

System Specification

Nachhilfebörse HTL Leonding  
  
  
Danijal Orascanin  
Eva Pürmayr  
  
4AHIF  
2016/17

Content

[**Initial Situation and Goal**](#_jjekvdybyd20)[**2**](#_jjekvdybyd20)

[Initial Situation](#_n8ahulnkju5h) [3](#_n8ahulnkju5h)

[Application Domain](#_htqsu5topy7j) [4](#_htqsu5topy7j)

[Glossary](#_u5hpi6nokaeo) [4](#_u5hpi6nokaeo)

[Model of the Application Domain](#_v3hfvsh3ebj0) [5](#_v3hfvsh3ebj0)

[Overview of the Business Processes](#_vubs1hevlx62) [6](#_vubs1hevlx62)

[Description of the Business Processes](#_egqp0it2y9xj) [7](#_egqp0it2y9xj)

[Goal Definition](#_8fagnwgsr5tj) [8](#_8fagnwgsr5tj)

[**Functional Requirements**](#_yeq9ys48yb63)[**9**](#_yeq9ys48yb63)

[Use Case Diagrams](#_pdlgvadmdzmr) [10](#_pdlgvadmdzmr)

[GUI to call the use case](#_vdxtooz8yvt3) [14](#_vdxtooz8yvt3)

[GUIs for the standard use](#_62mth2z42ghn) [16](#_62mth2z42ghn)

[Scenarios for non-standard uses (bad cases or work around cases)](#_4bbgzyjfwat) [17](#_4bbgzyjfwat)

[GUIs for the non-standard uses](#_js9chb4tx0nq) [20](#_js9chb4tx0nq)

[Workflow](#_tyry1nrwp0p6) [21](#_tyry1nrwp0p6)

[Open Points](#_rkpewzma0qvk) [22](#_rkpewzma0qvk)

[**Non-functional Requirements**](#_ux1r8r4sav3c)[**23**](#_ux1r8r4sav3c)

[**Quantity Structure**](#_b3capvkgtyym)[**26**](#_b3capvkgtyym)

[**System Architecture and Interfaces**](#_xeh6yhwp8ozt)[**26**](#_xeh6yhwp8ozt)

[**Acceptance Criteria**](#_9bl1ffs2vyc1)[**27**](#_9bl1ffs2vyc1)

[**List of Abbreviations**](#_hnyo0yi553iw)[**30**](#_hnyo0yi553iw)

[**References**](#_ch9dkran1jaz)[**30**](#_ch9dkran1jaz)

[**List of Figures**](#_odg4471nf0tx)[**30**](#_odg4471nf0tx)

# Initial Situation and Goal

## Initial Situation

At the HTL Leonding the organization and handling of a tutor exchange platform is under the responsibility of Ms. Keck. Her main task is to collect students who are willing to offer coaching services to other students.

Unfortunately this system has a number of shortcomings:

1. No images on the contact list -> students are sometimes too shy to ask for the chosen tutor in front of the whole class -> contact barrier

2. System is outdated, inefficient and time-consuming for the administrator

3. Displaced and ancient form of publication. Students who need coaching are mostly not aware that there is a tutoring market

4. No quality assurance of the data. The information on the list might be incorrect and everyone can get personal information about the tutors

5. No other information, like availability times, minimum remuneration etc.

*Available Software in the market*

There are various products in the market, here are three of them:

* **talentify.me**: An austrian-wide tutoring market, with an attractive web-design. One clear disadvantage of this platform is the lengthy sign-up procedure where a student id, a user name, and an extra password has to be provided.
* **lernquadrat.at**: A tutoring market for Austria, Germany and Switzerland, which offers plenty of possiblities of tutoring (single & group tutoring, crash courses) in a lot of convenient locations. One advantage of this website is the easy registration process, but on the other hand the user cannot choose his/her teacher and the design is not appealing.
* **www.betreut.at/nachhilfe**: An austrian-wide tutoring offering website. Clear disadvantages of this website are that the searching criterias are not specialised on tutoring (the user has to enter details of his/her family when he/she wants to sign up) and the website has a few bugs.

### Application Domain

In order to to collect all pupils, who want to give tutoring lessons, Mrs Keck hands out a list to the class teachers where the pupils can write down their name, subjects, class and E-Mail. This information will be entered in a Microsoft Access Database, where all tutors with their personal data are saved. Afterwards the list appears in the internal area of the Htl Leonding website in the Nachhilfebörse tab. Additionally the list is printed and then hung up in every first-grad-class and in front of the room E33.

When students want to address a possible coach they have to consult the list, find the student in his/her class or contact him/her via email and agree on a price and meeting times.

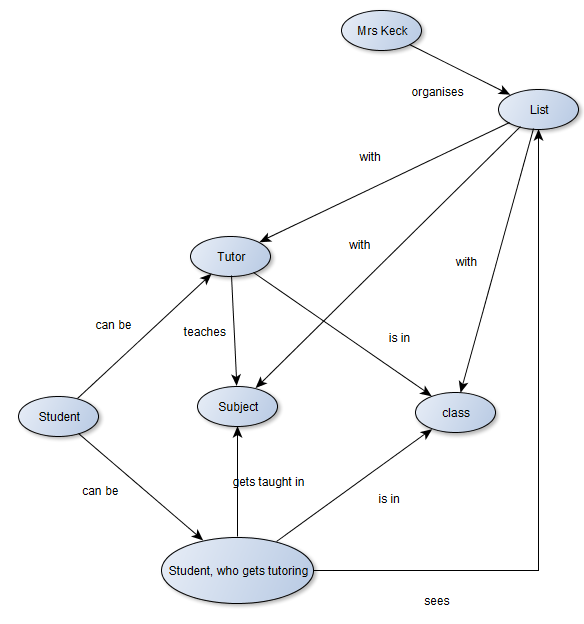
### Glossary

Tutoring: A student helping another younger student in a certain subject

Tutoring market: A place where students, who are searching for tutors, can find an appropriate tutoring teacher

