



DAS GROSSE SQL-SPIEL

THE SQL-ALCHEMIST

Software-Entwicklungspraktikum (SEP)
Sommersemester 2015

Pflichtenheft

Auftraggeber:

Technische Universität Braunschweig
Institut für Informationssysteme
Prof. Dr. Wolf-Tilo Balke
Mühlenpfordtstraße 23, 2.OG
D-38106 Braunschweig

Betreuer: Jan-Christoph Kalo

Auftragnehmer:

Name	E-Mail-Adresse
Gabriel Ahlers	g.ahlers@tu-braunschweig.de
Majid Dashtiepilehroud	m.dashtiepilehroud@tu-braunschweig.de
Ronja Friebe	r.friebe@tu-braunschweig.de
Stefan Hanisch	stefan.hanisch@tu-braunschweig.de
Fabio Luigi Mazzone	f.mazzone@tu-braunschweig.de
Nicole Naczk	n.naczk@tu-braunschweig.de
Denis Nagel	denis.nagel@tu-braunschweig.de
Luca Porcello	l.porcello@tu-braunschweig.de
Christian Reineke	c.reineke@tu-braunschweig.de
Christian Sander	christian.sander@tu-braunschweig.de
Carl Schiller	c.schiller@tu-braunschweig.de
Levent Muzaffer Üner	l.uener@tu-braunschweig.de
Sören van der Wall	s.van-der-wall@tu-braunschweig.de
Daniel Wolfram	d.wolfram@tu-braunschweig.de

Braunschweig, 13. Mai 2015

Inhaltsverzeichnis

1 Zielbestimmung	5
1.1 Registrierung	5
1.2 Spielerprofil	6
1.3 Story-Modus	7
1.4 Tutorial	8
1.5 Trivia-Modus	9
1.6 Shop	9
1.7 Hausaufgaben-Modus	10
1.8 Admin-Tool	10
1.9 Einstellungen	11
1.10 Ranglisten	11
1.11 Musskriterien	12
1.12 Sollkriterien	12
1.13 Kannkriterien	12
1.14 Abgrenzungskriterien	13
2 Produkteinsatz	15
2.1 Anwendungsbereiche	15
2.2 Zielgruppen	15
2.3 Betriebsbedingungen	16
3 Produktübersicht	17
3.1 Die Registration	17
3.2 Der Story-Mode	19
3.3 Das Spiel	19
4 Produktfunktionen	24
4.1 Nutzer registrieren	24
4.2 Nutzer anmelden	25
4.3 Nutzer abmelden	25
4.4 Profil einsehen	26
4.5 Benutzernamen ändern	26
4.6 Passwort ändern	27

4.7 Avatar ändern	28
4.8 Benutzer löschen	28
4.9 Audioeinstellungen bearbeiten	29
4.10 Spielstand zurücksetzen	29
4.11 Tutorial spielen	30
4.12 Story spielen	30
4.13 SQL-Trainer spielen	31
4.14 Minispiel spielen	32
4.15 Hausaufgaben bearbeiten	32
4.16 Ranglisten einsehen	33
4.17 Spieler suchen	33
4.18 Hausaufgabenergebnisse einsehen	34
4.19 Benutzer befördern	35
4.20 Einem Benutzer Adminrechte geben	35
4.21 Eine Trivia-Aufgabe erstellen	36
4.22 Benutzeraufgaben bewerten	36
4.23 Hausaufgaben erstellen	37
5 Produktdaten	38
6 Nichtfunktionale Anforderungen	43
6.1 Funktionalität	43
6.2 Sicherheit	43
6.3 Benutzbarkeit	44
6.4 Änderbarkeit	44
6.5 Qualitätsanforderungen	44
7 Benutzeroberfläche/Schnittstellen	46
8 Technische Produktumgebung	49
8.1 Software	49
8.2 Hardware	49
8.3 Produktschnittstellen	49
9 Glossar	50

Abbildungsverzeichnis

1.1	Veranschaulichung des Spielablaufes	14
3.1	Die Registration	17
3.2	Das Admintool	18
3.3	Spielablauf im Story-Modus	20
3.4	Login-/Registrationsvorgang	22
3.5	Der Großteil des eigentlichen Spiels	23
7.1	Login	46
7.2	Menüführung	47
7.3	Login zum Admin-Tool	47

1 Zielbestimmung

Die Aufgabe besteht darin, eine plattformübergreifende, interaktive Spielesoftware zu entwickeln, um den Studenten der Pflichtveranstaltung „Relationale Datenbanksysteme I (RDBI)“ den Umgang mit der Datenbankanfragesprache SQL vorlesungsbegleitend spielerisch zu vermitteln.

Es soll eine Möglichkeit geschaffen werden, praktische Aspekte im Bereich der Datenbankanfragen zu üben, da dies im Rahmen einer Lehrveranstaltung viele verschiedene Schwierigkeiten mit sich bringt. Zum Einen kann theoretisches Verständnis zwar gut vermittelt werden, jedoch ist es nur eingeschränkt möglich, genügend Übungsmaterial zur Verfügung zu stellen, damit eine komplexe Anfragesprache, wie SQL, geübt und somit auch verstanden werden kann. Um diese Probleme zu beheben soll die Software mit Fokus auf einen interaktiven SQL-Trainer entwickelt werden. Dieser wird in zwei verschiedene Spiel-Modi integriert.

Die für die Applikation gewählte Sprache ist Englisch. Deshalb werden im folgenden Programmablauf alle spielbezogenen Begriffe ebenfalls in englisch benannt.

1.1 Registrierung

Vor dem ersten Start der Anwendung muss eine Registrierung des Users erfolgen. Dafür steht ein „Sign In“-Fenster zur Verfügung, das einen Benutzernamen, ein Passwort, sowie eine E-Mail-Adresse oder für Studenten der TU Braunschweig die entsprechende y-Nummer verlangt. Diese Daten werden über eine gesicherte Leitung an das Back-End gesendet und dort wird zwischen y-Nummer und der E-Mail-Adresse unterschieden. Meldet sich der User mit einer E-Mail-Adresse an, so wird eine Bestätigungsanfrage inklusive Aktivierungslink an diese gesendet. Bestätigt der User die Registrierung, wird ein UserProfile-Object erstellt und in der Datenbank gespeichert. Ist der User Student, so wird direkt, ohne Bestätigung der y-Nummer, das UserProfile-Object gesichert. Eine Bestätigung ist in diesem Fall nicht nötig, da der User direkt über das LDAP der Technischen Universität Braunschweig verifiziert wird.

Ist die Registrierung abgeschlossen, wird der Nutzer zum „Login“-Fenster umgeleitet, wo er sich nun mit den vorher registrierten Daten anmelden kann. Hierbei reicht die y-Nummer beziehungsweise die E-Mail-Adresse und das Passwort. Die eingegebenen Daten werden mit denen in der Datenbank abgeglichen. Ist dies erfolgreich, so gelangt der User in das Hauptmenü.

Im Hauptmenü besteht die Möglichkeit aus folgenden Punkten zu wählen:

- Profile
- Story-Mode
- Trivia-Mode
- Shop
- Homework
- Leaderboard
- Settings

1.2 Spielerprofil

Das Spielerprofil beschreibt alle Nutzereigenschaften und soll dazu dienen eine Übersicht über diese darzustellen. Möchte der Spieler diese eventuell verändern, so muss er in den Reiter Settings wechseln um die Veränderung vorzunehmen.

Wählt der Benutzer nun „Profile“ aus, dann bekommt er einen Einblick über die Benutzerinformationen und die aktuellen persönlichen Statistiken. Diese Informationen werden vom Front-End angefragt und vom Back-End zur Verfügung gestellt. Übergeben wird hierbei das bei der Registration erstellte Userprofile-Object. Die persönlichen Statistiken beinhalten:

- Den Punktestand im Story- und Trivia-Mode, welcher der akkumulierten Punktezahl aus den gesammelten Lofi-Coins und der Beantwortung eines Statements entspricht. Hierbei bringen schwierigere Statements entsprechend mehr Punkte. Pro Lofi-Coin gibt es generell einen Punkt. Sind die Lofi-Coins in tieferen Stufen des Dungeons gesammelt worden, so wird der Wert des Lofi-Coins mit der erreichten Stufe im Dungeon multipliziert.
- Den prozentualen Spielerfortschritt im Story-Mode.
- Für den Fall, dass der Benutzer den Story-Mode mindestens einmal abgeschlossen hat, wird die geringste Anzahl an Durchgängen dargestellt.
- Die insgesamt verbrachte Zeit in den jeweiligen Modi. Die Zeit wird ab einer Abwesenheit von 2 Minuten nicht mehr mitberechnet.
- Die prozentuale Erfolgsquote des Benutzers beim Beantworten von SQL-Statements.
- Die gesamte Anzahl, sowie der aktuelle Stand an Lofi-Coins im Besitz des Benutzers.
- Das für den Tag festgelegte Limit an Lofi-Coins/Scrolls, die noch vom Benutzer eingesammelt werden können.

1.3 Story-Modus

Der Story-Modus ist der Spielmodus der dem Spieler die Möglichkeit bieten soll SQL in einer nicht ganz so trockenen, langweiligen Umgebung zu üben. Der Modus besteht demnach aus einer Mischung aus SQL-Trainer und einem Minispiel. Das Minispiel soll hierbei als Motivationsförderung dienen.

Wird nun der „Story-Mode“ gewählt, wird eine Anfrage an das Back-End gesendet, der Story-Content geladen und der Benutzer gelangt in das „Laboratory“. Der Story-Content beinhaltet sämtliche Spieldaten. Darunter befinden sich das Inventory, die Playerstatistics, die Scroll-/Lofi-Coin-Limits, die Scenetexts und die Scrollcollection (Erklärungen zu diesen Begriffen finden sich im Glossar). Die einzelnen Elemente, die sich im Laboratory befinden, sind: Das Character-Sheet, der Belt, die Scrollcollection, der Eingang in das Dungeon und der Ausgang, mit dem der Story-Mode verlassen werden kann.

Das Dungeon stellt das Minispielelement der Applikation dar und wird durch einen Endless-Runner repräsentiert. Dort werden dem Benutzer die verbliebene Anzahl seiner Leben, der Inhalt seines Belts und sein aktueller Punktestand angezeigt. Punkte erhält man durch das Einsammeln von Münzen und durch das Abschließen von Maps. Diese Maps werden zufällig, aber nach Schwierigkeitsgrad sortiert, von der Spielfigur durchlaufen. Nach jeder fünften Map gibt es eine Herausforderung, die ohne das Einsetzen von Potions, was wiederum das Bearbeiten von SQL-Anfragen einschließt, nicht bestanden werden kann. Je tiefer der Spieler in das Dungeon gelangt, desto schwieriger werden die Läufe und die Anfragen.

Die Spielfigur besitzt vier Attribute: Jump (Sprunghöhe), Speed (Laufgeschwindigkeit), Defense (Verteidigung) und Health (Lebenspunkte). Typische Hindernisse, die im Dungeon zu überwinden sind, sind Schluchten, die mit Jump oder Speed überwunden werden können und kleinere Gegner, die der Spielfigur bei Kollision Health stehlen, wenn nicht genug Defense vorhanden ist. Attribute können dauerhaft durch „Enchantments“ erhöht werden. Dafür müssen vorher entsprechende Scrolls im Dungeon gesammelt werden. Der Belt repräsentiert den Gürtel der Spielfigur. Dieser hat mehrere Slots, die mit den verschiedenen Potions gefüllt werden können. Potions sind Tränke, die sich der Benutzer brauen kann, wofür er, wie bei den Enchantments zunächst eine Scroll im Dungeon finden muss (die verschiedenen Scrolls unterscheiden sich hierbei in ihrer Farbgebung, dabei ist die Enchantment-Scroll mit einem roten Bändchen gekennzeichnet). Für jede Potion gibt es exakt eine Scroll, die so gesehen ein Rezept darstellt. Die Scrolls werden beim Brauen von Potions nicht verbraucht und werden in der Scrollcollection im Laboratory gesammelt.

