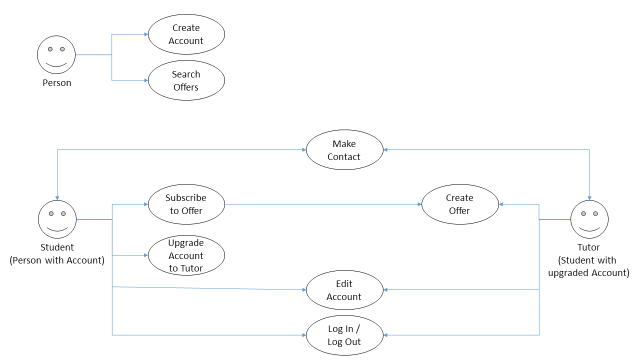
The system will allow the students to create a profile to present themselves. Tutors can create tutoring offers. A function to verify, if the tutor really has passed the course, for which they offer tutoring, and with which grade is not part of this project.

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

1. References

https://tutor24.ch/

# Diagram



# Use cases

1. **Create a student account**
   1. **Actors**

Person

* 1. **Description**

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

* 1. **Trigger**

Sign up page request to create a student account.

* 1. **Pre-conditions**

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

* 1. **Post-conditions**
     1. The system contains a personal account for the person.
     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 student 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 person is not logged in.

* 1. **Notes**

Since person may share the same name, the uniqueness of the accounts has to be validated with the email address or an artificial primary key,

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

Student

* 1. **Description**

Asa student 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**

Student pushes “Sign In” button.

* 1. **Pre-conditions**
     1. The student isn’t logged in yet.
     2. The student has an account.
  2. **Post-conditions**
     1. The student is logged into his personal account.
     2. If the log in didn’t work, enough information is displayed, that the student can log in correctly.
  3. **Main Scenario**
     1. The student requests login.
     2. The student enters username and password.
     3. The student sends a request.
     4. System checks if information is valid.
     5. The student 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 student fills out the form again.
3. The student 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 person is not logged in.

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

Student

* 1. **Description**

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

* 1. **Trigger**

Student pushes “Log Out” button.

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

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

Person

* 1. **Description**

Asa person I want to use a search box, where I can course names. And get a list with all the offers from 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 offers 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 offer and a list of related offers is displayed.
  2. **Alternative Scenarios**

3a.The search criteria matched to no offer.

* 1. A feedback message is displayed, that no offer exists which matched the search criteria.

1. **Offer 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 location, fee range and/or grade of the tutor.

* 1. **Trigger**

Advanced search is requested.

* 1. **Post-conditions**

The person sees of a list of the matching offers.

* 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 offer and a list of related tutors is displayed
  2. **Alternative Scenarios**

3a.The search criteria matched to no offer.

1. A feedback message is displayed, that no offer exists which matched the search criteria.
2. **Contacting / Request Tutoring**
   1. **Actors**

Student, Tutor

* 1. **Description**

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

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

* 1. **Trigger**

The student subscribes to an offer either found via the search or directly on the profile of a specific tutor.

* 1. **Post-conditions**

The request wasn’t accepted or the tutor and the student both receive the contact information of their counterpart.

* 1. **Main Scenario**
     1. Student subscribes to an offer (5.3). Whit this subscription, he requests tutoring.
     2. The tutor gets a notification about the new request.
     3. The tutor accepts the request.
     4. The student get notified that the tutor has accepted his request and receives the tutors contact information.
     5. The tutor receives the contact information of the student.

1. The student gets informed about the decline.
   1. Notes
      1. Communication between the parties is not handled by the system.
2. **Upgrade Account to Tutor**
   1. **Actors**

Student

* 1. **Description**

As a student who wants to tutor, I want to upgrade my account to tutor status, which allows me to add additional tutoring related information to my profile and add/manage tutoring offers.

* 1. **Trigger**

The trigger would be to find as an option on the account page.

* 1. **Post-conditions**

The student is now a tutor with his account upgraded to tutor status.

* 1. **Main Scenario**
     1. The student selects “Become Tutor”-Option in his account settings.
     2. Additional information becomes available on his account page:
        1. Fee
        2. Locations
        3. Tutoring Offers
     3. Additional options become available on his account page:
        1. “Add Offer”
        2. Ability to manage the offers.

1. **Edit Account**
   1. **Actors**

Student, Tutor

* 1. **Description**

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

* 1. **Trigger**

The student/tutor clicks the “Edit Profile”-button on his account page.

* 1. **Post-conditions**

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

* 1. **Main Scenario**
     1. The student/tutor triggers the editing mode (7.3).
     2. The student/tutor has the option to change account information:
        1. First Name, Last Name, Password, Email, Description, Username.
     3. The tutor has additional options to change information related to tutoring:
        1. Fee, Locations
     4. The student/tutor pushes “Save” Button.
     5. He is asked to confirm changes with his password.
     6. The changes are saved and the account is updated.
     7. The student/tutor gets notified that the changes are saved.
  2. **Alternative Scenarios**

3a.The student/tutor discards changes.

1. The profile is not be changed.
2. **Add/Create offer**
   1. **Actors**

Tutor

* 1. **Description**

As a tutor I want to create/add an offer to my profile.

* 1. **Trigger**

The tutor clicks the “Add Offer”-button on his account page.

* 1. **Pre-Conditions**
     1. The student has upgraded his account to tutor, and is on his account page.
  2. **Post-conditions**
     1. The offer is stored persistently.
     2. The tutor can see the offer on his account page.
     3. A person can find the offer via the search function.
  3. **Main Scenario**
     1. The tutor selects “Add Offer” on his account page.
     2. He is asked to select the course he wants to offer and to specify the grade he had in this course.
     3. He clicks the save button.
     4. The created offer gets displayed on his account page.
     5. The created offer can be found with the search function.
  4. **Notes**

The courses are organized per university.

# Specific Requirements

1. **Functional Requirements**
   1. **FR1 Student**

A basic member account consists of the following information.

1. Full name
2. Username
3. Email
4. Password
5. Description
6. Is the student a tutor?
   1. **FR2 Tutor**

An upgraded student account has additionally the following information:

1. Locations (Universities)
2. Fee
3. Tutoring offers (consisting of course and grade the tutor had in this course).
   1. **FR3 Search (quick)**

A quick search should find offers from tutors with the course name as search criteria.

* 1. **FR4 Search (Advanced)**

An advanced search should provide options for filtering and refining the search results:

* 1. Locations (Universities)
  2. Fee range
  3. Grade of the tutor.
  4. **FR5 Initial Contact**

The initial contact between student and tutor personally happens once the tutor accepts a request. Both parties will then receive the contact information of their counterpart.

* 1. **FR6 Account activation**

In order to be able to answer a request, a tutor must have an activated account. If the account is not activated the necessary information to complete the activation process are displayed. The activation process is not handled by the system.

* 1. **FR7 Privacy/Visibility of Information**

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

* 1. **FR8 Specificity**

The system should not support general subjects. A course is associated with one and only one university. That is, several courses on the same subject may exists for different universities.

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

The System should not reveal private information unauthorized. A person has data sovereignty of his information.

* 1. **NFR2 Responsiveness**

The use of the system feels responsive.

* 1. **NFR3 Mobile Ready**

The system is fully usable on modern mobile devices.

* 1. **NFR4 Compatibility**

The system is rendered correctly on up-to-date browsers Chrome and Firefox.

* 1. **NFR5 Display Responsiveness**

The system is rendered correctly and adaptive down to a device-width of 500px.

* 1. **NFR6 Specificity**

The system does not provide general tutoring. That is, tutoring is only possible for a specific course.