Projektmanagement und Softwareentwicklung – Schuljahr 2023/2024

Math Master

**Team: Billiard**

**Gruppe: Staffner Luca/Resch Lukas**

**Datum: 26.01.2024**

## Inhaltsverzeichnis:

[Inhaltsverzeichnis: 2](#_Toc157110968)

[1) Systemüberblick 3](#_Toc157110969)

[2) Unser System ermöglicht 4](#_Toc157110970)

[1) Use Case Diagramm 4](#_Toc157110971)

[2) Use Cases 5](#_Toc157110972)

[3) Klassendiagramm 13](#_Toc157110973)

[4) Datenbankdiagramm 14](#_Toc157110974)

[5) Preview / Gui Prototyp 15](#_Toc157110975)

[6) Timetable 18](#_Toc157110976)

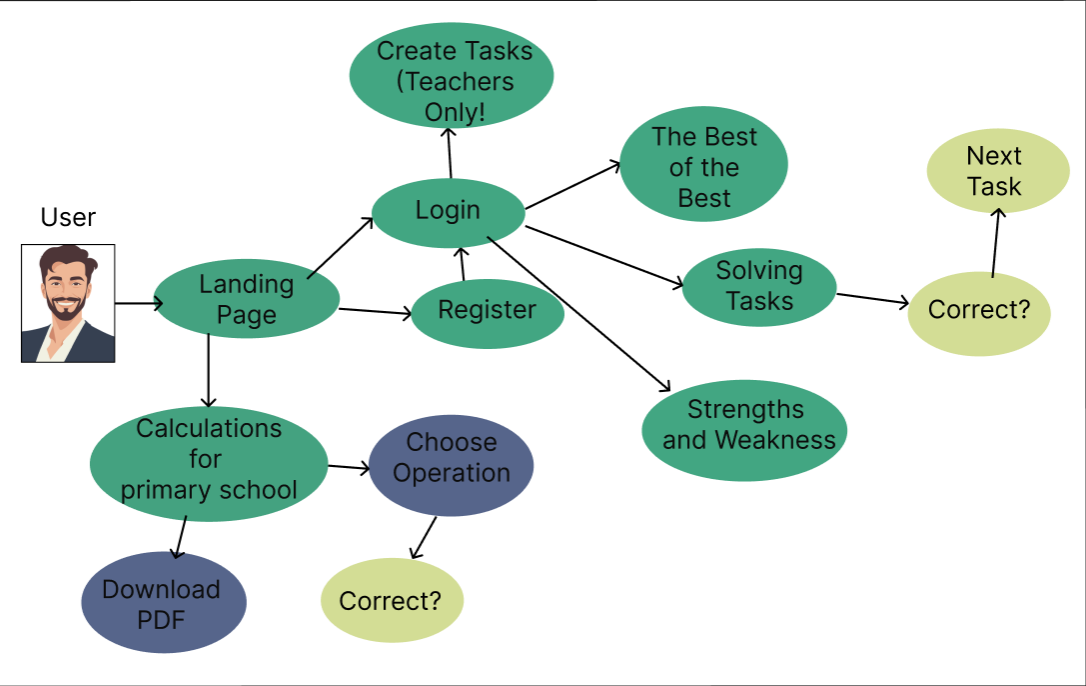
# Systemüberblick

Das Ziel dieser Anwendung ist es die Entwicklung von mathematischer Intuition zu erleichtern, indem es in der Schwierigkeit an den Nutzer angepasste Aufgaben präsentiert. Die Aufgaben kommen aus verschiedenen Quellen, wie der österreichischen Zentralmatura, oder auch verschiedenen Mathematikolympiaden.

Im Frontend wird alles zur Verfügung gestellt, was man braucht, um Aufgaben möglichst gut lösen zu können. Es passiert eine automatische Überprüfung der Lösung und es existiert ein Elo System, das nicht nur über die Verteilung der Aufgaben entscheidet, sondern auch dazu genutzt werden kann um Nutzer innerhalb einer Klasse und auch insgesamt miteinander zu vergleichen.

# Unser System ermöglicht

## Use Case Diagramm



## Use Cases

**Use Case “Rechnungen für Volksschulkinder im Web lösen”**

**Kurze Beschreibung**

In der Weboberfläche werden Rechnungen für Volkschulkinder angezeigt. Sie können auch direkt gelöst und kann kontrolliert werden.