Die Scrollcollection ist ein Rezeptbuch, in dem sich alle zur Verfügung stehenden Scrolls, Enchantments und Potions sammeln. Wählt man eine Scroll aus, fordert das Front-End eine SQL-Anfrage von dem Back-End. Daraufhin liefert das Back-End einen Task, der aus einer Fragestell-

lung und einem Datenbankschema besteht, aus. Der Spieler beantwortet die Anfrage, sie wird zurück an das Back-End gesendet, wird auf Korrektheit überprüft und inklusive eines Fehlerhinweises zurückgeschickt. Der Fehlerhinweis wird so umschrieben, dass die Möglichkeit für den Spieler besteht seine Anfrage zu korrigieren. Beantwortet der Spieler sie korrekt, erhält er die entsprechende Potion, beziehungsweise das Enchantment. Die Schwierigkeit des Tasks ist abhängig von der Scroll. Je tiefer die Scroll im Dungeon gefunden worden ist, sprich wie weit der Spieler in der Story fortgeschritten ist, desto schwieriger ist der Task. Außerdem erhält der Benutzer Lofi-Coins für das richtige Bearbeiten der Anfrage. Hierbei ist die Anzahl der erhaltenen Lofi-Coins abhängig vom Schwierigkeitsgrad und der Bearbeitungszeit. Die Veränderungen von Inventory und Lofi-Coin-Guthaben wird im Back-End gesichert. Das Character-Sheet zeigt den zurzeit ausgewählten Avatar und die Werte der oben beschriebenen Attribute des Spielers.

1.4 Tutorial

Beim ersten Aufruf des Story-Modes wird ein „Tutorial“ gestartet, welches verlustfrei übersprungen werden kann. Das gesamte Tutorial wird durch den narrativen Charakter des Spiels, einem raubbeinigen Zauberschrank namens „Secreterry“, gehalten. Dieses Tutorial besteht aus einer Einführung der einzelnen oben genannten Elemente des Laboratory und der Erklärung eines Spielablaufes. Hierbei werden die einzelnen Elemente visuell hervorgehoben und sowohl auditiv, als auch textuell beschrieben. Diese Informationen werden nach Anfrage direkt zu Anfang mit dem Story-Content über das Back-End geladen. Nachdem alle Elemente erklärt worden sind, wird dem Benutzer ein festgelegter Teil des Minispiele erklärt. Dieses beginnt mit einem ersten Lauf durch das Dungeon. In einer vorgefertigten Tutorial-Map erhält der Benutzer eine erste Scroll und stößt auf ein, zu diesem Zeitpunkt, unüberwindbares Hindernis und stirbt. Daraufhin erscheint ein „Game-Over“-Screen, in dem die eingesammelten Items aufgelistet werden, in diesem Fall eine Scroll, mit der der Spieler in der Lage ist eine Jump-Potion zu brauen. Nun führt das Tutorial zurück in das Laboratory und zeigt dem Benutzer wie die gesammelte Scroll in der Scrollcollection benutzt werden kann um die Potion zu brauen, sprich der Spieler beantwortet eine einfache SQL-Anfrage und bekommt seine Potion. Im nächsten Schritt des Tutorials wird dem Spieler durch Secreterry erklärt, wie die Potion in den Belt eingefügt wird. Dazu wechselt der Benutzer in das Untermenü Belt, wo er die Potion auswählt und in einen der dafür vorgesehenen Slots einfügt. Nun ist der User in der Lage das vorher genannte unüberwindbare Hindernis zu bewältigen. Am Ende dieses Levels erhält er eine weitere Scroll, welche diesmal das Rezept für ein Enchantment enthält.

Wenn das Tutorial erfolgreich beendet, beziehungsweise übersprungen, wurde stellt das Front-End die Anfrage an das Back-End den nächsten Part der Story inklusive Spieldaten zu laden. Dieser, wie auch alle folgenden, besteht wieder aus dem aktuellen Inventory, den Playerstatistics,

den (nun aktualisierten) Scroll-/Lofi-Coin-Limits, den jeweiligen Scenetexts und der Scrollcollection. Jeder Teil der Story besteht aus einer Einführung durch den narrativen Charakter, die dem Benutzer den aktuellen Stand des Spieles erläutert, mit Bezug auf den vorherigen Teil des Spieles. Durch weitere Durchläufe im Dungeon, das Einsammeln der Scrolls und das Benutzen dieser, ist der Benutzer in der Lage weitere Parts zu laden und immer weiter im Dungeon vorzudringen, bis man schließlich nach acht Parts zum Endgegner gelangt. Zuletzt sei zu erwähnen, dass der Benutzer zu jeder Zeit den Story-Mode verlassen kann, wobei sein Fortschritt und sein Inventory im Back-End gespeichert werden.

Zur Veranschaulichung des gesamten Spielablaufes inklusive Tutorial dient eine tabellarische Darstellung, die am Ende von Kapitel 1 zu finden ist. Diese Darstellung ist ebenfalls in ein Aktivitätsdiagramm überführt worden. Dieses ist wiederum in Kapitel 3 zu finden.

1.5 Trivia-Modus

Der Trivia-Modus ist für die Spieler, die reines SQL lernen möchten. Dazu kommt der SQL-Trainer zum Vorschein.

Wählt der Benutzer nun den „Trivia-Mode“ aus, so kann er SQL-Anfragen üben, ohne das Minispiel zu spielen. Geladen werden auch hier wieder, nach Anfrage durch das Front-End an das Back-End, sämtliche dafür benötigten Information. Diese bestehen demnach aus einem gesamten Task inklusive Datenbankschema und dem Anfragetext, zu welchem das SQL-Statement erstellt werden muss. Hierbei hat der Spieler von vornherein die Möglichkeit einen aus fünf Schwierigkeitsgraden auszuwählen. Hat er sich für einen Schwierigkeitsgrad entschieden, beginnt der gleiche Ablauf, wie bei den Tasks im Story-Mode. Das Front-End schickt eine Anfrage an das Back-End und ein kompletter Task wird übergeben. Aber auch in diesem Mode hat der Spieler die Möglichkeit, ohne Story-Content in das Dungeon zu gehen und nach Belieben mehr oder weniger viele, zufällig ausgewählte, Maps zu spielen. Dafür wird es einen eigens dafür vorgesehenen Menüpunkt geben, mit dem der Spieler zu jeder Zeit hin, aber auch wieder zurück wechseln kann.

1.6 Shop

Der Shop dient dazu, den Spielern ein wenig Raum für Personalisierung zu bieten. Wählt der Benutzer „Shop“, wird ihm die Möglichkeit gegeben, gesammelte Lofi-Coins gegen Spieler-Avatare einzutauschen, die gleichzeitig die Spielfigur im Dungeon repräsentieren. Außerdem soll es die Möglichkeit geben weitere Slots zum Belt der Spielfigur dazu zu kaufen. Auch weitere Leben können hier gegen Lofi-Coins erworben werden.

1.7 Hausaufgaben-Modus

Wählt der Spieler „Homework“, so kann er die in der Applikation verknüpften Hausaufgaben bearbeiten. Hierbei stellt wieder das Front-End die Anfrage an das Back-End. Dafür ist ein komplettes Aufgabenpaket mit unterschiedlich schwierigen SQL-Aufgaben vorgesehen. Die Hausaufgabe ist erst dann abgeschlossen, wenn das Aufgabenpaket unter einer bestimmten Abschlussbedingung erfüllt worden ist. Die Bedingung kann entweder ein Punkte-Limit oder ein Zeit-Limit sein. Ein Aufgabenpaket kann auch aus einer gewissen Anzahl an SQL-Statements bestehen. Diese Bedingung kann von einem Wissenschaftlichen Mitarbeiter des Dozenten von RDBI für die wöchentlichen Challenges ausgewählt werden und in der Datenbank hinterlegt werden. Der Benutzer hat also nur innerhalb einer Woche Zeit das gesamte Aufgabenpaket zu lösen, erst dann wird die Hausaufgabe als abgeschlossen markiert. Auch die Aufgabenstellung und die Statements der zu lösenden Aufgaben müssen von dem Mitarbeiter in die Datenbank eingefügt werden. Dafür steht das Admin-Tool des Back-Ends zur Verfügung. Ist ein Aufgabenpaket für eine bestimmte Woche erstellt worden, so kann sie von einem Studenten bearbeitet werden. Wählt der Student die aktuelle Hausaufgabe aus, geht die Anfrage für eine erste Aufgabenstellung raus und das Back-End sendet die Aufgabe zurück.

Schickt der Benutzer einmal ein Statement ab, so wird auch hier wieder eine Kontrolle durchgeführt und mit eventuellem Fehlerhinweis zurückgeschickt. Der User korrigiert sein Statement und sendet es erneut an das Back-End. Bei richtiger Beantwortung wird dieses Statement als abgeschlossen markiert und auf dem Server gesichert. Bevor die Aufgaben jedoch gelöst werden können, müssen sie in das Back-End eingepflegt werden. Dafür steht das Admin-Tool zur Verfügung. Dort kann auch jeder Student nachschauen ob er die Hausaufgaben, beziehungsweise die Challenges, bestanden hat oder nicht.

1.8 Admin-Tool

Das Admin-Tool stellt für verschiedene Benutzergruppen unterschiedliche Funktionen zur Verfügung. Studenten, die sich mit ihrer y-Nummer registriert haben, können hier die Ergebnisse aller von ihnen bearbeiteten Hausaufgaben einsehen. Wenn ein Spieler die vom Kunden gewünschten Kriterien erfüllt hat, können diese „befördert“ werden. Einem beförderten Benutzer ist es möglich selbst Aufgaben zu erstellen, die von anderen Spielern im Trivia-Mode bearbeitet werden können, wenn sie diese Funktion wahrnehmen möchten. Dieses kann jeder Spieler für sich entscheiden und im Menü im Trivia-Mode die dafür vorgesehene Checkbox aktivieren. Des Weiteren gibt es die Gruppe der Administratoren. Diese sind vor allem für die Benutzerverwaltung verantwortlich. Sie können Benutzer löschen, Challenge-Pakete und Hausaufgaben erstellen, sowie die Lösungen der Studenten einsehen.

1.9 Einstellungen

Wählt der Benutzer „Settings“ hat er Zugriff auf allgemeine Einstellungen. Diese werden Spieler-spezifisch im Back-End gespeichert und beim Aufruf von diesem erfragt. Der User hat nun die folgenden Einstellungsmöglichkeiten:

- Sound on/off: Aktiviert oder deaktiviert die Soundeffekte die beim Springen der Spielfigur oder beim Betätigen eines Buttons abgespielt werden.
- Music on/off: Aktiviert oder deaktiviert die Hintergrundmusik des Minispiele.
- Tutorial: Das Tutorial kann reaktiviert werden, damit die Spieler, die es vermeidlich übersprungen haben, dieses trotzdem nochmal ansehen können.
- Story-Reset: Setzt den Fortschritt des Story-Modes nach zurück. Dieses muss bestätigt werden.
- Change Username: Gibt dem User die Möglichkeit, seinen Benutzernamen zu ändern.
- Change Password: Gibt dem User die Möglichkeit, sein Passwort zu ändern.
- Delete Profile: Hiermit kann der User sein komplettes Profil löschen.

Erfolgt eine Änderung der Einstellungen, wird diese sofort ans Back-End geschickt, dort gespeichert und kurz darauf im Front-End aktualisiert.

