Marius Brinkmann, Florian Klocke, Felix Vössing / PBBFA23A

Pflichtenheft Studentenwürfel

Inhaltsverzeichnis

[Abbildungsverzeichnis 2](#_Toc192260243)

[1 Aufgabenstellung – Einführung 3](#_Toc192260244)

[1.1 Ist-Analyse 3](#_Toc192260245)

[1.1.1 Ist-Zustand 3](#_Toc192260246)

[1.1.2 Schwachstellen 3](#_Toc192260247)

[1.1.3 Visualisierung 4](#_Toc192260248)

[1.2 Soll-Analyse 4](#_Toc192260249)

[1.2.1 Zieldefinition 4](#_Toc192260250)

[1.2.2 Anforderungen 5](#_Toc192260251)

[1.2.3 Optionale Anforderungen 5](#_Toc192260252)

[1.2.4 Weiterentwickelung und Verbesserungen 6](#_Toc192260253)

[1.2.5 Teststrategie 6](#_Toc192260254)

[1.2.6 Projektmanagement 6](#_Toc192260255)

[1.2.7 Risikomanagement 6](#_Toc192260256)

[2 Produktumgebung 7](#_Toc192260257)

[2.1 Anwendungsbereiche 7](#_Toc192260258)

[2.2 Anwendergruppen 7](#_Toc192260259)

[2.3 Basismaschine 7](#_Toc192260260)

[2.4 Mengengerüst 8](#_Toc192260261)

[3 Produktmodell 8](#_Toc192260262)

[3.1 Anwendungsfälle der Software 8](#_Toc192260263)

[3.2 Ablaufanalyse 9](#_Toc192260264)

[3.3 Benutzeroberfläche 9](#_Toc192260265)

[3.4 Strukturelle Datenanalyse 10](#_Toc192260266)

[3.4.1 Datenmodellierung 10](#_Toc192260267)

[3.4.2 Datenobjekte und Attribute 10](#_Toc192260268)

[3.4.3 Datenflüsse 10](#_Toc192260269)

[3.4.4 Metadaten 11](#_Toc192260270)

[3.4.5 Datenqualität 11](#_Toc192260271)

[4 Qualitätsanforderungen 11](#_Toc192260272)

[5 Benutzerschnittstelle 12](#_Toc192260273)

[5.1 Benutzermodell 12](#_Toc192260274)

[6 Entwicklungsumgebung 12](#_Toc192260275)

# Abbildungsverzeichnis

[Abbildung 1: Darstellung der Ist-Situation 4](#_Toc192251957)

[Abbildung 2 : Klassendiagramm für die Anwendung 8](#_Toc192251958)

[Abbildung 3 : Use-Case-Diagramm zum Ablauf des Würfels 9](#_Toc192251959)

[Abbildung 4 : Benutzeroberfläche vom Studentenwürfel 9](#_Toc192251960)

# 1 Aufgabenstellung – Einführung

Geplant ist eine Verbesserung des aktuellen Studentenwürfels des bib international Colleges.

Dieser neue Würfel wird mit einer grafischen Oberfläche erstellt und kann mit mehreren konfigurierbaren Beschränkungen einen zufälligen Schüler aus einer Liste auswählen. Außerdem soll nach der Verwendung des Würfels ein Protokoll erstellt werden.

## Ist-Analyse

### 1.1.1 Ist-Zustand

Der Studentenwürfel ist eine Webanwendung im Intranet des bib International College. Ist der Lehrkörper bei dem Aufruf der Seite im Intranet angemeldet, wird der Würfel mit den Klassen der aktuellen Stunde vorinitialisiert.  
Sobald der Dozent den Würfel anklickt, wird ein zufälliger Schüler aus den konfigurierten Klassen ausgewählt und unter dem Würfel angezeigt.

Außerdem wird ein inoffizieller Würfel von manchen Dozenten verwendet. Dieser ist eine einfache Konsolenanwendung, welche nach manueller Klassenkonfiguration auch einen zufälligen Studenten ausgibt.

### 1.1.2 Schwachstellen

Sowohl der Studentenwürfel des Intranets als auch die Konsolenanwendung bieten dem Dozenten keine Konfigurationsmöglichkeiten, den Zufall einzuschränken.  
Des Weiteren bietet keine der beiden Lösungen eine Funktion zur Überprüfung und Nachverfolgung der Beteiligung an.