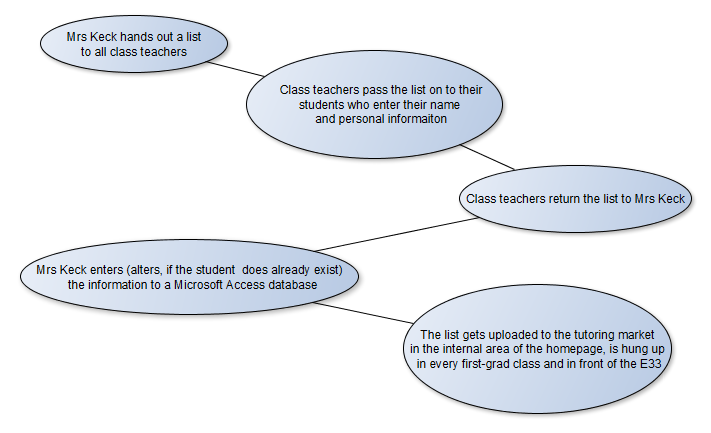
### 

### Model of the Application Domain

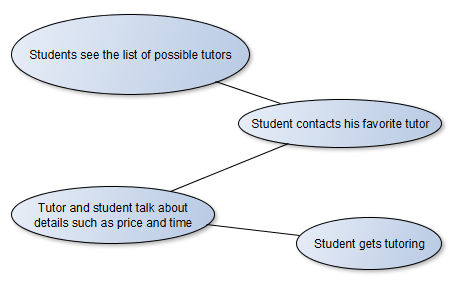


### Overview of the Business Processes

* Collect all tutors



* Students find an appropriate tutor



### 

### Description of the Business Processes

Collect all tutors

|  |  |
| --- | --- |
| Triggering Event: | Beginning of the school year |
| Result: | A complete and up-to-date list of tutors |
| Contributors: | Mrs, Keck, Class teachers, Tutors |

Student finds tutor

|  |  |
| --- | --- |
| Triggering Event: | Need of a tutor |
| Result: | A fitting team of a student and a tutor who improve the student’s grades |
| Contributors: | Found tutor, student |



## Goal Definition

The main idea of our project is to ease the process for searching for the right tutor and also decrease the work hours for the responsible persons.

If a student wants to use our website he/she first has to login with his/her school account, because the data privacy of our students if very important to our headmaster.

After logging in the student has to decide whether he/she wants to take or give tutoring. If he/she wants to give tutoring he/she has to make a detailed profile with his/her name, a picture of him/her, his/her department, his/her grade, the subjects he/she wants to give tutoring in and more information about himself/herself. If he/she wants to take tutoring lessons he/she can [immediately](https://www.dict.cc/englisch-deutsch/immediately.html) start searching for the right teacher.

On the technical side, the tutors data gets saved in a database. When the user now starts to filter the teachers to find the right one he/she always gets a refreshed list of possible matching tutors. This list can be filtered by department, grade, subject, rating or remuneration and then the list can be sorted by grade, remuneration or rating.

The user also can send the tutor a request via email or SMS.

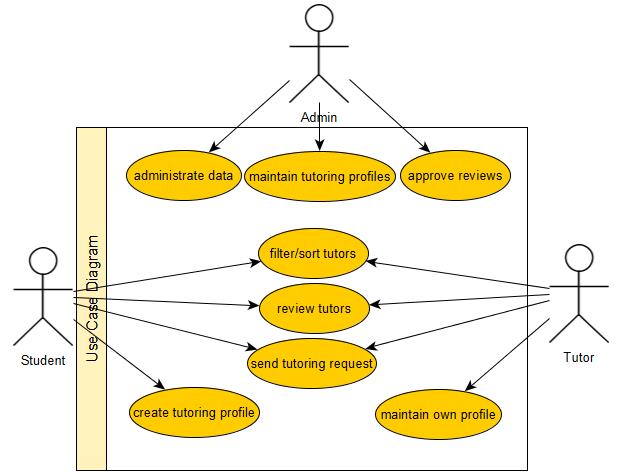
We also want to implement a rating system so every student can rate his tutor. The user can rate the tutor with 1-5 books (5 books is the best) and with a comment, which is required. With this function we want to differ good from bad teachers. Furthermore it helps students to get a more precise impression of the tutoring teacher. Before the ratings will actually appear on the website, the administrator(Mrs Keck) will have a look on the ratings and check if there are no fake or insulting ratings.

Of course our website is going to be responsive so the students can use it on their phones too.

# Functional Requirements

## Use Case Diagrams





* 1. Use-Case Details

**Administrate data:** The admin account maintains the master data for example if the subjects change in a few years.

**Maintain tutoring profiles:** The administrator is going to be able to edit the tutors profiles (only the subjects can be edited) and also delete whole profiles if it is a fake account or inappropriate or inactive or the student is not anymore at school.

**Approve reviews:** The administrator can approve or reject reviews. He will be able to delete ratings to pretend insulting or inappropriate ratings .

**Filter/Sort tutors:** The user can filter the list of teachers by department, grade, subject, gender, rating or remuneration. He also can sort the list by grade, remuneration or rating.

**Review tutors:** The users can also rate their tutors with 1-5 books (5 books are the best) and a comment, which is required.

**Send tutoring request:** The user can send a request to the tutor via email or if indicated via sms.

**Create tutoring profile:** If a student wants to become a tutor he has to create a profile. He has to enter Information about himself like his class, year, name, expected wage, time, teachers, email, age, subjects and a picture.

