Ausbildung Projektarbeit

**Projektdokumentation zum Projekt:**

**Rotten Tomatoes But Better (RTBB)**

**Namen der Mitglieder:** Cassiano Speroto, Benis Manxhuku, Philip Weinand, Marlon Zwank, Denis Becker

**Projektleiter:** Cassiano Speroto

**Projektabnahmedatum:** 06.07.2023 – 19Uhr

**Ausbildungsberuf:** Fachinformatiker Anwendungsentwicklung

**Einsatzgebiet:** Berufsbildende Schule Gestaltung und Technik Trier

**Ausbildungsbetrieben:** Universität Trier und Verbandsgemeinde

**1. Ausgangssituation**

**1.1. Ausgangslage**

Angesichts des bekannten Programms 'Higher Lower' und der Vielzahl an Informationen, die es zu Filmen gibt, haben wir uns dazu entschieden, eine Filmdatenbank zu entwickeln. Diese ermöglicht es uns, Filme und die dazugehörigen Daten zu erfassen und abzurufen. Zusätzlich entwerfen wir eine webbasierte Benutzeroberfläche, die es dem Nutzer ermöglicht, Filme anhand verschiedener Kriterien miteinander zu vergleichen. Darüber hinaus stellen wir ein weiteres Webinterface bereit, über das Nutzer Wünsche und Feedback äußern können. Außerdem können sie eigene Favoriten anhand des 'Higher Lower'-Prinzips bestimmen und diese in ihrem Profil anzeigen lassen.

**1.2. Projektziele**

**Filmdatenbank (IMDB) mit Interaktion:** Das Hauptziel dieses Projekts besteht darin, eine umfassende Filmdatenbank zu entwickeln, die auf dem bewährten Modell der International Movie Database (IMDB) basiert. Die Datenbank wird es den Nutzern ermöglichen, nach Filmen zu suchen, detaillierte Informationen abzurufen und Interaktionen wie Bewertungen, Kommentare und Favoriten hinzuzufügen.

**Webinterface für Administratoren und Benutzer:** Wir werden ein benutzerfreundliches Webinterface entwickeln, das sowohl Administratoren als auch Benutzern zugänglich ist. Administratoren erhalten privilegierten Zugriff, um Filmdaten zu verwalten, neue Filme hinzuzufügen, Daten zu aktualisieren und Benutzerfeedback zu überprüfen. Benutzer werden in der Lage sein, die Filmdatenbank zu durchsuchen, Filme zu vergleichen, ihre Favoriten zu speichern und ihre eigenen Bewertungen und Kommentare abzugeben.

**Verschiedene Modi/Spiele:** Um den Unterhaltungswert zu steigern, werden wir verschiedene Modi und Spiele innerhalb der Filmdatenbank implementieren. Dazu gehören beispielsweise Quiz-Spiele, in denen die Benutzer ihr Filmwissen testen können, und Empfehlungsmodi, die auf ihren Vorlieben basieren. Diese Funktionen werden die Benutzererfahrung bereichern und die Bindung an die Plattform stärken.

**Profil mit Ratingsystem:** Jeder Benutzer wird ein personalisiertes Profil haben, das es ihnen ermöglicht, ihre Aktivitäten zu verfolgen und ihre Erfahrungen innerhalb der Filmdatenbank zu personalisieren. Ein integriertes Ratingsystem erlaubt es den Benutzern, Filme zu bewerten und ihre Bewertungen mit anderen Nutzern zu teilen. Dadurch entsteht eine Community, in der sich Filminteressierte austauschen und inspirieren lassen können.

