Business Requirements Set

<ESE Project: Platform for students and tutors>

ESE Team 2

Cockpit				
Phase	**Deadline**	**Status**	**Date**	**Person**
Creation	07.10.2015	SRS 0.5	03.10.2015	M. Wenger [MW]
Accepted		SRS 0.9 ☐	09.12.2015	E. Mendoza [EM]
Finish	09.12.2015	SRS 2.0 ☒	09.12.2015	E. Mendoza [EM]

Change log

Versio n	Date	Person	Type of change
0.5	03.10.2015	Mischa Wenger	Create (first, not review ready version)
0.52	05.10.2015	Marc Jost	Extended intro, use cases, assumptions and more
0.53	05.10.2015	Marc Jost	Added further Use Cases
0.54	06.10.2015	Eve Mendoza Quiros	Added Use Cases
0.55	06.10.2015	Cyrill Portmann	Added Use Case Diagrams
0.56	07.10.2015	Eve Mendoza Quiros	Changed Use Case Set up a contract
0.6	14.10.2015	Cyrill Portmann	Changed Use Case Diagrams (after meeting)
0.61	14.10.2015	Eve Mendoza Quiros	Changed Use Cases Change profile, Set up contract (after meeting)
0.62	21.10.2015	Eve Mendoza Quiros	Changed Use Case Register (after meeting)
0.63	28.10.2015	Eve Mendoza Quiros	Changed Use Case Change profile (after meeting)
0.7	10.11.2015	Eve Mendoza Quiros	Updated complete SRS (before release 1.0)
0.71	11.11.2015	Eve Mendoza Quiros	Changed Use Cases Change profile, Search for tutor, Communication (after meeting)
0.72	17.11.2015	Eve Mendoza Quiros	Updated Use Cases
0.73	18.11.2015	Eve Mendoza Quiros	Changed Use Case Set up contract (after meeting)
0.8	06.12.2015	Eve Mendoza Quiros	Update several Use Cases
0.82	08.12.2015	Eve Mendoza Quiros	Insert new updated UML

List of distributors / stakeholders

Function	**Name** [& shortcut]	**OE/Firma**	**Review**	**Info**
Project-Team				
Request Owner	Andrea Caracciolo [AC]		x	
Dev. Team	Cyrill Portmann [CP]		x	
Dev. Team	Eve Mendoza Quiros [EM]		x	
Dev. Team	Marc Jost [MJ]		x	
Dev. Team	Mischa Wenger [MW]		x	

Content

1 Introduction

Every year there are new technologies and features developed all over the world. The modern bachelor student of computer science has to deal with all these changes. In this immense jungle of information, it's hard to find out what to learn, how to learn and when to learn.

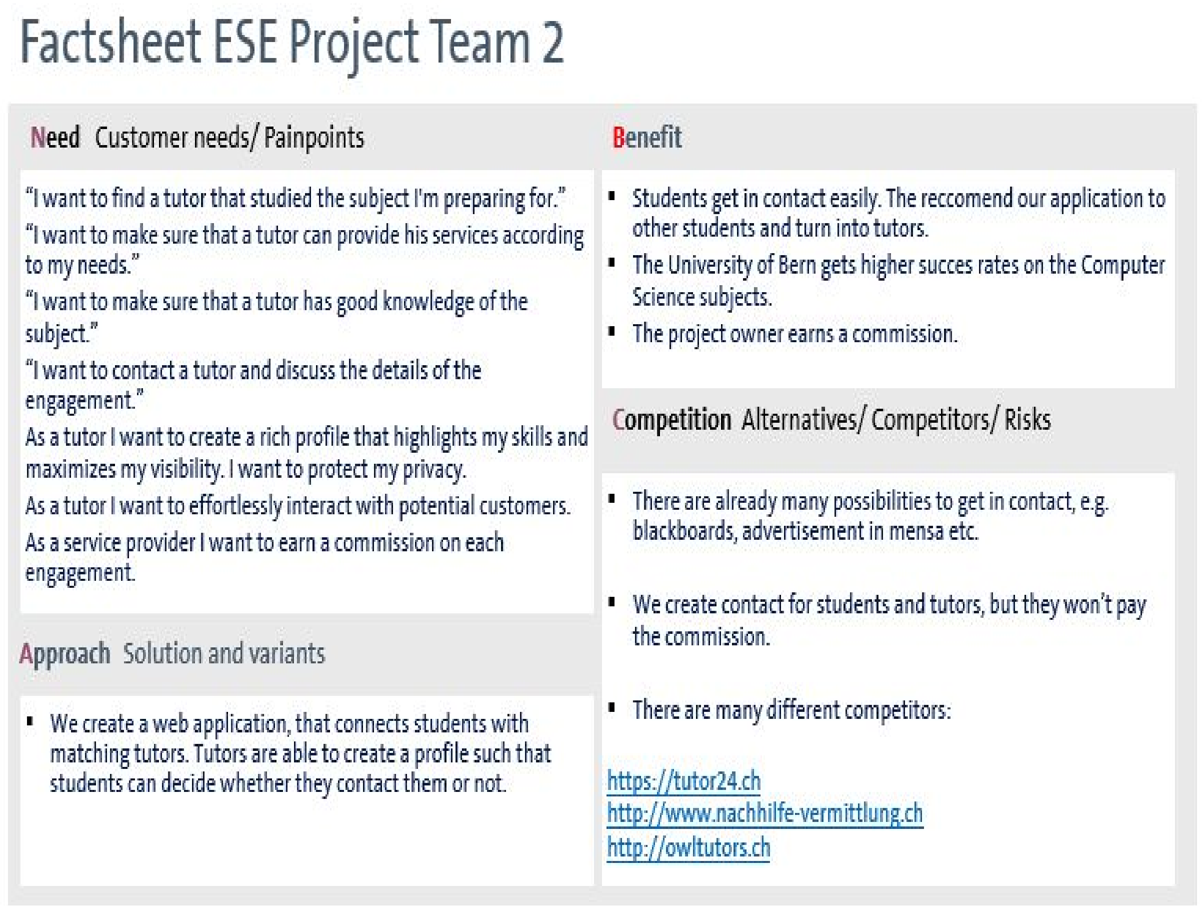
With the ESE Team 2 Platform, we offer a possibility for students to easily get in touch with qualified tutors who can support the students during the semester and/or during exam preparations. These tutors display their skills by featuring grades on their profile pages. Along with a profile picture, a description of the tutor’s person and the demanded rate a student is able to make an informed decision, whether to contact the tutor or not. The to-be-developed platform offers a simple meeting-arrangement-feature allowing students and tutors to arrange their first lesson through the platform. The website furthermore offers a payment system through a third party provider, through which students can pay their tutors without needing to know detailed banking information for all their tutors. The project owner will collect a commission on every payment, making the platform beneficial for all involved parties.