1.10 Ranglisten

Die Leaderboards dienen zusätzlich zu dem Minispiel als Motivationsförderung. Wählt der Benutzer „Leaderboards“ bekommt er eine Einsicht in Ranglisten, in denen die Statistiken aller Benutzer in den verschiedenen Modi dargestellt werden. Folgende Ranglisten gibt es sowohl für den „Story-Mode“ als auch für den „Trivia-Mode“:

- Gesamtanzahl Lofi-Coins (total lofi-coins)
- Gesamtanzahl Läufe (total runs)
- geringsten Anzahl an benötigten Durchgängen für die Story (min. runs per story)
- insgesamt im Spiel verbrachten Zeit (time spent)
- Anzahl an abgeschlossenen Stories (story count)
- prozentuale Erfolgsquote beim Beantworten von Anfragen
- prozentuale Erfolgsquote der beim ersten Versuch korrekt beantworteten Anfragen

Die „Top Ten“ jeder dieser Ranglisten werden direkt als Liste dargestellt. Ist der Benutzer unter diesen, wird er markiert, falls nicht, wird sein Rang unter dieser Liste angezeigt. Der Benutzer hat die Möglichkeit sowohl seinen eigenen Rang, als auch den anderer Benutzer zu sehen, sofern er deren Benutzernamen kennt. Die Daten für die Ranglisten werden vom Back-End zusammengestellt und auf Anfrage ans Front-End gesendet.

1.11 Musskriterien

- ⟨RM1⟩ Es wird ein „Jump&Run“-Minispiel für die Motivationsförderung der User geben.
- ⟨RM2⟩ Ein grafisch aufbereitetes Webinterface, in dem die Menüführung eingebettet ist.
- ⟨RM3⟩ Ein SQL-Trainer, das aus einem Aufgabenstellungsfenster und einem Eingabefenster besteht.
- ⟨RM4⟩ Das Back-End wird entsprechende Nutzerdaten der Spieler speichern, sichern.
- ⟨RM5⟩ Das Back-End wird ein Admin-Tool zur Userverwaltung zur Verfügung stellen.
- ⟨RM6⟩ Das Back-End wird ein Admin-Tool zur Aufgabenverwaltung zur Verfügung stellen.
- ⟨RM7⟩ Es wird die drei verschiedenen Spielmodi geben (Trivia-, Story und Homework-Mode).
- ⟨RM8⟩ Die Software wird an das LDAP der Technischen Universität Braunschweig angebunden.
- ⟨RM9⟩ Die Game-Engine ist in das Back-End integriert.

1.12 Sollkriterien

- ⟨RS1⟩ Das Minispiel kann in verschiedenen Schwierigkeitsgraden gespielt werden.
- ⟨RS2⟩ Die Applikation soll auf mobilen Endgeräten ausführbar sein.
- ⟨RS3⟩ Der Avatar, beziehungsweise die Spielfigur sollen ersetzt oder personalisiert werden können.
- ⟨RS4⟩ Es können Highscores generiert und im Profil visualisiert werden.
- ⟨RS5⟩ Auch Leaderboards sollen generiert und in einem eigenen Menüpunkt visualisiert werden können.

1.13 Kannkriterien

- ⟨RC1⟩ Es könnten neben den Hausaufgaben-Paketen auch Übungspakete erstellt werden können.
- ⟨RC2⟩ Auch andere Universitäten könnten über ein entspechendes LDAP angebunden werden.
- ⟨RC3⟩ Es könnte ein abgespecktes Tutorial für den Trivia-Mode geben.

⟨RC4⟩ In den Shop könnten weitere Features integriert werden.

⟨RC5⟩ Die Steuerung des Minispiel kann auf Tasten, statt Mausklicks gelegt werden.

⟨RC6⟩ Es soll Freundeslisten geben können.

1.14 Abgrenzungskriterien

⟨RW1⟩ Die Verknüpfung zu Social-Networks soll nicht realisiert werden.

⟨RW2⟩ Es wird keinen Mode geben, in dem der Fokus nur auf dem Minispiel liegt.

⟨RW3⟩ Innerhalb der Applikation wird es keine Möglichkeit für In-Game-Käufe geben.

⟨RW4⟩ Es wird keine kostenpflichtigen Zugriffsmöglichkeiten für die Software geben.

⟨RW5⟩ Es werden keine Hausaufgabenergebnisse in das HMS des Instituts für Informationssysteme übertragen.

DAS GROSSE SQL-SPIEL

The SQL-Alchemist

Action/Site	Show	Data From User	From	Server	To	Result	Redirect
						Successful	Failure
						Successful	Failure
Sign In	"Sign In" Window	Username, ID, Password	-	Username, ID, Password	-	Redirect To Login(Successfull)	-
Login	"Login" Window	ID, Password	-	ID, Password	Userprofile-Object	Redirect To Home	-
Home	Haupmenü						
Story							
if	Tutorial	Show Lab Tutorial	Get Story Context	?Intro, Inventory, Playerstats, Scroll-/Coin Limit, Texte, levelcontent, Rezeptbuch	confirmation	correctness	Failure
	1. Run	Dungeon		Collected Items			
	SQL	Statement	SQL Statement For Scroll X Strength y	Send User Statement			
	2.Run	Dungeon		Collected Items	confirmation		
else if	Next Part		Get Story Context	?Intro, Inventory, Playerstats, Scroll-/Coin Limit, Texte, levelcontent, Rezeptbuch	confirmation	correctness	
	1.RUN	Dungeon		Collected Items	confirmation		
	SQL	Statement	SQL Statement For Scroll X Strength y	Send User Statement	correctness		
	2.Run	Dungeon		Collected Items	confirmation	correctness	
	SQL	Statement	SQL Statement For Scroll X Strength y	Send User Statement	correctness		
else if						

Abbildung 1.1: Veranschaulichung des Spielablaufes

2 Produkteinsatz

2.1 Anwendungsbereiche

The SQL Alchemist wird entwickelt, um den Umgang und das praktische Anwenden von SQL-Anfragen einfach und spielerisch üben, beziehungsweise lernen zu können. Dies ist oftmals nicht ohne Weiteres möglich, da es ein großer Aufwand ist sich selbst einen Datenbankserver zu installieren, ihn mit Daten und eigenem geeigneten Schema zu füllen und sich dafür eigene Aufgaben auszudenken, sowie diese auf Korrektheit zu überprüfen. Die Anwendung wird nicht explizit Bezug auf die Vorlesung nehmen, sondern weitestgehend autonom funktionieren, mit Ausnahme des „Homework-Mode“, der durch die Wissenschaftlichen Mitarbeiter der TU Braunschweig genutzt werden kann, um den Studenten der RDB1-Vorlesung SQL-Hausaufgaben zu stellen. Dieser Modus ist nur durch die Anmeldung als Student mit bekannter y-Nummer erreichbar.

2.2 Zielgruppen

Die Studenten der TU Braunschweig sind die Hauptzielgruppe. Um das Modul „RDB1“ erfolgreich abzuschließen ist es notwendig, die geforderten Pflichthausaufgaben im Homework-Mode zu absolvieren. Damit die Studenten motivierter lernen können, enthält der Story-Modus ein „Jump&Run“-Minigame. Will sich der Student jedoch optimal auf die Klausur vorbereiten, um die SQL-Anfragen zu vertiefen, ist es hilfreich so viel wie möglich zu üben. Dies ist durch den Trivia-Mode möglich, in dem der Spieler Aufgaben in verschiedenen, wählbaren Schwierigkeitsgraden bearbeiten kann.

Auch für die wissenschaftlichen Mitarbeiter ist die Anwendung von Nutzen. Die gestellten Hausaufgaben müssen nicht mehr von Hand einzeln kontrolliert werden, da die Hausaufgaben in der Anwendung selbst geprüft und automatisch als abgeschlossen markiert werden, wenn alle Aufgabenteile korrekt bearbeitet wurden. Sie müssen so nur die wöchentlichen Aufgaben in die Datenbank eintragen.

Für die Professoren ist die Anwendung insofern interessant, dass sie einen neuen Weg darstellt, um die Studenten für etwas sehr theoretisches, aber vorlesungsrelevantes zu begeistern und ihnen so eine motivierende Möglichkeit geboten wird, diese wichtigen Fähigkeiten auf angenehme

Weise zu verinnerlichen. Da SQL aber nicht nur an der Universität wichtig ist, sondern vor allem in der Arbeitswelt eine wesentliche Rolle spielt, spricht die Anwendung auch private Nutzer an. Diese können ihr Wissen auf diesem Gebiet erweitern und festigen oder sogar völlig neu zu erlernen. Neue Nutzer, also keine Studenten, können sich mit ihrer E-Mail-Adresse anmelden. Für diese Gruppe fällt der Homework-Mode weg und ist somit nicht nutzbar. Eventuell könnte die Anwendung später auch von anderen Universitäten in ihrem Vorlesungsbetrieb unterstützend eingesetzt werden. Das Minispiel ist nur in Verbindung mit dem Lernen verwendbar. Ein Einzelmodus nur für das Minispiel gibt es nicht.

2.3 Betriebsbedingungen

Um die Anwendung nutzen zu können, wird ein internetfähiges Endgerät mit installiertem Webbrowser benötigt, zum Beispiel ein Computer, Smartphone oder Tablet. Durch die Plattformunabhängigkeit kann sie auch unterwegs auf dem Smartphone verwendet werden. Da es zudem keine App ist, die auf das Gerät geladen werden muss, sind keine manuellen Aktualisierungen durch den Benutzer nötig. Alle Updates können durch die Administratoren durchgeführt werden, ohne dass sie erst durch eine Handlung des Nutzers wirksam gemacht werden müssen. Der Homework-Mode erfordert während des Semesters wöchentliche Betreuung durch einen wissenschaftlichen Mitarbeiter, der die aktuellen Aufgaben in die Datenbank einfügt. Auch die Serverapplikation, sprich das Admin-Tool, ist über einen einfachen Internetbrowser erreichbar.

3 Produktübersicht

3.1 Die Registration

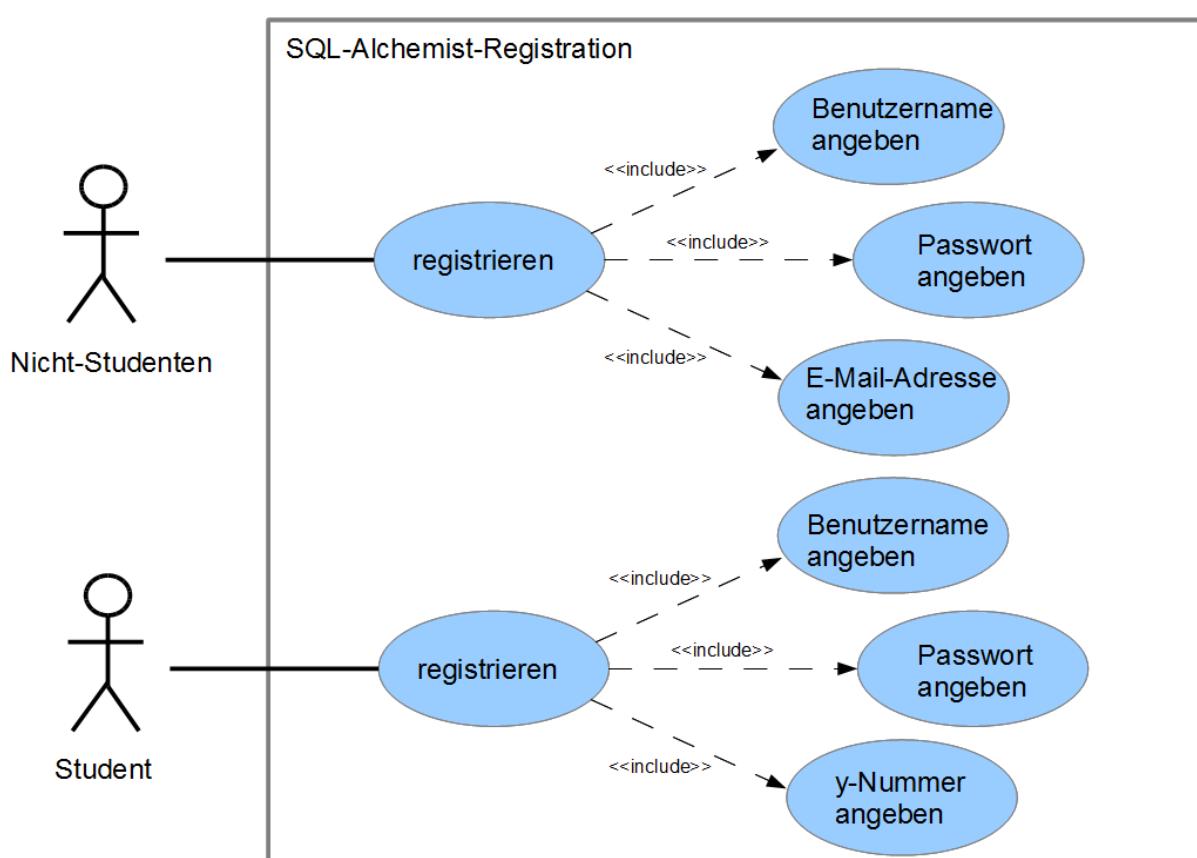


Abbildung 3.1: Die Registration

In diesem Use-Case-Diagramm werden die Abläufe aus Nutzersicht beschrieben, die zur Registration nötig sind. Dabei muss zwischen Studenten und Nicht-Studenten unterschieden werden. Studenten müssen zur Registration ihre y-Nummer, einen Benutzernamen und das zu ihrer y-Nummer gehörige Passwort angeben.

