

# Project Requirements

**Project Name: STutor**

**Team: ESE Team 8**

**Customer: Niklaus**

## Revision History

Version	Date	Revision Description
0.1	06.10.2015	Regrouping of the first Project Requirements
0.2	14.10.2015	Added missing requirements.
0.21	20.10.2015	Refined use cases to the actual UI, added a few NFR
0.22	28.10.2015	Fixed Use Case Diagramm
0.23	11.11.2015	Refined use cases to the actual UI, added a few NFR

Date: November 11, 2015

# 1 INTRODUCTION

## 1.1 *Purpose*

This document serves as a description of the project. It includes an overview of the system to be developed, its possible uses, user goals and characterizations and needed specific requirements.

## 1.2 *Stakeholders*

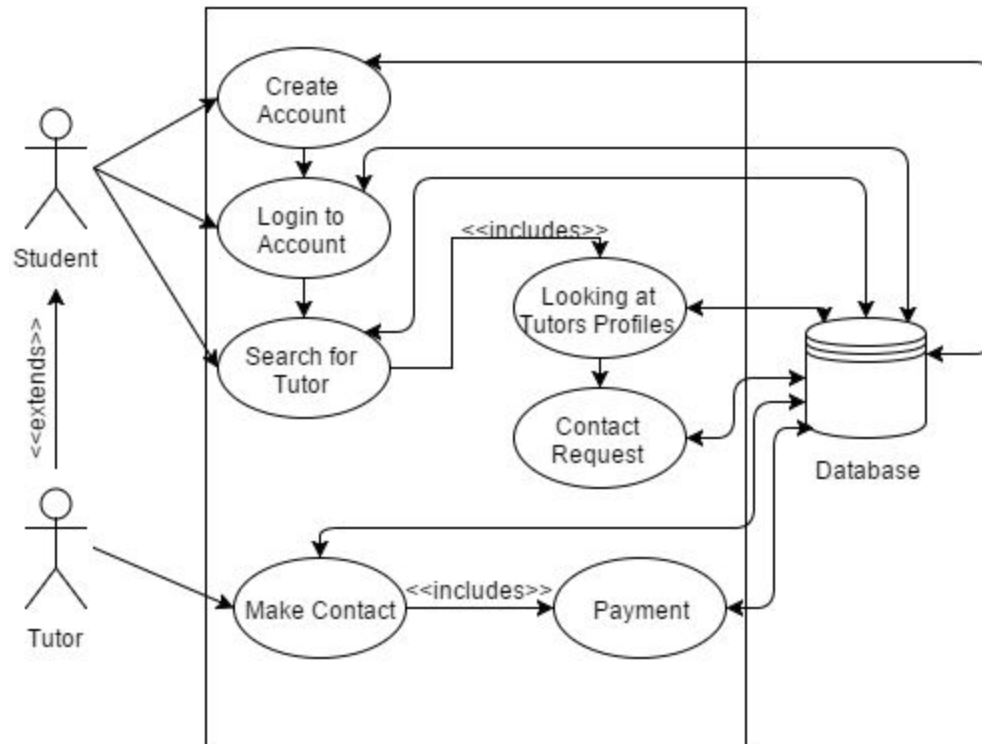
Aside from the developers, this project is of interest to the stakeholders, namely the ESE assistants who are client and head of the project, but also the targeted user base: students and tutors (see 1.3).

## 1.3 *System Overview*

The software system will be a website targeted at university students, who are looking for tutoring in a specific university course, and tutors, who will offer their expertise in those subjects to teach willing students. The platform will enable students to find matching tutors based on their search criteria and connect with them. For this service it will demand a commission on each connect.

## 2.1 Diagram

### Use Case Diagram (V1.3)



## 2.2 Use cases

### 1. Create Account

#### a. Actors

Student  
Tutor  
Database (secondary)

#### b. Description

A User wants to create a new account

#### c. Trigger

A User clicks on the 'Sign Up' Button on the log in page.

#### d. Pre-conditions

User shouldn't be registered yet

#### e. Post-conditions

##### i. For Student

1. Username and email must stay unique
2. Password shouldn't be null
3. New User should be saved in Database

##### ii. For Tutor

1. -

#### f. Main Scenario

- i. User opens the registration page.
- ii. The new account page opens with its registration form.
- iii. Enters first and last name, username, a password and email.
- iv. If a tutor wants to sign up, the checkbox "Tutor" has to be checked.
- v. Clicks on the "Sign Up" button.
- vi. Data is saved in Database
- vii. Opens new Page with: "Sign Up Complete" to confirm registration.

#### g. Alternative Scenarios

User Closes the Window or reloads it, all entered data are deleted

4a. Username or Email already exist in Database

1. Stay on the Subscribe Page
2. Notifies user that username/email are already used

4b. User's Email is invalid

1. Notifies user that email is invalid

#### h. Special Requirements

-

#### i. Notes

-

## 2. Fill in account for Tutor

### a. **Actors**

Tutor

Database (secondary)

### b. **Description**

As a tutor I want to create a rich profile that highlights my skills and maximizes my visibility.

### c. **Trigger**

A User clicks on the “Add Lecture” or “Add Timelaps” Button

### d. **Pre-conditions**

User must be logged in as a tutor

### e. **Post-conditions**

-

### f. **Main Scenario**

- i. User clicks on “Add Lectures” or “Add Timelaps”
- ii. Enters Lecture, Subject, University and Grade or day and hours
- iii. Clicks on the Button “Add this lecture” or “Add Timelaps”
- iv. Data is Saved in Database
- v. Opens profile page with updated information

### g. **Alternative Scenarios**

User Closes the Window or reloads it, all entered data is deleted

### h. **Special Requirements**

-

### i. **Notes**

-

### 3. **Login to Account**

#### a. **Actors**

Student  
Tutor  
Database (secondary)

#### b. **Description**

A User wants to login

#### c. **Trigger**

User enters the URL address of the Website and gets redirected to the login page

#### d. **Pre-conditions**

User is registered

#### e. **Post-conditions**

User enters his/her account if username matches password.