This document specifies the project order and the software requirements.

2 Pitch for the project (incl. vision)

Hook:

To connect students and tutors we create an interactive platform for students and tutors. Students can easily select their preferred tutor by browsing for the course they require help for. As a tutor, I don't have to create advertisements or to check the blackboards all the time – the students contact me instead. Additionally, the scope of the tutoring is clearly defined in form of a University lecture which doesn't change all too often. As a student, I don't have to browse through all the other generic tutoring websites which cover a too wide area of studies. Instead, I can visit the platform which is tailored to the Bachelor in computer science at the University of Berne. As a student, I can relate to the tutor's past experience by examining his/her grades in the same lecture I am currently taking. Since the tutor already took the exam, a student can also learn about the exam itself, further enhancing his/her chances for success.



Factsheet ESE Project Team 2

Need Customer needs/ Painpoints

"I want to find a tutor that studied the subject I'm preparing for."
"I want to make sure that a tutor can provide his services according to my needs."
"I want to make sure that a tutor has good knowledge of the subject."
"I want to contact a tutor and discuss the details of the engagement."
As a tutor I want to create a rich profile that highlights my skills and maximizes my visibility. I want to protect my privacy.
As a tutor I want to effortlessly interact with potential customers.
As a service provider I want to earn a commission on each engagement.

Approach Solution and variants

- We create a web application, that connects students with matching tutors. Tutors are able to create a profile such that students can decide whether they contact them or not.

Benefit

- Students get in contact easily. The reccomend our application to other students and turn into tutors.
- The University of Bern gets higher succes rates on the Computer Science subjects.
- The project owner earns a commission.

Competition Alternatives/ Competitors/ Risks

- There are already many possibilities to get in contact, e.g. blackboards, advertisement in mensa etc.
- We create contact for students and tutors, but they won't pay the commission.
- There are many different competitors:

https://tutor24.ch
http://www.nachhilfe-vermittlung.ch
http://owltutors.ch

Close:

This platform is specialized for bachelor students in computer science of the University of Bern. It is a more focused version of the existing platforms, which convinces with ease of use and professionality.