Zudem ist für die Nutzung des Intranet-Würfels eine Internetverbindung zwingend nötig.

Der Konsolen-Würfel hingegen benötigt keine Serververbindung, besitzt allerdings keine grafische Oberfläche. Diese bietet wenig Benutzerfreundlichkeit und wirkt auf weniger technisch affine Dozenten eher abschreckend.

### 1.1.3 Visualisierung

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

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 1: Darstellung der Ist-Situation

## 1.2 Soll-Analyse

### 1.2.1 Zieldefinition

Ziel ist das zufällige Auswürfeln eines Studenten aus einer oder mehreren Klassenlisten.

Dem Dozenten werden zwei Konfigurationsmöglichkeiten geboten, den Zufall etwas zu steuern.

Eine erste Option verhindert das mehrfache Auswürfeln desselben Schülers innerhalb einer Unterrichtseinheit.  
Die zweite Konfiguration ist etwas weniger restriktiv und verhindert lediglich, dass derselbe Student in Folge von dem Würfelwerkzeug ausgewählt wird.

Die zur Auswahl der Schüler benötigten Klassenlisten werden in einem dafür vorgesehenen Ordner lokal abgelegt. Diese Liste muss ein CSV-Dateiformat einhalten, sodass das fehlerfreie Einlesen gewährleistet werden kann.

Außerdem wird die Beteiligung der vom Zufall aufgerufenen Schüler in einem Verlaufsprotokoll beim Beenden des Programms festgehalten. Dieses Protokoll wird mit der Klassenbezeichnung, dem Block und einem Datum versehen und, wie die Schülerlisten, in einem vorgesehenen Ordner abgelegt.

Zusätzlich soll dem Dozenten die Möglichkeit geboten werden, einen Schüler manuell aufzurufen und dessen Namen über ein Eingabefeld dem Protokoll hinzuzufügen.

Optional sollte eine Verbindung zu einem lokalen Server, der die Klassenlisten beherbergt, simuliert werden.  
Diese Listen werden vom Server mit einem GET-Request, der eine Liste an Klassennamen als Parameter erhält, abgefragt.

### 1.2.2 Anforderungen

Das Programm muss offline verwendbar sein.

Beim Starten des Würfels werden die CSV-Klassendateien aus dem dafür vorgesehenen Ordner automatisch eingelesen. Falls keine Klassendateien vorhanden sind, soll der Lehrkörper den Ordner über ein Steuerelement auf der Benutzeroberfläche öffnen können. Dort kann er manuell Klassendateien hinzufügen.

Sobald die Klassenlisten vorhanden sind, kann der Dozent die Klassen über ein Dropdown-Menü auswählen und der Würfel ist einsatzbereit.

Der Zufall muss nach den konfigurierten Regeln eingeschränkt werden und diese Konfigurationen müssen für den nächsten Start des Programms zwischengespeichert werden.

Der vom Zufall bestimmte Schüler wird in die Protokolldatei eingetragen.

### 1.2.3 Optionale Anforderungen

Optional soll eine Serverschnittstelle simuliert werden.

Diese soll hauptsächlich die initiale Einrichtung des Programms erleichtern. Auf dem lokalen Server werden ein paar Beispielklassendateien abgelegt sein.

Sollte der für die Klassenlisten vorgesehene Ordner bei Start des Studentenwürfels leer sein, wird eine GET-Anfrage an den Server gestellt, der die Klassenlisten abruft. Dies kann weiter durch einen Log-in des Lehrers eingeschränkt werden, wird aber vorerst nicht berücksichtigt.

Des Weiteren ist eine Ansicht für das Protokoll im Programm selbst erstrebenswert. Das Protokoll wird vorerst nur als .txt-Datei in der Ordnerstruktur abgelegt, allerdings würde eine zusätzliche Anzeige die Nutzerfreundlichkeit erheblich steigern.

Außerdem soll es weitere Konfigurationsmöglichkeiten geben. Diese dienen zum Anheben der Moral der Schüler. Die erste Option zeigt beim Auswürfeln eine kurze Animation eines zwanzigseitigen Würfels. Die zweite Option soll bei der Wahl des Schülers einen kleinen Motivationsspruch anzeigen.