Nicht-Studenten müssen zur Registration eine E-Mail-Adresse, einen Benutzernamen und ein Passwort angeben.

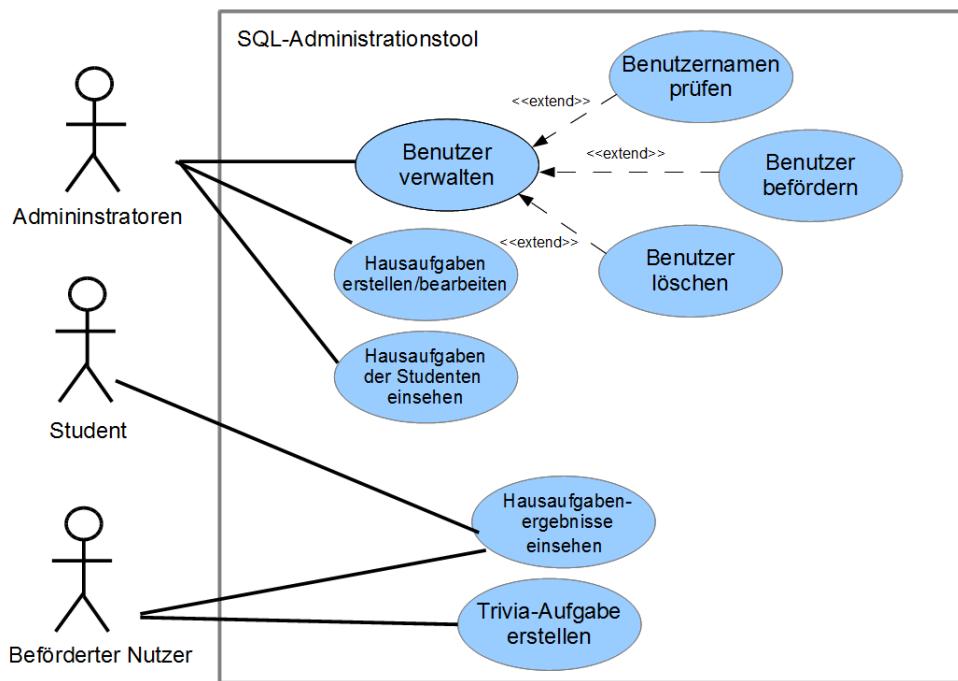


Abbildung 3.2: Das Admintool

In diesem Use-Case-Diagramm ist zu sehen, welche Möglichkeiten das Admin-Tool bereit stellt. Dabei gibt es verschiedene Nutzergruppen mit verschiedenen Rechten. Externe registrierte Benutzer haben standardmäßig keinen Zugriff auf das Admin-Tool.

Registrierte Studenten können im Admin-Tool einsehen, wie sie in ihren Hausaufgaben abgeschnitten haben. Das schließt sowohl die aktuellen, als auch die voran gegangenen ein. Beförderte Benutzer haben, zusätzlich zu ihren Standardfunktionen (externe keine, Studenten Hausaufgaben-Übersicht) weiterhin die Möglichkeit, selber Aufgaben/-pakete zuerstellen und die erstellten Aufgaben von anderen beförderten Benutzern zu bewerten.

Administratoren erhalten die Rechte zur Nutzer-, Inhalts- und Hausaufgabenverwaltung.

3.2 Der Story-Mode

Das folgende Aktivitätsdiagramm zeigt detailliert den Ablauf der Story. Um die Story zu spielen loggt sich der User ein und wählt im Hauptmenü den Reiter „Story“ aus. An diesem Punkt hat er bereits die Möglichkeit sich umzuentcheiden. Möchte er tatsächlich die Story spielen, so gelangt man in das so genannte Laboratory, wählt ein Rezept bzw. Scroll aus und braut sich seinen gewünschten Trank, indem er via SQL -Statement die dafür gebrauchten Materialien aus dem Schrank „Secreterry“ anfordert. Anschließend hat er die Möglichkeit entweder aus dem Story-Modus rauszugehen oder das Minispiel im Dungeon zu starten. Dort muss er alle Hürden überwinden und kann sich dann dazu entscheiden den Prozess des Trankbrauens zu wiederholen oder den Modus zu verlassen.

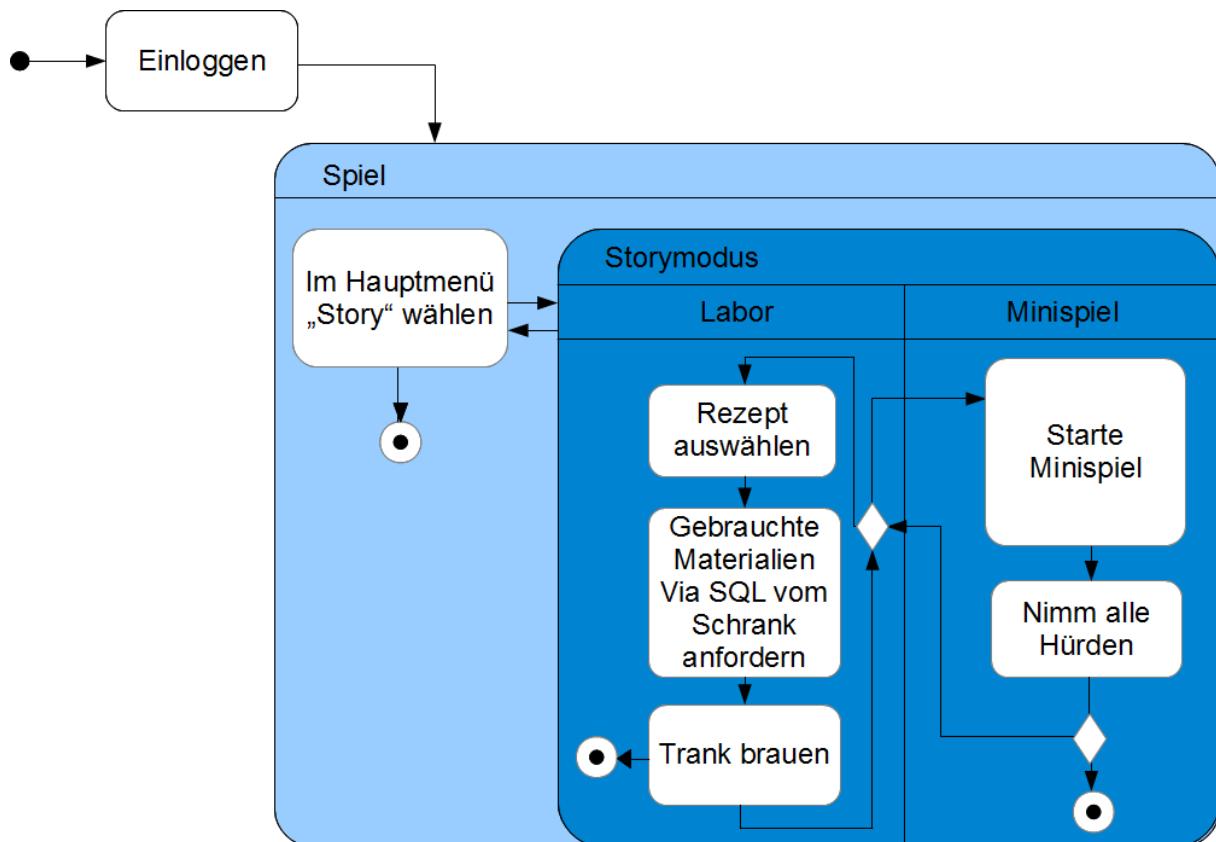


Abbildung 3.3: Spielablauf im Story-Modus

3.3 Das Spiel

Im folgenden Aktivitätsdiagramm ist der Vorgang des Logins und der Registration beschrieben. Nach dem Spielstart ist der Login-Bildschirm zusehen. Ist der Nutzer noch nicht registriert kann

er dies, nach Druck auf den „Sign up“ Knopf, tun. Hier muss er, je nach dem ob er Student ist oder nicht, entweder seine y-Nummer, das dazugehörige Passwort, oder eine E-Mail-Adresse und ein selbst ausgedachtes Passwort angeben. Dazu gehört in beiden Fällen noch ein Benutzername. Nach erfolgreicher Registrierung wird man in das Hauptmenü weitergeleitet.

Sind E-Mail-Adresse, bzw. y-Nummer, und Passwort schon registriert, kann man diese im Login-Bildschirm angeben und gelang danach ins Hauptmenü. Der Benutzername wird für den Login-Vorgang nicht benötigt.

Das folgende Use-Case-Diagramm zeigt den Großteil des Spielablaufs. Loggt sich der User ein, so gelangt er in das Hauptmenü. In diesem Menü hat er verschiedene Möglichkeiten zu interagieren. Zum einen kann er sein Profil einsehen. Dort hat er die Option sowohl seinen Avatar zu ändern, als auch seine Benutzerdaten einzusehen. Möchte der User wissen welche Nutzerrechte er hat, so kann er dies ebenfalls dort tun. Zum anderen kann der User unterschiedliche Ranglisten einsehen, andere Spieler suchen und sich deren Rang anzeigen lassen.

Eine weitere Möglichkeit in dem Menü voranzuschreiten, ist die, persönliche Einstellungen zu verändern. Diese sind:

- Der User kann Audio-Optionen ändern.
- Er kann seinen Spielstand zurücksetzen.
- Er hat auch die Möglichkeit sein Profil komplett zu löschen.
- Er kann sein Passwort ändern.
- Er kann seinen Benutzernamen löschen.

Außerdem kann der User im Shop unterschiedliche Erweiterungen für sein Spielfigur erwerben.

Des Weiteren kann der Benutzer einen von drei verschiedene Spiel-Modi wählen. Diese sind alle untereinander verknüpft. Betritt der Spieler den Hausaufgaben-Modus, so kann er dort, die von einem Wissenschaftlichen Mitarbeiter des Dozenten der Vorlesung RDB1 gestellten Aufgaben lösen. Möchte man diese bearbeiten, so schließt das die Benutzung des SQL-Trainers ein. Dieser ist ebenfalls aus dem Trivia-Modus erreichbar. Der Trainer ist auch fester Bestandteil des Story-Modes. Möchte der User die Story spielen, so impliziert das ebenfalls das Spielen des Minispiele. Im Story-Modus ist dabei die Möglichkeit gegeben, vor Start der eigentlichen Story, ein Tutorial zu absolvieren. Dieses ist hierbei aber eine freiwillige Option.