#### f. **Main Scenario**

1. User enters the URL address of the website and gets access to the login page
2. User enters username and password
3. System verifies match with data in Database
4. Data matches, and a new page opens with the user logged in

#### g. **Alternative Scenarios**

3a. Username doesn't exist in Database

1. Goes back to last step with notification to User

3b. Password doesn't match Username in Database

1. Goes back to last step with notification to User

#### h. **Special Requirements**

-

#### i. **Notes**

-

#### 4. **Search for Tutor**

##### a. **Actors**

Student

Database (secondary)

##### b. **Description**

As a student I want to find a tutor that studied the subject I'm preparing for.

As a student I want to make sure that a tutor can provide his services according to my needs.

As a student I want to make sure that a tutor has good knowledge of the subject.

##### c. **Trigger**

Student clicks on "Search Tutor" Button

##### d. **Pre-conditions**

- i. Student is logged in

##### e. **Post-conditions**

- List of Tutors is Displayed (or none if no one corresponds)

##### f. **Main Scenario**

1. Student clicks on "Search Tutor" Button
2. Student applies filter
3. Student looks at a Tutor Profile

##### g. **Alternative Scenarios**

2a. No Tutor matches the filter

1. System prompts an excuse message

##### h. **Special Requirements**

-

##### i. **Notes**

-

## 5. Contact Request (from Student to Tutor)

### a. **Actors**

Student

Database (secondary)

### b. **Description**

As a student I want to contact a tutor and discuss the details of the engagement. In order to do that, I first need to send a request to the tutor, which he has to accept.

### c. **Trigger**

Student has clicked on the "Make Contact" button on a Tutor profile page

### d. **Pre-conditions**

- i. Student is logged in

### e. **Post-conditions**

- 1. Contact request sent to Tutor

### f. **Main Scenario**

- 1. Student has clicked on the "Make Contact" button on a Tutor profile page
- 2. System sends a notification to the Tutor
- 3. UI redirects the Student on home Page
- 4. The tutor receives a notification

### g. **Alternative Scenarios**

-

### h. **Special Requirements**

-

### i. **Notes**

-



## **6. Make Contact (Tutor to Student)**

### **a. Actors**

Tutor

### **b. Description**

As a tutor, I want to effortlessly interact with potential customers

As a service provider, I want to earn a commission on each engagement

### **c. Trigger**

Tutor receives notification Email that he got a request by a student without revealing private information about this student so the Tutor cannot contact the student without paying the commission.

### **d. Pre-conditions**

- Tutor is logged in

### **e. Post-conditions**

- Tutor has paid money to the service provider
- Tutor receives information about the student so that a direct contact can be established

### **f. Main Scenario**

1. Tutor receives notification Email with a link but no private information about the student
2. Tutor clicks on the link in the Email. This links him to the website where he has the choice to either accept (and thus will have to pay) or decline the offer
3. Tutor accepts the offer, which makes him pay the common fee
4. Tutor pays the commission to the service provider
5. Tutor receives Student contacts via Email as soon as the provider got the money

### **g. Alternative Scenarios**

- 3a. Tutor does not accept the offer and thus has to pay nothing
  1. Student receives Notification Email that the request was declined

### **h. Special Requirements**

-

### **i. Notes**

# 3 REQUIREMENTS

## 3.1 Functional requirements

### 3.1.1 Tutor

A user can create a tutor profile. For signing up, only the username, full name, password and e-mail address are required. Optionally, a tutor can add a lecture, subject, grade (multiple subjects are possible) and university and also his gender. He can also add times when he would be free to tutor.

The public tutor profile consists of the following information: Username, rating, gender, what lectures he can tutor and when he's free.

### 3.1.2 Student

A user can create a student profile. Required sign up information is the email address, a password and a username, as well as his first and last name. The public student profile page consists of the username and email of the user.

### 3.1.3 Search

Search for tutors by lecture, subject, gender, minimum grade and university. The student should be able to see the tutor's free times in order to decide whether he fits his needs or not.

### 3.1.4 Rating

A student can rate his tutor after he's been accepted by this tutor. The scale goes from 1 to 6, only integers. A future student should then be able to see the medium rating of a tutor without having to contact him.

### 3.1.5 Initial contact

Initial contact happens within the application. The application contacts the tutor when he was requested by a student. The tutor can accept this request and thus has to pay a given fee before getting the student's email address for further contact.

### 3.1.6 Access to account

A user can always login into his created account with his username and matching password, or log out of it.

## **3.2 Non-functional requirements**

### **3.2.1 Privacy**

Privacy of the tutor and the student: Full name and contact information should only be revealed after successfully establishing contacts.

### **3.2.2 Commission**

The tutor needs to pay a commission to be able to answer requests.

### **3.2.3 Portability**

The system should run on the desktop browsers Firefox and Chrome, but also be working correctly on the iPad.

### **3.2.4 Accessibility**

The website is available 24/7. Multiple people can access it simultaneously. It shouldn't be down for more than an hour in one month.

### **3.2.5 Response time**

Every page not related to the searching of tutors is loaded within 2 seconds. Pages related to the searching of a tutor, including refreshing the filters, will be loaded within 10 seconds.

### **3.2.6 Security**

Passwords are stored safely encrypted in the database, so that even a security breach will not reveal the user passwords.