

**Hausarbeit**im Bildungsgang

„Staatlich geprüfte/r Wirtschaftsinformatiker/in“

gemäß §5 der Ausbildungs- und Prüfungsordnung

Entwicklung einer Entertainmentsoftware zur Visualisierung von Musik/Sound „MuseSpectra“

vorgelegt von: Alexander Melchior

Klasse: HBFS-WI22ZAS

Adresse: Grubenstraße 2

Ort: 66265 Heusweiler

E-Mail: [alex.melchior1811@gmail.com](mailto:alex.melchior1811@gmail.com)

Abgabetermin: 27.05.2024

Betreuer/in: Herr Lukas

Inhalt

[Einführung 3](#_Toc167454705)

[IST-Situation 3](#_Toc167454706)

[SOLL-Situation 3](#_Toc167454707)

[Sollzustand 3](#_Toc167454708)

[Systemanforderungen 4](#_Toc167454709)

[Qualitätsanforderungen 4](#_Toc167454710)

[Schnittstellen 4](#_Toc167454711)

[Risikoabschätzung 4](#_Toc167454712)

[Abnahmekriterien 5](#_Toc167454713)

[MUSS-Kriterien 5](#_Toc167454714)

[KANN-Kriterien 5](#_Toc167454715)

[Use-Case-Diagramm 6](#_Toc167454716)

[Projektplan 7](#_Toc167454717)

[Produktumgebung 7](#_Toc167454718)

[Skizze des GUI 8](#_Toc167454719)

[DB-Entwurf 9](#_Toc167454720)

[Link zu gehosteten Git-Repository 9](#_Toc167454721)

[Testplan 10](#_Toc167454722)

[Benutzte Werkzeuge und Quellen 13](#_Toc167454723)

# Einführung

In einer nicht allzu fernen Zukunft hat die Welt sich zu einem Ort entwickelt, in dem Kreativität und Technologie miteinander verschmelzen. Das innovative Unternehmen a.m.prdcts hat sich das Ziel gesetzt eine revolutionäre Software namens „MuseSpectra“ zu entwickeln, die die Grenzen zwischen Musik und visueller Kunst aufheben soll. Durch die Zusammenarbeit von Programmierern, Künstlern und Tontechnikern entsteht ein kreativer Prozess, der die Grenzen der konventionellen Softwareentwicklung sprengt. Die Software soll nicht nur einfache Visualisierungen bieten, sondern auch den individuellen Stil jedes Künstlers einfangen und in dynamische visuelle Meisterwerke umwandeln während dabei sichergestellt wird, dass die Anwendung benutzerfreundlich bleibt, damit sie von jedem, ohne Vorkenntnisse, genossen werden kann. Da ein Projekt solchen Ausmaßes viel mehr Zeit in Anspruch nimmt, als uns zur Verfügung steht, handelt es sich hierbei lediglich um einen Prototypen.

# IST-Situation

Zurzeit haben unsere Kunden nur begrenzte Möglichkeiten, ihre Musik zu visualisieren. Es fehlt an einer spezialisierten Software, die ihren Bedarf an innovativen visuellen Darstellungen erfüllen kann. Die Idee für das Projekt entstand aus dem Wunsch, eine einzigartige Software zu entwickeln, die Künstler eine neue, inspirierende Möglichkeit bietet, ihre Musik durch visuelle Kunstwerke zu präsentieren.

# SOLL-Situation

## Sollzustand

Die fertige Software ermöglicht es dem Kunden, ihre Musik auf beeindruckende Weise visuell darzustellen. Der Mehrwert besteht in der intuitiven Benutzeroberfläche, die künstlerische Freiheit fördert und es Anwendern ermöglicht, ihre individuelle Kreativität auszudrücken. Die Software generiert eine immersive Erfahrung, die das Publikum tiefer in die Musik eintauchen lässt.