**Auslöser**

Aufrufen der Seite Calculations durch den Nutzer.

**Akteure**

nicht angemeldeter Benutzer

**Vorbedingungen**

System bereit

**Schritte**

1. Aufrufen der Seite Calculations
2. Auswahl der passenden Art von Rechnungen
3. Eingeben der Lösungen
4. Kontrollieren

**Involvierte Klassen**

PrimarySchoolTask

**Use Case “Aufgaben lösen”**

**Kurze Beschreibung**

Die Fragen verschiedener Aufgaben können angezeigt und die Lösung kann in ein Feld geschrieben werden, wo eine automatische Kontrolle der Lösung erfolgt und je nach dem welches Ergebnis dieses bringt wird die Elo des Nutzers entweder erhöht oder gesenkt. Die Elo entscheidet darüber welchen Aufgaben dem Nutzer als nächstes zugewiesen werden.

**Auslöser**

Nutzer ruft die Main Seite auf.

**Akteure**

Angemeldete Benutzer

**Vorbedingungen**

System bereit

**Schritte**

1. Seite aufrufen
2. Antwort in Feld eintragen
3. Auf Überprüfen klicken
4. Wenn Falsch wird die richtige Lösung angezeigt
5. Nächste Aufgabe laden durch Button Click

**Involvierte Klassen**

SQL

**Use Case „Login“**

**Kurze Beschreibung**

Der Benutzer meldet sich bei dem System an, es wird überprüft, ob der Benutzer existiert und ob das Passwort richtig ist.

**Auslöser**

Benutzer drückt auf den Login Button und gibt seine Daten dann bei dem Login Formular ein.

**Akteure**

nicht angemeldeter Benutzer

**Vorbedingungen**

System bereit

**Schritte**

1. Anmeldeformular wird geöffnet
2. Nutzer gibt Nutzernamen und Passwort ein
3. Eingaben werden überprüft

**Involvierte Klassen**

SQL

**Use Case „Account erstellen“**

**Kurze Beschreibung**

Nutzer legt einen neuen Account an, der in die Tabelle Players geschrieben wird.

**Auslöser**

Benutzer drückt auf den Register Button und legt danach seine Account-Daten auf dem Register-Formular fest.

**Akteure**

Nutzer ohne Account

**Vorbedingungen**

System bereit

**Schritte**

1. System lädt Formular zur Registrierung
2. Neuer Nutzer gibt Nutzernamen und Passwort ein
3. Passwörter werden abgeglichen
4. Eingaben werden in der Tabelle Users gespeichert

**Involvierte Klassen**

SQL

**Use Case „Bestenliste“**

**Kurze Beschreibung**

Nutzer können in einer Tabelle ihre eigenen Punkte in einem gewissen Zeitraum mit denen von anderen Leuten vergleichen. Es soll auch möglich sein eine Bestenliste für ein einzige Gruppe zu erstellen.

**Auslöser**

Aufrufen der Bestenliste.

**Akteure**

Angemeldete Benutzer

**Vorbedingungen**

System bereit

**Schritte**

1. Aufruf der Seite
2. Auswahl der Vergleichsgruppe
3. Scrollen innerhalb der Liste

**Involvierte Klassen**

SQL

**Use Case „Stärken-Schwächen Analyse“**

**Kurze Beschreibung**

Analyse wie gut der Nutzer bei verschiedenen Arten von Aufgaben abschneidet. Es werden Möglichkeiten angeboten explizit nur diese Art von Aufgaben zu üben.

**Auslöser**

Aufrufen der Stärken-Schwächen Analyse.

**Akteure**

Angemeldete Benutzer

**Vorbedingungen**

System bereit

**Schritte**

1. Aufruf der Seite

2. Auswahl der Bereich, um mehr Details zu sehen

3. Aufgaben aus dem Schwächsten Bereich machen

**Involvierte Klassen**

SQL

**Use Case „Aufgaben erstellen“**

**Kurze Beschreibung**

Über ein Formular werden Aufgaben erstellt, die entweder allen oder nur denen einer bestimmten Gruppe zur Verfügung gestellt werden. Eine Aufgabe, die jeder sehen kann, muss noch einmal von einem Admin überprüft werden.

