28.10.2020

Shenia Scherer, Andrei Oleniuc, Noah Grand I3A

IMS Eignungstest

IDPA Hauptprojekt

Inhaltsverzeichnis

[1. Teil 1 2](#_Toc54791962)

[1.1 Team und Projektdetails 2](#_Toc54791963)

[1.2 Projektarbeit 2](#_Toc54791964)

[1.2.1 Ausgangslage 2](#_Toc54791965)

[1.2.2 Aufgabenstellung 2](#_Toc54791966)

[1.2.3 Projektorganisation 2](#_Toc54791967)

[1.2.4 Projektmanagementmethode 3](#_Toc54791968)

[1.3 Deklaration der Vorkenntnisse 3](#_Toc54791969)

[1.4 Deklaration der Vorarbeiten 3](#_Toc54791970)

[1.5 Deklaration der benutzten Standards 3](#_Toc54791971)

[1.6 Zeitplan 5](#_Toc54791972)

[1.7 Arbeitsjournal 6](#_Toc54791973)

[1.8 Organisation der Arbeitsergebnisse 7](#_Toc54791974)

[2. Teil 2 8](#_Toc54791975)

[2.1 Kurzfassung des IDPA Projektes 8](#_Toc54791976)

[2.2 Abschreibungsprogramm nach IPERKA 8](#_Toc54791977)

[2.2.1 Informieren 8](#_Toc54791978)

[2.2.2 Planen 9](#_Toc54791979)

[2.2.3 Entscheiden 11](#_Toc54791980)

[2.2.4 Realisieren 11](#_Toc54791981)

[2.2.6 Auswerten 11](#_Toc54791982)

# 1. Teil 1

## 1.1 Team und Projektdetails

**Projektname:** IMS Eignungstest

**Startdatum:** 28.10.2020

**Enddatum:** 13.01.2020

**Projektmitarbeiter:**

* Shenia Scherer
* Noah Grand
* Andrei Oleniuc

**Projektbetreuer:**

* Michael Schneider
* Corrado Olivito

## 1.2 Projektarbeit

### 1.2.1 Ausgangslage

TODO

### 1.2.2 Aufgabenstellung

TODO

### 1.2.3 Projektorganisation

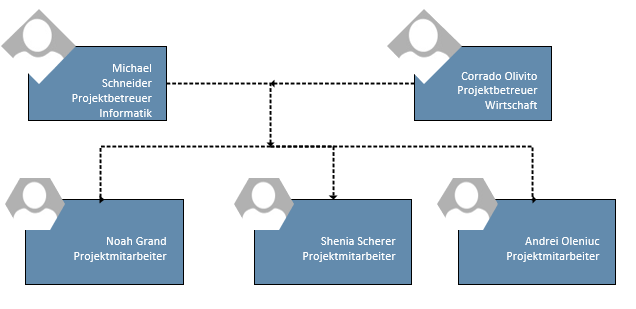


Abbildung 1 Projektorganisation

Für die Darstellung unseres Teams wurde ein Organigramm erstellt. Die Projektbetreuer sind Michael Schneider, zuständig für die Informatik und Corrado Olivito, zuständig für die Wirtschaft. Die Projektmitarbeiter sind alle auf einer Ebene, weil wir gemeinsam an einem Projekt arbeiten, in welchem es keine tieferen Strukturen braucht.

### 1.2.4 Projektmanagementmethode

Als Projektmanagementmethode haben wir IPERKA gewählt. Ihr Aufbau ist folgend aufgelistet:

1. **Informieren**
2. **Planen**
3. **Entscheiden**
4. **Realisieren**
5. **Kontrollieren**
6. **Auswerten**

Diese Methode hat viele Vorteile, insbesondere die klare Trennung zwischen Informieren und Planen. So hat jedes Gruppenmitglied die Möglichkeit sich individuell über Themen zu informieren, welche man vielleicht noch nicht so versteht. Bei IPERKA ist es auch zwingend nötig sich über die in der Planung gesammelten Ideen und Konzepte zu entscheiden. Dies kann Konflikte verhindern und eine effiziente Arbeit unterstützen. Der für die Teamarbeit wichtigste Punkt ist der Letzte, nämlich das Auswerten. Dort können sich alle Teammitglieder nochmals Gedanken über die Arbeit machen und diese dokumentieren. Es ist auch der Zeitpunkt, wo man sich eventuelle Verbesserungsmöglichkeiten und Negatives notiert, um dies bei zukünftigen Projektarbeiten zu verhindern.