Die letzte Option die sich im Hauptmenü noch bietet, ist die, sich auszuloggen.

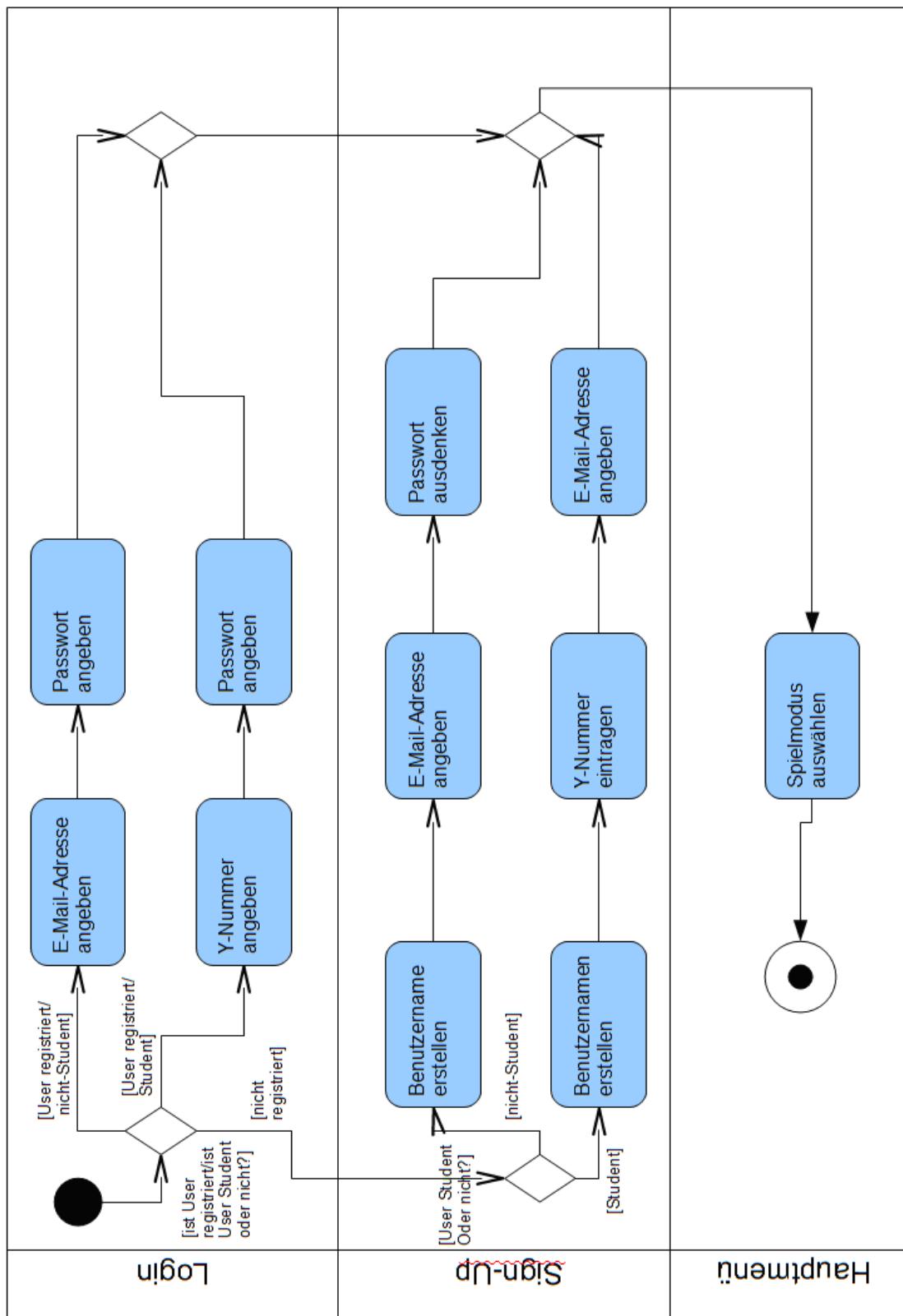


Abbildung 3.4: Login-/Registrationsvorgang

DAS GROSSE SQL-SPIEL

The SQL-Alchemist

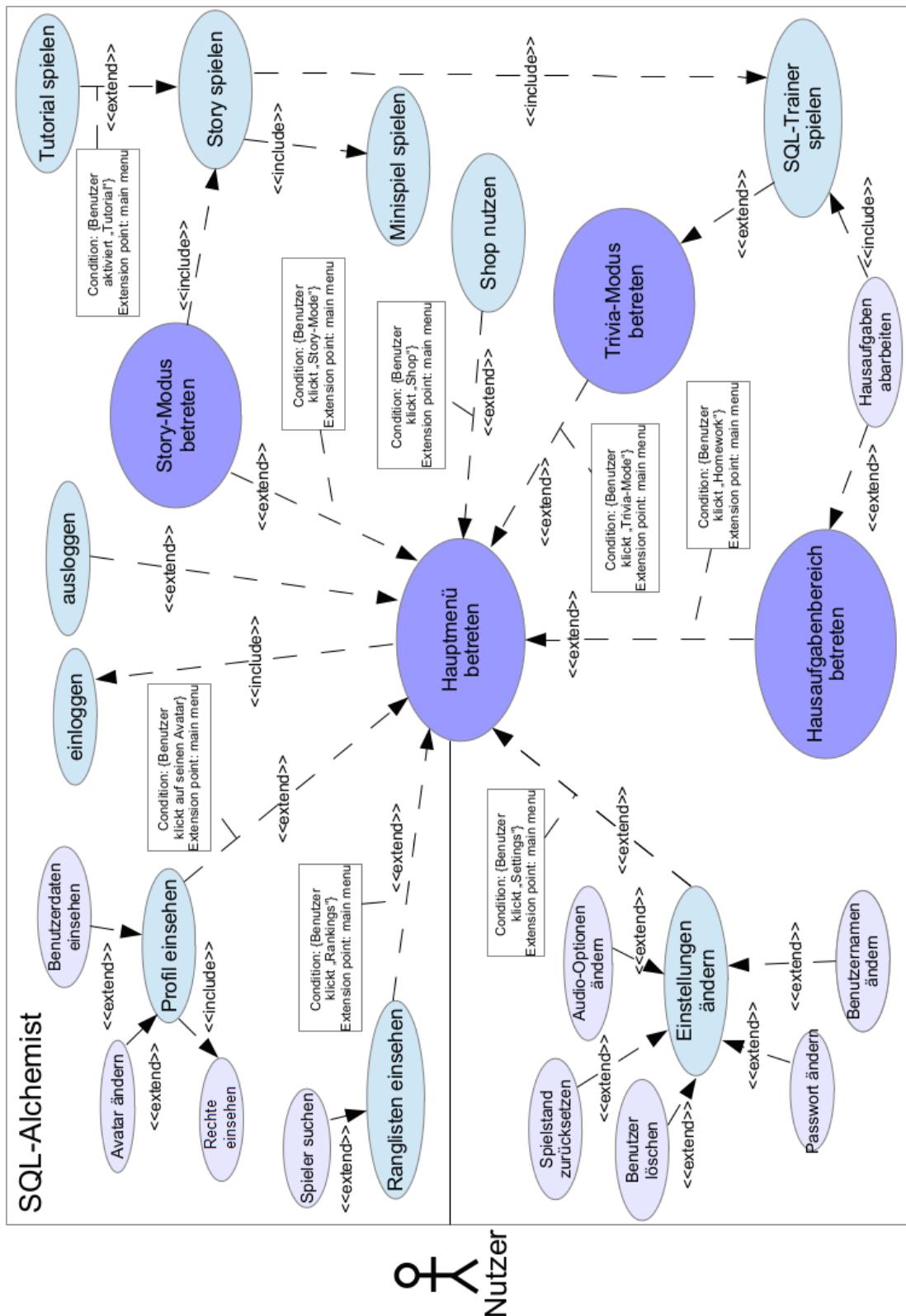


Abbildung 3.5: Der Großteil des eigentlichen Spiels

4 Produktfunktionen

4.1 Nutzer registrieren

Nutzer registrieren *(F10)*

Anwendungsfall: Ein neuer Benutzer möchte Zugang zum Spiel erhalten.

Anforderung: *(RM4), (RM8), (RC2)*

Ziel: Der Benutzer wird registriert und erhält eigene Login-Daten.

Vorbedingung: Der Nutzer ist noch nicht registriert.

Nachbedingung Erfolg: Der Nutzer verfügt über eigene eindeutige Login-Daten und wird zum Login-Bereich weitergeleitet.

Nachbedingung Fehlschlag: Die eingegebenen Daten waren nicht vollständig und die Registrierung schlägt fehl; es wird eine entsprechende Fehlermeldung angezeigt.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat die zum Spiel gehörende URL eingegeben und klickt auf „Sign Up“.

Beschreibung:

1. Der Nutzer gibt in die vorgesehenen Texteingabefelder einen Benutzernamen, ein Passwort, sowie eine E-Mail-Adresse oder y-Nummer ein.
2. Der Nutzer bestätigt seine Eingaben mit einem Klick auf einen Button.
3. Verfügt der Nutzer über eine y-Nummer, so ist er automatisch verifiziert, andernfalls erhält er eine E-Mail mit einem Aktivierungslink, den er zur Verifikation anklicken muss.
4. Der Nutzer wird anschließend zum Login-Bereich weitergeleitet.

4.2 Nutzer anmelden

Nutzer anmelden $\langle F20 \rangle$

Anwendungsfall: Einloggen des Nutzers um Zugang zum Spiel zu erhalten.

Anforderung: $\langle RM8 \rangle$, $\langle RC2 \rangle$

Ziel: Der Nutzer wird angemeldet.

Vorbedingung: Der Nutzer muss sich registriert und dadurch einen Account erstellt haben.

Nachbedingung Erfolg: Die eingegebenen Daten wurden als korrekt verifiziert und der Nutzer wird auf den Hauptbildschirm weitergeleitet.

Nachbedingung Fehlschlag: Die eingegebenen Daten waren nicht korrekt und dem Nutzer wird der Zugang verweigert; es wird eine entsprechende Fehlermeldung angezeigt.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat die zum Spiel gehörende URL eingegeben und wurde zum Login-Bildschirm weitergeleitet.

Beschreibung:

1. Der Nutzer gibt seine Email-Adresse/y-Nummer in das dafür vorgegebene Feld ein.
2. Der Nutzer gibt sein Passwort in das dafür vorgesehene Feld ein.
3. Der Nutzer klickt auf „Einloggen“.
4. Das System gleicht die eingegebenen Daten mit den Nutzerdaten in der Datenbank ab.
5. Der Nutzer wird zum Hauptbildschirm weitergeleitet.

4.3 Nutzer abmelden

Nutzer abmelden $\langle F30 \rangle$

Anwendungsfall: Der Nutzer will das Spiel verlassen und sich abmelden.

Anforderung: -

Ziel: Der Nutzer wird abgemeldet.

Vorbedingung: Der Nutzer muss sich im Hauptmenü befinden.

Nachbedingung Erfolg: Die Abmeldung ist erfolgt und der Nutzer wird zum Anmeldebereich weitergeleitet. Ohne erneute Anmeldung ist keine Aktivität im Spiel mehr möglich.

Nachbedingung Fehlschlag: Es gab einen Fehler beim Ausführen der Abmeldung und eine entsprechende Meldung wird ausgegeben.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat im Hauptmenü auf den „Logout“-Button geklickt.

Beschreibung:

1. Der Nutzer wird abgemeldet und verlässt das Spiel

4.4 Profil einsehen

Profil einsehen $\langle F40 \rangle$

Anwendungsfall: Der Nutzer möchte seine bisher gemachten Fortschritte nachverfolgen und ruft zu diesem Zweck sein Profilfenster auf.