**Maintain own profile:** The tutor can edit his own profile if some of his informations have changed. Of course he is also able to delete his profile.

**Characteristic Information**

|  |  |
| --- | --- |
| Superior business process: | 1: Administrate data |
| Goal: | Master data is up to date |
| Precondition: | Master has to be changed |
| Postcondition: | Master data (e.g. available subjects) are up to date |
| Involved User: | Administrator |
| Triggering Event: | Available subjects have changed |

|  |  |
| --- | --- |
| Superior business process: | 2: Maintian tutoring profiles |
| Goal: | Incorrect or inappropriate information of a tutor must be edited; tutor’s profile is deleted |
| Precondition: | Information (e.g. tutor’s subject) of the tutor must be changed (e.g. a teacher told the administrator that the tutor is not good enough to be a tutor) or the tutor left the school |
| Postcondition: | The information about the tutor is correct |
| Involved User: | administrator, tutor |
| Triggering Event: | Tutor’s profile is inappropriate, Tutor left the school, recommendation against the tutor |

|  |  |
| --- | --- |
| Superior business process: | 3: Approve reviews |
| Goal: | Review is rejected/approved |
| Precondition: | A user rates a tutor |
| Postcondition: | Rating is shown on the tutor’s profile/Rating is deleted |
| Involved User: | Administrator, User, Tutor |
| Triggering Event: | User rates a tutor |

|  |  |
| --- | --- |
| Superior business process: | 4. Filter/Sort tutors |
| Goal: | A filtered/sorted list of tutors is displayed |
| Precondition: | The user wants to search a tutor by a certain criteria or wants to see a sorted list |
| Postcondition: | A sorted/filtered list of tutors appears |
| Involved User: | User |
| Triggering Event: | The user starts searching and wants a filtered/sorted list |

|  |  |
| --- | --- |
| Superior business process: | 5. Review tutors |
| Goal: | A rating of a tutor is saved in the database |
| Precondition: | A user wants to give feedback, so that other pupils can see the ratings of the tutor |
| Postcondition: | The rating appears on the tutor’s profile |
| Involved User: | User, admin (has to check the ratings) |
| Triggering Event: | User rates tutor |

|  |  |
| --- | --- |
| Superior business process: | 6. Send tutoring request |
| Goal: | The tutor gets a message, which tells him that a user is interested in tutoring |
| Precondition: | A user wants to contact a tutor |
| Postcondition: | Tutor is informed about the user’s interest |
| Involved User: | Tutor, user |
| Triggering Event: | A user wants to contact a tutor |

|  |  |
| --- | --- |
| Superior business process: | 7. Create tutoring profile |
| Goal: | A tutoring profile is created |
| Precondition: | A user wants to become a tutor |
| Postcondition: | The user has her/his own tutoring profile |
| Involved User: | User |
| Triggering Event: | The user want to become a tutor |

|  |  |
| --- | --- |
| Superior business process: | 8. Maintain own proprofilel |
| Goal: | The data of the tutor is up to date |
| Precondition: | Information of the tutor has changed |
| Postcondition: | The information is up to date again |
| Involved User: | Tutor |
| Triggering Event: | Personal information has changed |

### 

### GUI to call the use case

Scenario for the standard use (good case)

**Administrate data**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | admin | Choose master data |
| 2 | admin | Change master data |
| 3 | server | Save new data on server |

**Maintain tutoring profiles**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | admin | Choose a tutors profile |
| 2 | admin | Changes or deletes the profile |
| 3 | server | Saves the changed profile on server or deletes the profile from server |

**Approve reviews**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | admin | Choose a rating |
| 2 | admin | Approve/Deletes the rating |
| 3 | server | Admin approves the rating => server saves the rating  Admin deletes the rating => server deletes the rating from server |

**Filter/Sort tutors**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student / tutor | Choose a filter for the list |
| 2 | server | Filters the list of tutors |
| 3 | server | New filtered list gets displayed |
| 4 | student / tutor | Chooses a criteria for sorting the list |
| 5 | server | Sorts the list of tutors |
| 6 | server | New sorted list gets displayed |

**Review tutors**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student / tutor | Choose a tutor |
| 2 | student / tutor | Reviews the tutor |
| 3 | server | Save the review |

**Send tutoring request**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student / tutor | Choose a tutor |
| 2 | student / tutor | Sends a request |
| 3 | server | Sends the request |