**Auslöser**

Benutzer drückt auf den Aufgaben erstellen Button und gibt die benötigten Daten für die Aufgabe ein

**Akteure**

Angemeldete Benutzer der Gruppe Teacher

**Vorbedingungen**

System bereit

**Schritte**

1. Eingabe der Daten

2. Bestätigung der Eingabe und Speicherung in der Tabelle Tasks

**Involvierte Klassen**

SQL

**Use Case „Download PDF“**

**Kurze Beschreibung**

Es werden PDFs von verschiedensten Mathezetteln, die für Volksschulkinder entwickelt wurden, erstellt.

**Auslöser**

User drückt Download Button im Frontend auf der Seite Calculations.

**Akteuere**

nicht angemeldeter Benutzer

**Vorbedingungen**

System bereit

**Schritte**

1. Calculations-Reiter öffnen

2. Auf den Button Download drücken

3. Dokumente öffnen

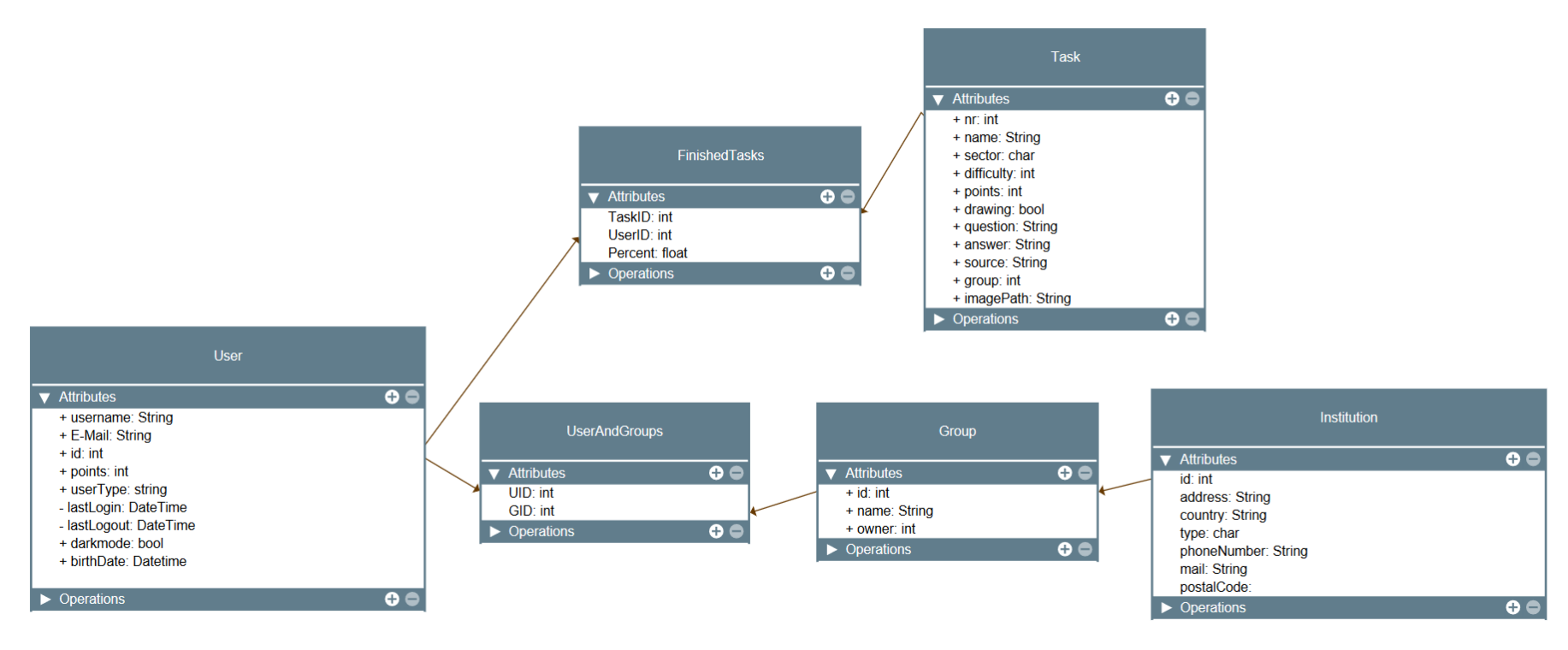
4. Entweder öffnen und ausdrucken oder sonst eine Aktion machen

**Involvierte Klassen**

PrimarySchoolTask.cs

# Klassendiagramm

# Datenbankdiagramm



# Preview / Gui Prototyp

Main Design:

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung

Stärken-Schwächen Analyse:

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Rechenzettel für die Volksschulkinder:

Ein Bild, das Text, Handschrift, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Accountsettings:

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

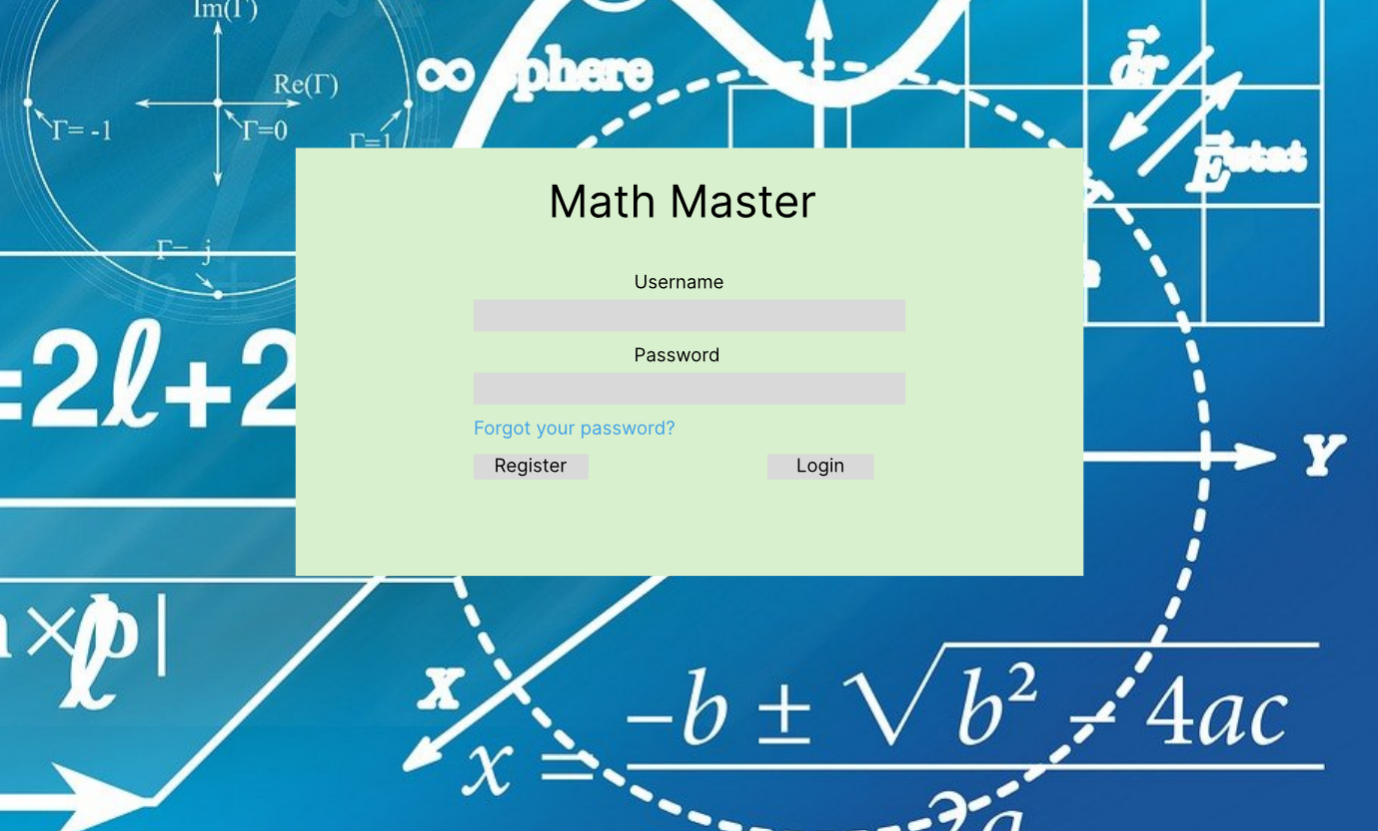
Automatisch generierte Beschreibung

All Exercises:

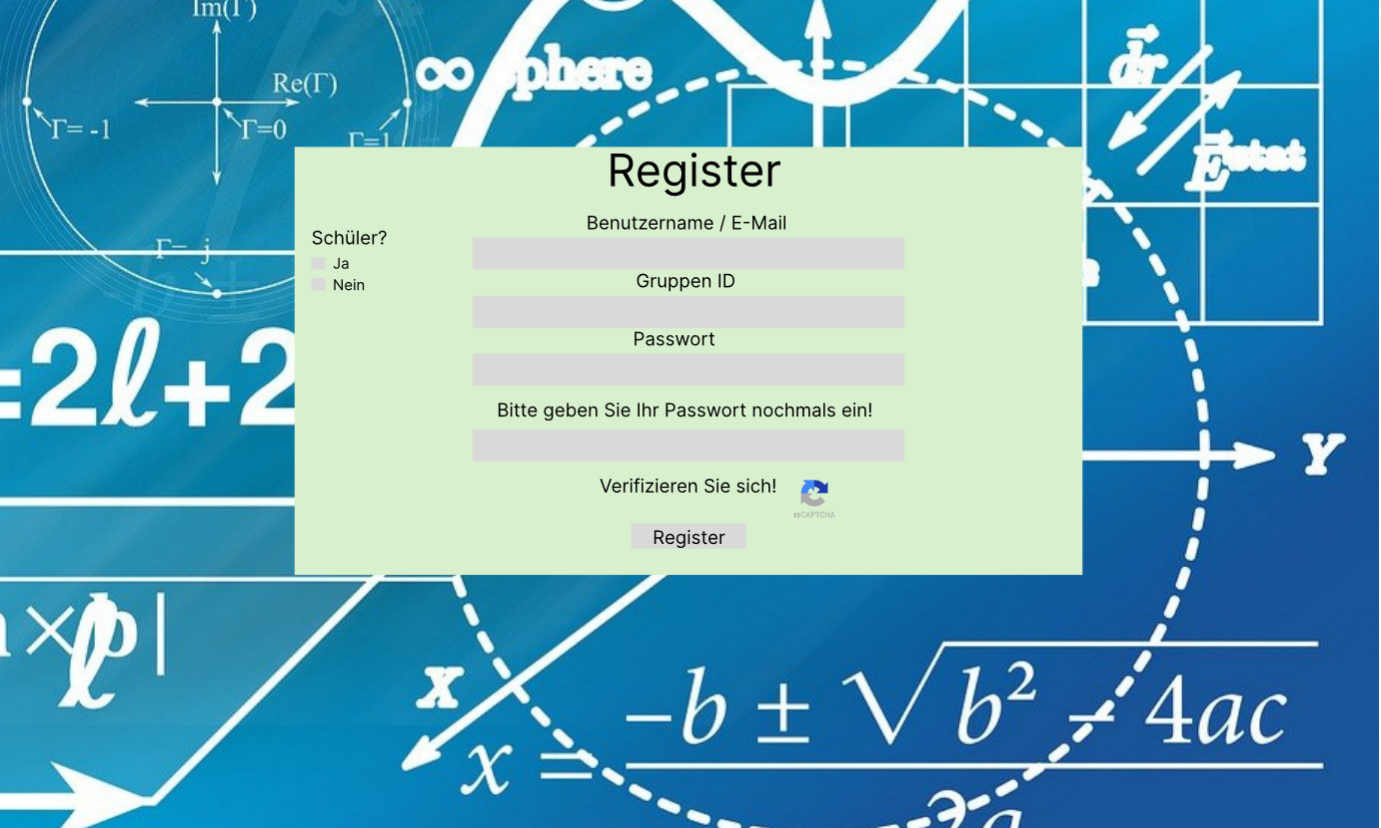
Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Login:



Register:



Studentslist:

Ein Bild, das Text, Screenshot, Rechteck, Reihe enthält.

Automatisch generierte Beschreibung

The best of the best

Ein Bild, das Text, Screenshot, Rechteck, Design enthält.

Automatisch generierte Beschreibung

# Timetable

|  |  |
| --- | --- |
| ***Was wir machen möchten*** | ***Bis wann wir es erledigt haben möchten*** |
| ***Winter - Semester*** | |
| PDF-Creation | Dezember |
| Einarbeiten in React | Dezember |
| Einarbeitung in ASP.net | Jänner |
| ***Sommer- Semester*** | |
| Database creation | Februar |
| Main Page first draft in React | Februar |
| Login and register page | März |
| Lehrerfunktionen | März |
| Webscraper Tasks | Mai |
| Testen und sonstige Features | Mai |
| Fertiges Projekt | Juni |