## Systemanforderungen

* Anmelden und Speichern von Benutzerdaten (Username, Password, Profile) (25 Stunden)
* Anpassbare Benutzeroberfläche für individuelle Bedürfnisse (20 Stunden)
* Vielfältige, auf unterschiedliche Musikgenres abgestimmte, anpassbare, visuelle Effekte zur Darstellung von Musik (50 Stunden)
* Export-/Importmöglichkeit von erstellten Visualisierungen in benennbaren Profildateien (15 Stunden)
* Integration von bestehenden Musikdateien und Streaming-Diensten (30 Stunden)

## Qualitätsanforderungen

* Ansprechendes Design / Hochauflösende Texturen (15 Stunden)
* Einfache Bedienbarkeit der GUI (20 Stunden)
* Schnelle Reaktionszeit für eine flüssige und immersive Nutzererfahrung (25 Stunden)

## Schnittstellen

Die Software soll Schnittstellen zu gängigen Musikplattformen wie Spotify und SoundCloud ermöglichen, um eine reibungslose Integration von Musikdateien zu gewährleisten.

## Risikoabschätzung

* Technologische Herausforderungen bei der Integration mit verschiedenen Streaming-Diensten (Auftraggeber trägt das Risiko)
* Verzögerungen aufgrund unerwarteter Anpassungen im Designprozess (Dienstleister trägt das Risiko)

# Abnahmekriterien

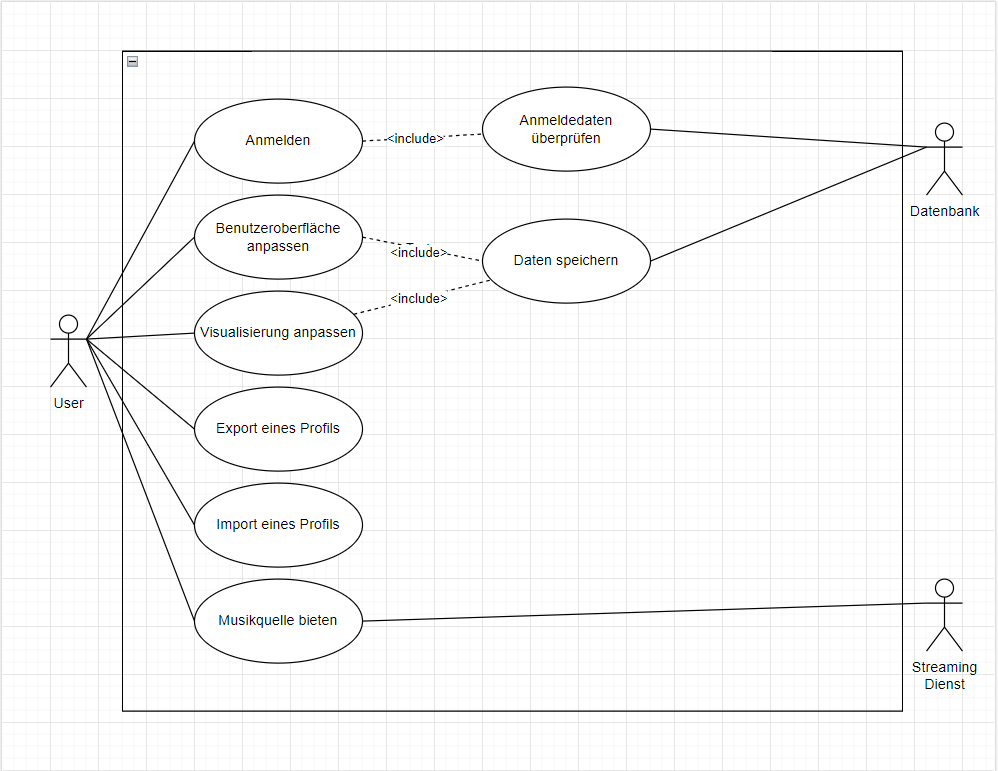
## MUSS-Kriterien

* Anmelden und Speichern von Benutzerdaten (Username, Password)
* Anpassbare Benutzeroberfläche für individuelle Bedürfnisse
* Vielfältige visuelle Effekte zur Darstellung von Musik
* Exportmöglichkeit von erstellten Visualisierungen in benennbaren Profildateien
* Importmöglichkeit von Profildateien
* Aufnahme aus Direct-Input