**Create tutoring profile**

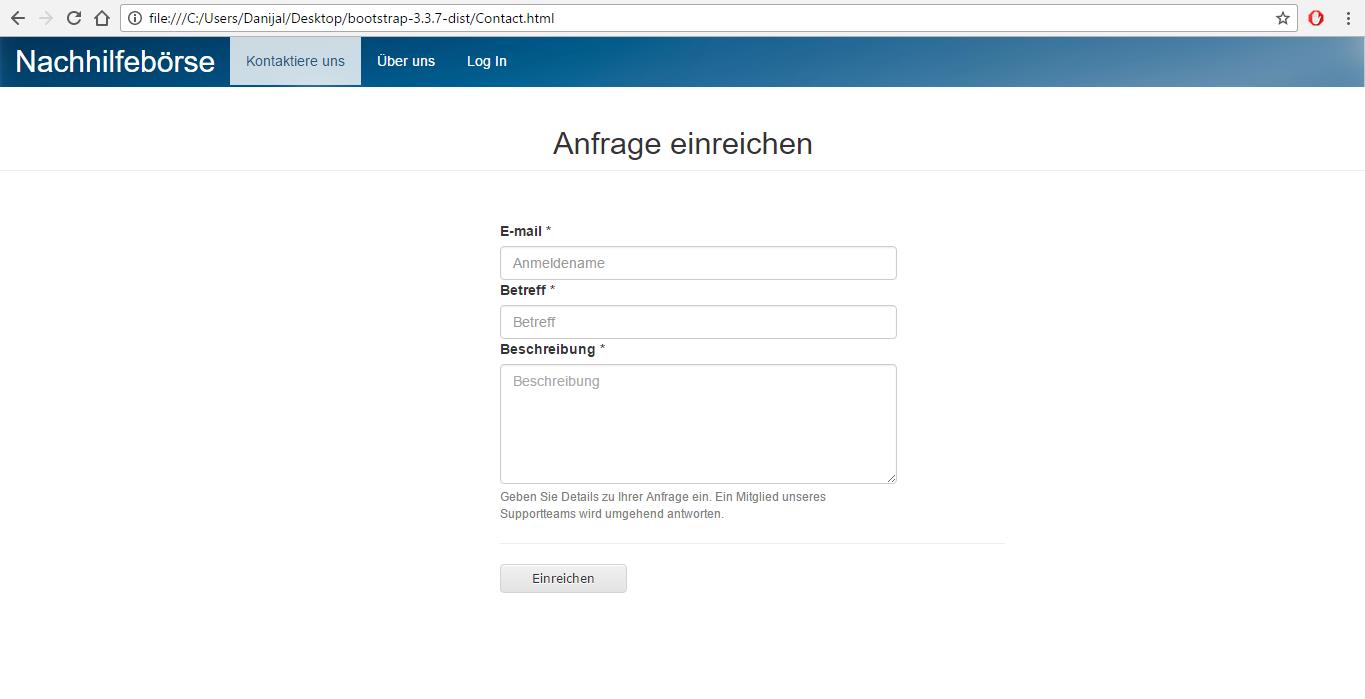
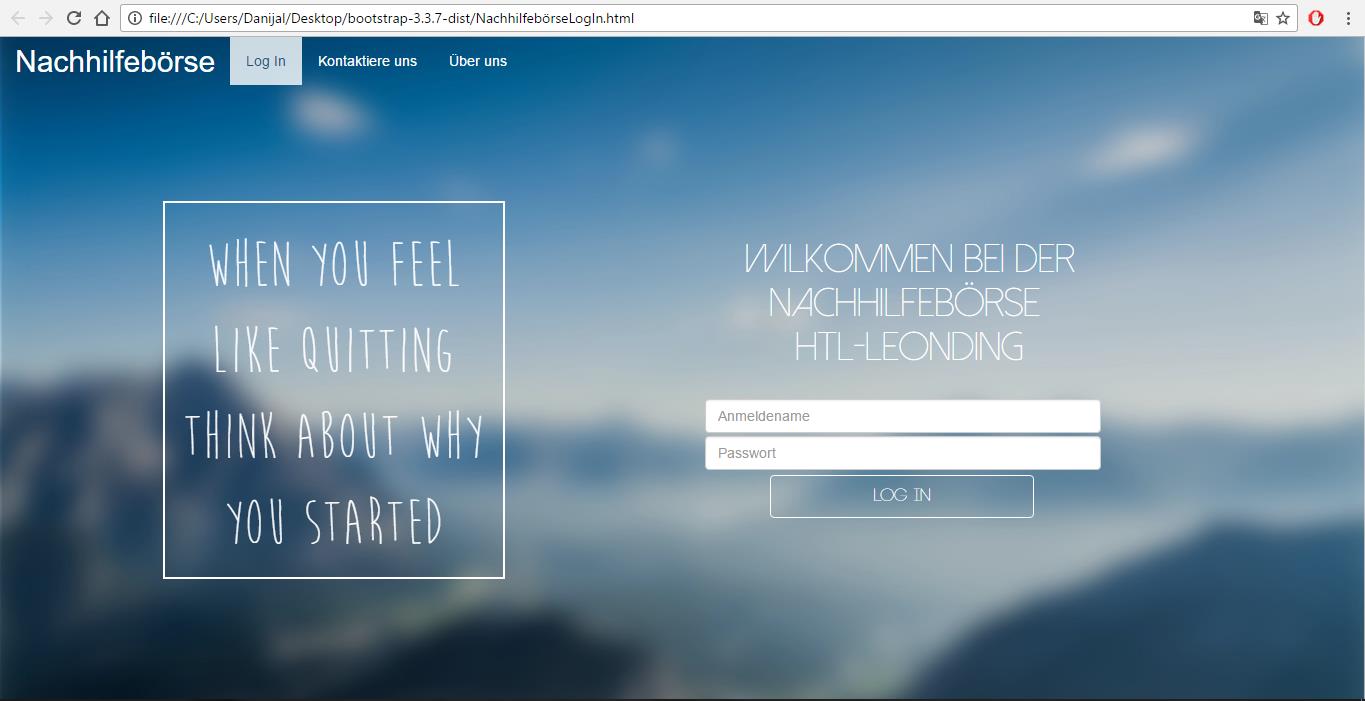
|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student | Fill in information |
| 2 | student | Press register button |
| 3 | server | Save tutor on server |