## 1.3 Deklaration der Vorkenntnisse

|  |  |
| --- | --- |
| Teammitglied | Vorkenntnisse |
| Shenia Scherer | Programmierkenntnisse in Java und JavaScript, SQL, Git |
| Noah Grand | Programmierkenntnisse in Java und JavaScript, Erfahrung mit React, Git, SQL, Spring |
| Andrei Oleniuc | Programmierkenntnisse in Java und JavaScript, Erfahrung mit React, Git, SQL, Spring, SCSS |

## 1.4 Deklaration der Vorarbeiten

TODO

## 1.5 Deklaration der benutzten Standards

TODO

## 1.6 Zeitplan

TODO

## 1.7 Arbeitsjournal

TODO

## 1.8 Organisation der Arbeitsergebnisse

Als primäres Mittel für den Dateiaustausch verwendeten wir Microsoft Teams in einem eigenen Chat. Über dieses konnten wir auch unsere Dokumentation gemeinsam bearbeiten, in dem wir es alle mit SharePoint geöffnet hatten. Der Programmcode haben wir auf GitHub gespeichert. Dieses Tool garantierte uns, dass jeder von uns mit der aktuellen Version gearbeitet hat. Diese Dateien waren für uns immer zugreifbar und auch durch Versionierung dementsprechend gesichert.

1.9 Anleitung Installation und Bedienung

**Installation**

TODO

**Bedienung**

TODO

1.10 Projektverträge, Disposition

**Vertrag zum Vorprojekt**

|  |  |
| --- | --- |
| **Projektname:** | IMS Eignungstest |
| **Gruppenmitglieder mit Klasse und E-Mail:** | Noah Grand I3a, [noah.grand@students.ksba.ch](mailto:noah.grand@students.ksba.ch)  Andrei Oleniuc I3a, [andrei.oleniuc@students.ksba.ch](mailto:andrei.oleniuc@students.ksba.ch)  Shenia Scherer I3a, [shenia.scherer@students.ksba.ch](mailto:shenia.scherer@students.ksba.ch) |
| **Betreuende Lehrpersonen und E-Mail:** | Herr Schneider, [michael.schneider@kanti-baden.ch](mailto:michael.schneider@kanti-baden.ch)  Herr Olivito, [corrado.olivito@kanti-baden.ch](mailto:corrado.olivito@kanti-baden.ch) |
| **Titel:** | IMS Eignungstest |
| **Beschreibung:** | Der Abschreibungsrechner kann Anlangen auf verschiedene Methoden abschreiben (linear, degressiv, direkt, indirekt). |
| **Startdatum des Vorprojekts:** | 28.10.2020 |
| **Abgabe mit Zeitpunkt:** |  |
| **Anforderungen:** | TODO |
| **Datum und Unterschrift der Mitglieder:** | 28.10.2020, Andrei Oleniuc  28.10.2020, Noah Grand  28.10.2020, Shenia Scherer |
| **Datum und Unterschrift der Lehrpersonen:** |  |

# 2. Teil 2

## 2.1 Kurzfassung des IDPA Projektes

**Ausgangssituation**

TODO

**Umsetzung**

TODO

**Ergebnis**

TODO

## 2.2 Abschreibungsprogramm nach IPERKA

### 2.2.1 Informieren

#### 2.2.1.1 Personas

Unsere Personas richten sich an die gesamte Anwendung.