## KANN-Kriterien

* Speichern der erstellten Profile des Benutzers
* Laden von verschiedenen Skins
* Punkte sammeln für das rhythmische Klicken auf Effekte
* Profil bearbeiten
* Profil bewerten/favorisieren
* Integration von bestehenden Musikdateien und Streaming-Diensten

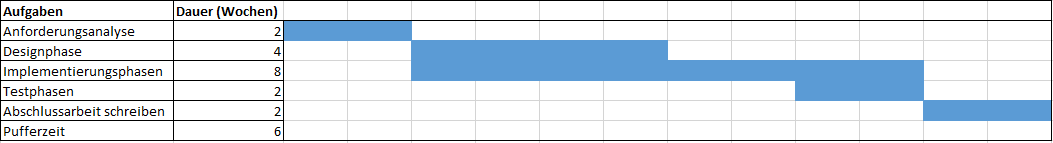
# Use-Case-Diagramm



|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | **Beschreibung** |
| UC01 | Anmelden | Benutzer meldet sich an, um auf seine gespeicherten Daten zugreifen zu können. |
| UC02 | Benutzeroberfläche anpassen | Benutzer stellt ein wie die GUI dargestellt werden soll. |
| UC03 | Visualisierung anpassen | Benutzer stellt ein, wie Audio dargestellt werden soll. |

|  |  |  |
| --- | --- | --- |
| UC04 | Export eines Profils | Benutzer speichert Einstellungen zur Darstellung in einer lokal gespeicherten Profildatei. |
| UC05 | Import eines Profils | Benutzer lädt Einstellungen zur Darstellung aus einer lokal gespeicherten Profildatei. |
| UC06 | Musikquelle bieten | Benutzer wählt eine lokal gespeicherte oder eine aus den angebotenen Streaming-Diensten bereitgestellte Audiodatei. |
| UC07 | Daten speichern | Sämtliche Änderungen des Benutzers werden automatisch, für diesen Benutzer, von der Datenbank erfasst. |