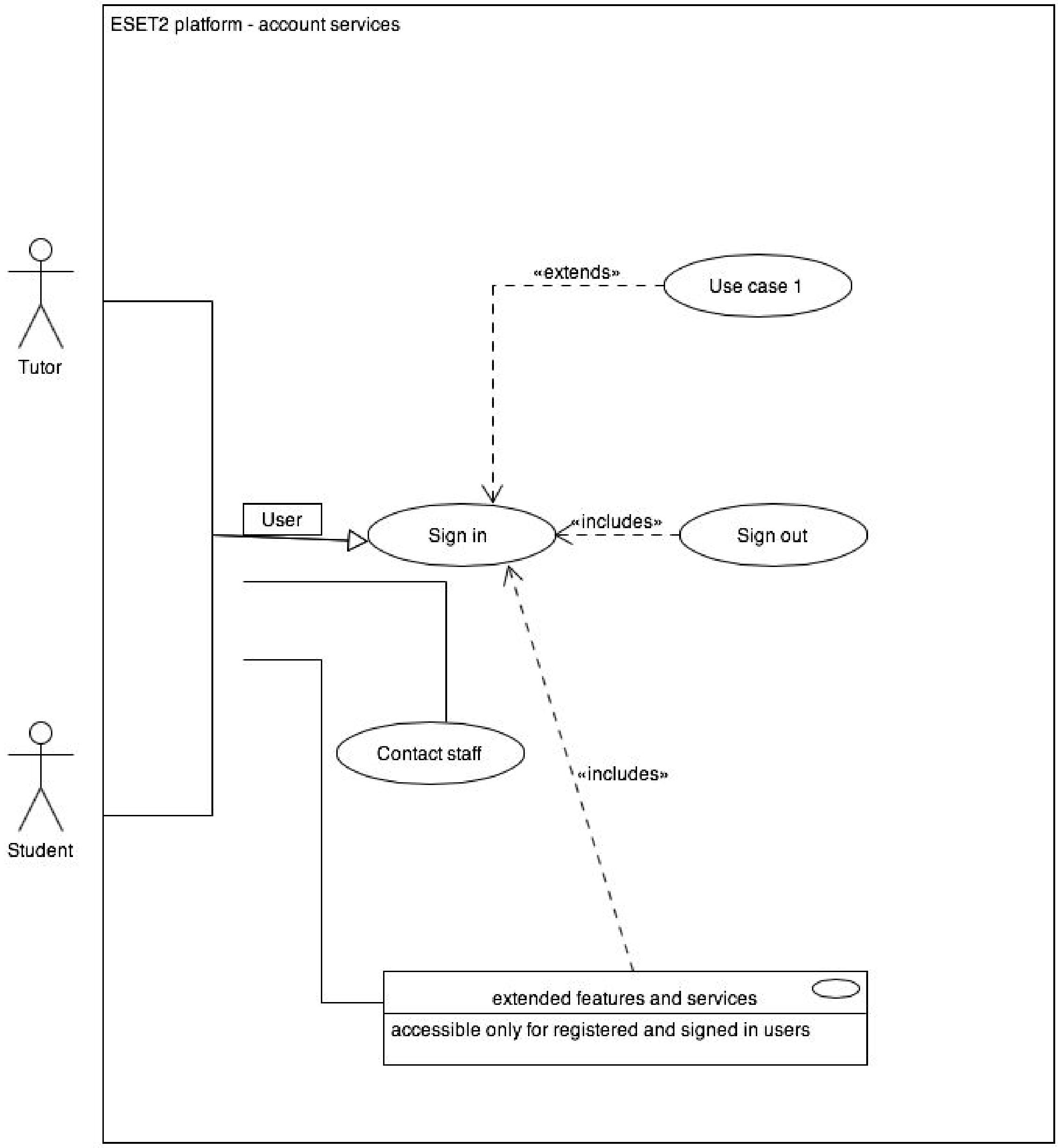
3 Customer needs

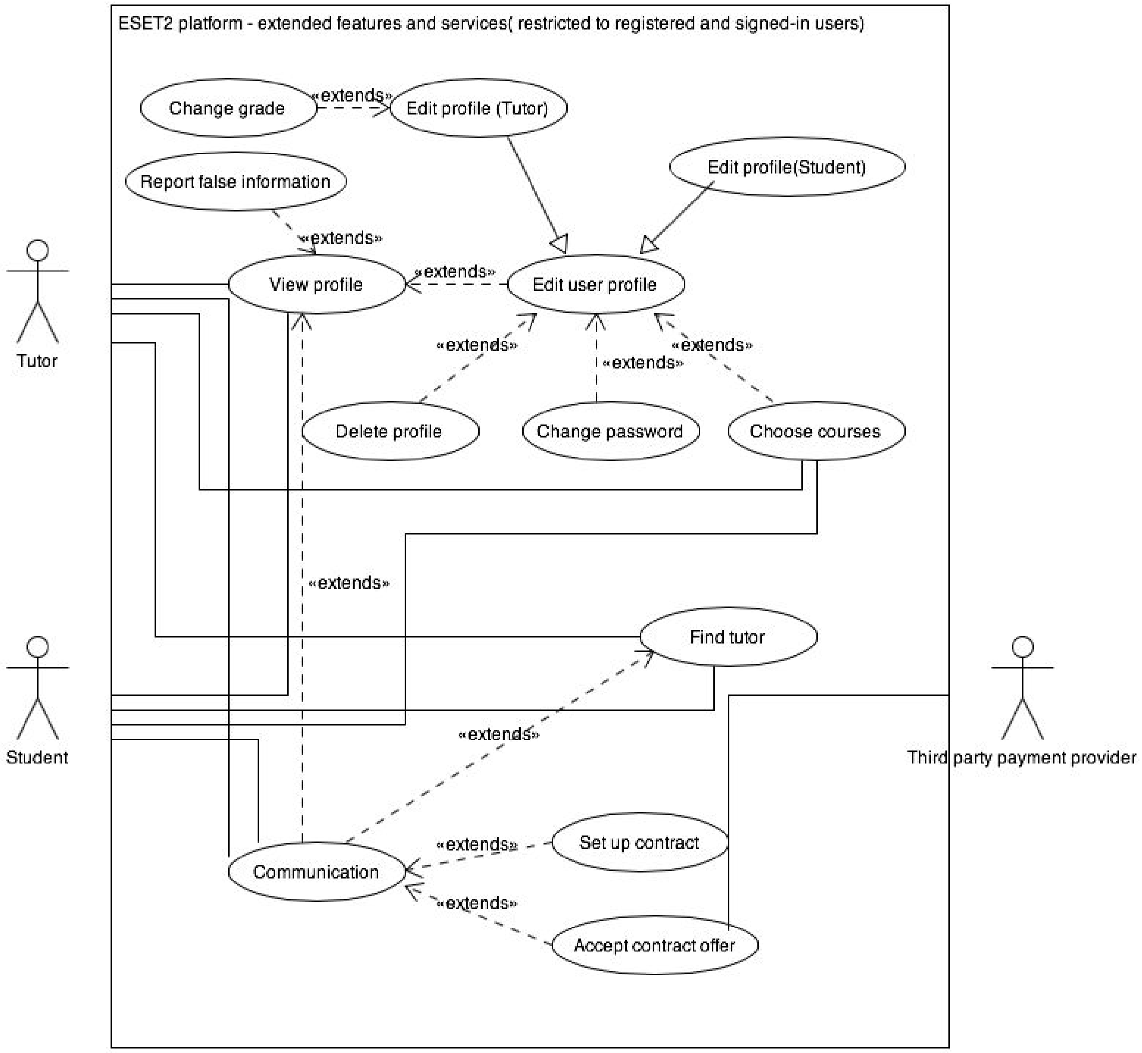
This project addresses these customer needs:

No.	Description of the need	Field	
		Customer needs *(User-Stories)*	**Project owner**
N-01	I want to find a tutor that studied the subject I'm preparing for.	X	
N-02	I want to make sure that a tutor can provide his services according to my needs.	X	
N-03	As a student I want to make sure that a tutor has good knowledge of the subject.	X	
N-04	As a student I want to contact a tutor and discuss the details of the engagement.	X	
N-06	As a tutor I want to effortlessly interact with potential customers.	X	
N-07	As a tutor I want to create a rich profile that highlights my skills and maximizes my visibility.	X	
N-08	As a tutor I want to protect my privacy.	X	
N-09	As a service provider I want to earn a commission on each engagement.		X

4 Use cases

In the following the most important use cases are listed and shown in diagrams. This set of use cases may not be complete and will be subject to change according to new information.


ESET2 platform - account services
Tutor
Student
User
«extends»
Use case 1
Sign in
«includes»
Sign out
Contact staff
«includes»
extended features and services
accessible only for registered and signed in users


ESET2 platform - extended features and services(restricted to registered and signed-in users)
Change grade
«extends»
Edit profile (Tutor)
Edit profile(Student)
Report false information
«extends»
View profile
«extends»
Edit user profile
«extends»
«extends»
«extends»
Delete profile
Change password
Choose courses
Tutor
«extends»
Find tutor
Student
Third party payment provider
«extends»
«extends»
Set up contract
Communication
«extends»
Accept contract offer

4.1 Registration:

Actors:

Students

A person enrolled at the University of Berne in Computer Sciences is called a student. A student visiting this platform is in need of a tutor who can support him/her in a subject.

Tutors

A person who is (or was) enlisted at the University of Berne in Computer Sciences and provides tutoring services to current students is called a tutor. Tutors have typically passed the exams they are offering their tutoring services for.

Description:

Certain features, such as contacting a tutor and owning a profile are limited to registered users only. Certain profile information (e.g. grades, contact details) is also hidden to the public, requiring users to register in order to view this more sensitive information.

- As a student I want to register on the platform, so I can use all the features it offers.
- As a tutor I want to register on the platform, so I can use all the features it offers.

Trigger:

A student or a tutor visits the platform, chooses to register and visits the registration page.

Pre-conditions:

1. Both student and tutor don't have an existing account on the platform.
2. Both student and tutor must have and provide an active University of Berne email-address in order to register.
3. Both student and tutor must have a matriculation number.

Post-conditions:

1. The student receives his student account and can now use all platform features.
2. The tutor receives his tutor account and can now use all platform features.

Main Scenario:

1. A person (user) visits the platform and choses to register
2. The user visits the registration page
3. The user enters his E-Mail address, his password and confirms the password
4. The user chooses if he/she wants to register as tutor or as a student
5. The user provides all necessary information according to the chosen account type:
    a. Student: First and Last Name, Matriculation Number, Address(Street, Zip, city), phone number

   b. Tutor: First and Last Name, Matriculation Number, Address(Street, Zip, city), phone number

6. The system validates the provided information
7. The user enters the captcha to verify himself
8. The user clicks the register button to conclude the registration
9. The system creates a new account and profile in the database
10. The user is on the main page again and can log into his account.

Alternative Scenarios:

3a. User didn't provide a valid email-address (limited to @students.unibe.ch):

1. System will prompt the user to provide a valid email-address
2. User enters a valid email-address
3. Use case resumes at step 4

3b. User provides an email-address already in use:

1. System displays a message stating the email is already in use
2. System asks the user to either log in or provide an alternative email-address
3. Use case resumes at step 4

5a. User didn't provide a valid matriculation number (format: xx-xxx-xxx):

1. System will prompt the user to provide a valid matriculation number
2. User enters a valid matriculation number
3. Use case resumes at step 5

1-7a User cancels the registration:

1. System cancels the registration process and discards all entered data

Special Requirements:

User Registration at step 9 cannot take more than two seconds.

