Zeeguu Requirements Document

Version 1.0

Authors :
Oliver Holder
Christian Grier Mulvenna
Henry Salas
Tai-Ting Chen
Evi Xhelo
Ai Deng
Jakob Vokac

Clients :
Mircea Lungu
Carlos

Supervisor :
Lars Holdijk

University of Groningen
February 2018

1 Introduction

Zeeguu is an innovative new tool for learning languages. By integrating multiple translating software, such as Google Translate and Microsoft Translator, Zeeguu allows its users to read articles with real time translation at their finger tips and practice their vocabulary with interactive exercises. Right now, Zeeguu supports five different languages to learn (English, Dutch, German, Spanish and French) and three base languages (English, Dutch and Chinese).

Right now, the interaction in Zeeguu is still rather simple. Anyone, who wishes to use it, simply signs up with an access key and inputs some basic information (your name, email address, password, which language you wish to learn and what your base language is). Once the account is created, the user can choose to read articles, practise their vocabulary or look at the words they have learned.

Our project is to expand Zeeguu to allow teachers to supervise their students as they learning using it. We would allow teachers to have their seperate accounts, be able to invite students into 'classes' and oversee their progress with an intuitive visual interface. Apart from these basic requirements, we can expect to integrate other features that we will find out about in the near future as well.

2 Critical Functional Requirements

1. Teachers are able to create classes which hold students.
   — Teachers have real-world classes and they want to record these classes in the system in order to track performance, view statistical information, and assign work.
2. Teachers are able to add students to the class by invite code.
   — Invites codes provide an easy way to allow students access to a class.
3. Teachers are able to select a source.
   — Different sources which provide reading material are available. The teacher should be able to assign a source to a class where the students can select an article from.
4. Teachers are able to see the time students spent on a certain activity.
   — Teachers want to see how much time a student has spent reading a particular article.
   — Teachers want to see how much time a student spent practicing vocabulary.
5. Teachers are able to see weak words by individual students, classes and all students.
   — Weak words should be that students and or classes fail with a certain selected percentage.
   — Individual students, individual classes and multiple classes should be analyzable.

3 Important Functional Requirements

1. Teachers are able to see this data in visualizations.
   - These visualizations could consist of Pie charts, Bar graphs and other visualization techniques.
   - These visualizations should be available on the level of student, class and multiple classes.

4 Useful Functional Requirements

1. Teachers are able to see which words no one understood.
   - There could be words that almost no one understood in which case the students click on them. The teachers will be able to see which are these words.
2. Teachers can give names to classes.
   - Just as in real life schools there are classrooms, in ZD teachers are going to be able to have separate classes for students and be able to name those classes.
   - A list of the words that every student knew, i.e no one clicked on for translation, is going to appear if the teacher is going to choose this option.
3. Teachers are able to see charts on the exercises.
   - Based on the data that is gathered, teachers will also be able to see the performance of the students on the exercises they solve.
4. Teachers are able to see which words everyone knows.
   - A list of the words that every student knew, i.e no one clicked on for translation, is going to appear if the teacher is going to choose this option.

5 Non-functional Requirements

5.1 User interface

1. UI is simple and modern.
2. UI is functional.
3. UI is understandable - Ease of use must be high.

5.2 Performance

1. Creating a class should take very little time.
2. Generating statistics should take a short amount of time.

5.3 Security

1. Teachers only have access to data of classes they teach.

6 Won't Do

1. Article distribution system.
2. Personal translations from teachers.
3. Statistics will split words into nouns/verbs/etc.

7 Implementation details

- Use Python as programming language.
- Use flask as model for application.
- Use SCRUM as development framework.

8 Open Questions

1. Can a teacher move/delete a student from a class (to another class) ?
2. How does the code invitation work ? Do students send the request and the teachers have to accept/decline ? Or does the teacher send all the invites ?
3. What specific data is recorded in the database ? (Needed to be more specific about the visualizations of data)
4. Security concerns with regards to students data.
5. Could there be more than one teacher per class ?
6. What is the relation between the data and the teacher ? Should the student be prompted to be made aware their data is being used ?

9 Meeting Log

First meeting on the 23rd at 6pm.

10 Change Log

Created requirements document on the 27th at 5pm.

When	**Which Section**	**What**
Feb 27th, 15 :30 - 18 :00	The document	Started the requirement document