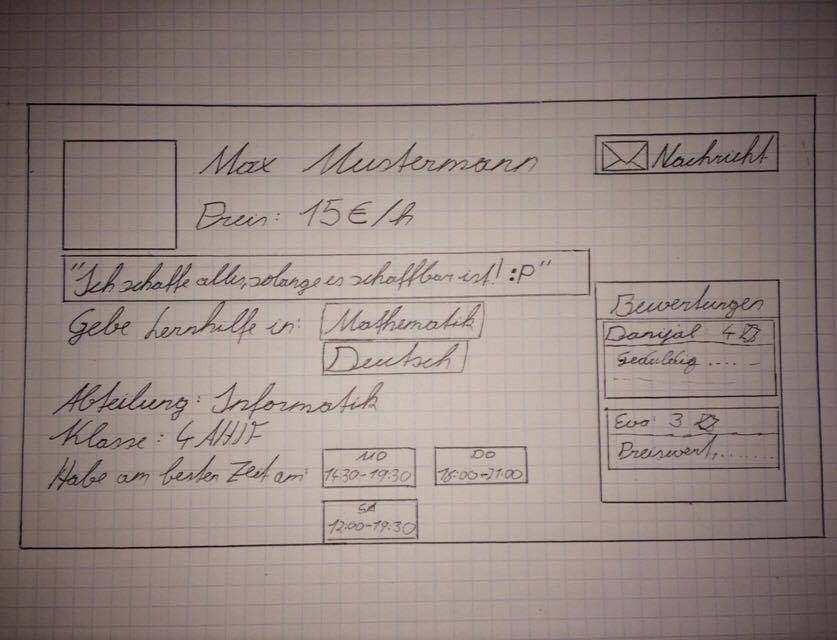
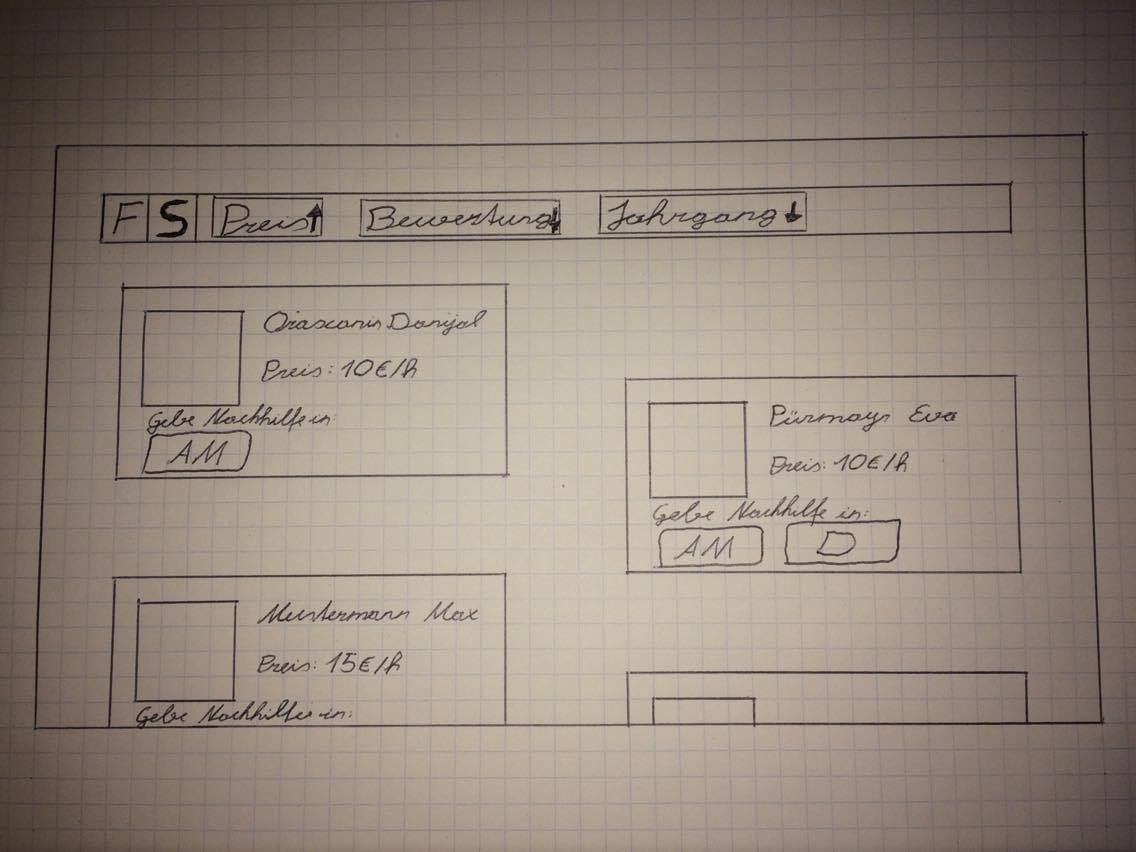
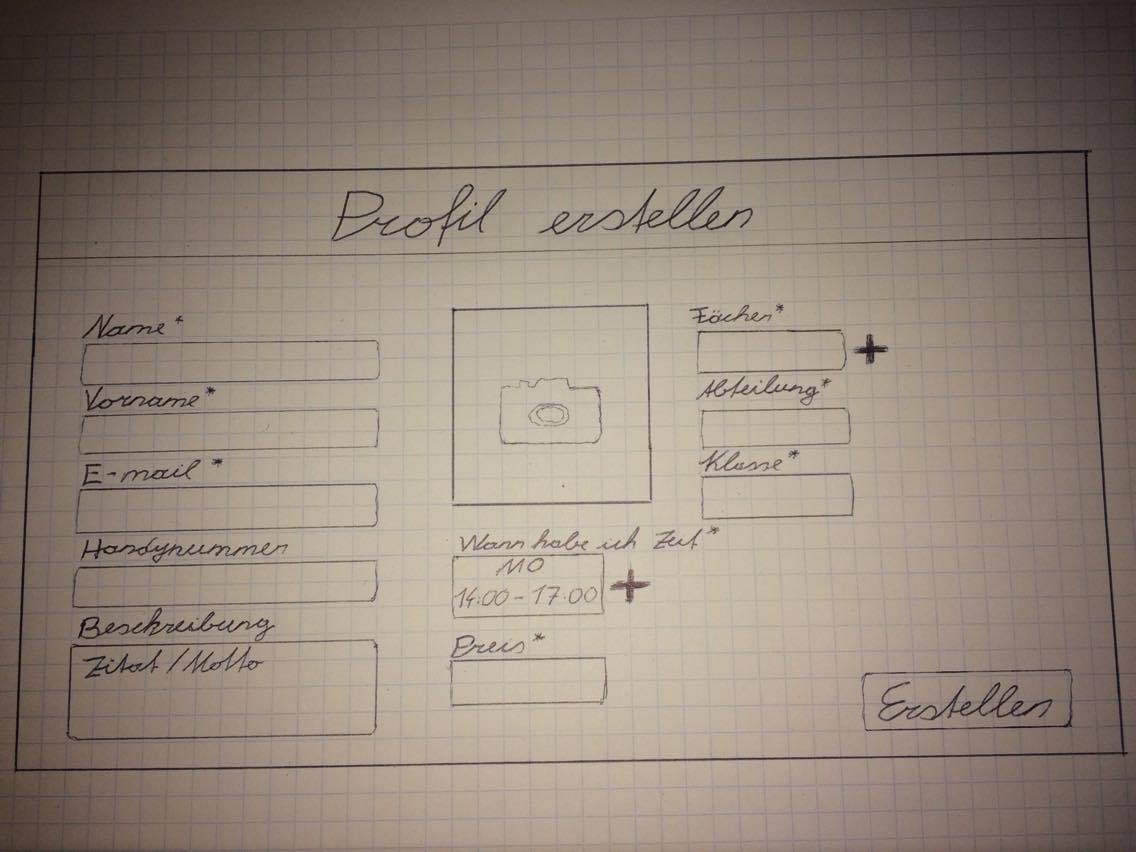
**Maintain own profile**

