

# AHIF Academy

---

## Grobe Idee

Eine Lernapp für verschiedene Fächer mit angepassten Fragen und Aufgaben.

## Must-Have

- User Profiles
- Loginsystem
- Benutzerfreundliches und ansprechendes GUI
  - Mehrere Themes (Darkmode,...)
- Benutzerdefinierte Fragen
- Eigene Pages für die Fächer
- Verschiedene Aufgabentypen:
  - Karteikarten Englisch
  - Normale Matheaufgaben
  - Deutsch Fehler finden, Zeichensetzung,...

## Nice-To-Have

- Fragen nach Schulstufen filtern
- Mehr Fächer und Aufgabentypen
- Schöne Matheformeln
- User Profiles:
  - Achievement
  - Mit "Geld" Avatar bearbeiten/Minispiele spielen
- AHIF Pass (Battle Pass)
- Vorbereitete Lerndokumente
- Minigames

## Wont-Have

- Globale Datenbank
- Globales Loginsystem
- Online Multiplayer
- Mehrere Sprachen

## Aufteilung

### Arbeitsteilung

Marlon	Ensar
Klassen	GUI
User Profiles Statistiken	Login-System

Marlon	Ensar
Mathe, Deutsch	Karteikarten
Benutzerdefinierte Fragen	User Profiles
Fragen organisieren	Fragen organisieren

Grobe zeitliche Einteilung

Zeitraum	Marlon	Ensar
Woche 1	Klassen	GUI
Woche 2	Mathe und Deutsch	
Woche 3	Benutzerdefinierte Fragen	
Woche 4	User Profiles inkl. Statistiken	

Klassen

- MultipleChoice
- YesNo
- TextInput
- FlashCard
- User
- ChangeTheme

Klassenhierarchie

Alle Frage-Klassen (MultipleChoice, YesNo, TextInput) haben eine Parent-Klasse "Question" von welcher sie einige Properties und Methoden erben.

Question

```
@startuml
class Question{
    + Answerpressed: bool
    + btnNextQuestion: Button
    + textblockQuestion: TextBlock
    + Subject: string
    + Text: string
    + string[3]: subjects
    + CorrectAnswer: string
    + LastUsed: DateTime
    + Counter: int
    + abstract Draw(Grid grid): void
    + CheckAnswer(string answer): void
    + abstract Copy(): void
}
@enduml
```

## Multiple Choice

```
@startuml
class MultipleChoice{
+ Answers: string[4]
+ ans1: Button
+ ans2: Button
+ ans3: Button
+ ans4: Button
+ MultipleChoice()
+ MultipleChoice(string text, string ans1, string ans2, string ans3, string ans4,
string correct, string subject, int counter, DateTime lastUsed)
+ EVENT Click(object sender, RoutedEventArgs e): void
}
@enduml
```

## YesNo

```
@startuml
class YesNo{
+ yes: Button
+ no: Button
+ YesNo()
+YesNo(string text, string subject, string correctAnswer, int counter, DateTime
lastUsed)
}
@enduml
```

## TextInput

```
@startuml
class TextInput{
+ WrongText: string
+ submit: Button
+ textBoxAnswer: RichTextBox
+ TextInput()
+ TextInput(string text, string subject, string answer, string falseAnswer, int
counter, DateTime lastUsed)
+ EVENT Submit_Click(object sender, RoutedEventArgs e): void
}
@enduml
```

## FlashCard

```
+ German: string
+ English: string
+ FlashCard()
```

## User

```
@startuml
class User{
+ Username: string
+ Password: string
+ QuestionsAnsweredMath: int
+ QuestionsAnsweredGerman: int
+ QuestionsAnsweredCorrectMath: int
+ QuestionsAnsweredCorrectGerman: int
+ StatisticsDraw()
}
```

## QuestionList

```
@startuml
class QuestionList{
- questions: List<Question>
- random: Random
+ CurrentSubject: string
+ FilterBySubject(string subject): QuestionList
+ Add(Question question): void
+ Remove(Question question): void
+ DeserializeFromJSON(): void
+ GetRandomQuestion(): Question
+ SerializeToJSON(): void
}
@enduml
```