**Heinrich-Emanuel-Merck-Schule Darmstadt**

**Lastenheft**

|  |  |
| --- | --- |
| **Projektbezeichnung** | SolveMeDis |
| **Projektleiter** | Norbert Grothe |
| **Erstellt am** | 02.11.2020 |
| **Letzte Änderung am** | 14.11.2020 |
| **Status** | [in Bearbeitung/~~fertiggestellt/pausiert/abgebrochen/Prüfung~~] |
| **Aktuelle Version** | 1.4 |

**Änderungsverlauf**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr.** | **Datum** | **Version** | **Geänderte Kapitel** | **Art der Änderung** | **Autor** | **Status** |
| 1 | 02.11.2020 | 1.1 | Alle | Erstellung | Eva Zimmermann | - |
| 2 | 12.11.2020 | 1.2 | Alle | Ergänzung | Eva Zimmermann | i.B. |
| 3 | 14.11.2020 | 1.3 | 1; 3.1.1; 6 | Ergänzung | Eva Zimmermann | i.B. |
| 4 | 15.11.2020 | 1.4 | Alle | Verbesserung | Paul Pusch | i.B. |
|  |  |  |  |  |  |  |

Inhalt

[1 Einleitung 4](#_Toc56346411)

[2 Allgemeines 4](#_Toc56346412)

[2.1 Ziel und Zweck des Dokuments 4](#_Toc56346413)

[2.2 Ausgangssituation 4](#_Toc56346414)

[2.2.1 Vorbemerkungen 4](#_Toc56346415)

[2.2.2 Kernpunkte 4](#_Toc56346416)

[2.3 Projektbezug 5](#_Toc56346417)

[3 Konzept 5](#_Toc56346418)

[3.1 Ziel(e) des Anbieters 5](#_Toc56346419)

[3.1.1 Warum Kopfrechnen? 5](#_Toc56346420)

[3.2 Ziel(e) und Nutzen des Anwenders 6](#_Toc56346421)

[3.3 Zielgruppe(n) 6](#_Toc56346422)

[4 Funktionale Anforderungen 6](#_Toc56346423)

[4.1 **System-Anforderungen** 6](#_Toc56346424)

[4.1.1 *Grundanforderung* 6](#_Toc56346425)

[4.1.2 Betriebssystem 6](#_Toc56346426)

[4.1.3 Berechtigungen 6](#_Toc56346427)

[**4.2** **UI-Anforderungen** 6](#_Toc56346428)

[4.2.1 *Grundfunktion* 6](#_Toc56346429)

[4.2.1.1 Quiz-Item 7](#_Toc56346430)

[4.2.1.2 Antwort 7](#_Toc56346431)

[4.2.1.2.1 Mutliple choice 7](#_Toc56346432)

[4.2.1.3 Bewertung 7](#_Toc56346433)

[4.2.1.3.1 Belohnung 7](#_Toc56346434)

[4.2.2 Fortsetzung 7](#_Toc56346435)

[4.2.3 Schwierigkeitsstufen 7](#_Toc56346436)

[4.3 **Modell-Anforderungen** 7](#_Toc56346437)

[4.3.1 *Grundanforderung* 7](#_Toc56346438)

[4.3.2 Aufgabenauswahl 7](#_Toc56346439)

[4.3.3 Schwierigkeit 7](#_Toc56346440)

[4.3.4 Aufgabenart 7](#_Toc56346441)

[5 Nichtfunktionale Anforderungen 8](#_Toc56346442)

[5.1 **Allgemeine Anforderungen** 8](#_Toc56346443)

[5.1.1 Bedienbarkeit 8](#_Toc56346444)

[5.1.1.1 übersichtlich 8](#_Toc56346445)

[5.1.1.2 intuitiv 8](#_Toc56346446)

[5.1.2 Visuelles Design 8](#_Toc56346447)

[5.1.3 Quellcode-Anforderungen 8](#_Toc56346448)

[5.1.3.1 Objektorientierung 8](#_Toc56346449)

[5.1.4 Projektmanagement 8](#_Toc56346450)

[6 Ausblick 8](#_Toc56346451)

# Einleitung

SolveMeDis, so nennen wir unser App-Projekt. Wir, das sind Paul Pusch, Steven Pleyer und Eva Zimmermann. Kleines Problem, schnelle Lösung: SolveMeDis.

# Allgemeines

## Ziel und Zweck des Dokuments

Dieses Dokument beschäftigt sich mit der Anforderungsanalyse für eine Anwendung, die im Rahmen des Berufsschulunterrichts entwickelt wird. Es wird ebenfalls im Berufsschulunterricht von den Auszubildenden selbst erstellt, wie es im Rahmenlehrplan vorgesehen ist.