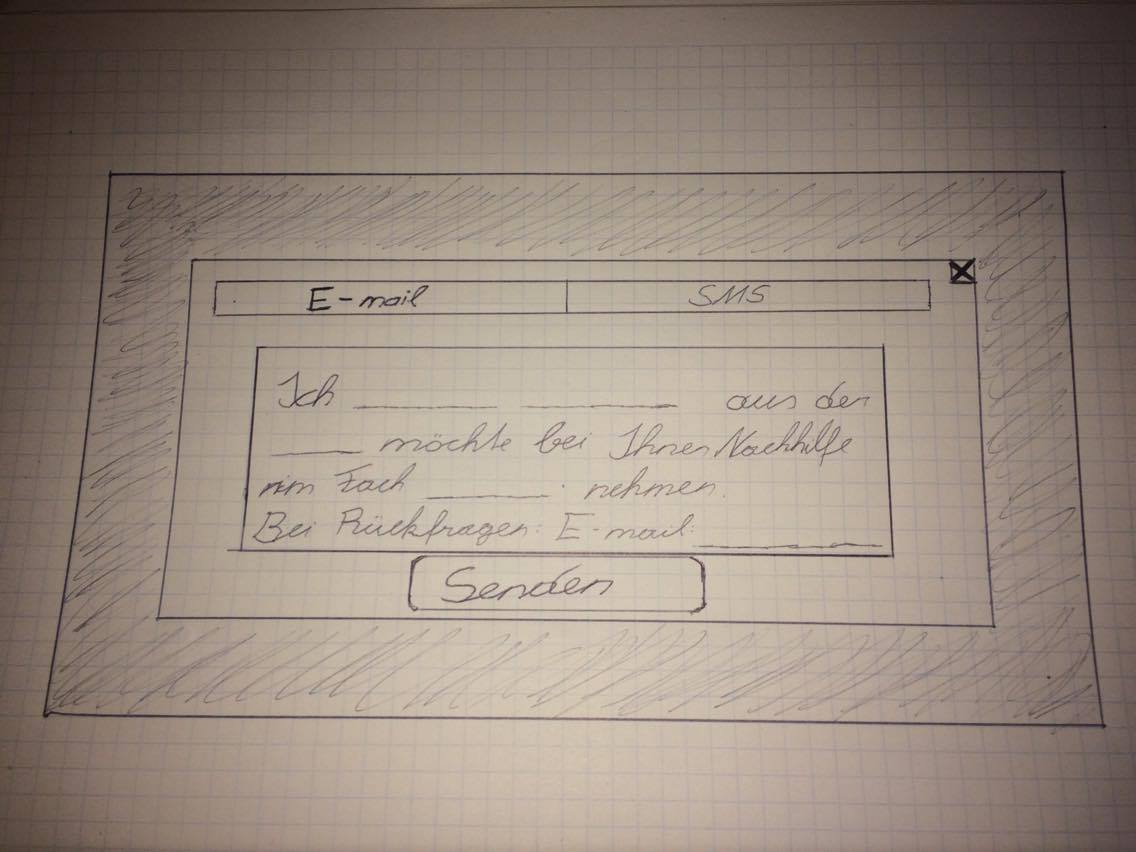
|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | tutor | Edit or delete profile |
| 2 | server | Saves the changed profile on server or deletes the profile from server |

### 

### GUIs for the standard use







### Scenarios for non-standard uses (bad cases or work around cases)

**Administrate data**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | admin | Choose master data |
| 2 | admin | Change master data |
| 3 | server | * Connection to server failed => the data will not be changed * Required fields are not filled => “Bitte füllen Sie die notwendigen Felder ein” * Timeout: Changes will not saved |

**Maintain tutoring profiles**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | admin | Choose a tutors profile |
| 2 | admin | Changes or deletes the profile |
| 3 | server | * Connection to server failed => the changes will not be saved * List of subjects is empty => “Bitte füllen Sie die notwendigen Felder ein” * Timeout: Changes will not be saved |

