Projekt: FitMeUp!

Software Architektur Spezifikation (Software Architecture Document)

[Dokumentstruktur basiert auf RUP „Software Architecture Document“]

1 Dokumentinformationen

1.1 Änderungsgeschichte

Datum	*Version*	*Änderung*	*Autor*
10.05.2022	1.0	Erstellung der Software Architektur Spezifikation	Krüger, Schill, Strauß
17.06.2022	2.0	Anpassung der Software Architektur Spezifikation	Schill

1.2 Inhalt

2 Einführung (Introduction)

2.1 Definitionen und Abkürzungen (Definitions, Acronyms, Abbreviations)

- GUI = Graphical User Interface
- Nutzer = Kunde, der auf die FitMeUp-Software zugreift
- UML = Unified Modeling Language
- CRUD = Create, Read, Update, Delete

2.2 Referenzen (References)

Literatur: Craig Larman, UML 2 und Patterns angewendet
Vorlesungsunterlagen: Datenbanken (Hr. Payer), Software Engineering (Prof. Dr. A. Doering)

2.3 Übersicht (Overview)

In diesem Dokument werden die Art und der Aufbau der Architektur beschrieben. Hierbei werden die verschiedenen Ebenen und Packages und deren Schnittstellen betrachtet und näher erläutert. Die Applikation wird mittels eines UML-Klassendiagramm technisch beschrieben.

3 Architektonische Darstellung (Architectural Representation)

Die Softwarearchitektur wird nach der dreischichten Architektur entworfen. Die Architektur wird hierfür softwareseitig in drei Schichten unterteilt, einmal in die Präsentationsschicht, die Logikschicht und die Datenhaltungsschicht. Die Präsentationsschicht visualisiert die Daten und Benutzereingaben und ist somit die Benutzerschnittstelle für den Kunden. In der Logikschicht werden die Daten verarbeitet. Die Datenhaltungsschicht enthält die MySQL-Datenbank, die für alle CRUD-Operationen verantwortlich ist. Des Weiteren wird ein MV-Pattern umgesetzt. Dieses dient als Model-Präsentation für die GUI. Die Eingabe des Nutzers wird verarbeitet und dem Model übermittelt. Hier wird die View mit den Anwenderdaten dem Nutzer auf einem Bildschirm dargestellt.

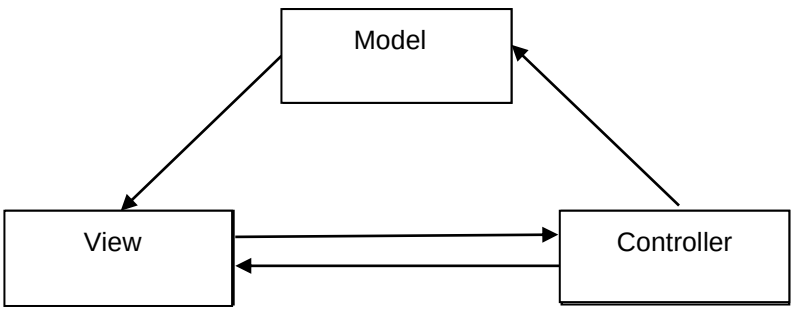
4 Architektonische Ziele & Einschränkungen (Architectural Goals and Constraints)

In unserem Programm verwenden wir die MVC-Architektur, welche sich aus den Komponenten Model, View und Controller zusammensetzt. Dabei bestehen Abhängigkeiten zwischen den genannten Komponenten. Dadurch können Änderungen in den einzelnen Komponenten vorgenommen werden, ohne das gesamte System ändern zu müssen.

5 logische Architektur (Logical View)

Im Folgenden wird die angewandte MV-Architektur näher erläutert.