### 1.2.4 Weiterentwickelung und Verbesserungen

Das Programm könnte die Vorkonfigurationsfunktion der Klassen, wie der Studentenwürfel des Intranets, übernehmen.

Des Weiteren sollten den Lehrkräften zwei weitere Konfigurationsmöglichkeiten angeboten werden, welche das Auswürfeln der Schüler etwas ansprechender gestalten sollen.

Die erste Konfiguration ist das Einschalten einer Würfelanimation eines zwanzigseitigen Würfels. Der Studierende wird daraufhin in der Mitte des Würfels angezeigt.

Die zweite Option ist das Hinzufügen von Aufforderungssprüchen. Diese sollen zusammen mit dem ausgegebenen Schüler angezeigt werden.

Außerdem könnte der Würfel weitere Sprachen unterstützen und somit barrierefreier sein.

### 1.2.5 Teststrategie

Unsere Teststrategie beruht auf dem Vier-Augen-Prinzip. Jede fertige Änderung wird einem Projektmitglied vorgelegt, um die Qualität und Richtigkeit des neu entstandenen Codes mit einem Integrationstest gezielt auszuprobieren.

### 1.2.6 Projektmanagement

Das Management dieses Softwareprojekts erfolgt über die Versionsverwaltungssoftware GitHub. Dadurch wird ein praxisnahes Umfeld geschaffen und die Befolgung der Teststrategie vor der Integration gewährleistet.

### 1.2.7 Risikomanagement

Da die Projektzeit sehr knapp bemessen ist, sind Ausfälle durch Krankheit ein hohes Risiko, welches den Erfolg dieses Projekts stark gefährdet. Sollte es zu krankheitsbedingten Ausfällen kommen, werden Aufgaben mit einer hohen Priorität auf die verbleibenden Mitglieder neu verteilt.

Für die optionale Simulation einer Serververbindung besteht ein Sicherheitsrisiko, sollte sie nicht nach Projektabschluss überarbeitet werden. Die Implementierung sieht zurzeit keine Authentifizierungsmöglichkeit vor. Dadurch kann unautorisierter Zugriff nicht abgewendet werden und das Würfelprogramm erhält bei Start, mit einem leeren Klassenlisten-Ordner, alle sich auf dem Server befindenden Listen.

# 2 Produktumgebung

## 2.1 Anwendungsbereiche

Der Studentenwürfel ist für den Einsatz in Berufs- und Schulklassen vorgesehen. Er dient als Hilfsmittel für Lehrkräfte, um auf einfache Weise eine zufällige Auswahl von Studierenden zu treffen. Dies kann beispielsweise genutzt werden, um sicherzustellen, dass alle Schüler oder Studierenden gleichermaßen in den Unterricht eingebunden werden. Die zufällige Auswahl kann etwa für mündliche Beiträge, das Beantworten von Fragen oder die Zuteilung von Aufgaben erfolgen. Das Programm trägt somit zu einer fairen und gleichmäßigen Beteiligung am Unterricht bei.

## 2.2 Anwendergruppen

Die Software richtet sich in erster Linie an Lehrkräfte und Dozenten, die sie in ihrem Unterricht oder in Vorlesungen einsetzen können. Sie soll eine einfache und effiziente Möglichkeit bieten, um Studierende ohne subjektive Einflussnahme oder aufwendige manuelle Verfahren zufällig auszuwählen.

## 2.3 Basismaschine

Damit das Programm einwandfrei funktioniert, sind bestimmte Mindestanforderungen an das Betriebssystem und die Hardware zu erfüllen. Es ist kompatibel mit Windows 10 oder neueren Versionen, MacOS ab Version 10.13 sowie Linux-Distributionen mit einem Kernel der Version 4.x oder höher. Die erforderliche Hardware umfasst einen Prozessor mit mindestens zwei Kernen und einer Taktrate von mindestens 1,8 GHz. Darüber hinaus benötigt das Programm mindestens 512 Megabyte Arbeitsspeicher sowie 100 Megabyte freien Festplattenspeicher. Diese Anforderungen stellen sicher, dass die Software auf einer Vielzahl moderner Computer problemlos genutzt werden kann.

