FIAE20J – LF12a & Deutsch

Fachinformatiker für Anwendungsentwicklung

Dokumentation zum Projekt



**Avalanche**

**Programm zur Ansicht und Verwaltung von Snowboardern**

Abgabetermin: 13.11.2022

# Inhaltsverzeichnis

[Inhaltsverzeichnis I](#_Toc113535560)

[Abkürzungsverzeichnis II](#_Toc113535561)

[1 Einleitung 1](#_Toc113535562)

[1.1 Projektbeschreibung 1](#_Toc113535563)

[1.2 Projektziel 1](#_Toc113535564)

[2 Projektplanung 1](#_Toc113535565)

[2.1 Ressourcenplanung 1](#_Toc113535566)

[2.2 Entwicklungsprozess 1](#_Toc113535567)

[3 Entwurfsphase 1](#_Toc113535568)

[3.1 Git 1](#_Toc113535569)

[3.2 Entwurfsmuster 1](#_Toc113535570)

[3.3 Entwurf der Benutzeroberfläche 1](#_Toc113535571)

[4 Implementierungsphase 2](#_Toc113535572)

[4.1 Git 2](#_Toc113535573)

[4.2 Implementierung von MVC 2](#_Toc113535574)

[4.3 Implementierung der Benutzeroberfläche 2](#_Toc113535575)

[4.4 Implementierung der Datenbank 2](#_Toc113535576)

[4.5 Abweichungen gegenüber dem erwarteten Ergebnis 2](#_Toc113535577)

[5 Dokumentation 2](#_Toc113535578)

[6 Fazit 2](#_Toc113535579)

[6.1 Soll-/Ist-Vergleich 2](#_Toc113535580)

[A Anhang i](#_Toc113535581)

[A.1 Abbildung ER-Diagramm Datenbank i](#_Toc113535582)

# Abkürzungsverzeichnis

ISC *International Snowboarding Community*

MVC *Model-View-Controller*

# Einleitung

Die folgende Projektdokumentation beschreibt den Ablauf unseres Snowboarder Projekts, welches wir für die Notenvergabe in Deutsch und Lernfeld 12 anfertigen.

## Projektbeschreibung

Um die Wettkämpfe der Snowboarder zu Verwalten hat die International Snowboarding Community ein Programm in Auftrag gegeben.

## Projektziel

Ziel des Projekts ist es ein Programm zu entwickeln, welches die Wettkämpfer, deren Sponsoren sowie andere relevante Informationen speichern und ausgeben kann. Geplant ist dafür die Verwendung einer Datenbank zum speichern der Informationen, sowie eine Möglichkeit diese Benutzerfreundlich zu bedienen.

# Projektplanung

## Ressourcenplanung

Für die Umsetzung des Projektes stand uns die Zeit bis zum 13 November zur Verfügung. Das Verfügbare Personal besteht dabei aus Simon, Vladimir und Can.  
Aufgrund der nahenden Abschlussprüfungen wurde im Team vereinbart den 16 September als neue Frist zu nutzen um die Abgabe frühzeitig von unserer To-Do Liste zu streichen.  
Als Entwicklungstools nutzen wir Visual Studio und SQLite.

## Entwicklungsprozess

Um die Entwicklung und das Projekt möglichst zügig voranzutreiben, wurde eine Teilung der Arbeit vereinbart.  
Die Teilung folgte in die folgenden Bereiche: Dokumentation, Datenbank und Benutzeroberfläche. Diese Verantwortlichkeiten haben wir bei uns im Team verteilt. Keiner der Teile kann dabei Selbstständig funktionieren, sodass ein ständiger Austausch im Team und der aktuellen Arbeit gewährleistet ist.

# Entwurfsphase

## Git

Als Quellcodeverwaltungssystem wird Git verwendet. Das Projekt soll in mehreren Git-Branches entwickelt werden um einen parallelen Fortschritt zu ermöglichen. Das Ganze wird dabei in der Cloud gespeichert, mit GitHub als freizugänglichen Dienstleister.