5.1 Übersicht (Overview)



5.2 Design Pakete (Architecturally Significant Design Packages)

5.2.1 Package Präsentationsschicht

5.2.1.1 Beschreibung des Packages

Die GUI visualisiert dem Nutzer Informationen und stellt eine Interaktion zur Verfügung. Der Nutzer kann die GUI nutzen, um dem System Informationen zur Verarbeitung weiterzuleiten, aber auch bereits bestehende Daten abzurufen, zu ändern oder zu entfernen. Der Nutzer kann diese Funktionen über die Anwendungslogik durchführen. Die Bedienung der FitMeUp-Applikation durch den Nutzer erfolgt über die GUI. Dadurch ist das Erkennen von Nutzereingaben und Interaktionen, sowie die Verarbeitung und Weiterleitung von Daten an die Logikschicht von Grund auf essenziell.

5.2.1.2 Schnittstellen

Die Schnittstellen der Präsentationsschicht bestehen aus einer Schnittstelle zur Anwendungslogik des Systems und den Nutzer. Die Anwendungsschnittstelle prüft und verarbeitet alle Daten, die zur Prüfung und Verarbeitung notwendig sind, sowie bereits bestehende Daten und solche, die durch den Nutzer entstehen. Der Nutzer kann über die GUI mit dem gesamten System interagieren.

5.2.1.3 Operationen

Die Applikation lässt sich in drei Haupt-Views gliedern. Beim Aufruf der Applikation wird das Login-View angezeigt. Dieses beinhaltet die Operation Einloggen. Sollte ein Nutzer noch nicht registriert sein, wird bei der Operation Registrierung-View geöffnet und die Eingabe der Stammdaten des Nutzers verlangt. Sobald sich ein Nutzer einloggt, erscheint die Nutzer-View. Sie beinhaltet Operationen wie die Eingabe von Wasser-, Kalorien- und Fitnesseinheit, Einstellungen und Challenges.

5.2.2 Package Logikschicht

5.2.2.1 Beschreibung des Packages

Die Logikschicht beinhaltet die gesamte Logik des Systems. In diesem Package werden Funktionen durch Dienste für alle weiteren Packages zur Verfügung gestellt. In der Logikschicht werden die wesentlichen Anforderungen aus der Anforderungsspezifikation realisiert.

5.2.2.2 Schnittstellen

Die Logikschicht ist mit der Präsentationsschicht, als auch mit der Datenhaltungsschicht verknüpft. Verbunden sind diese durch die GUI, die Dienste und Funktionalitäten zur Verfügung stellt.

5.2.2.3 Operationen

Die GUI wird durch den Nutzer gesteuert. Die Klassen Eingabe, LogicLoginSystem, Übungsplan und Challenges realisieren die Anwendung. Diese Klassen verwalten die Eingaben des Nutzers und haben eine direkte Verbindung zur Datenbank.

5.2.3 Package Datenhaltungsschicht

5.2.3.1 Beschreibung des Packages

Die Datenhaltungsschicht dient der Speicherung aller anfallenden und bestehenden Daten. Die Schicht besteht aus einer Menge von Tabellen einer relationalen Datenbank.