**Approve reviews**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | admin | Choose a rating |
| 2 | admin | Approves/Deletes the rating |
| 3 | server | * Connection to server failed => admin must approve/remove the rating again |

**Filter/Sort tutors**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student / tutor | Choose a filter for the list |
| 2 | server | Filters the list of tutors |
| 3 | server | * Connection failed => List will be displayed without the filter * No tutors in the filtered list => “Leider entspricht kein Nachhilfelehrer den vorgegebenen Kriterien” * “Filter” button is pressed but no criteria is selected => “Sie haben kein Filterkriterium eingegeben” |
| 4 | student / tutor | Chooses a criteria for sorting the list |
| 5 | server | Sorts the list of tutors |
| 6 | server | * Connection failed => List will be displayed, but unsorted |

**Review tutors**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student / tutor | Choose a tutor |
| 2 | student / tutor | Reviews the tutor |
| 3 | server | * Connection failed => The review will not be saved Timeout: Changes will not be saved * “Bewerten”-Button is pressed but amount of ‘books’ is not selected => “Bitte bewerten Sie den Nachhilfelehrer, indem Sie 1-5 Bücher vergeben” * “Bewerten”-Button is pressed but comment field is empty => “Bitte füllen Sie das Kommentarfeld aus” |

**Send tutoring request**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student / tutor | Choose a tutor |
| 2 | student / tutor | Sends a request |
| 3 | server | * Connection failed => Request will not be sent * Tutor did not enter his mobile phone number => Request can only be sent as an email * Student deleted the suggested text and the text box is empty => “Bitte geben Sie Ihre Nachricht ein. <Link>Mustertext wiederherstellen<Link> * Student did not enter his name/class/email/phone number => “Bitten füllen Sie die notwendigen Felder ein” |

**Create tutoring profile**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | student | Fill in information |
| 2 | student | Press register button |
| 3 | server | * Connection failed => Tutoring profile will not be created * Timeout =>Changes will not be saved * Missing information => “Bitte füllen Sie die notwendigen Felder ein” |

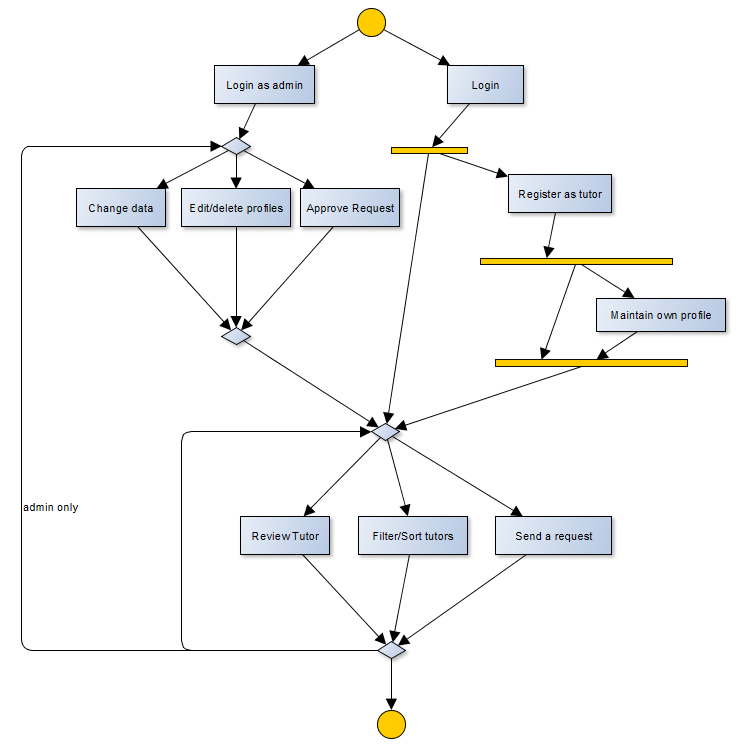
**Maintain own profile**

|  |  |  |
| --- | --- | --- |
| Step | User | Activity |
| 1 | tutor | Edit or delete profile |
| 2 | server | * Connection failed / information is not valid => changes will not be saved * Timeout => Changes will not be saved * Empty fields => “Bitte füllen Sie die notwendigen Felder aus” |

### 

### 

### Workflow



### 

### Open Points

* Report user: If a tutoring account seems to be created for fun (for example:an unserious picture or an inappropriate description of a tutor), then there should be a possibility for other users to report this tutor.

# 

# 

# Non-functional Requirements

|  |  |
| --- | --- |
| ID: | NFR\_001 |
| Name: | Modern Design |
| Type: | USE |
| Description: | For reaching a high number of users, the design of the application must be modern |
| Assigned use cases: | filter/sort tutors, create tutoring profile |

|  |  |
| --- | --- |
| ID: | NFR\_002 |
| Name: | Data security |
| Type: | SEC |
| Description: | The system must not let anyone but the students to see private data of the tutors |
| Assigned use cases: | filter/sort tutors, create tutoring profile |

|  |  |
| --- | --- |
| ID: | NFR\_003 |
| Name: | correct Data |
| Type: | SEC |
| Description: | The tutor’s db ata eg. phone number, name must be correct so that they can be contacted by interested students. |
| Assigned use cases: | send tutoring request |

|  |  |
| --- | --- |
| ID: | NFR\_004 |
| Name: | Waiting time |
| Type: | EFFIC |
| Description: | The system must respond after three seconds. |
| Assigned use cases: | filter/sort tutors |

|  |  |
| --- | --- |
| ID: | NFR\_005 |
| Name: | Changeable Data |
| Type: | MAINT |
| Description: | Data (for example available subjects) must be changeable without editing the code. |
| Assigned use cases: | administrate data |