Matriculation number needs to be in format: xx-xxx-xxx

Email address needs to be a @students.unibe.ch

The username is firstname.lastname

Notes:

-

4.2 Sign-In

Actors:

Students and Tutors (Users)

Description:

As a user, I want to log into the platform so I can see and edit my profile, view more sensitive data and engage with other users.

Trigger:

User visits the website and completes the login form.

Pre-conditions:

1. User is not already logged in
2. User has an active account

Post-conditions:

1. User is logged into the System

Main Scenario:

1. User visits the website
2. User fills out the login form (Username and password)
3. User clicks the login button
4. System validates the provided information
5. System logs the user in
6. System removes the login form and shows a Sign-Out link instead

Alternative Scenarios:

4a. User provides false credentials:

1. System displays an error message
2. System prompts the user to provide valid credentials
3. Use case resumes at step 2

Special Requirements:

Credential Validation and Login (step 4 and 5) cannot take more than two seconds.

Notes:

1. Should the login procedure be in AJAX so the page doesn’t have to be reloaded? No
2. How long is an active user session (session timeout) ?30 minutes

4.3 Sign-Out

Actors:

Students and Tutors (Users)

Description:

As a User I want to logout of the platform.

Trigger:

User clicks on the logout link.

Pre-conditions:

User must be logged in and have an active session.

Post-conditions:

User is logged out of the platform and can only use it the same way as an unregistered user.

Main Scenario:

1. User clicks on the logout link
2. System logs the user out
3. System removes the logout link and displays the login form instead

Alternative Scenarios:

2a. The User is already logged out because his/her session has expired.

1. Use case resumes at step 3

Special Requirements:

-

Notes:

-

4.4 Edit Profile (Student)

Actors:

Students

Description:

As a student I want to be able to change my profile information.

Trigger:

Student views their profile and clicks the “Edit profile ” link.

Pre-conditions:

1. Student is logged into their account
2. Student is on the view of their profile

Post-conditions:

1. Student has changed profile information
2. Student is back on the view of their profile

Main Scenario:

1. Student visits their view profile page
2. Student clicks on “Edit profile” link
3. System toggles profile display mode to editing mode, transforming labels into text fields
4. Student can now change the following information: Profile description, first and last name, matriculation number, address(street, Zip, city), phone number, password
5. Student can enter a link to his profile picture.
6. The student can also delete courses out of the list of his courses.
7. System validates entered information during editing
8. Student clicks “Save” button
9. System persists updated information into the database
10. System toggles profile editing mode back to viewing mode

Alternative Scenarios:

6a. The student does not have any courses in his list yet and wants to add some:

1. The student hits the “Add courses” button
2. The system navigates the student to the chooses courses page
3. The use case resumes as the use case 4.6 “Choose course”

7a. Student provides false credentials:

1. System displays an error message next to the text field containing false information
2. System prompts the Student to provide valid information
3. Student provides accordingly
4. Use case resumes at step 7

8a. Student aborts editing:

1. System discards all changes
2. Use case resumes at step 10

8b. Student hits delete profile link:

1. We continue at use case 4.13 delete profile step 2

Special Requirements:

Persisting data and returning to view mode (step 9 and 10) cannot take more than two seconds.

Notes:

-

4.5 Edit Profile (Tutor)

Actors:

Tutors

Description:

As a Tutor I want to be able to change my profile information.

Trigger:

Tutor visits his/her profile and clicks the “Edit profile” link.

Pre-conditions:

1. Tutor is logged in with his/her account
2. Tutor is on the view of his profile

Post-conditions:

1. Tutor has changed profile information
2. Tutor is back on the view of his profile

Main Scenario:

1. Tutor views their profile page
2. Tutor clicks on “Edit profile” link
3. System toggles profile display mode to editing mode, transforming labels into text fields
4. Tutor can now edit following information: first name and last name, matriculation number, address(street, Zip, city), phone number, password

5. Tutor can insert link to his profile picture.
6. In the list of his chosen courses the tutor can edit, his grade, his fee, and when he passed the exam. He can also delete courses from his list
7. System validates entered information during editing
8. Tutor clicks “Save” button
9. System persists updated information into the database
10. System toggles profile display mode back to viewing mode

Alternative Scenarios:

6a. The tutor does not have any courses in his list yet and wants to add some:

4. The tutor hits the “Add courses” button
5. Use case continues as use case 4.6 “Choose course”

7a. Tutor provides false credentials:

1. System displays an error message next to the text field containing false information
2. System prompts the Tutor to provide valid information
3. Tutor provides accordingly
4. Use case resumes at step 7

8a. Tutor aborts editing

1. System discards all changes
2. Use case resumes at step 10

8b. Tutor hits delete profile link:

2. We continue at use case 4.13 delete profile step 2

Special Requirements:

Persisting data and returning to view mode (step 9 and 10) cannot take more than two seconds.

Notes:

-

4.6 Choose course from database

Actors:

Students and Tutors (Users)

Description:

As a User, I want to be able to choose the courses I need (Students) or offer (Tutors) tutoring.

Trigger:

User is editing his/her profile and clicks “add courses”.
Or the user visits the page ”Courses” and selects a course to add to their profile.