5.2.3.2 Schnittstellen

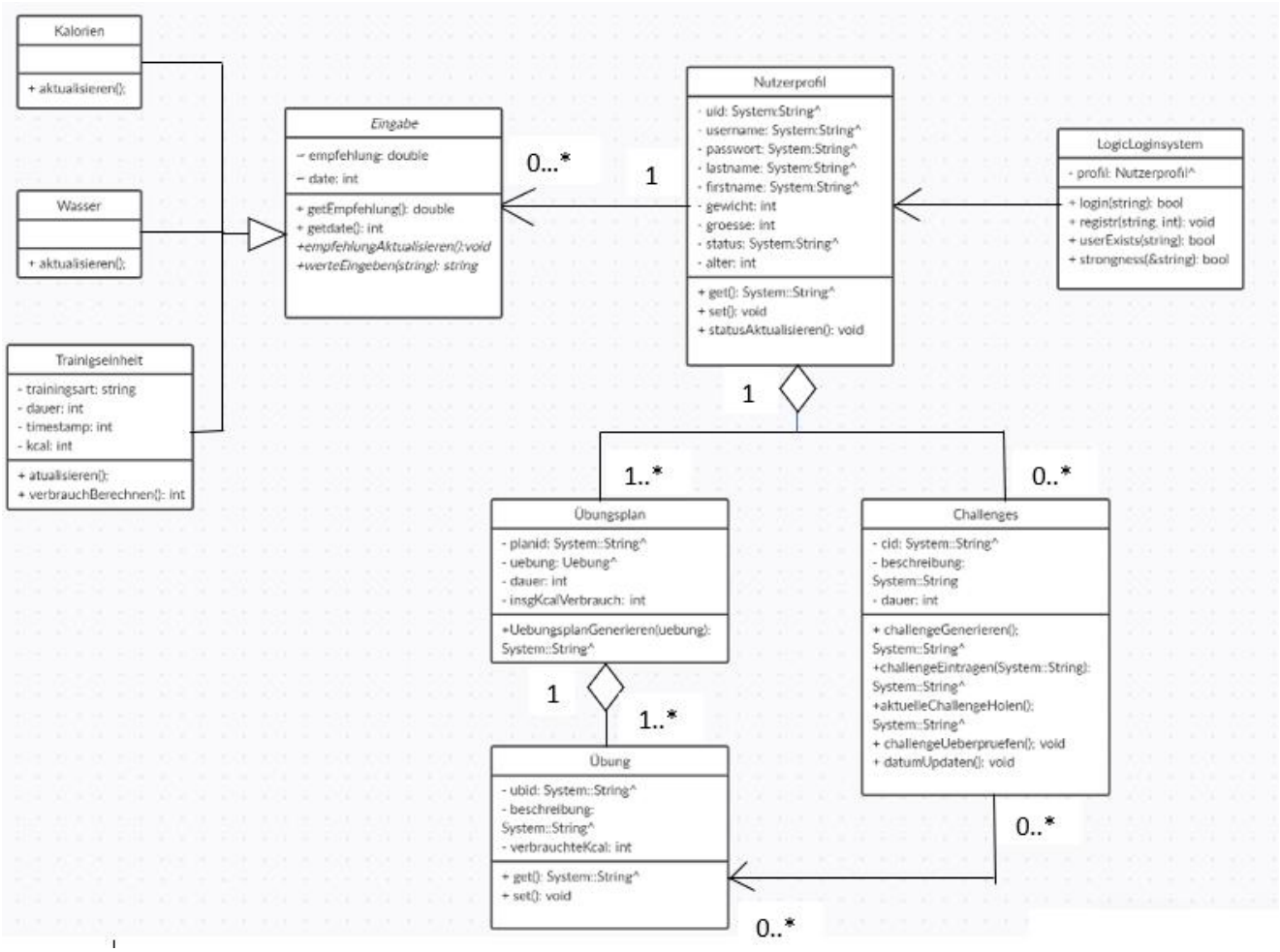
Die Datenhaltungsschicht hat eine Schnittstelle zur Präsentationsschicht. Die Schnittstelle wird über die GUI-Klassen der Anwendungsschicht definiert. In diesen werden CRUD-Funktionen definiert, die den Datenbankzugriff steuern.