# Projektplan

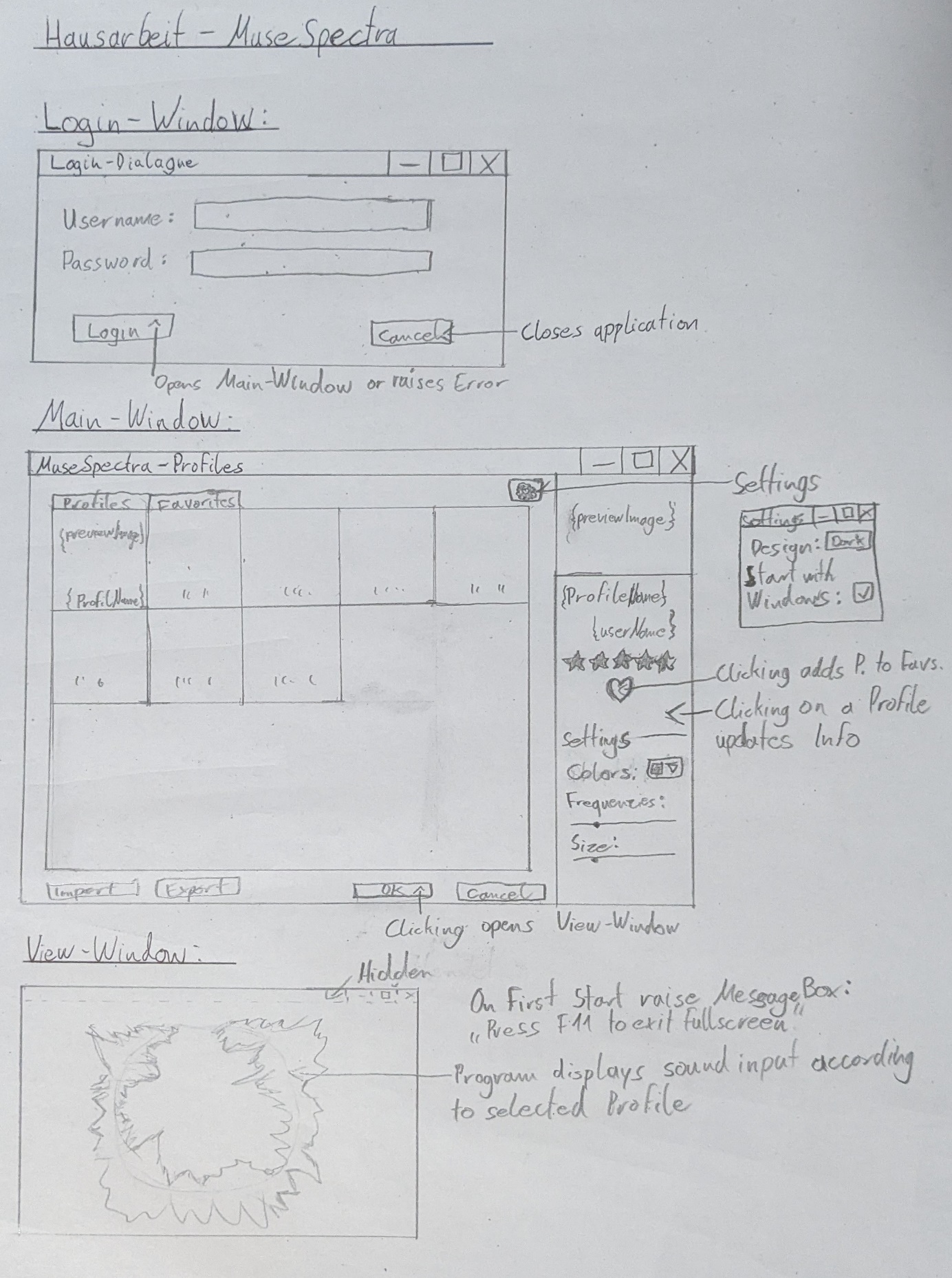


# Produktumgebung

Die Software wird in einer modernen, plattformunabhängigen Umgebung entwickelt, um eine breite Zugänglichkeit zu gewährleisten. Die folgenden Technologien werden in der Produktumgebung verwendet:

* Programmiersprache: C#
* Benutzeroberfläche: WPF-Anwendung
* Datenbank: Microsoft SQL Management Studio, relationale Datenbank
* Frameworks: .NET-Framework für die Anwendungslogik und Entity Framework für die Datenbankanbindung
* Test-Frameworks: Microsoft Test Framework mit Unit Tests

# Skizze des GUI

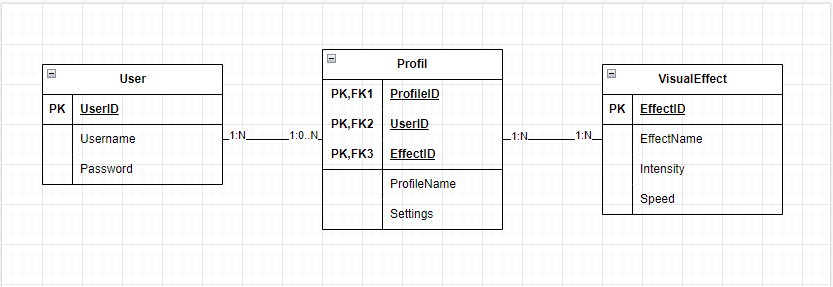


Cancel

Apply

Register

# DB-Entwurf



# Link zu gehosteten Git-Repository

[a-m-prd/MuseSpectra: Meine Hausarbeit für HBFSWI. (github.com)](https://github.com/a-m-prd/MuseSpectra)

