

Robert-Bosch Berufskolleg Dortmund

Blockwoche (20.01. – 24.01.2020) ITM7

Fachinformatiker für Anwendungsentwicklung

Dokumentation zum Blockwochenprojekt

**Fancy YOOLOO**

Make YOOLOO fancy again!

Abgabetermin: 24.01.2020

**Autoren**:

Philipp Gahlen

Lukas Loenser

Lukas Luzius

Falco Weihrauch



**fancy.IT**

Inhalt

[1 Einleitung 1](#_Toc30419821)

[1.1 Projektbeschreibung 1](#_Toc30419822)

[1.2 Projektziel 1](#_Toc30419823)

[1.3 Projektumfeld und Begründung 1](#_Toc30419824)

[1.4 Projektschnittstellen 1](#_Toc30419825)

[1.5 Projektabgrenzung 1](#_Toc30419826)

[2 Projektplanung 1](#_Toc30419827)

[2.1 Projektstruktur 1](#_Toc30419828)

[2.2 Projektablauf 1](#_Toc30419829)

[2.3 Kapazitätsplanung 1](#_Toc30419830)

[2.4 Qualitätsplanung 1](#_Toc30419831)

[3 Analysephase 1](#_Toc30419832)

[3.1 Ist-Analyse 1](#_Toc30419833)

[3.2 Lastenheft 1](#_Toc30419834)

[4 Entwurfsphase 1](#_Toc30419835)

[4.1 Zielplattform 1](#_Toc30419836)

[4.2 Architekturdesign 1](#_Toc30419837)

[4.3 Entwurf der Benutzerschnittstelle 1](#_Toc30419838)

[4.5 Programmlogik 1](#_Toc30419839)

[4.6 Pflichtenheft 1](#_Toc30419840)

[5 Implementierungsphase 1](#_Toc30419841)

[5.1 Pflichtmodul 1: bla bla 1](#_Toc30419842)

[5.2 Pflichtmodul 2: bla bla 1](#_Toc30419843)

[5.3 Pflichtmodul 3: bla bla 1](#_Toc30419844)

[5.4 Optionalmodul 1: bla bla 1](#_Toc30419845)

[6 Testphase 1](#_Toc30419846)

[6.1 Test der Funktionalität 1](#_Toc30419847)

[6.2 Test der Benutzerfreundlichkeit 1](#_Toc30419848)

[7 Dokumentation 1](#_Toc30419849)

[8 Fazit 1](#_Toc30419850)

[8.1 Soll- / Ist-Vergleich 2](#_Toc30419851)

[8.2 Retroperspektive 2](#_Toc30419852)

[A. Anhang i](#_Toc30419853)

[A.1 asd i](#_Toc30419854)

# Abbildungsverzeichnis

# Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden. Tabellenverzeichnis

[Tabelle 1: Modulzuteilung 1](#_Toc30421804)

[Tabelle 2: Zeitplanung 1](#_Toc30421805)

# Einleitung

## Projektbeschreibung

## Projektziel

## Projektumfeld und Begründung

## Projektschnittstellen

## Projektabgrenzung

# 2 Projektplanung

## 2.1 Projektstruktur

Um das Projekt sinnvoll bearbeiten zu können macht es Sinn die verschiedenen Aufgaben in Module zusammenzufassen und auf die Entwickler aufzuteilen. Hierzu haben sich die Entwickler vor der Umsetzung darauf geeinigt wer welche Aufgabe übernehmen will. Ein weitere Vorteil dieses Vorgehens ist es, das nicht jeder Entwickler jeden Teilbereich des Projektes analysieren muss, um eine Einschätzung abgeben zu können, wie viel Aufwand die Umsetzung in Anspruch nimmt. Nachfolgend findet sich eine Übersicht in Tabelle 1: Modulzuteilung, welcher Entwickler welches Modul bearbeitet hat.

|  |  |  |
| --- | --- | --- |
| Art des Moduls | Name des Moduls | Entwickler |
| Pflichtmodul | Spielweise | Lukas Loenser |
| Pflichtmodul | Regeleinhaltung | Philipp Gahlen |
| Pflichtmodul | Spielerkonto | Lukas Luzius |
| Optionalmodul | Bot | Falco Weihrauch |
| Optionalmodul | Logger | Alle |

Tabelle 1: Modulzuteilung

## 2.2 Projektablauf

Für die Umsetzung des Projektes standen der Gruppe 34 Stunden Arbeit zur Verfügung. Diese wurden vor Beginn der Umsetzung in mehrere Phasen aufgeteilt, die bei der Umsetzung von Software-Projekten typischerweise durchlaufen werden. Folgend lässt sich aus Tabelle 2: Zeitplanung entnehmen, wie viele Stunden welcher Phase des Projektes zugeordnet wurden. Anzumerken ist hier, dass es sich um Arbeitsstunden handelt, d.h. da die Gruppe aus 4 Entwicklern bestand ergibt sich eine insgesamte Zeit von 4 Entwickler \* 34 Stunden = 136 Stunden.

|  |  |
| --- | --- |
| Projektphase | Geplanter Zeitaufwand (Stunden) |
| Analysephase | 15 |
| Entwurfsphase | 20 |
| Implementierungsphase | 45 |
| Testphase | 16 |
| Dokumentation | 40 |
| Gesamt | 136 |

Tabelle 2: Zeitplanung

Außerdem lässt sich in Anhang TODOTODOTODO eine Übersicht des Projektablaufs als Gantt-Diagramm finden um den zeitlichen Ablauf besser nachvollziehen zu können.

## 2.3 Kapazitätsplanung

## 2.4 Qualitätsplanung

Gerade bei der Bearbeitung von Aufgaben durch mehrere Entwickler, wird die Kontrolle der Qualität von Code und Ergebnis immer wichtiger.

# 3 Analysephase

## 3.1 Ist-Analyse

## 3.2 Lastenheft

# 4 Entwurfsphase

## 4.1 Zielplattform

## 4.2 Architekturdesign

## 4.3 Entwurf der Benutzerschnittstelle

## 4.5 Programmlogik

## 4.6 Pflichtenheft

# 5 Implementierungsphase

## 5.1 Pflichtmodul 1: bla bla

## 5.2 Pflichtmodul 2: bla bla

## 5.3 Pflichtmodul 3: bla bla

## 5.4 Optionalmodul 1: bla bla

# 6 Testphase

## 6.1 Test der Funktionalität

## 6.2 Test der Benutzerfreundlichkeit

# 7 Dokumentation

# 8 Fazit

## 8.1 Soll- / Ist-Vergleich

## 8.2 Retroperspektive

# Anhang

## A.1 asd