|  |  |
| --- | --- |
| Name: | Jason Meier |
| Alter: | 14 |
| Ziele: | Will wissen, ob er nach der Bezirksschule in die IMS gehen sollte |
| Aufgaben: | Muss sich entscheiden, ob er in die IMS will |
| Kenntnisse: | Programmiert schon in seiner Freizeit, weiss jedoch noch nicht viel über die IMS |
| Einstellung: | Offen, ob er in die IMS oder in die Kantonsschule geht. |

#### 2.2.1.2 Anforderungsanalyse

|  |  |
| --- | --- |
| Nummer | Anforderung |
| 1 | TODO |

#### 2.2.1.3 Use Case Diagramm

TODO

#### 2.2.1.4 Zustandsdiagramm

TODO

### 2.2.2 Planen

#### 2.2.2.1 Konzept

TODO

#### 2.2.2.2 GUI Design

TODO

Abbildung 5 GUI Design

Für das GUI-Design haben wir uns an die Richtlinien der ISO 9241-110 orientiert. Diese geben Designvorgaben an. Wir haben dies so gemacht, weil wir dieses In der Schule kennengelernt haben.

**Aufgabenangemessenheit**

TODO

**Selbstbeschreibungsfähigkeit**

TODO

**Erwartungskonformität**

TODO

**Lernförderlichkeit**

TODO

**Steuerbarkeit**

TODO

**Fehlertoleranz**

TODO

**Individualisierbarkeit**

TODO

#### 2.2.2.3 Klassendiagramm

TODO

Abbildung 6 Klassendiagramm

#### 2.2.2.4 Schnittstellen / Projektumfeld

Systemdarstellung:

TODO

Abbildung 7 Systemdarstellung

#### 2.2.2.5 Datenmodell

Datenmodell des Abschreibungsrechners:

TODO

Abbildung 8 Datenmodell

Erklärung:

TODO

#### 2.2.2.6 Fehlerbehandlung

TODO

#### 2.2.2.7 Coding-Style

Unser Java Code wird nach der Standardformatierung von NetBeans formatiert. Diese wird mit Rechtsklick in den Code -> Format erreicht. Alle Variablen und Attribute in unserem Projekt sind nach Camel-Case benannt, alle mit einem Kleinbuchstaben beginnend, wie in der Java Namenskonvention.

Die Codedokumentation setzten wir mithilfe des Javadocs um, welches in einer lokalen Webseite präsentiert wird.

TODO

#### 2.2.2.8 Testkonzept

Name des Programms: IMS Eignungstest

Testinfrastruktur

* **Testsystem:** Acer Aspire A715 71G
* **Testhilfsmittel:** TODO

Anforderungen

|  |  |
| --- | --- |
| Nummer | Anforderung |
| 1 | TODO |

Testfallspezifikationen

|  |  |
| --- | --- |
| Testfallnummer: | 1 |
| Anforderung: | 1 |
| Voraussetzungen: | TODO |
| Eingabe: |  |
| Ausgabe: |  |

### 2.2.3 Entscheiden

#### 2.2.3.1 Entscheidungsmatrix

Unsere Entscheidungen haben wir mit der Hilfe von der Entscheidungsmatrix getroffen. Sie hilft Objektive Entschlüsse zu treffen und zeigt auch auf, nach welchen Kriterien gewertet wurde. Damit können auch Dritte diese Entscheidungen analysieren und nachvollzeihen.

TODO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TODO** | Gewichtung | Punkte | | | Total |
| Kriterium |  | 3 | 2 | 1 |  |
| Verständlichkeit | 2 |  | x |  | 4 |
| Professionalität | 2 | x |  |  | 6 |
| Aufwand  (weniger = besser) | -1 | x |  |  | -3 |
| Komplexität  (weniger = besser) | -2 |  | x |  | -4 |
|  |  |  |  |  | **3** |

Erläuterung

TODO

### 2.2.4 Realisieren

**Woche 1**

TODO

## 2.2.6 Auswerten

#### 2.2.6.1 Testbericht

Die Tests verliefen reibungslos, es sind keine Probleme aufgetreten. Das Programm wird ohne Fehler ausgeführt und kann nun freigegeben werden.