# Testplan

|  |  |
| --- | --- |
| **ID:** | **T01** |
| Beschreibung: | Anmelden |
| Vorbedingung: | Das Programm ist im Anmeldedialog. |
| Test-Schritte: | 1. Im Feld “Name” wird der Name eines Accounts eingegeben. 2. Im Feld “Passwort” wird ein passendes Passwort eingegeben. 3. Klick auf „Login“/„Register“. |
| Erwartetes Resultat: | Main-Window öffnet sich, Nutzer ist eingeloggt/wurde angelegt, sieht seine Bewertungen/Favoriten und kann auf seine Profile zugreifen. |

|  |  |
| --- | --- |
| **ID:** | **T02** |
| Beschreibung: | Benutzeroberfläche anpassen |
| Vorbedingung: | Das Programm ist im Einstellungendialog. |
| Test-Schritte: | 1. Im Feld “Design” wird ein Design ausgewählt. 2. Klick auf „Apply“. |
| Erwartetes Resultat: | Die Benutzeroberfläche hat das Design entsprechend geändert. |

|  |  |
| --- | --- |
| **ID:** | **T03** |
| Beschreibung: | Visualisierung anpassen |
| Vorbedingung: | Das Programm ist im Main-Window und ein Profil ist ausgewählt. |
| Test-Schritte: | 1. Einstellungen werden vorgenommen. 2. Klick auf „Ok“. |
| Erwartetes Resultat: | View-Window öffnet sich, Programm stellt Audio im entsprechenden Format dar. |

|  |  |
| --- | --- |
| **ID:** | **T04** |
| Beschreibung: | Export eines Profils |
| Vorbedingung: | Das Programm ist im Main-Window und das zu exportierende Profil ist ausgewählt. |
| Test-Schritte: | 1. Klick auf „Export“. 2. Dateisystem öffnet sich. 3. Speicherort und Namen festlegen. |
| Erwartetes Resultat: | Die Profil.txt wurde erfolgreich im angegebenen Speicherort mit dem richtigen Namen angelegt. |

|  |  |
| --- | --- |
| **ID:** | **T05** |
| Beschreibung: | Import eines Profils |
| Vorbedingung: | Das Programm ist im Main-Window. |
| Test-Schritte: | 1. Klick auf „Import“. 2. Dateisystem öffnet sich. 3. Auswählen der zu importierenden Profil.txt |
| Erwartetes Resultat: | Das Profil wurde der Liste erfolgreich hinzugefügt. |

|  |  |
| --- | --- |
| **ID:** | **T06** |
| Beschreibung: | Musik darstellen |
| Vorbedingung: | Das Programm ist im View-Window. |
| Test-Schritte: | 1. Nutzer spielt Audio ab. |
| Erwartetes Resultat: | Programm stellt Audio im entsprechenden Format dar. |

|  |  |
| --- | --- |
| **ID:** | **T07** |
| Beschreibung: | Daten speichern |
| Vorbedingung: | Das Programm wurde von Nutzer genutzt. |
| Test-Schritte: | 1. Anmelden. 2. Überprüfen ob Einstellungen gespeichert wurden. |
| Erwartetes Resultat: | Einstellungen wurden gespeichert und beim Start übernommen. |

# Benutzte Werkzeuge und Quellen

- Schulmaterial

- <https://learn.microsoft.com/en-us/dotnet/api/microsoft.win32.registry?view=net-8.0>

- <https://github.com/naudio/NAudio>

- <https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.notifyicon?view=windowsdesktop-8.0>

- <https://playground.com> (Zur Erstellung des Icons)

|  |
| --- |
| **„Hiermit erkläre ich an Eides statt, dass ich diese Hausarbeit selbständig und ohne fremde Hilfe angefertigt habe und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe.“** |