Pre-conditions:

1. User is logged in
2. Is on the courses page

Post-conditions:

1. User has added courses to their profile.

Main Scenario:

1. User is on courses page
2. User makes his/her choices
3. User clicks “Add” Button
4. The system adds the course to the users database and profile
5. The user remains on the “Courses” page and can add further courses if desired
6. When the user has added all the courses he wants to add he can return to his profile by clicking the “Profile” button.

Alternative Scenarios:

3a. User does not want to add any courses:

1. User hits any button other than the add courses or courses button
2. User leaves the courses page

Special Requirements:

Fetching from and persisting course selection data into the database cannot take longer than two seconds.

Notes:

Once a user has already selected a course the add button for that course disappears.

4.7 Enter grades

Actors:

Tutors

Description:

As a Tutor, I want to enter my grades for my courses so I can display my expertise.

Trigger:

Tutor is editing his/her profile and clicks “Edit grade” on one of his/her courses.

Pre-conditions:

1. Tutor is editing his/her profile
2. Tutor has already selected at least one course

Post-conditions:

1. Tutor has entered a grade for his/her course

Main Scenario:

1. Tutor edits his/her profile
2. Tutor clicks “Edit grade” on one of his/her courses
3. System allows Tutor to enter a grade for the course
4. Tutor confirms the grade by pressing the check sign
5. System validates grade formatting
6. The user is back on the edit of his profile

Alternative Scenarios:

4a. Tutor discards changes by click or by pressing the x sign

1. System discards changes
2. The user is back on the edit of his profile

5a. Tutor enters invalid grade formating

1. System displays error message
2. System prompts Tutor to provide correct formatting
3. Use Case resumes at step 4

Special Requirements:

-

Notes:

Grades can be entered freely and can’t be officially validated. See Assumptions.
To finalize the saving of the grade the tutor continues with the use case 4.5”Edit profile(tutor)” step 8

4.8 Contact staff for help

Actors:

All website visitors (Visitors)

Description:

As a Visitor, I want to be able to contact the website's staff in case I need help or want to leave a comment.

Trigger:

Visitor is on the contact page and fills in the contact information

Pre-conditions:

Visitor is on contact page

Post-conditions:

A message is sent to the staff describing the visitor's problem/concern/review.
Visitor is on confirmation page.

Main Scenario:

1. Visitor clicks on "Contact" link
2. Visitor enters his/her e-mail into the "Email" field
3. Visitor enters subject
4. Visitor describes the problem
5. Visitor clicks "Send" button
6. System sends a message to the staff
7. System redirects the visitor to the send confirmation page.

Alternative Scenarios:

3a. Visitor is already logged in as Student or Tutor (User)

1. System automatically completes the "Email" field with the User's data
2. Use Case resumes at step 3

6a. Visitor hits cancel button or aborts the Use Case otherwise

1. System discards entered information

Special Requirements:

The redirecting to the confirmation page can not take more than 2 seconds.

Notes:

-

4.9 Communication

Actors:

Students and Tutors (Users)

Description:

To set up a meeting with a tutor the student gets into contact with the tutor. He does so through a internal communication system on the website. Through a button on the tutor's profile the student can write him a message and when the tutor replies the student receives a message in his inbox.

Trigger:

A student is on a tutor's profile and decides to send him a message by hitting the "Send message" button.

Pre-conditions:

The student must be signed into his account on the website.
The tutor must have a profile, which he has once he creates an account.
The student and the tutor are not in contact yet.

Post-conditions:

The student and the tutor are now in contact and can set up meetings.

Main Scenario:

1. A student is on a tutor's profile and hits the send message button.
2. The student is redirected to the "Create message" page where he can enter his message and subject to the tutor
3. The sender and receiver are inserted automatically
4. The student then hits the send button.
5. The tutor receives the message in their inbox and can read it.
6. The tutor hits the reply button, which allows him to reply to the students message.
7. The tutor hits the the send button.
8. Now the student has a new message in their inbox and the steps 4 to 6 can be repeated over and over.

Alternative Scenarios:

4a. Student/ tutor decides to cancel the message:

1. Student/tutor hits the cancel button on the “Create message” page
2. The message content is discarded
3. The student/tutor is back to the view of the original message

4b. Student/Tutor decides to hit the return to inbox button:

1. Student/tutor is redirected to their inbox
2. Any entered data is discarded

4c. Tutor hits the set up contract button:

1. The use case resumes with the use case 4.10 Set up contract

Special Requirements:

After pushing the send message button, the user will be redirected to the “Create message” page after a maximum of 2 seconds.

Notes:

-

4.10 Set up a contract

Actors:

Students and Tutors (Users)

Description:

Once the student and tutor have been in contact(by messaging) and they have decided on a first day to meet up, the tutor fills out a form which then serves as contract between the student and the tutor but also the project owner. With agreeing to this contract the student makes the payment, of which a percentage goes to the project owner and the rest to the tutor as a guaranty.

Trigger:

A tutor hits the set up contract button on the received message from a student and fills out a contract form.

Pre-conditions:

Both student and tutor have an existing account on the platform and the tutor is signed in.
The student and the tutor have been in contact and have agreed on a first meeting.

Post-conditions:

The student pays for the first meeting.
A contract for the first meeting has been set up.