#### 2.2.6.2 Probleme

Beim Testen sind keine Probleme aufgetreten.

#### 2.2.6.3 Persönliches Fazit

**Noah**

TODO

**Shenia**

TODO

**Andrei**

TODO

#### 2.3 Abbildungsverzeichnis

[Abbildung 1 Projektorganisation 3](#_Toc51157071)

[Abbildung 2 Beispiel Ausführen des Programmes 5](#_Toc51157072)

[Abbildung 3 Use case diagramm 9](https://kantonsschulebaden-my.sharepoint.com/personal/benjamin_yildirim_students_ksba_ch/Documents/Microsoft%20Teams-Chatdateien/DOKUMENTATION.docx#_Toc51157073)

[Abbildung 4 Zustandsdiagramm 10](https://kantonsschulebaden-my.sharepoint.com/personal/benjamin_yildirim_students_ksba_ch/Documents/Microsoft%20Teams-Chatdateien/DOKUMENTATION.docx#_Toc51157074)

[Abbildung 5 GUI Design 11](#_Toc51157075)

[Abbildung 6 Klassendiagramm 12](https://kantonsschulebaden-my.sharepoint.com/personal/benjamin_yildirim_students_ksba_ch/Documents/Microsoft%20Teams-Chatdateien/DOKUMENTATION.docx#_Toc51157076)

[Abbildung 7 Systemdarstellung 13](#_Toc51157077)

[Abbildung 8 Datenmodell 13](#_Toc51157078)

[Abbildung 9 "add" Methode 20](#_Toc51157079)

[Abbildung 10 Calculations Konstruktor 21](#_Toc51157080)

[Abbildung 11 Singleton Pattern 21](#_Toc51157081)

[Abbildung 12 Calculate Methode 22](#_Toc51157082)

[Abbildung 13 Befüllen der Pagination 23](#_Toc51157083)

[Abbildung 14 Unittest 24](#_Toc51157084)

[Abbildung 15 Unittest Resultate 25](#_Toc51157085)

#### 2.4 Quellenverzeichnis

TODO

#### 2.5 Glossar

TODO

|  |  |
| --- | --- |
| Begriff | *Erklärung* |
| A |  |
|  |  |
| B  Backend |  |
| Der Teil eines Programmes, der für die Berechnungen zuständig ist und für den Benutzer nicht sichtbar ist. |
|  |  |
| C |  |
| CSS | Cascading Style Sheets für das Styling des GUIs |
|  |  |
| D |  |
|  |  |
| E |  |
|  |  |
| F |  |
|  |  |
| G |  |
| GUI | Graphical User Interface (Benutzerschnittstellt) |
|  |  |
| H |  |
|  |  |
| I |  |
|  |  |
| J |  |
| Javadoc | Tool für Java Codedokumentation |
|  |  |
| K  Konstruktor |  |
| Die Methode einer Klasse, die beim Erstellen eines Objektes aufgerufen wird. |
|  |  |
| L |  |
| ListView | JavaFX GUI-Element |
|  |  |
| M |  |
| MVVM | Softwaredesignpattern |
| MVC | Softwaredesignpattern |
| Model | Bestandteil von MVC zur Datenhaltung |
| Maven | Buildtool für Java Programme |
|  |  |
| N |  |
|  |  |
| O |  |
| ObservableList | Spezielle FXML Collection, welche zum speichern von Werten in einer ListView genutzt werden kann. |
|  |  |
| P |  |
| Pagination | JavaFX GUI-Element |
|  |  |
| Q |  |
|  |  |
| R |  |
|  |  |
| S |  |
| Singleton | Softwaredesignpattern |
|  |  |
| T |  |
|  |  |
| U |  |
| Unittest | Programmierte, automatische Tests für implementierte Lösungen |
|  |  |
| V |  |
|  |  |
| W |  |
|  |  |
| X |  |
|  |  |
| Y |  |
|  |  |
| Z |  |