**1.3. Teilaufgaben**

In den ersten Gesprächen mit den Projektmitgliedern wurden der Ist- und der Soll-Zustand der Website im Pflichtenheft festgehalten. Daraus ergaben sich die folgenden Schritte:

**Einrichten der Entwicklungsumgebung**

Wir sollten eine Anwendung in der gewünschten Sprache entwickeln. Die Aufgaben werden auf die einzelnen Mitglieder verteilt, um den Entwicklungsprozess zu beschleunigen. Wir haben uns für ein Framework von Python entschieden, weil es die Entwicklung der Anwendung wesentlich einfacher machen würde, da wir so oder so eine Sprache lernen mussten

**Implementierung einer Datenbank**

Aufgrund früherer Erfahrungen schlug eines unserer Mitglieder die Verwendung eines VPS (Virtual Private Server) vor. Auf diese Weise könnten wir nur eine Datenbank haben, die von uns allen gemeinsam genutzt wird, anstatt eine auf jedem Gerät mit unterschiedlichen Daten zu speichern.

**1.4. Projektabgrenzung**

**Erfassung wichtiger Filmdaten:** Das Hauptziel dieses Projekts besteht darin, wichtige Informationen über Filme zu sammeln, darunter der Name, das Jahr, der Regisseur, Auszeichnungen und andere relevante Daten. Durch die Zusammenstellung dieser Daten möchten wir eine umfassende und genaue Sammlung von Filmdaten aufbauen.

**Integration in eine Datenbank:** Die gesammelten Filmdaten werden in eine strukturierte Datenbank eingelesen, um eine effiziente Verwaltung und den einfachen Zugriff auf die Informationen zu ermöglichen. Die Datenbank wird es uns ermöglichen, die Filmdaten effektiv zu organisieren, zu durchsuchen und zu aktualisieren.

**Erstellung einer Webseite:** Wir werden eine ansprechende und benutzerfreundliche Webseite entwickeln, auf der die Nutzer auf die gesammelten Filmdaten zugreifen können. Die Webseite wird eine intuitive Navigation und Suchfunktionen bieten, um den Benutzern as einfache Finden und Erforschen von Filmen zu ermöglichen.

**Entwicklung eines interaktiven Interfaces:** Das Projekt beinhaltet die Erstellung eines interaktiven Interfaces, das den Benutzern eine reibungslose und ansprechende Erfahrung bietet. Wir werden Funktionen wie Bewertungen, Kommentare, Favoritenlisten und Empfehlungen implementieren, um die Interaktivität und den Mehrwert für die Nutzer zu erhöhen.

**2. Ressourcen- und Ablaufplanung**

**2.1. Verfügbare Hard- und Software**

Die Planungs- und Dokumentationsphase wurde auf den privaten Laptops unserer Mitglieder durchgeführt.

Die verwendete Software ist kostenneutral. Wir haben folgende Softwares verwendet:  
Visual Studio Code  
Notepad++

GitHub Desktop

**2.2. Zeit- und Projektplanung**

**2.2.1 Meilensteine**

**Meilenstein 1: Flask lernen**  
In diesem Meilenstein konzentrieren wir uns auf das Erlernen des Flask-Frameworks. Wir werden uns mit den grundlegenden Konzepten und Funktionen von Flask vertraut machen, um eine solide Grundlage für die Entwicklung unserer Filmdatenbank und des Webinterfaces zu schaffen.

**Meilenstein 2: Erstellung der Datenbank**  
Im zweiten Meilenstein werden wir die Datenbankstruktur für unsere Filmdatenbank definieren und erstellen. Wir werden die erforderlichen Tabellen und Beziehungen entwerfen, um die Filmdaten effizient zu speichern und abzurufen. Die korrekte Integration der Datenbank ist von entscheidender Bedeutung für den reibungslosen Betrieb unserer Anwendung.

**Meilenstein 3: Datenbankinteraktionen erstellen**  
Nachdem die Datenbank eingerichtet ist, werden wir uns im dritten Meilenstein auf die Implementierung der Datenbankinteraktionen konzentrieren. Dies umfasst das Hinzufügen, Aktualisieren und Löschen von Filmdaten sowie das Abrufen von Informationen aus der Datenbank. Wir werden sicherstellen, dass die Interaktionen effizient und fehlerfrei funktionieren.