## 2.4 Mengengerüst

Das Programm ist so konzipiert, dass es keine Begrenzung hinsichtlich der maximalen Anzahl an Studierenden gibt. Es kann sowohl in kleinen Schulklassen mit wenigen Teilnehmern als auch in großen Vorlesungen mit mehreren Hundert Studierenden ohne Einschränkungen eingesetzt werden. Unabhängig von der Anzahl der erfassten Namen bleibt die Leistung des Programms stabil und gewährleistet eine zuverlässige Nutzung in allen Unterrichtssituationen.

# 3 Produktmodell

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

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 2 : Klassendiagramm für die Anwendung

## 3.1 Anwendungsfälle der Software

Der Studentenwürfel wird benutzt, wenn der Dozent einen zufälligen Schüler auswählen möchte. Beispielsweise wenn sich kein Schüler meldet, der Dozent aber einen Schüler zum Vorstellen einer Aufgabe benötigt. Der Würfel geht dann die Liste der Schüler durch und wählt zufällig einen Schüler aus. Außerdem werden ausgewählte Namen in einer Protokoll-Datei ausgegeben, damit es leichter ist, die mündliche Mitarbeit für die Stunde zu beurteilen.

## 3.2 Ablaufanalyse

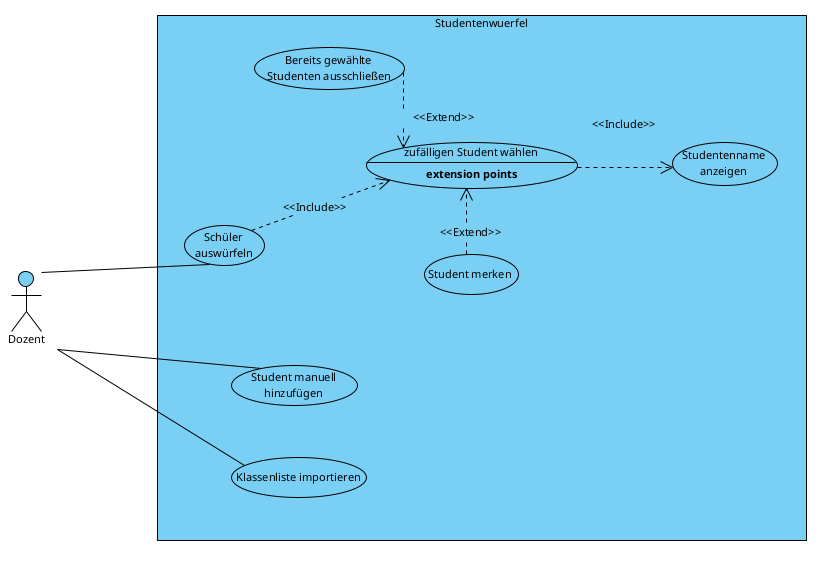


Abbildung 3 : Use-Case-Diagramm zum Ablauf des Würfels

## 3.3 Benutzeroberfläche

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

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 4 : Benutzeroberfläche vom Studentenwürfel

## 3.4 Strukturelle Datenanalyse

### 3.4.1 Datenmodellierung

Das Datenmodell basiert auf einfachen CSV-Dateien, die sowohl die Namen der Schüler als auch den Namen der Klasse als Dateiname enthalten. Diese CSV-Dateien dienen als grundlegende Datenquelle für das System. Jede Datei ist strukturiert, um die relevanten Informationen übersichtlich und leicht zugänglich zu machen. Die Vornamen und Nachnamen sind in separaten Spalten organisiert, sodass das Programm die Daten effizient einlesen und verarbeiten kann. Die erste Zeile der CSV-Dateien wird nicht beachtet, da sie lediglich den Kopfbereich darstellt und keine weiteren Daten enthält. Dieses Modell ermöglicht eine einfache Verwaltung und Aktualisierung der Daten, da CSV-Dateien leicht zu bearbeiten und zu erweitern sind. Außerdem erstellt das Programm eine Protokoll-Textdatei, damit der Dozent leichter nachvollziehen kann, welche Schüler drangenommen wurden. Die ausgewählten Optionen werden in einer JSON-Datei festgehalten, damit sie beim nächsten Starten des Programms bestehen bleiben.

