<Overtime>

Pflichtenheft <0.1>

Verfasser:

<Felix Jopkiewicz >

<Fabio Boran >

<Dejan Sunaric >

<Filip Josipovic >

<Eldi Neziri >

**Anmerkung:**

Das vorliegende Dokument ist eine Vorlage zur Erstellung eines **Pflichtenheftes** für „kleine“ Projekte. Die im Dokument enthaltenen Textteile in blauer Schrift sind als Erläuterungen zu verstehen; sie müssen bei der Erstellung gelöscht und durch projektspezifische Angaben ersetzt werden.

Dokumentenverwaltung

**Dokument-Historie**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Status** | **Datum** | **Verantwortlicher** | **Änderungsgrund** |
| **0.1** | **Entwurf** | **24\04\2020** | **Felix Jopkiewicz** | **Erstellung 1. Entwurf** |
|  |  |  |  |  |
|  |  |  |  |  |

**Dokument wurde mit folgenden Tools erstellt:**

<Pflichtenheft> Microsoft Word

<Websitetool> HTML, CSS

<Programmiertool> Visual Studio, Notepad++

<Repositorytool> Github

<Zeitplanungstool> GanttProject

<Präsentationstool> PowerPoint

Inhalt

1 Einleitung 4

1.1 Zweck und Gültigkeit des Dokuments 4

1.2 Zusammenhang mit anderen Dokumenten 4

2 Allgemeine Beschreibung des gewünschten Systems 4

2.1 Zweck 4

2.2 Überblick Funktionalität 4

2.3 Abgrenzung und Einbettung 4

2.4 Allgemeine Vorgaben und Einschränkungen 4

2.5 Anforderungsquellen und Zielgruppen 4

3 Detaillierte Beschreibung der Leistungsmerkmale des Systems 4

3.1 Funktionaler Lieferumfang 4

3.2 Interaktionen mit Umgebung 4

3.3 Geforderte Funktionen 4

3.4 Struktur und Verhalten 5

3.5 Benutzerschnittstelle 5

3.6 Datenbank-Schnittstelle 5

3.7 Sonstige Schnittstellen 5

3.8 Sonstige entwicklerorientierte Anforderungen 5

4 Vorgaben des Auftraggebers an die Projektabwicklung 5

4.1 Anforderungen an die Realisierung 5

4.2 Abnahme- und Lieferbedingungen 5

4.3 Gewährleistung 5

5 Verpflichtungen des Auftraggebers 6

6 Literatur 6

7 Begriffe und Abkürzungen 6

8 Anhang 6

# Einleitung

## Zweck und Gültigkeit des Dokuments (für ganzes Projekt)

Zielsetzung des Dokuments und Adressatenkreis.

Für welche Anwendungsdomäne: Gesamtes Projekt? Basissystem? Delta-Entwicklung?

Der Zweck dieses Dokuments ist die Umschreibung des Umfelds und des Hintergrunds des gesamten Projekts.

## Zusammenhang mit anderen Dokumenten

Bezug zu anderen Dokumenten (Lastenheft, System Spezifikation, ...). Aufbauend? Teil?

Es gibt einen Bezug zum Lastenheft, welches uns von unserem Arbeitgeber ausgehändigt wurde.

# Allgemeine Beschreibung des gewünschten Systems

## Zweck (wie Projekthintergrund, Sinn, Wieso mach ich das)

Ziel und Einsatz des Produkts aus Auftraggebersicht.

Unser Auftraggeber will nicht nur während den regulären Schulstunden ein sicheres Umfeld für seine Schüler bieten, sondern auch nach dem regulären Unterricht, welchen die Schüler in der Schule verbringen.

## Überblick Funktionalität(Was wird das ganze können/nicht können)

Hier werden die wesentlichen Leistungen aufgelistet, die das System erbringen wird (Kurzfassung von Kapitel 3).

Falls vorgesehen, auch die Planung von Ausbaustufen.

Gegebenenfalls auch die Leistungen, die das System nicht erbringen wird.

Es bietet den Schülern die Möglichkeit nach ihrem regulären Unterricht länger in der Schule zu bleiben und dies den Autoritätspersonen mitzuteilen. Zudem bietet es Personen mit Autorität eine klare Übersicht, über Schüler die länger bleiben wollen.

## Abgrenzung und Einbettung (Personalstem, externe Systeme, Webuntis zum anschließen, wahrscheinlich keine Einbettung)

Abgrenzung zu anderen Systemen in der Umgebung.

HW- und SW-Umgebung des Zielsystems.

Zu den Systemen die jeder aus dem Projektteam Overtime benutzen wird zählen: Visual Studio, Notepad++, HTML, CSS, Microsoft 365, GanttProject und Github.