## Types of non-functional requirements

|  |  |  |
| --- | --- | --- |
| * 1. Type | * 1. Name | * 1. Description |
| USE | Usability requirement | This requirement is to make the target group as described in section 1 is liking to work with that system. |
| EFFIC | Efficiency requirement | Run-time and memory efficiency. What are the constraints under which the system has to run. |
| MAINT | Maintenance and portability requirement | Which maintenance or porting effort is expected in the future? Internationalization expected? Porting to different hardware platform?... |
| SEC | Security requirement | Security requirements comprise confidentiality, data integrity, and availability. How much do we have to consider that data is not accessible to unauthorized persons? Is the correctness and/or consistency of data to be guaranteed? How severe are total system faults? |
| LEGAL | Legal requirement | Are there any standards or legal constraints to be considered? |

Typen von Produktcharakteristiken

Typ USE: Benutzbarkeitsanforderung

Die in Abschnitt 1 beschriebene Zielgruppe liegt diesen Anforderungen zugrunde. Wie muß die Software beschaffen sein, damit diese Zielgruppe gerne damit arbeitet?

Beispiel: Die Software soll flexibel für unterschiedliche Arbeitsweisen einsetzbar sein.

ODER

Die Software soll dem Erscheinungsbild anderer Produkte des Herstellers

entsprechen.

Typ EFFIC: Effizienzanforderung

Hier geht es sowohl um Laufzeit- als auch um Speichereffizienz. Was wird unter dem sparsamen Einsatz dieser Ressourcen verstanden?

Beispiel: Die Berechnung darf nicht länger als 0,25 Sekunden dauern.

Typ MAINT: Wartbarkeits- und Portierbarkeitsanforderung

Welcher Grad an Änderbarkeit wird gefordert? Hier werden, soweit wie möglich, kommende Anpassungen und Erweiterungen vorhergesehen.

Beispiel: Das Produkt soll später auch in englischer Sprache verfügbar sein.

Typ SEC: Sicherheitsanforderung

Zu den Sicherheitsanforderungen gehören die Aspekte Vertraulichkeit, Datenintegrität und Verfügbarkeit. Wie sehr müssen die Daten vor dem Zugriff durch Dritte geschützt werden? Ist es entscheidend, die Korrektheit der erfassten Daten und ihre Konsistenz zu gewährleisten? Dürfen Systemausfälle vorkommen?

Beispiel: Das System muss gewährleisten, dass Daten nie verändert werden können.

Typ LEGAL: Gesetzliche Anforderung

Welche Standards und Gesetze müssen beachtet werden?

Beispiel: Das Produkt muss die ISO 9000 Norm erfüllen.

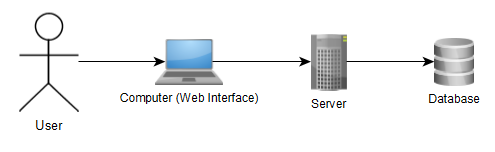
# 

# Quantity Structure

The expected number of tutors is about 100 tutors (4 departments \* 2nd-5th grade = 4 grades \* 2 classes on average \* 3 tutors in a class) .Per tutor an image, the name, class, department, prefered times and remuneration, subjects, given ratings and additional information must be saved. This indicates the use of a database.

# System Architecture and Interfaces

.



The tutoring market will be available online, via a web interface. We will also use a server and a database, but due to the missing information about the final implementation in the school system, we do not have further details about that.

# 

# Acceptance Criteria

Login

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| Login attempt with correct username or password | User is logged in | Username or password is incorrect |
| Login attempt with incorrect username or password | Error message | Username and password are correct |

Sort/filter tutors

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| Choose a sort category & press “Search” | Sorted list appears | No tutor is registered |
| Choose filter categories & press “Search” | Filtered list appears | No tutor with those filter categories can be found |
| Press “Search” without any filter categories | Standard list appears | - |

Show tutor’s profile

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| Pick a tutor and click on it | User gets linked to a page, where the selected tutor is shown | - |

Create tutoring profile

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| User clicks on the button “Become a tutor” | A page appears, where the user can enter his personal data | - |
| User fills in the required data and presses the “Save” button | User becomes a tutor and his registration is successful | Information is missing |

Maintain tutoring profile

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| Tutor clicks on the button “My profile” | A page appears, where the user can edit her/his personal data | - |
| User edits information clicks on the “Save” button | Changes were successful | Information is missing or incorrect |
| User clicks on the “Delete my profile” | Profile gets deleted and user gets linked to the start page | - |

Review tutors

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| User enters the number of books, adds a comment and clicks on the button “Rate” | The rating is added to the list, which must be approved by the administrator | Rating criterias are not filled in or comment field is empty |

Send tutoring request

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| User clicks on the “Contact” button and then on the “Email” button | Tutor gets an automatic email, which informs her/him about the requesting user | User has already sent a contact request |
| User clicks on the “Contact” button and then on the “SMS” button | Tutor gets an automatic sms, which informs her/him about the requesting user | User has already sent a contact request |

Approove reviews

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| Administrator goes to the administrator section and clicks on the button “Pending ratings” | Pending ratings are shown | There are no pending ratings |
| Administrator reads the ratings and clicks “Approve” | On the tutors page, the rating is listed | - |
| Administrator reads the ratings and clicks “Decline” | The ratings is deleted | - |

Adminsitrate data

|  |  |  |
| --- | --- | --- |
| * 1. Test Step | * 1. Expected Behaviour | * 1. Test fails if |
| Administrator goes to the administrator section and clicks on the button “Edit permanent information” | Permanent data isshown | - |
| Administrator selects data which should be changed and clicks “Change” | The permanent data is changed | Data which must be filled in is missing |

# 

# 