**Meilenstein 4: Interface erstellen**  
Im vierten Meilenstein steht die Erstellung des Webinterfaces im Fokus. Wir werden ein ansprechendes und benutzerfreundliches Design entwerfen und die erforderlichen Seiten und Funktionen entwickeln, um den Benutzern die intuitive Navigation durch die Filmdatenbank zu ermöglichen. Dabei legen wir Wert auf eine konsistente Benutzeroberfläche und eine gute Benutzererfahrung.

**Meilenstein 5: Spiele implementieren**  
Der fünfte Meilenstein beinhaltet die Implementierung verschiedener Spiele oder Modi innerhalb unserer Filmdatenbank. Wir werden Funktionen wie Quiz-Spiele oder Empfehlungsmodi entwickeln, um den Unterhaltungswert für die Benutzer zu steigern und ihnen zusätzliche Möglichkeiten zur Interaktion mit der Plattform zu bieten.

**Meilenstein 6: Profile erstellen**  
Im letzten Meilenstein werden wir die Erstellung von Benutzerprofilen implementieren. Jeder Benutzer wird ein personalisiertes Profil haben, in dem er seine Bewertungen, Favoriten und andere Informationen verwalten kann. Dies ermöglicht es den Benutzern, ihre Filmerfahrungen anzupassen und ihre Aktivitäten innerhalb der Plattform zu verfolgen.

**2.2.2 Projektablaufplan**

Das folgende Schaubild zeigt die verschiedenen Projektphasen mit ihrer Dauer und ihren Abhängigkeiten zueinander. Die Berufsschultage sind nicht explizit dargestellt.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **May** | | | | | **Juni** | | | | | | | | | | | | | | | | | | | | | | | | | **Juli** | | | | |
| **Vorgang** | **KW21** | | | | | **KW22** | | | | | **KW23** | | | | | **KW24** | | | | | **KW25** | | | | | **KW26** | | | | | **KW27** | | | | |
| **Flask Lernen** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Analyse** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Planung** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Entwicklung** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Test** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Dokumentation** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Abnahme** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**2.2.3 Zeitplan**

Das Zeitkontingent ist wie folgt auf die Projektphasen aufgeteilt.

|  |  |  |
| --- | --- | --- |
| **Aufgaben** | **KW** | **Zeit** |
| **Vorbereitung und Planung** |  | **8h** |
| Ermittlung des Ist- und Soll-Zustands |  | 3h |
| Festlegung der einzelnen Durchführungsschritte |  | 3h |
| Einrichtung der Entwicklungsumgebung |  | 2h |
|  |  |  |
| **Realisierung des Soll-Zustands** |  | **50h** |
| Entwicklung des Frontends |  | 35h |
| Entwicklung des Backends |  | 5h |
| Entwicklung der Datenbank |  | 10h |
|  |  |  |
| **Qualitätssicherung** |  | **9h** |
| Tests |  | 2h |
| Bug Behebung |  | 7h |
|  |  |  |
| **Dokumentation** |  | **8h** |
| Verfassen |  | 6h |
| Korrektur |  | 2h |
|  |  |  |
| **Gesamt** | **7 Wochen** | **75h** |

**2.3. Termine**

Während der Projektphase finden mehrere Treffen mit den Projektmitgliedern statt, um den Fortschritt gemeinsam zu besprechen. Bei eventuellen Schwierigkeiten kann eine Lösung gefunden werden.

In der folgenden Tabelle sind die Termine aufgeführt, die mit allen Mitgliedern abgehalten wurden:

|  |  |
| --- | --- |
| Datum | Thema |
| 26.05.2023 | Aufgabe Verteilung |
| 02.06.2023 | Datenbank erstellen |
| 09.06.2023 | CRUD, Login und Homepage Entscheidungen |
| 16.06.2023 | Spiele Erstellung |

**3. Durchzuführende Prozessschritte**

**3.1. Planung**

Das Projekt begann am 22.05.2022 mit der Aufgabenstellung durch Herrn Maier. Die Anforderungen an die Anwendung, wie z.B. welche Sprache wir verwenden sollen, was die Funktionen unserer Anwendung sind und die Termine wurden alle noch am selben Tag definiert.