Main Scenario:

1. A tutor and a user have decided to set up a contract.
2. The tutor hits the set up contract form in a message between him and the student.
3. The system automatically fills in the name of the tutor and the student.
4. The user provides all necessary information: course, fee, date of first meeting(day, time), message(if desired)
5. The system validates the provided information
6. The tutor clicks the send offer button.
7. The systems sends out the offer in an message to the student
8. The student confirms the contract(more detailed in use case 4.16)
9. The system sends out a request to the third party payment system, which checks if the student’s payment has been made.
10. The third party payment service notifies the system that the payment has been successful

Alternative Scenarios:

6a. Tutor decides to cancel the contract:

4. Student/tutor hits the cancel button in the set up contract form
5. The entered date is discarded
6. The tutor is back to the message where he first hit the set up contract button

6b. Tutor decides to hit the return to inbox button:

1. Tutor is redirected to their inbox
2. Any entered data is discarded

9a. The payment is not successful

1. The systems sends an email to the tutor and student saying that the contract has been declined

Special Requirements:

Payment process is started automatically, once the student agrees to the contract.
The validation of the information should take no more than 2 seconds.

Notes:

-

4.11 Finding a tutor

Actors:

Students and Tutors (Users)

Description:

As a student i want to find a good tutor,to find these tutors the website offers suggestions to me on the find tutor website,once i have chosen the courses for which i need help. This saves the student time by offering him ideal options.

Trigger:

A student chooses to look for a tutor by going to the look for tutor page.

Pre-conditions:

Both student and tutor have an existing account on the platform and the student is signed in.
The student has chosen a course out of the catalogue and put it on his profile.
The student is on the look tutor page.
The tutor offers tutoring for that course.

Post-conditions:

The student gets to look at different tutors profile and gets to see which one is fitting for him.
If he chooses this tutor he can send him a message.

Main Scenario:

1. A student chooses to go to the search for tutor page.
2. There all the students courses and the tutors available for those courses are listed.
3. We have the following information about the tutor available: exam passed in, grade, fee, city
4. Student can then choose to sort the table by the different attributes
5. The student chooses a tutor and clicks on the go to profile link
6. The system takes the student to the profile of the tutor
7. The student browses through the profile
8. The student hits return and looks at the table again
9. Student repeats steps 6-9 until he finds a tutor that suits him

Alternative Scenarios:

10a.The student can not find a tutor that matches

1. Student hits refresh button
2. System updates tutor table
3. Use case resumes at step 3

Special Requirements:

Loading table can take up to 5 seconds maximum

Notes:

-

4.12 Report false information

Actors:

Students and Tutors (Users)

Project owner
The people who own and host the website are in charge of dealing with technical problems and false information. Also they are in charge of deleting profiles.

Description:

Since the information on the profiles of the students and the student is not verified, some might be tempted to post false information. For that purpose we include a report false information button on every profile, so the project owner can intervene. That way the profiles are more reliable.

Trigger:

A student or tutor chooses to report a profile, by clicking the report button.

Pre-conditions:

The student/tutor needs to know what information is false.
The student/tutor must have an account and be signed in.

Post-conditions:

The profile with the false information is reported.
The project owner acts as he sees it, by removing the false information or even deleting the profile

Main Scenario:

1. A user sees false information on a profile and decides to report it
2. The user hits the report button
3. The system redirects the user to the contact page
4. The user enters the exact detail of the false information
5. The user hits send
6. The system redirect the user to the confirmation page

Alternative Scenarios:

4a. The user decides to abort the report:
1. User hits the cancel button
2. The system discards all the entered information

Special Requirements:

User should be redirected within 2 seconds

Notes:

-

4.13 Delete profile

Actors:

Students and Tutors (Users)

Project owner
The people who own and host the website are in charge of dealing with technical problems and false information. Also they are in charge of deleting profiles.

Description:

If a user no longer needs the services of the website and decides to delete his account they can do so by hitting the deactivate button on the “Edit profile” page.

Trigger:

A user chooses to delete his profile and hits the deactivate button in the edit of his profile.

Pre-conditions:

The user must have an account and be signed in.
The user is on the “Edit profile” page.
The user does not need the services of the website anymore.

Post-conditions:

The profile of the user will be deactivated and marked as such in the profile.

Main Scenario:

1. A user wants to delete his profile and hits the delete profile button
2. User is asked if he really wants to delete the profile.
3. User hits the yes button
4. The system deactivates the profile
5. The project owners change their statistics

Alternative Scenarios:

3a. The user decides not to delete the profile:

1. User hits the cancel button when asked if he really wants to delete the profile
2. The user is taken back to the “Edit profile” page

Special Requirements:

-

Notes:

-

4.14 Change password

Actors:

Students and Tutors (Users)

Description:

If a user wishes to change his password he can do so while signed in and editing his profile.

Trigger:

A user is on the editprofile page and changes the data in the password field.

Pre-conditions:

The user must have an account and be signed in.
User must be editing his profile.

Post-conditions:

The user has changed his password and next time he signs in he needs to use the new password.

Main Scenario:

1. A user is editing his profile
2. User enters new password in password field
3. User hits the save button
4. User is back on the view of their profile

Alternative Scenarios:

3a. User aborts editing password:

1. System discards all changes
2. User leaves the edit profile page

3b. User hits delete profile link:

1. We continue at use case 4.13 delete profile step 2

Special Requirements:

-

Notes:

-

4.15 View Profile

Actors:

Students and Tutors (Users)

Description:

As a user I want to be able to look at profiles, including mine.