Wir entscheiden uns für eine eher frei gestaltete Anforderungsspezifikation, die Aspekte eines Lasten- sowie eines Pflichtenheftes vereint.

## Ausgangssituation

## Vorbemerkungen

Wir sehen einen Zukunftstrend der App-Entwicklung in einer zunehmenden Nachfrage nach schlanken Anwendungen, die lizenzfrei und quelloffen für alle zur Verfügung stehen.

Auf Anwenderseite wächst das Bedürfnis nach kleinen Freizeitbeschäftigungen für zwischendurch, die schnelle Belohnung für wenig Investition bieten.

## Kernpunkte

Thematisch liegt unser Fokus klar auf dem Themenfeld MINT (Mathematik, Informatik, Naturwissenschaft und Technik), das auch gesamtgesellschaftlich an Bedeutung gewinnt. Dabei wächst das betriebs- wie auch volkswirtschaftliche Bedürfnis danach, den Zugang breiterer Bevölkerungsgruppen zum MINT-Bereich zu erleichtern.

Aus Untersuchungen der pädagogischen Psychologie ist bekannt, dass ein spezifisches Interesse an Zahlen, Maschinen, Computern oder anderen stereotypischen MINT-Gegenständen zwar einen positiven Faktor für hohe Leistungsfähigkeit im MINT-Bereich darstellt, aber keinen entscheidenden.  
Positiv korrelierte Faktoren sind daneben auch angenehme Vorerfahrungen in den beteiligten Schulfächern, geringe motivationale Hürden sowie positive Kognitionen der eigenen Leistungsfähigkeit in Bezug auf MINT-Themen.

Dieses Projekt will einen kleinen positiven Beitrag leisten: Eine App, die beim Entwickeln wie auch beim Anwenden Spaß macht und einen spielerischen Ansatz verfolgt – klein, schnell und intuitiv.

## Projektbezug

Das vorliegende Projekt ist ein unabhängiges Projekt, das von einer frei arbeitenden Gruppe von Auszubildenden Mathematisch-technischen Softwareentwicklern bearbeitet wird.

Die Bearbeitung findet innerhalb des Unterrichts an der HEMS in der Schulklasse BS12MTS1 statt und wird von Norbert Grothe betreut.

Wir entscheiden uns dazu, das Projekt quelloffen auf GitHub zu veröffentlichen. Die Nutzung von GitHub erleichtert uns die Versionsverwaltung. Quelloffene Entwicklung halten wir für eine empfehlenswerte und wichtige Praxis.

# Konzept

## Ziel(e) des Anbieters

Allgemein gesagt entsteht eine kleine Android-App, die auch alle Klassenkameraden einmal ausprobieren können.

Das Projekt kann innerhalb des Schulhalbjahres verwirklicht und zu einer vorläufigen Produktreife gebracht werden.

Modellierung und Implementierung beinhalten mathematische Aspekte, was für die Ausbildung relevant ist.

Die App dieser Gruppe ermöglicht eine niedrigschwellige Nutzung und bietet kleine Aufgaben an, die einfach zu lösen sind.

## Warum Kopfrechnen?

Mit einfachem Kopfrechnen entdeckt kaum jemand neue mathematische Konzepte oder trainiert mathematisches Modellieren, deshalb einige Worte zu den Vorteilen dieser Aufgaben.

Leistungen in Kopfrechenaufgaben weisen Zusammenhänge auf zu Leistungen im Bereich des logischen Denkens und des räumlichen Vorstellungsvermögens. Mathematisches Vorwissen dagegen hängt kaum damit zusammen. Kopfrechnen trainiert also grundlegende Fähigkeiten, die gute Leistungen in Mathematik – und den Spaß dabei – wahrscheinlicher machen.

Beim Kopfrechnen nutzen zudem alle ihre eigenen Tricks und Vorstellungen. Eine stellt sich häufig den Zahlenstrahl vor, ein anderer denkt viel in Fünferpäckchen, wieder eine andere überschlägt meistens zuerst. Kopfrechenaufgaben weisen eine gewisse Offenheit auf und ermöglichen das selbständige Auswählen von Strategien.

Wir entscheiden uns zunächst für Additions- und Subtraktionsaufgaben im Zahlenraum bis 110. Diese Aufgaben liegen in einem Schwierigkeitsbereich, der keine starke Assoziation mit der Zuhilfenahme eines Taschenrechners hat – „Das kann ich auch im Kopf rechnen!“

## Ziel(e) und Nutzen des Anwenders

Der Anwender kann die Anwendung ohne hohe Einstiegsmotivation starten. Er kann sofort beginnen und schnell abbrechen. Die Anwendung ähnelt im Format einem Quiz und bietet spielerische Belohnungsaspekte.