Zu den externen Systemen zählt der Schulserver der htl-donaustadt, auf dem unsere fertige Website gehosted wird. Die Website soll als eigener Tab auf der Schulwebsite der htl-donaustadt vorhanden sein.

## Allgemeine Vorgaben und Einschränkungen (Speicher, CPU Generation, Windows 10, Spezifische Angaben)

Vorgaben zu HW, SW, Schnittstellen, Standards, Methoden.

* Software
  + Betriebssysteme: Windows 10 64 bit
  + Arbeitstools: Visual Studio, Notepad++, HTML, CSS, Microsoft 365,
  + Koordinationstools: GanttProject
  + Repositorytool: Github
* Hardware
  + CPU: Intel Core i5 und aufwärts
  + Speicher: 128 GB und aufwärts
  + Schnittstellen: USB
* Schnittstellen zu anderen Geräten
  + Server der htl-donaustadt

## Anforderungsquellen und Zielgruppen (Schüler, Benutzer, Stakeholder)

Benutzer des gewünschten Systems.

* Schüler
* Head of htl-donaustadt - mister Bonatz
* Housekeeper

Andere Stakeholder.

* Parents
* Mister Dassler
* Police/firefighters
* Teachers

# Detaillierte Beschreibung der Leistungsmerkmale des Systems

## Funktionaler Lieferumfang (Welche Funktionen Windows Applikation mit 5 Aktionen, User anlegen etc)

Beschreibung der zu liefernden Hauptfunktionalität.

* Variante 1
  + Login mit eigenem Account (Benutzername und Passwort). Benutzername und Passwort entsprechen den gleichen Login Daten der Schuldienste (Schul-Mail, Moodle).
    - Ansicht eines Schülers
      * Als Schüler kommt man nun auf seinen eigenen Account und kann nun **taggleich** bestimmen von wann bis wann man länger bleiben möchte.
      * Wenn ein Schüler die Schule früher verlässt als geplant muss er sich noch ein Mal einloggen und seinen Eintrag löschen.
      * Er hat aber auch die Möglichkeit nach den Extrastunden noch länger in der Schule zu bleiben, sofern er dies auch einträgt.
      * Wenn die Zeit der Extrastunden abgelaufen ist und der Schüler sich an diese Zeit hält, muss er sich nicht manuell austragen, sondern wird automatisch vom ausgetragen
      * Die Einträge, die der Schüler tätigt, gelten nur für den **taggleichen Tag**, dass bedeutet, dass der Schüler nicht planen kann ob er zum Beispiel morgen auch länger bleibt
  + Autoritätsperson
    - Autoritätspersonen müssen sich nicht anmelden, sondern bekommen einen Link, mit dem sie auf die Übersicht über die Schüler, die länger bleiben, zugreifen können.
* Variante 2
  + Hier erfolgt die Anmeldung über Moodle. Damit bestätigt man, dass man ein Schüler der htl-donaustadt ist. Auf einem Allgemeinverzeichnis auf Moodle gibt es dann einen Link der einen zu der Overtime Website führt. Anders als bei Variante 1 trägt man hier nun seinen eigenen Namen ein, anstatt dass man bereits angemeldet ist. Alle weiteren Funktionen sind wie bei Variante 1.
    - Nachteil
      * Man kann nun einen beliebigen Namen eingeben welcher nicht dem eigenen entsprechen muss. Dies kann zu falschen Informationen führen.
  + Für Autoritätspersonen gibt es wie bei Variante 1 einen Link der zur Übersicht über alle Schüler führt.

Beschreibung der zu liefernden System-Bestandteile

* Notepad++ files
  + Ein HTML file für das Grundgerüst der Loginwebsite
  + Ein HTML file für das Grundgerüst der Hauptübersicht
  + Insgesamt drei CSS files für das individuelle Aussehen der HTML files
  + Ein PHP file für die Programmierung
* Visual Studio
  + Ein SQL file mit der Datenbank

Falls Ausbaustufen, dann Bezugnahme auf die Ausbaustufe.

## Interaktionen mit Umgebung (logischer Ablauf: 1. Manager meldet sich an 2: Manager wählt eine Woche aus 3: Manager verschickt Dienstplan)

Typische Szenarien (Abläufe) beschreiben.

Z.B. mit Anwendungsfall-Diagrammen darstellen.