Anforderung: $\langle RM2 \rangle$, $\langle RM4 \rangle$, $\langle RS4 \rangle$, $\langle RS5 \rangle$

Ziel: Der Nutzer kann sein Profil einsehen.

Vorbedingung: Der Nutzer muss sich eingeloggt haben.

Nachbedingung Erfolg: Dem Nutzer wird sein Profilfenster, in welchem dessen Fortschritte in Form von Statistiken festgehalten werden, angezeigt.

Nachbedingung Fehlschlag: Es gab einen Fehler beim Aufruf und eine entsprechende Meldung wird ausgegeben.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat auf seinen Avatar geklickt.

Beschreibung:

1. Das Profilfenster wird geöffnet
2. Der Nutzer sieht seine Statistiken und Nutzerdaten ein.

4.5 Benutzernamen ändern

Benutzernamen ändern $\langle F50 \rangle$

Anwendungsfall: Der Nutzer will seinen Benutzernamen ändern.

Anforderung: $\langle RM4 \rangle$

Ziel: Der Benutzername, der im Spiel und in den Ranglisten verwendet wird, wird geändert.

Vorbedingung: Der Nutzer muss sich eingeloggt haben und sich im „Settings“-Menü befinden.

Nachbedingung Erfolg: Dem Nutzer wird sein Profilfenster mit dem neuen Benutzernamen angezeigt.

Nachbedingung Fehlschlag: Es gab einen Fehler beim Ändern des Benutzernamens und eine entsprechende Meldung wird ausgegeben. Der Nutzer behält seinen alten Benutzernamen.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt auf den „Change Username“-Button.

Beschreibung:

1. Es öffnet sich ein Texteingabefenster, in dem der Nutzer seinen neuen Benutzernamen eingibt.
2. Er bestätigt seine Eingabe, anschließend prüft das Programm, ob der Benutzername verwendet werden darf (also noch nicht vergeben ist).

4.6 Passwort ändern

Passwort ändern *(F60)*

Anwendungsfall: Ein Nutzer will sein Passwort ändern.

Anforderung: *(RM6)*

Ziel: Das Passwort wird geändert.

Vorbedingung: Der Nutzer muss eingeloggt sein und sich im „Settings“-Menü befinden.

Nachbedingung Erfolg: Der Nutzer hat sein neues Passwort zugewiesen bekommen und gelangt zurück ins „Settings“-Menü.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgegeben. Der Nutzer behält sein bisheriges Passwort.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im „Settings“-Menü auf den Button „Change Password“

Beschreibung:

1. Es öffnet sich ein Texteingabefeld, in das der Nutzer das neue Passwort eingibt.
2. Der Benutzer muss sein altes Passwort zur Verifikation in einem weiteren Texteingabefeld eingeben.
3. Der Nutzer bestätigt seine Eingaben.

4.7 Avatar ändern

Avatar ändern *(F70)*

Anwendungsfall: Ein Nutzer will seinen Avatar (Spielfigur) ändern.

Anforderung: *(RM4), (RS3)*

Ziel: Der Avatar des Nutzers wird geändert.

Vorbedingung: Der Nutzer muss eingeloggt sein und sich im „Profile“-Menü befinden.

Nachbedingung Erfolg: Der neue Avatar des Nutzers wird übernommen.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgetragen. Der Nutzer behält seinen bisherigen Avatar.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im „Profile“-Menü auf den Button „Change Avatar“.

Beschreibung:

1. Es wird eine Liste angezeigt, die alle möglichen Avatare enthält.
2. Der Nutzer wählt einen Avatar aus der Liste aus.
3. Die Auswahl wird über einen Button bestätigt.

4.8 Benutzer löschen

Benutzer löschen *(F80)*

Anwendungsfall: Ein Nutzer will seinen Account löschen.

Anforderung: *(RM4), (RM5)*

Ziel: Der Nutzeraccount wird gelöscht.

Vorbedingung: Der Nutzer muss eingeloggt sein und sich im „Settings“-Menü befinden.

Nachbedingung Erfolg: Der Account des Nutzers wurde, inklusive aller auf diesen bezogenen Daten (ausgenommen die von diesem Nutzer erstellten SQL-Abfragen), aus der Nutzerverwaltung gelöscht. Der Nutzer wird automatisch ausgeloggt.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgetragen. Der Account bleibt unverändert.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im „Settings“-Menü auf den Button „Delete User“.

Beschreibung:

1. Der Nutzer wird gefragt, ob er wirklich seinen Account löschen möchte.

2. Der Nutzer bestätigt sein Vorhaben und sein Account wird gelöscht, außerdem wird er automatisch ausgeloggt.

4.9 Audioeinstellungen bearbeiten

Audioeinstellungen bearbeiten $\langle F90 \rangle$

Anwendungsfall: Ein Nutzer will die Soundeffekte oder die Musik aktivieren/deaktivieren.

Anforderung: -

Ziel: Die Audioeinstellungen werden geändert.

Vorbedingung: Der Nutzer muss eingeloggt sein und sich im „Settings“-Menü befinden.

Nachbedingung Erfolg: Die Audioeinstellungen wurden geändert und deren Status wurde im Back-End gespeichert.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgegeben. Die bisherigen Einstellungen bleiben bestehen.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im „Settings“-Menü auf den Button „Audio Settings“.

Beschreibung:

1. Es öffnet sich ein Fenster, in dem der Nutzer die Soundeffekte und die Musik unabhängig voneinander an- und abschalten kann.
2. Der Nutzer bestätigt die Änderungen.

4.10 Spielstand zurücksetzen

Spielstand zurücksetzen $\langle F100 \rangle$

Anwendungsfall: Ein Nutzer möchte das Spiel erneut von vorne beginnen.

Anforderung: -

Ziel: Der Spielstand wird zurückgesetzt.

Vorbedingung: Der Nutzer muss eingeloggt sein und sich im „Settings“-Menü befinden.

Nachbedingung Erfolg: Der Spielstand ist zurückgesetzt.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgegeben. Der bisherige Speicherstand bleibt bestehen.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im „Settings“-Menü auf den Button „Reset Story“.

Beschreibung:

1. Der Nutzer wird gefragt, ob er wirklich seinen Spielstand zurücksetzen möchte.
2. Der Nutzer bestätigt das Löschen.

4.11 Tutorial spielen

Tutorial spielen $\langle F110 \rangle$

Anwendungsfall: Ein Nutzer startet den „Story-Mode“.

Anforderung: -

Ziel: Der Nutzer lernt die Grundsteuerung des Spieles kennen.

Vorbedingung: Der Nutzer muss eingeloggt sein und den „Story-Mode“ gestartet haben. Außerdem muss das Tutorial aktiviert sein.

Nachbedingung Erfolg: Der Nutzer hat das Tutorial erfolgreich absolviert und der Anfang der Story wird gestartet.

Nachbedingung Fehlschlag: Es gab einen Fehler und das Tutorial konnte nicht gestartet werden.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer startet den „Story-Mode“.

Beschreibung:

1. Das Tutorial wird gestartet.
2. Der Nutzer erhält Anweisungen, die dieser befolgen muss.
3. Nach Abschluss aller Aufgaben wird das Tutorial abgeschlossen.

4.12 Story spielen

Story spielen $\langle F120 \rangle$

Anwendungsfall: Ein Nutzer möchte den „Story-Mode“ spielen.

Anforderung: $\langle RM7 \rangle$

Ziel: Der Nutzer spielt die Story.

Vorbedingung: Der Nutzer muss eingeloggt sein und sich im Hauptmenü befinden.

Nachbedingung Erfolg: Der Nutzer spielt den „Story-Mode“ bis er von selbst abbricht oder die Story bis zum Ende absolviert hat.

Nachbedingung Fehlschlag: Es gab einen Fehler und die Story konnte nicht gestartet werden. Eine entsprechende Meldung wird ausgegeben.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im Hauptmenü auf den Button „Story-Mode“.

Beschreibung:

1. Der Story-Durchlauf wird gestartet.
2. Der Nutzer spielt die Story, bis er sie zu einem beliebigen Zeitpunkt abbricht.

4.13 SQL-Trainer spielen

SQL-Trainer spielen *(F130)*

Anwendungsfall: Der Nutzer spielt den „Trivia-“ oder „Story-Mode“ oder er bearbeitet seine Hausaufgaben.

Anforderung: *(RM3)*

Ziel: Der Nutzer löst, abhängig vom Modus, zufällige oder vorher festgelegte Aufgaben.

Vorbedingung: Der Nutzer muss eingeloggt sein und einen der Spielmodi spielen.

Nachbedingung Erfolg: Der Nutzer löst die verschiedenen Aufgaben und die Ergebnisse werden in einer Nachbesprechung angezeigt.

Nachbedingung Fehlschlag: Das Spiel weist einen Fehler beim Aufruf der Aufgaben auf und wirft eine Fehlermeldung aus.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer startet einen der verschiedenen Spielmodi.

Beschreibung:

1. Das Nutzer wird zum SQL-Modul weitergeleitet.
2. Die erste Aufgabe wird angezeigt und vom Spieler bearbeitet.
3. Es folgen weitere Aufgaben, bis das aktuelle Aufgabenpaket abgeschlossen ist.
4. Dem Nutzer werden die Ergebnisse angezeigt.

4.14 Minispiel spielen

Minispiel spielen $\langle F140 \rangle$

Anwendungsfall: Der Nutzer spielt den „Story-Mode“.

Anforderung: $\langle RM1 \rangle$, $\langle RS1 \rangle$, $\langle RC5 \rangle$

Ziel: Der Nutzer spielt das Minispiel und versucht die jeweiligen Level zu lösen.

Vorbedingung: Der Nutzer muss eingeloggt sein und den vorhergehenden SQL-Abschnitt gelöst haben. Zusätzlich muss er sich „Story-Mode“ befinden.

Nachbedingung Erfolg: Der Nutzer spielt die jeweils gestellten Level und die Ergebnisse werden in einer Nachbedingung angezeigt. Dann wird er in den nächsten SQL-Abschnitt weitergeleitet.

Nachbedingung Fehlschlag: Das Spiel weist einen Fehler beim Aufruf des Minispiele auf und wirft eine entsprechende Meldung aus.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat den vorhergehenden SQL-Abschnitt gelöst.

Beschreibung:

1. Das Minispiel wird gestartet.
2. Die ersten vier Level werden nacheinander vom Spieler gelöst.
3. Ein Endlevel wird aufgerufen und vom Spieler gelöst.

4.15 Hausaufgaben bearbeiten

Hausaufgaben bearbeiten $\langle F150 \rangle$

Anwendungsfall: Der Nutzer ruft die ihm zugeordneten Hausaufgaben auf um diese zu bearbeiten.

Anforderung: $\langle RM3 \rangle$, $\langle RM7 \rangle$

Ziel: Der Nutzer bearbeitet die vom Spiel gestellten Aufgaben.

Vorbedingung: Der Nutzer muss eingeloggt sein und sich im „Homework-Mode“ befinden.

Nachbedingung Erfolg: Der Nutzer bearbeitet die gestellten Aufgaben und die Ergebnisse werden in einer Nachbesprechung angezeigt.

Nachbedingung Fehlschlag: Es gab einen Fehler beim Aufruf der Aufgaben und eine entsprechende Meldung wird ausgegeben.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat im „Homework“-Menü auf den Button „Do Homework“ geklickt.

Beschreibung:

1. Der Hausaufgabenmodus wird gestartet.
2. Die erste Aufgabe wird angezeigt und vom Spieler bearbeitet.
3. Es folgen weiter Aufgaben solange bis das komplette Hausaufgabenpaket abgearbeitet wurde.
4. Die Nachbesprechung wird angezeigt.