Der Anwender löst eine kleine Aufgabe nach der anderen und findet dabei wenig Hürden vor, einfach weiter zu machen. Andererseits kann er auch einfach aufhören, wenn es ihm zu langweilig ist. Dann kann er vielleicht eine höhere Schwierigkeitsstufe ausprobieren oder das Handy einfach weglegen.

## Zielgruppe(n)

Die Projektgruppe möchte eine Anwendung zur Verfügung stellen, die alle nutzen können, denen sie Spaß macht. Sie soll klein, schnell und einfach zugänglich sein. Ihre Funktionalität soll überschaubar sein, sodass Einstiegshürden vermieden werden.

# Funktionale Anforderungen

## **System-Anforderungen**

## *Grundanforderung*

Die Anwendung soll eine breit nutzbare Android-App sein.

## Betriebssystem

Die Anwendung sollte auf gängigen aktuellen Android-Versionen stabil lauffähig sein.

## Berechtigungen

Die Anwendung soll in Android keine kritischen Berechtigungen benötigen und sollte insgesamt möglichst wenige Berechtigungen benötigen.

## **UI-Anforderungen**

## *Grundfunktion*

Die Anwendung soll einfache Rechenaufgaben anbieten und die Antwort nach Richtigkeit bewerten.

## Quiz-Item

Die Anwendung sollte durch Tippen auf einen Button ein Aufgaben-Item anbieten.

## Antwort

Die Anwendung soll eine Antwort des Anwenders entgegennehmen. Dies kann unterschiedlich erfüllt werden, durch multiple choice oder Eingabefeld.

## Mutliple choice

Bei multiple choice kann die Anwendung die korrekte und mehrere plausible Falschantworten zur Verfügung stellen. Falschantworten können auf häufigen Fehlertypen basieren, die aus der Didaktik bekannt sind.

## Bewertung

Die Anwendung soll die Antwort des Anwenders nach Richtigkeit bewerten und diese Bewertung anzeigen.

## Belohnung

Die Anwendung kann Belohnungsanreize bieten und dafür beispielsweise die Bewertungen als Zähler anzeigen.

## Fortsetzung

Die Anwendung soll nach Beenden eines Aufgaben-Items das nächste Item anbieten, nach kurzer Zeit oder durch einfaches Tippen.

## Schwierigkeitsstufen

Die Anwendung kann mehrere Tabs für unterschiedliche Schwierigkeitsstufen zur Auswahl anbieten.

## **Modell-Anforderungen**

## *Grundanforderung*

Die Anwendung soll Rechenaufgaben zur Verfügung stellen, die für einen durchschnittlich intelligenten erwachsenen Anwender im Kopf innerhalb einiger Sekunden mit mittlerer Anstrengung und Konzentration zu berechnen sind.

## Aufgabenauswahl

Die Anwendung kann Aufgaben auswählen, die sich für den Anwender eher zufällig anfühlen, wenige Wiederholungen enthalten und ähnliche Schwierigkeit aufweisen.

## Schwierigkeit

Im Fall von mehreren Schwierigkeitsstufen sollte die einfachste Stufe eine besonders geringe Einstiegshürde für den Anwender darstellen.

## Aufgabenart

Die Aufgaben können Grundrechenarten und andere einfache mathematische Konzepte beinhalten.

# Nichtfunktionale Anforderungen

## **Allgemeine Anforderungen**

## Bedienbarkeit

Die Anwendung soll intuitiv bedienbar sein.

## übersichtlich

Das UI soll schlank und auf Notwendiges beschränkt sein, ausgenommen motivierende Funktionalitäten.

## intuitiv

Die Funktionalität soll flach aufgebaut sein, alles soll in wenigen Schritten erreichbar sein.

## Visuelles Design

Die visuelle Gestaltung sollte ästhetisch ausgewogen aber eher zurückhaltend sein.

## Quellcode-Anforderungen

Der Quellcode sollte gut lesbar, wartbar und änderbar sein.

## Objektorientierung

Der Quellcode soll objektorientiert modelliert sein und eine übersichtliche Architektur aufweisen.

## Projektmanagement

Der Quellcode wird auf GitHub unter Versionsverwaltung gestellt. Die App wird lauffähig gehalten und iterativ erweitert. Releases können kurzfristig stattfinden.

# Ausblick

Die folgenden Features sind bereits angedacht:

Highscores – Ein erreichter Punktewert wird in den Highscores mit Zeitstempel gespeichert. Ein Eintrag in den Highscores löst ein Popup aus. Der Anwender kann die Highscores jederzeit einsehen.

Daten speichern – Zwischenspeichern ist möglich in den verschiedenen Modi.

Optionen – Zur Erhöhung der Nutzerfreundlichkeit können einige Bedienfunktionen angepasst werden.

Neue Modi – z.B. Time Attack: Der Anwender spielt gegen die Zeit. Falschantworten geben Punktabzug.