## Geforderte Funktionen (Wenn oben steht user anlegen muss hier stehen was es bedeutet, also wenn man User anlgt passiert das, im hintergrund wir xzy gemacht etc)

Das Produkt wird aus funktionaler Sicht anhand von Anwendungsfällen (Use Cases) beschrieben. Jeder Anwendungsfall wird z.B. in Form einer Tabelle spezifiziert:

|  |  |
| --- | --- |
| **Bezeichnung** | *Name des Anwendungsfalls* |
| **Zusammenfassung** | *Kurzbeschreibung (was wird bei erfolgreicher Ausführung erreicht)* |
| **Akteure** | *Akteure (Personen oder andere Systeme), die den Anwendungsfall auslösen oder bei der Durchführung involviert sind* |
| **Vorbedingung** | *Voraussetzungen, die zur Ausführung des Anwendungsfalls erfüllt sein müssen* |
| **Ablaufbeschreibung** | *Strukturierte Ablaufbeschreibung (alle Aktionen des Standardfalls)* |
| **Verwendungen**  **(Include-Beziehungen)** | *Auflistung aller Anwendungsfälle, die von dem Anwendungsfall verwendet werden* |
| **Erweiterungen**  **(Extend-Beziehungen)** | *Auflistung aller Anwendungsfälle, die den Anwendungsfall erweitern.* |
| **Alternativen** | *Optionen und Alternative Ausführung einzelner Aktionen* |
| **Nachbedingung** | *Resultate, die der Anwendungsfall liefert* |
| **Fehlschlag** | *Erwarteter Zustand, wenn das Ziel nicht erreicht werden kann* |

## Struktur und Verhalten (Daweil noch leer lassen)

Beschreibung der verschiedenen statischen Strukturaspekte des Systems (Klassen-, Paket-, Komponenten-, Verteilungsdiagramm).

Beschreibung der Dynamik, der internen Abläufe und des Zusammenspiels der Systemteile (Aktivität-, Sequenz-, Zustand-, Timingdiagramm).

## Benutzerschnittstelle (eher Grafiken verwenden, Prototypen zeichnen, UI 1 Login, UI2 Create Name)

Beschreibung der Benutzerschnittstelle. User-Interface-Konzept.

## Datenbank-Schnittstelle (wie werden die Daten gespeichert die in der Applikation anfallen, wir verwenden Levels(spiel assets) abgelegt, wie sieht die Speicherungsmethode aus, ER Diagramm(falls es eine Datenbank ist))

Beschreibung der Datenbankschnittstelle. ER-Diagramm.

## Sonstige Schnittstellen (Andere HW/SW verwendet und daten bezieht muss diese Schnittstelle genau angegeben werden und wie diese funktioniert)

Beschreibung weiterer Schnittstellen zu Soft- und Hardwaresystemen.

## Sonstige entwicklerorientierte Anforderungen (Ob die Speicherung gesichert ist, Ob alle zugriff brauchen, Unter folgenden Anforderungen garantieren wir 60 fps)

Spezifikation von Anforderungen hinsichtlich Performance, Ressourcen, Safety (Schutz und Sicherheit), Datensicherheit, Portabilität, Reliability, Wartung, Wiederverwendung, Usability, Serviceability, Qualität, zu berücksichtigende Normen.

# Vorgaben des Auftraggebers an die Projektabwicklung

## Anforderungen an die Realisierung (Developer Workstations, Welche Software inklusive Versionen, Wasserfallmodell bei Entwicklungsmethode)

Z. B. Angaben über zu verwendende Software, Hardware, Entwicklungsmethode, Termine, zugekaufte Produkte.

Z. B. Standardsoftware, Software des Auftraggebers, wiederverwendete eigene Software, Betriebssysteme, ...

## Abnahme- und Lieferbedingungen (Liefern eine HW mit oder ohne USB, wie viele Levels)

Bedingungen des Auftraggebers für die Abnahme: Wie? Welche Unterlagen?

Lieferplan mit Terminen, Form der Lieferungen.

Geforderte Dokumentation

## Gewährleistung (Falls Fehler auftaucht wie lang bis man das kostenfrei aufhebt)

Gewährleistungsdauer, Umfang der Gewährleistung, Fehlermeldungsverfahren, ...

# Verpflichtungen des Auftraggebers (Antwortet der Auftraggeber, Wirkt dieser mit, testen lassen)

Z.B.: gestellte Hardware / Software, Schulung von Entwicklern oder von Auftraggeber personal, Ansprechpartner, zur Verfügung stellen von Räumen, Rechenzentrum, Reaktionszeiten des Auftraggebers auf Anfragen.

# Literatur (Syp buch, Quellen)

# Begriffe und Abkürzungen (Alle Abkürzungen müssen hier stehen zb GUI Graphical User Interface)

Die Begriffsdefinitionen sind deshalb so besonders wichtig, weil das Pflichtenheft die gemeinsame Verständnisbasis von Auftraggeber und Auftragnehmer darstellt.

# Anhang (Wenn vielleicht etwas hinzugefügt werden muss, wo man dieses noch nicht hinzugefügt hat)

Relevante Dokumente, die zusätzliche Informationen über das zu erstellende System geben, z.B. Modelle, detaillierte Spezifikationen, …