**Planung des Frontends**

Wir planten, wie wir die Aufgaben anhand der Ansichten für unsere Anwendung verteilen würden.

**Philip Weinand** würde sich um die Login-Funktionen und die Sicherheit kümmern.

**Marlon Zwank** und **Benis Manxhuku** entwickeln eine Homepage mit integriertem Spiel.

**Denis Beckers** Aufgabe war es, ein Impressum zu erstellen und Unterstützung bei der Entwicklung der anderen Ansichten zu bieten.

**Cassiano Speroto** entwickelte die Struktur des Projektes und die Entwicklung einer CRUD-Ansicht.

**Planung des Backends**

Für die Entwicklung des Backends war die Aufgabe aufgrund der Fähigkeit von Flask, Modelle zu verarbeiten, sehr einfach. Wir mussten die Schemata für die Datenbank erstellen, aber aufgrund der Einfachheit unserer Anwendung waren nur sehr wenige Zeilen Code notwendig, um das Backend zu konfigurieren.

**Planung der Datenbank**

Für die Planung im Zusammenhang mit dem Backend wurde ein ER-Modell über die benötigten Entitäten erstellt und dem Kunden vorgestellt. Die Attribute zu den Entitäten wurden ebenfalls in dem Modell festgehalten und erläutert.

**3.2. Durchführung**

**Realisierung des Frontends**

Da wir weder mit Python noch mit Flask Erfahrung hatten, nahmen wir uns ein paar Tage Zeit, um die Sprache so weit zu lernen, dass wir unser Projekt weiterführen konnten. Aufgrund der Anzahl der Mitglieder entschieden wir uns für die Nutzung von GitHub, um die Entwicklung der Ansichten sauberer zu halten und gleichzeitig laufen zu lassen. Wir erstellten eine kleine Struktur unseres Projekts, die wir Tag für Tag mit neuem Code ergänzten, um die Entwicklung des Projekts organisch zu verbessern. Das Routing-System in Flask ist sehr einfach und mächtig und erlaubte uns, POST- und GET-Anfragen einfach zu verwalten. Durch die Verwendung von Flask können viele Funktionen, deren Erstellung und Implementierung viel Zeit in Anspruch nehmen würde, jetzt mit einer einfachen Flask/Python-Funktion in viel weniger Code ausgeführt werden.

**Realisierung des Backends**

Die Verbindung zwischen unserer Datenbank und unserer Anwendung war keine Aufgabe, auch wenn nur 1 Zeile Code in Flask notwendig war. Die gesamten Verbindungsprobleme wurden gelöst, als wir unsere Datenbank in einem VPS (Virtual Private Server) implementierten.

Das Hinzufügen neuer Tabellen zur Datenbank war keine große Herausforderung, da Flask, wie bereits erwähnt, mit Modellen arbeitet. Bei der Entwicklung der Anwendung wird kein externes Dokument erzeugt oder benötigt, alles wird im Code behandelt.

**Realisierung der Datenbank**

Wir haben uns aufgrund der Realisierbarkeit für PostgreSQL als Datenbank entschieden. Durch die Installation der GUI und des pgServers in unserem VPS haben wir es geschafft, nur einen Ort zu haben, um mit der Installation, der Konfiguration und den Datenänderungen zu interagieren.

**3.3. Testen**

Während der Entwicklung haben wir ständig Usability- und Stabilitätstests durchgeführt. Außerdem wurden die Funktionen im Backend durch Testfälle geprüft.

**3.4. Dokumentation**

Die Projektdokumentation wurde fortlaufend während der Projektarbeit ergänzt.

4. Projektergebnisse

4.1. Soll-Ist-Vergleich

4.2. Zeitanalyse

4.3. Probleme bei der Projektumsetzung

4.4. Ausblick

5. Quellen

6. Anlage