Trigger:

User visits his/her profile.

Pre-conditions:

User is logged into his/her account

Post-conditions:

User has looked at their profile information

Main Scenario:

1. User visits his/her profile page
2. User looks at the information on his profile(First name, last name, address(street, Zip, city), phone number, profile picture, chosen courses, friends).
3. User has the option to click the “Edit” button or the “Look for courses” button.
4. User has looked at all his information.
5. User leaves view profile page by hitting any button or link.

Alternative Scenarios:

-

Special Requirements:

A profile is subject to three privacy states. The first privacy state only shows general information to guests, whereas the second privacy state shows some contact details only to registered users. The third and most sophisticated privacy state is achieved by becoming friends with the user the profile belongs to and displays all available information.

Notes:

-

4.16 Accept contract offer

Actors:

Students

Description:

As a student I can decide whether i agree to the terms of the contract the tutor sent me or whether i want to decline the offer.

Trigger:

Student checks his inbox and sees an offer, he clicks on the offer and is shown the offered contract.

Pre-conditions:

Student logged into their account.
Student has an offer from a tutor in his inbox.

Post-conditions:

Student either accepted the offer and now has a contract with a tutor and pays them through the third party provider.
Or student declines offer and is back in his inbox, no contract has been made.
Once the payment has been made over the third party provider, the tutor and the student are friends and it is displayed on their profiles

Main Scenario:

1. Student clicks on offer in inbox
2. The system opens up the offer, so the student can view it.

3. The student agrees with the terms of the offer.
4. Student clicks the accept button
5. The system send a payment request to the third party payment service.
6. The system redirects the student to the confirmation page, from which they can return to their inbox.

Alternative Scenarios:

4a. Student hits reply button:

1. The system opens up a message field
2. Student can type his message to the tutor
3. The contract is not accepted

4b. Student hits return to inbox:

1. System redirects the student back to his inbox
2. The contract is not accepted

4c. Student hits cancel button or aborts otherwise:

1. The system redirects the student back to the inbox or which ever other navigation link he clicked
2. The contract is not accepted

Special Requirements:

An offer can only be accepted once.

Notes:

-

5 Project order

A web-application for interacting students and tutors is developed.

The solution is available on the 9th December 2015.

The requirements are detailed available on our ESE2015-team2 board on trello.

The project contains any implementations and changes for all correlating systems and all user interface and database applications specified on the ESE2015-team2 board on tello.

6 Trade-offs and conflict potential

The following elements could lead to trade-offs or conflicts, these points have to be discussed in an early phase with all stakeholders:

- How are changing courses managed?
- How do you ensure the truthfulness of grades?
- How do you prevent students from paying tutors directly at their physical meeting?
- Who will include new courses into the system?

- How is the tutor’s expertise measured and validated? Are grades the only indicator?

7 Assumptions

- The finished platform will be incorporated into the developer’s server farm
- The developer will provide the necessary infrastructure, alleviating the project owner from high initial capital investments
- The developer will maintain the platform and continue development after release
- With no initial design specifications, the developer will provide prototypes and iteratively enhance the design towards the finished product
- During development, the developer will strife to always have a representative, running version of the platform available to the customer on http://digital-front.ch:8880/ESET2/index.xhtml
- With no further mechanics in place, the platform heavily relies on trust regarding tutor’s expertise and payments between students and tutors.
- Tutors can not take the role of students on website at the same time
- The project owner is not allowed to own payment information of tutors, the payment service provider can identify tutors at the time of payment
- Payment is provided automatically at time of payment to project owner
- The project owner does not keep records of payments
- Passwords are encrypted

8 Requirements

8.1 Priority of requirements (must / should / nice to have)

The priorities are done according must / should / nice to have model as far as possible.

Requirements with priority MUST, have to be in the release.

Requirements with priority SHOULD, have also to be in the release, however it's possible to use manual workarounds to achieve the requirement.

Requirements with priority NICE TO HAVE can be thrown out in case of missing resources.

The priorities are listed and managed directly on the trello ESE2015-team2 board.

8.2 Hints

The requirements of the SRS are structured according topic and use case on Trello:
Every use case card on trello has in its description a list of linked requirement cards. These are the resulting requirements of the use case.

8.3 List of requirements

The list of the requirements is managed directly on the trello ESE2015-team2 board.

9 Risk management

9.1 Project risks

ID	Risks	Probability[1]	Impact[2]
1	Missing project resources	frequently	high
2	User journeys are not clear / Interface isn't intuitive	possible	high (reduces the customer experience)
3	Missing of deadlines	possible	undefined

9.2 Project risks - plan

ID	Risks	Massnahmen
1	Missing project resources	Work as a team, bottlenecks have to be shown as early as possible!
2	User journeys are not clear / Interface isn't intuitive	For every use case concrete user journeys are defined and tested.
3	Missing of deadlines	Communication and organisation is the key here..

9.3 Product Risks

ID	Risks	Probability	Impact
1	Customers get in contact but do not pay the fee.	occasionally	medium

[1] Probability: occasionally / possible / frequently / occurs

[2] Impact : small / medium / high / catastrophic