### 3.4.2 Datenobjekte und Attribute

Die CSV-Dateien haben zwei Spalten, eine für den Vornamen und eine für den Nachnamen des Schülers. Der Klassenname wird dem Dateinamen der CSV-Datei entnommen.

### 3.4.3 Datenflüsse

Die im System hinterlegten CSV-Dateien, die die verschiedenen Klassen und deren Schüler enthalten, werden zunächst vom Programm eingelesen. Der Dozent hat dann die Möglichkeit, eine oder mehrere Klassen aus diesen Dateien auszuwählen, je nach Bedarf. Nachdem die Auswahl der zutreffenden Klassen abgeschlossen ist, durchsucht das Programm die entsprechenden Daten, speichert die Schüler in einem Array und wählt zufällig einen Schüler aus der getroffenen Auswahl aus. Dieser zufällig ausgewählte Schüler wird anschließend dem Dozenten angezeigt. Dieser Prozess ermöglicht es dem Dozenten, auf einfache und effiziente Weise einen Schüler aus einer bestimmten Klasse oder mehreren Klassen zu bestimmen, ohne manuell durch die Listen gehen zu müssen.

### 3.4.4 Metadaten

Die Metadaten umfassen Informationen über die Struktur der CSV-Datei:

* Dateiformat: CSV
* Dateiname: Name der Klasse, z.B. pbbfa23a.csv
* Kopfzeile: Vorname, Nachname
* Spalten: Vorname, Nachname
* Trennzeichen: Semikolon (;)

Informationen über die Struktur der Protokoll-Datei:

* Dateiformat: Text-Datei
* Dateiname: protokoll.txt
* Zeilen: Vorname, Nachname, Klassenname
* Trennzeichen: Semikolon (;)

Informationen über die Struktur der Option-Datei:

* Dateiformat: JSON
* Dateiname: options.json
* Objekt: “optionen”
* Schlüssel: “mehrmalsProStunde”, “nacheinander”

### 3.4.5 Datenqualität

* Schülernamen müssen eindeutig sein.
* Klassennamen müssen einer bestehenden Klasse zugehörig sein.
* Spalten müssen für jeden Eintrag ausgefüllt sein.

# 4 Qualitätsanforderungen

Das Definieren der Qualitätsanforderungen stellt sicher, dass das Programm unseren Erwartungen entspricht. Das Programm muss in der Lage sein, eine CSV-Datei einzulesen, ohne dass Daten verloren gehen. Es muss leichtgewichtig sein, damit andere Anwendungen des Nutzers nicht verlangsamt werden. Die Benutzeroberfläche muss intuitiv und leicht verständlich sein. Weiterhin soll das Programm schnell starten und nutzungsbereit sein.

# 5 Benutzerschnittstelle

## 5.1 Benutzermodell

Der Studentenwürfel wird für Dozenten am bib international College entwickelt. Da das bib eine IT-Schule ist, setzt die Benutzung des Programms Basiswissen über die Bedingung eines Computers und die Navigation mit der Maus voraus.

Des Weiteren wird für das Aufrufen der Protokolle Kenntnisse über die Navigation durch den Windows Explorer erwartet.

# 6 Entwicklungsumgebung

Das Projekt wird auf verschiedenen Systemen entwickelt, darunter Notebooks mit Windows 10 sowie Ubuntu, sowie auf bib-PCs. Als Programmiersprache kommt Java in der Version 17 zum Einsatz. Die Entwicklungsumgebung für das Projekt ist IntelliJ, das eine effiziente und benutzerfreundliche Umgebung für die Programmierung bietet.

Für die Versionsverwaltung wird GitHub verwendet, wodurch eine strukturierte und nachvollziehbare Entwicklung gewährleistet wird. Änderungen am Code können somit dokumentiert, gemeinsam bearbeitet und bei Bedarf rückgängig gemacht werden.

Die grafische Benutzeroberfläche wird mit JavaFX implementiert, da dieses Framework eine moderne und flexible Gestaltung von Oberflächen ermöglicht. Das Design der Anwendung wird mit Figma entworfen, um eine ansprechende und intuitive Benutzererfahrung sicherzustellen.