## Entwurfsmuster

Zu Beginn der Aufgabe haben wir uns zudem auf ein Entwurfsmuster festgelegt, welches wir für die Benutzeroberfläche Nutzen wollen. Unsere Entscheidung fiel dabei auf Model-View-Controller.

## Entwurf der Benutzeroberfläche

Die Benutzeroberfläche wird als ASP.net Anwendung umgesetzt. Dies erlaubt den Zugriff auf die Datenbank von verschiedensten Geräten. Da die ISC eine Internationale Vereinigung ist, erlaubt unser Ansatz für eine Einfache Erreichbarkeit unserer Services Weltweit.  
Optisch wird die Anwendung einer einfachen Website gleichen, mit Möglichkeiten zur Abfrage und Eingabe von Daten aus und in die Datenbank.

# Implementierungsphase

## Git

Zu Beginn der Implementierungsphase wurde für ein Projekt in Visual Studio erstellt, und dann in die Cloud von GitHub gepusht. Anschließend wurden die nötigen Rechte an alle Teammitglieder vergeben. Dem Basisprojekt folgten 3 Branches um die Fortschritte getrennt zu erarbeiten.

## Implementierung von MVC

~~Für alle im Folgenden angelegten Klassen stehen Basis-Klassen aus dem hauseigenen Core-Framework zur Verfügung, durch die grundlegende Funktionalitäten bereitgestellt werden.~~

## Implementierung der Benutzeroberfläche

~~Ein Screenshot der implementierten Benutzeroberfläche befindet sich im Anhang A.17: Screenshot der Benutzeroberfläche auf Seite xii.~~

## Implementierung der Datenbank

~~Für alle im Folgenden angelegten Klassen stehen Basis-Klassen aus dem hauseigenen Core-Framework zur Verfügung, durch die grundlegende Funktionalitäten bereitgestellt werden.~~

## Abweichungen gegenüber dem erwarteten Ergebnis

~~Aufgrund der parallelen Entwicklung an der Firmware und dem intensiven Austausch mit dem~~ [~~DDE~~](#_Abkürzungsverzeichnis) ~~ist es zu Verzögerungen in der Implementierungsphase gekommen. Die Implementierung der Tests wurde zum Teil auf einen späteren Zeitpunkt verschoben.~~

# Dokumentation

Um die Zeitlichen Anforderungen einzuhalten, wurde die Dokumentation parallel zur Entwicklung geschrieben.  
Zu Beginn haben wir ein Grundgerüst der Dokumentation angefertigt, anschließend das Ganze in Git mit eingecheckt. Jedoch merkten wir schnell, dass es so zu Problemen bei unterschiedlichen Dokumentationsständen kommen könnte. Als Antwort auf das Problem haben wir uns entschlossen die Doku künftig gemeinsam in Teams zu pflegen, wo eine parallele bearbeiten von Dokumenten Benutzerfreundlich ermöglicht wird.

# Fazit

Zum Abschluss des Projekts zieht der Autor ein Fazit über das Gelernte und gibt einen Ausblick auf die Zukunft des Projekts.

## Soll-/Ist-Vergleich

~~Das~~ [~~SMCU5~~](#_Abkürzungsverzeichnis) ~~soll in unterschiedlichen Projekten mit unterschiedlichen Maschinen verwendet werden. Zum Zeitpunkt dieser Dokumentation werden Firmware und Maschinenkonfigurationen weiterhin erarbeitet. Die parallele Entwicklung des~~ [~~SMCU5~~](#_Abkürzungsverzeichnis) ~~und der stetige Austausch mit dem~~ [~~DDE~~](#_Abkürzungsverzeichnis) ~~haben die Implementierungsphase verlängert. Es wurde die Entscheidung getroffen, die Unit-Tests für den~~ [~~UI~~](#_Abkürzungsverzeichnis)~~-Service zu einem späteren Zeitpunkt zu implementieren.~~

1. Anhang
2. Abbildung ER-Diagramm Datenbank