4.16 Ranglisten einsehen

Ranglisten einsehen *(F160)*

Anwendungsfall: Aufruf der Ranglisten durch den Nutzer.

Anforderung: *(RM4), (RM7), (RS4), (RS5)*

Ziel: Die Ranglisten wird dem Nutzer angezeigt.

Vorbedingung: Der Nutzer muss sich eingeloggt haben und sich im Hauptmenü befinden.

Nachbedingung Erfolg: : Dem Nutzer werden die Ranglisten angezeigt.

Nachbedingung Fehlschlag: Es gab einen Fehler beim Aufruf der Ranglisten und es wird eine entsprechende Meldung angezeigt.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat auf den „Leaderboard“-Button geklickt.

Beschreibung:

1. Das Ranglistenfenster wird geöffnet.

4.17 Spieler suchen

Spieler suchen *(F170)*

Anwendungsfall: Der Nutzer möchte einen Nutzer anhand des Benutzernamens suchen.

Anforderung: *(RM4)*

Ziel: Dem Nutzer werden die Ergebnisse des gesuchten Benutzers angezeigt.

Vorbedingung: Der Nutzer muss sich eingeloggt haben und sich im Hauptmenü befinden. Er muss den Benutzernamen kennen, nach dem er suchen will.

Nachbedingung Erfolg: : Dem Nutzer wird den gefundenen Nutzer angezeigt.

Nachbedingung Fehlschlag: Es gab einen Fehler beim Suchen der Nutzer in den Ranglisten und es wird eine entsprechende Meldung angezeigt.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer hat auf den „Search User“-Button geklickt.

Beschreibung:

1. Es öffnet sich ein Texteingabefeld, in dem der zu suchende Benutzername eingetragen wird.
2. Der Nutzer bestätigt seine Eingabe.
3. Ihm wird das Ergebnis der Suche angezeigt.

4.18 Hausaufgabenergebnisse einsehen

Hausaufgabenergebnisse einsehen *(F180)*

Anwendungsfall: Ein Nutzer will seinen Stand bei den Hausaufgaben abrufen.

Anforderung: *(RM4), (RM6), (RM7)*

Ziel: Die Hausaufgabenergebnisse werden angezeigt.

Vorbedingung: Der Nutzer muss sich im Administrationstool angemeldet haben.

Nachbedingung Erfolg: Der Student erhält eine Übersicht über seine bisherigen Ergebnisse bei den Hausaufgaben.

Nachbedingung Fehlschlag: Bei der Ausgabe der Ergebnisse gab es einen Fehler und eine entsprechende Meldung wird ausgegeben.

Akteure: Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im Administrationstool auf „View Results“.

Beschreibung:

1. Es wird eine Liste angezeigt, die die Ergebnisse aller bisher bearbeiteten Hausaufgaben anzeigt.

4.19 Benutzer befördern

Benutzer befördern *(F190)*

Anwendungsfall: Ein Admin will einen Studenten oder einen regulären Nutzer befördern, damit dieser eigene SQL-Aufgaben erstellen darf.

Anforderung: *(RM4), (RM5)*

Ziel: Der beförderte Nutzer kann eigenständig SQL-Aufgaben erstellen.

Vorbedingung: Der zu befördernde Nutzer muss registriert sein und der Admin muss Zugang zum Admin-Tool haben.

Nachbedingung Erfolg: : Der Student oder regulärer Nutzer ist befördert und wird nun als beförderter Nutzer behandelt.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgetragen.

Akteure: Admin und Programm

Auslösendes Ereignis: Der Admin hat im Admin-Tool auf den Button „Promote User“ geklickt.

Beschreibung:

1. Der Admin gibt in einem Texteingabefeld den Benutzernamen des zu befördernden Studenten oder regulären Nutzers ein.
2. Die Eingabe wird mit einem Button bestätigt.

4.20 Einem Benutzer Adminrechte geben

Benutzer Adminrechte geben *(F200)*

Anwendungsfall: Ein Admin will einem anderen Nutzer Adminrechte geben.

Anforderung: *(RM4), (RM5)*

Ziel: Der ausgewählte Nutzer wird zum Admin.

Vorbedingung: Der zu befördernde Nutzer muss registriert sein und der Admin muss Zugang zum Admin-Tool haben.

Nachbedingung Erfolg: Der Nutzer ist befördert und wird nun als Admin behandelt.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgetragen.

Akteure: Admin und Programm

Auslösendes Ereignis: Der Admin hat im Admin-Tool auf den Button „Promote User“ geklickt.

Beschreibung:

1. Der Admin gibt in einem Texteingabefeld den Benutzernamen des zu befördernden Studenten oder regulären Nutzers ein.
2. Die Eingabe wird mit einem Button bestätigt.

4.21 Eine Trivia-Aufgabe erstellen

Trivia-Aufgabe erstellen $\langle F210 \rangle$

Anwendungsfall: Ein beförderter Nutzer will eigene Aufgaben für den „Trivia-Mode“ erstellen.

Anforderung: $\langle RM6 \rangle, \langle RM7 \rangle$

Ziel: Die bestehende Aufgabensammlung wird um die erstellte Aufgabe erweitert.

Vorbedingung: Der beförderte Nutzer muss sich im Admin-Tool angemeldet haben.

Nachbedingung Erfolg: Der beförderte Nutzer hat seine Aufgabe erstellt und gelangt zurück ins Admin-Tool.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgetragen.

Akteure: beförderter Nutzer und Programm

Auslösendes Ereignis: Der Nutzer klickt im Admin-Tool auf den Button „Create New Task“.

Beschreibung:

1. Es öffnet sich ein Texteingabefeld, in das der Nutzer die Aufgabenstellung mit der entsprechenden Lösung eingibt.
2. Anschließend bestätigt der Nutzer seine Eingaben.

4.22 Benutzeraufgaben bewerten

Benutzeraufgaben bewerten $\langle F220 \rangle$

Anwendungsfall: Ein Admin oder ein beförderter Nutzer möchte von anderen beförderten Benutzern erstellte Aufgaben bewerten.

Anforderung: $\langle RM4 \rangle, \langle RM5 \rangle$

Ziel: Von beförderten Nutzern erstellte Aufgaben sollen bewertet werden.

Vorbedingung: Der Admin oder beförderte Nutzer muss sich im Admin-Tool befinden.

Nachbedingung Erfolg: Die Aufgabe ist bewertet und die neue Bewertung wird mit den bereits vorhandenen Bewertungen verrechnet und der Nutzer wird auf den Hauptbildschirm weitergeleitet.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgetragen.

Akteure: Admin (oder beförderter Nutzers) und Programm

Auslösendes Ereignis: Der Admin oder beförderte Nutzer klickt im Admin-Tool auf den „Rate Tasks“-Button.

Beschreibung:

1. Dem Admin oder befördertem Benutzer erscheint eine Liste an (von beförderten Nutzern erstellte) Fragen.
2. Beim Klick auf eine Frage öffnet sich ein Fenster, in dem die Bewertung vorgenommen werden kann.

4.23 Hausaufgaben erstellen

Hausaufgaben erstellen *(F230)*

Anwendungsfall: Ein Admin erstellt neue Hausaufgaben, die von den Nutzern bearbeitet werden sollen.

Anforderung: *(RM6), (RM7)*

Ziel: Die Hausaufgaben müssen allen Studenten zugänglich sein.

Vorbedingung: Der Admin muss sich im Admin-Tool befinden.

Nachbedingung Erfolg: Die Hausaufgabe ist erstellt, wird in der Datenbank gespeichert und kann von den Studenten abgerufen werden.

Nachbedingung Fehlschlag: Es gab einen Fehler und eine entsprechende Meldung wird ausgetragen.

Akteure: Admin und Programm

Auslösendes Ereignis: Der Admin klickt im Admin-Tool auf den „Create Homework“-Button.

Beschreibung:

1. Der Admin gibt die Aufgabenstellung, die Lösung, die Anzahl möglicher Bearbeitungsversuche je Student und den Bearbeitungszeitraum in Eingabefelder ein.
2. Er bestätigt seine Eingabe.

5 Produktdaten

Produktdaten entsprechen sämtlichen relevanten Daten, die laufzeitpersistent gespeichert werden müssen. Dabei werden für jeden Nutzer einzeln die folgenden Daten benötigt:

Userdaten $\langle D10 \rangle$

Persönliche Daten des Nutzers:

- Profil (Referenz)
- E-Mail
- E-Mail bestätigt
- E-Mail Bestätigungscode
- y-Nummer (nur bei Studenten der TU BS)
- Matrikelnummer (nur bei Studenten der TU BS)
- Passworthash
- Passwort vergessen Code
- Nutzerberechtigung

Sitzungsdaten $\langle D20 \rangle$

Verbindungsdaten zum Client:

- User (Referenz)
- Cookie-Identifikationsnummer
- IP-Adresse
- Erstellungsdatum (Timestamp)
- Auslaufdatum (Timestamp)

Avatar $\langle D30 \rangle$

Avatar-Zuordnung:

- Name
- File-URL

- Skin-URL
- Preis

Profildaten *(D40)*

Daten des Nutzerprofils:

- Username
- Avatar (Referenz)
- Münzen
- Musikeinstellungen
- Toneinstellungen
- Characterattribute (Health, Defense, Speed, Jump, Beltslots)
- Neue Story

Avatarkauf *(D50)*

Speichert den Kauf eines Avatars:

- Profil (Referenz)
- Avatar (Referenz)
- Kaufdatum (Timestamp)

Aufgabenpakete *(D60)*

Zusammenstellung mehrerer Aufgaben:

- Urheber (Referenz zu User)
- Paketname
- Paketbeschreibung
- Erstelltdatum (Timestamp)
- Aufgabentyp (Hausaufgabe, allgemeines Trivia-Paket, Story-Paket)
- Lösungsbedingung (Begrenzung durch Zeit, Anzahl an Versuchen)
- Auslaufdatum (Timestamp)
- Zufällige oder festgelegte Reihenfolge

Aufgabenpaket-Fortschritt *(D70)*

Speicherung des Nutzerfortschrittes im Aufgabenpaket:

- Profil (Referenz)

- Aufgabenpaket (Referenz)
- Fortschritt (Aufgabennummer)
- Bearbeitungsbeginn (Timestamp)
- Bearbeitungsende (Timestamp)

Trank *(D80)*

Speichert die Daten der einzelnen Tränke:

- Trank-Art (Healing, Defense, Speed, Jump)
- Trank-Name

Tasche *(D90)*

Inventar des Benutzers:

- Profil (Referenz)
- Trank (Referenz)
- Trank-Stärke
- Position im Gürtel

Aufgabe *(D100)*

Einzelne SQL-Aufgaben:

- Urheber (Referenz zu User)
- Erstellungsdatum (Timestamp)
- Schwierigkeit
- Trank (Referenz)
- Rating
- Bearbeitungsdatum (Timestamp)
- Verfügbarkeit
- Vom Nutzer erstellt

Aufgabe in Aufgabenpaket *(D110)*

Beschreibt die Aufgaben im Aufgabenpaket

- Aufgabe (Referenz)
- Aufgabenpaket (Referenz)
- Reihenfolge

Aufgabenbearbeitung *(D120)*

Zeigt, ob ein Benutzer eine Aufgabe bereits gelöst hat

- Profil (Referenz)
- Aufgabe (Referenz)
- Lösungsdatum (Timestamp)
- Erfolgreich
- Bearbeitungsdauer

Aufgabenbewertung *(D130)*

Speichert Bewertungen zu Aufgaben von anderen Benutzern:

- Profil (Referenz)
- Aufgabe (Referenz)
- Bewertung
- Datum (Timestamp)

Aufgabenkommentar *(D140)*

Speichert Benutzerkommentare zu Aufgaben

- Profil (Referenz)
- Aufgabe (Referenz)
- Kommentar
- Datum (Timestamp)

Texte *(D150)*

Vorgefertigte Texte, die der narrative Character benutzt:

- Typ (Abhängig von Zeit oder Richtigkeit des Statements)
- Text

Story-Text *(D160)*

Textelemente der Story:

- Text
- Bedingung

Text in Aufgabenpaket *(D170)*

Texte für die Aufgabenpakete:

- Challenge (Referenz)

- Story-Text (Referenz)
- Reihenfolge

Schriftrolle *(D180)*

Speicherung der verschiedenen Schriftrollen Arten:

- Name
- Typ (Rezept, Buff)
- Stärkeindikator
- Attribut (Health, Defense, Speed, Jump)
- Potion (Referenz falls Typ = Rezept)
- Benutzt (Falls Typ = Buff)

Spieler besitzt Schriftrolle *(D190)*

Speichert, welche Schrifrollen bereits eingesammelt wurden:

- Schriftrolle (Referenz)
- Profil (Referenz)
- Besitzt

6 Nichtfunktionale Anforderungen

6.1 Funktionalität

Produktqualität	sehr gut	gut	normal	nicht relevant
Angemessenheit		x		
Richtigkeit	x			
Interoperabilität		x		
Ordnungsmäßigkeit	x	x		

Besonders penibel wird mit Fehlern in der SQL-Auswertung umgegangen, denn diese können falsche Lerneffekte erzeugen. Die Webapplikation soll die geforderten Produktfunktionen (Spielesoftware) umsetzen. Für die Motivationsförderung der Benutzer wird es ein „Minispiel“ geben, welches in verschiedenen Modi gespielt werden kann. Für die Analyse sowie die Lösung der Aufgaben wird die externe Schnittstelle zum Teamprojekt verwendet.

