**Project Requirements**

**Project Name:**

**Team: Nr.9**

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

Revision History

| **Version** | **Date** | **Revision Description** |
| --- | --- | --- |
| .01 |  |  |
| .02 |  |  |
| .. |  |  |
| 1.0 |  |  |
|  |  |  |
|  |  |  |

Date: October 14, 2015

# Introduction

1. The Purpose Of This Document

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

For this reason, it serves as contract between us, and the assistants of this course.

We will attempt to build a system, which provides the functionality that is described in this document. And are not obligated to add other functionalities to our system, when not specified here.

1. Stakeholders

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

1. Definitions

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.

For now, we assume that the students will be from the University of Bern (i.e. the available

courses for tutors to subscribe to will be the same as here).

In the following, we will refer to the students looking for a tutor as students and for those who want to tutor as tutors. And if we refer to tutors, and students, we call them users.

1. System Overview

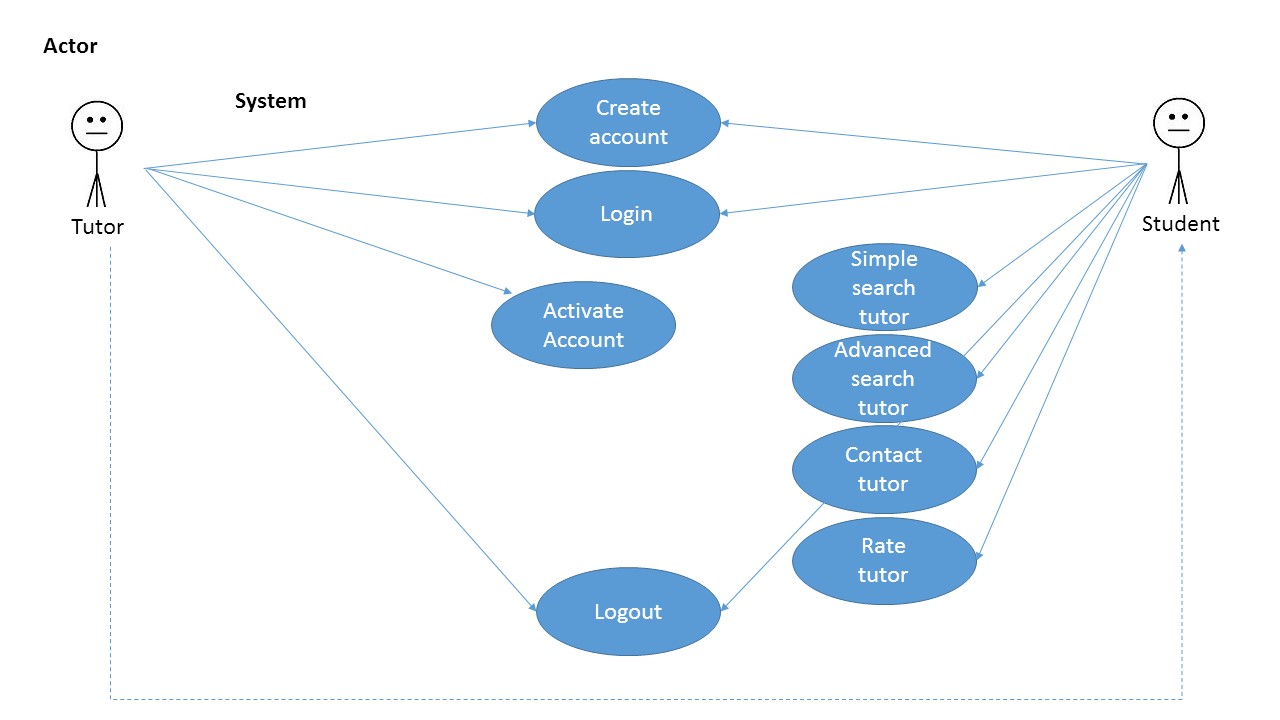
The system will allow the tutors to create a profile to present themselves. Among other things, this profile will display the courses for which they offer tutoring, and the grade they had in those courses. A function to verify, if the tutor really has passed the course

and with which grade is not part of this project.

On the other side, the system will 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, 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 once he paid a commission to the service provider.

# Diagram



# Use cases

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

Person who wants to become a user.

* 1. **Description**

As a person who wants to become a user, I want to create a personal account, to use the service.

* 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. Not all necessary fields are filled out or not in the right 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**
   2. **Notes**
3. **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 for 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 logged in.
     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, an 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, under the according field.
2. User fills out the form again.
3. User sends request to log in.
4. Resume on step 4.
   1. **Special Requirements**
   2. **Notes**