

|  |  |
| --- | --- |
| **Use Case name** | **Take Test** |
| Goal in context | Test the user’s knowledge of German. |
| Successful End condition | Test is presented to the user and then marked. |
| Failed End condition | n/a. |
| Primary Actors | Student, Database manager. |
| Trigger | User wishes to test knowledge. |
| Main Flow | 1. User chooses what category they wish to be tested on. 2. Database manager selects a random row from the database and creates a question. 3. User is then presented with a vocab question with multiple choice answers. 4. User completes questions until test ends. |
| Extensions | View Score   1. User is present with a score relative to the amount of questions he got correct. |

|  |  |
| --- | --- |
| **Use Case name** | **View Individual Grades** |
| Goal in context | Review individual user progress. |
| Successful End condition | Pages is displayed showing all completed tests and the grades that the student achieved. |
| Failed End condition | No grades found as no tests have been taken. |
| Primary Actors | Student, Database manager. |
| Trigger | User wishes to view his progress. |
| Main Flow | 1. User navigates to the grades page. 2. Grades table is called via database manager. 3. If the user is logged in as a student, they will be presented with the grades that they have achieved via completing tests. |

|  |  |
| --- | --- |
| **Use Case name** | **View High Scores** |
| Goal in context | Compare individual progress with the progress of other students. |
| Successful End condition | Students with the highest scores on each test is displayed in descending order. |
| Failed End condition | No high scores found as no tests have been taken. |
| Primary Actors | Student, Database manager. |
| Trigger | User wishes to compare his progress to other students. |
| Main Flow | 1. User navigates to the high scores page. 2. Database of all student’s scores is queried. 3. User is presented with the other students that have scored the highest on the quizzes for each category. |

|  |  |
| --- | --- |
| **Use Case name** | **Change Password** |
| Goal in context | To allow the user to change his password to something more secure and unique. |
| Successful End condition | Old password is replaced by new password. |
| Failed End condition | Original password incorrect or new password invalid. Password not changed. |
| Primary Actors | Student, Database manager. |
| Trigger | User wishes to change his password. |
| Main Flow | 1. User navigates to change password page. 2. User inputs old password and new desired password. 3. Database entry is updated. |

|  |  |
| --- | --- |
| **Use Case name** | **Change Username** |
| Goal in context | To allow for personalisation so the user feels more in touch with the website. |
| Successful End condition | Old username is replaced by new username. |
| Failed End condition | Original username incorrect or new username invalid. Username not changed. |
| Primary Actors | Student, Database manager. |
| Trigger | User wishes to change his username. |
| Main Flow | 1. User navigates to the change username page. 2. User inputs new desired username. 3. Database of users is updated to contain new username. |

|  |  |
| --- | --- |
| **Use Case name** | **View All Grades** |
| Goal in context | To allow the teachers to monitor the progress of all students. |
| Successful End condition | Grades of all students for all tests is displayed. |
| Failed End condition | No grades displayed as no tests have been taken. |
| Primary Actors | Teacher, Database manager. |
| Trigger | User wishes to view the grades and monitor the progress of all students. |
| Main Flow | 1. User navigates to the grades page. 2. Database is queried. 3. Grades of all students are presented. |

|  |  |
| --- | --- |
| **Use Case name** | **Edit word** |
| Goal in context | Allow the teacher to correct any mistakes or update with any changes. |
| Successful End condition | Database entry for specific word is updated with new data. |
| Failed End condition | Original word fields incorrect or new word fields invalid. Word not changed. |
| Primary Actors | Teacher, Admin, Database manager. |
| Trigger | User wishes to update the entry of a specific word. |
| Main Flow | 1. User navigates to the admin page. 2. If user is signed in as a teacher or an admin they are allowed to proceed. 3. User fills in details for the word they wish to change and what they wish to change it to. 4. If word exists and the fields are valid, word is updated in the vocab database. |

|  |  |
| --- | --- |
| **Use Case name** | **Add Word** |
| Goal in context | To allow the teachers/admins to increase the word pool so the website can be continuously used into the future. |
| Successful End condition | New word with all relevant fields is added to the database |
| Failed End condition | New word invalid or word already exists. New word not entered into database. |
| Primary Actors | Teacher, Admin, Database manager. |
| Trigger | User wishes to add another word to the database. |
| Main Flow | 1. User navigates to the admin page. 2. If user is signed in as a teacher or an admin they are allowed to proceed. 3. User fills in details for the word they wish to add. 4. If word doesn’t exist in the current database and the fields are valid, word is added to the vocab database. |

|  |  |
| --- | --- |
| **Use Case name** | **Delete Word** |
| Goal in context | To allow teachers/admins to remove words that are no longer relevant to the students learning. |
| Successful End condition | Specified word is removed from the database. |
| Failed End condition | Word is not found in the database. |
| Primary Actors | Teacher, Admin, Database manager. |
| Trigger | User wishes to remove a word from the database. |
| Main Flow | 1. User navigates to the admin page. 2. If user is signed in as a teacher or an admin they are allowed to proceed. 3. User fills in details for the word they wish to delete. 4. If word does exist in the current database and the fields are valid, word is deleted from the vocab database. |
| Extensions |  |

|  |  |
| --- | --- |
| **Use Case name** | **Register User** |
| Goal in context | To allow teachers/admins to add a new student to the database. |
| Successful End condition | New entry added to the users database which a new student can use to log in. |
| Failed End condition | User is already found in the database. |
| Primary Actors | Teacher, Admin, Database manager. |
| Trigger | User wishes to add another user to the database. |
| Main Flow | 1. User navigates to the admin page. 2. If user is signed in as a teacher or an admin they are allowed to proceed. 3. User then fills in the credentials of the student account they wish to create. |

|  |  |
| --- | --- |
| **Use Case name** | **Login** |
| Goal in context | Allows any user to access the features relevant to their access level. |
| Successful End condition | Checks the provided credentials with the user database, then allows the user to proceed onto the rest of the site if the entry is found. |
| Failed End condition | Username and password combination not found in the database. Login fails. |
| Primary Actors | Teacher, Student, Admin, Database manager. |
| Trigger | User wishes to proceed onto the rest of the site. |
| Main Flow | 1. User loads the web page. 2. User then enters username and password. 3. If the user and pass are correct they will be allowed to continue. 4. If not they will be asked to return and try again. |