6.2 Sicherheit

Produktqualität	sehr gut	gut	normal	nicht relevant
Zuverlässigkeit	x			
Reife		x		
Fehlertoleranz	x			
Wiederherstellbarkeit		x		

Das Programm wird sehr zuverlässig sein, damit es in den Vorlesungs- und Hausaufgabenbetrieb eingebunden werden kann. Programminterne Fehler sind in Spielen ein großer Grund für Spaß- und Motivationsverlust und werden deshalb bestmöglich vermieden.

Die Sicherheit des Programms beschränkt sich auf den Zugriff zur Webapplikation. Der Zugriff ist nur durch die y-Nummer, beziehungsweise der E-Mail Adresse, und einem Passwort möglich.

6.3 Benutzbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Verständlichkeit		x		
Erlernbarkeit	x			
Bedienbarkeit			x	
Effizienz		x		
Zeitverhalten		x		
Verbrauchsverhalten		x		

Da dies im Fokus eine Lernsoftware ist, sollte auch das Erlernen der Benutzung schnell zu beherrschen sein. Dabei muss der Nutzer Vorkenntnisse in SQL haben, um die Aufgaben verstehen zu können.

6.4 Änderbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Analysierbarkeit		x		
Modifizierbarkeit		x		
Stabilität	x			
Prüfbarkeit			x	
Übertragbarkeit	x			
Anpassbarkeit				x
Installierbarkeit				x
Konformität				x
Austauschbarkeit				x

Das Programm sollte sehr stabil laufen. Abstürze würden den Hausaufgabenbetrieb extrem behindern, was sowohl für die Lehrenden als auch für den Nutzer extrem ärgerlich wäre. Die Übertragbarkeit ist dadurch gegeben, dass es sich um eine Webapplikation handelt und somit in gängige Umgebungen portiert werden kann. Der Quellcode des Programms sollte gut analysierbar sein, um Änderungen ohne großen Zeitaufwand implementieren zu können. Mögliche Änderungen am Quellcode sollen keinen Einfluss auf die Stabilität der Spielsoftware haben, sodass ein fehlerfreier Betrieb gewährleistet werden kann.

6.5 Qualitätsanforderungen

- ⟨Q10⟩ Das Produkt soll Benutzerfreundlich sein.

- $\langle Q20 \rangle$ Das Programm soll zuverlässig und stabil laufen, um einen reibungslosen Ablauf der Hausaufgaben zu garantieren. $\langle F20 \rangle$
- $\langle Q30 \rangle$ Der SQL-Frage-Antwort Ablauf soll nach dem aktuellen SQL-Standard exakt kontrolliert werden.
- $\langle Q40 \rangle$ Das Produkt soll plattformunabhängig sein

7 Benutzeroberfläche/Schnittstellen

Benutzeroberfläche:

Login $\langle UI10 \rangle$

Der Login-Screen bietet dem User die Möglichkeit, sich in sein Profil einzuloggen, oder eine Registration durchzuführen.

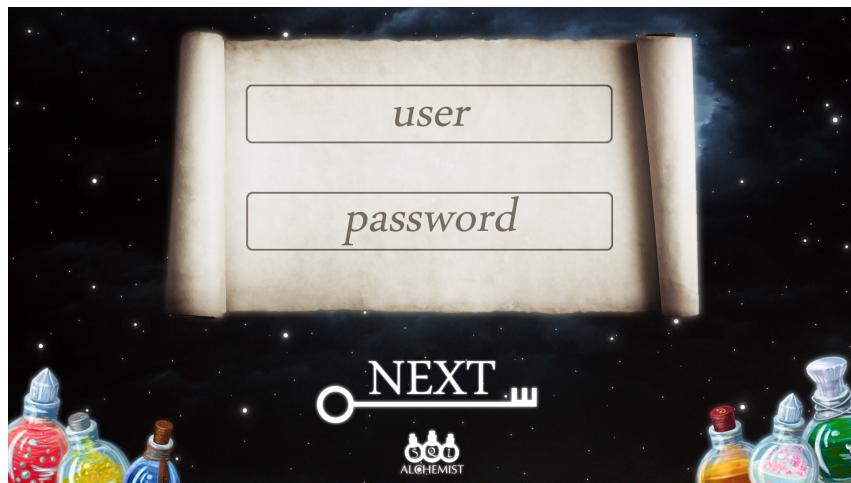


Abbildung 7.1: Login

Minispiel-HUD $\langle UI30 \rangle$

Das Minispiel wird über die Maus gesteuert. Durch Klicks kann der Spielcharakter springen. Durch das Klicken der Potions im linken unteren Rand des Screens können diese ebenfalls eingesetzt werden. Durch das Klicken des „Beenden“-Buttons, dargestellt durch ein X, kann das Minispiel befrüht beendet werden.

SQL-Trainer-Screen $\langle UI40 \rangle$

Hier wird es dem User möglich sein, eine Aufgabe zu lösen. Dies geschieht über die Eingabe in ein Textfeld. Über einen Button kann das eingegebene Statement auf Korrektheit überprüft werden. Eventuelle Fehlerhinweise werden ebenfalls in einem Textfeld visualisiert.

Menüführung $\langle UI20 \rangle$

Das Hauptmenü ist das zentrale Userinterface zur Kontrolle des Programmablaufes.

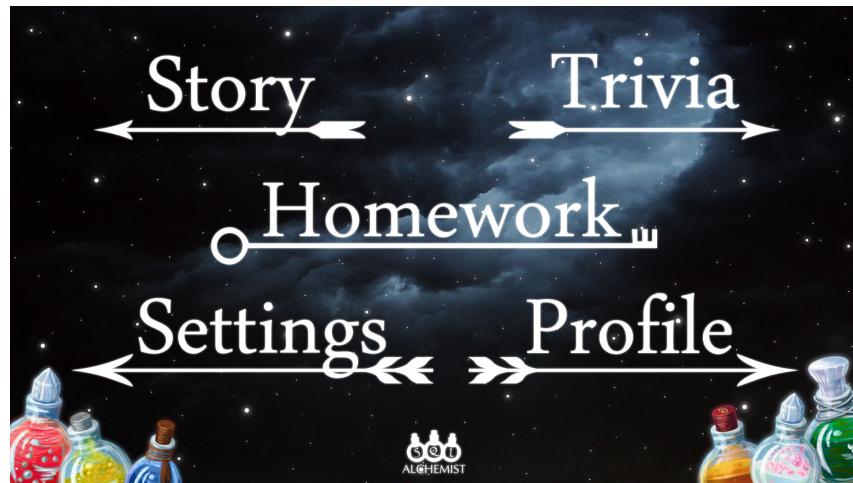


Abbildung 7.2: Menüführung

Hier kann der User über Mauseingaben entweder weitere Teilmenüs aufrufen, wie den Trivia-, den Storymodus. Auch die Ranglisten können in diesem Menü eingesehen werden. Weiterhin bietet sich den Studenten unter den Usern die Möglichkeit, sich ihren Hausaufgaben zu widmen. Hierfür ist ebenfalls eine eigener Menüpunkt vorgesehen. Außerdem gibt es die Möglichkeit sich in das Untermenü der Settings zu begeben, wo der User sämtliche Einstellungen verändern kann.

Admin-Tool (UI50)

Das Admin-Tool ist das zentrale Userinterface zur Kortrolle der User- und Aufgabenverwaltung.

A screenshot of a web-based login form titled 'Sign In'. It features two input fields: 'E-Mail or Y-ID' and 'Password', both with placeholder text. To the right of the password field is a green 'Login' button.

Abbildung 7.3: Login zum Admin-Tool

Hier kann sich der User einloggen um nachzuschauen ob er die Hausaufgaben bestanden hat oder nicht. Des Weiteren kann sich hier der Admin anmelden um Einstellungen zu verändern

oder Aufgabenpakete in die Datenbank einzupflegen. Jeder beförderte User wird die Möglichkeit haben über diesen Login selbst SQL-Aufgaben zu erstellen.

8 Technische Produktumgebung

8.1 Software

Front-End/ Minispiel:

- beliebiges Betriebssystem
- ein gängiger Browser
- aktiviertes Javascript im Browser

Back-End/ Administration:

- ein gängiger Browser
- Datenbank
- LDAP-Anbindung zur Studentenverifizierung

8.2 Hardware

Front-End:

- Standard PC oder browserfähiges Gerät

Back-End:

- Server

8.3 Produktschnittstellen

Die Teamprojekt-Software wird nativ in die Serverapplikation integriert.

Die Serveranwendung besitzt eine Kommunikations-Schnittstelle zum LDAP der Technischen Universität Braunschweig.

9 Glossar

Belt: Die vom Spieler ausgewählten Potions, die er im Dungeon verwenden kann.

Character-Sheet: Eine Übersicht über die Attribute der Spielfigur.

Dungeon: Eine Aneinanderreihung von Leveln, die das Minispiel repräsentieren.

Inventory: Im Inventory wird die Anzahl der aktuell vorhandenen Potions gespeichert.

Scrollcollection: Beschreibt den Fortschritt des Spielers indem es die bereits gesammelten Scrolls speichert.

Herausforderung im Dungeon: Jede fünfte Map besitzt eine solche Herausforderung. Sie ist nur überwindbar wenn mehrere Attribute der Spielfigur auf erhöhten Stand sind. Beispiele wären eine extrem weite Schlucht oder einen besonders großen Gegner.

Playerstatistics: Attribute des Spielers, die Auswirkungen auf das Minispiel haben (Sprunghöhe, Geschwindigkeit, Leben und Stärke).

Potions: Die im Spiel verwendete Bezeichnung für Tränke, die dem Spieler temporäre Vorteile verschaffen. Sie werden durch das richtige Bearbeiten von SQL-Anfragen erstellt werden können.

Scrolls: Einmalig im Spiel im einzusammelnde Objekte. Besitzt man diese, ist man in der Lage durch das Lösen von SQL-Statements Potion zu erstellen oder die Playerstatistics dauerhaft zu erhöhen.

Scroll-/Coin Limit: Gesetztes Limit, das die Anzahl an einsammelbaren Coins beziehungsweise Scrolls pro Tag bestimmt.

Scenetexts: Texte, die die Story und das Tutorial erzählen.