5.2.3.3 Operationen

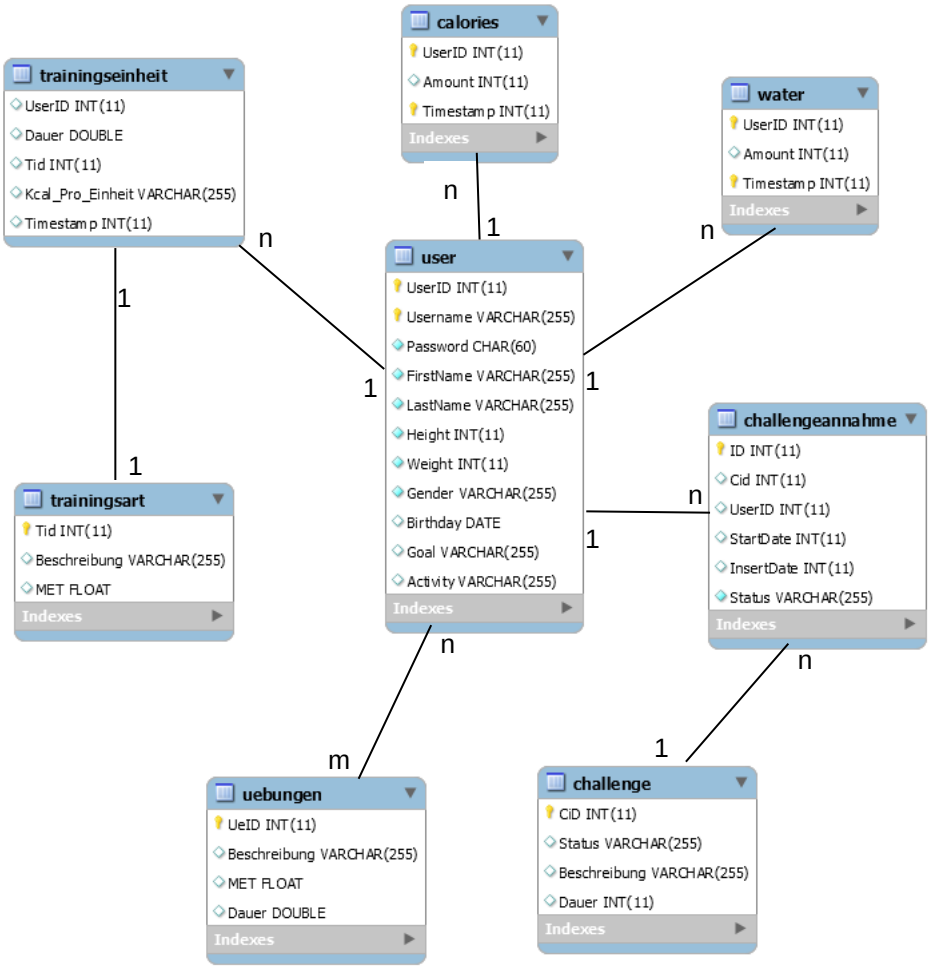
Die Datenbankverbindung wird in der Klasse DB_Response realisiert. Die GUI-Klassen (siehe oben) greifen auf die Datenbank zu.

6 Physikalische Sicht (Physical View)

6.1 UML-Klassendiagramm



Klasse	Nutzen
LogicLoginsystem	Dient zur Anmeldung im Programm
Nutzerprofil	Enthält alle Nutzerdaten und berechnet Status
Eingabe	Dient zum Berechnen der Empfehlung von Kalorien und Wassereinnahmen
Kalorien	Dient zum Eintragen der Kalorien
Wasser	Dient zum Eintragen des zu sich genommenen Wassers
Trainingseinheit	Dient zum Eintragen und Berechnen des Kalorienverbrauchs pro Trainingseinheit
Übung	Enthält verschiedene Übungen
Übungsplan	Benutzerdefinierter Übungsplan der Übungen Enthält
Challenges	Verwaltet Funktionalitäten der Challenges



7 Datenspeicherung (Data View)

Alle Daten, die erstellt, gelesen, bearbeitet oder gelöscht werden, werden in der Datenbank gespeichert. Dabei wird die Datenhaltung gesichert. Der Zugriff auf die MySQL-Datenbank auf unserer Windows virtuellen Maschine erfolgt über Microsoft Visual Studio mit einer externen Bibliothek. Dabei wird eine VPN-Verbindung mit dem OTH-Netz benötigt. MySQL ist ein Server, das eine Datenbankmodul implementiert. Zur Modellierung und Visualisierung der rationalen Datenbank wurde ein ER-Modell verwendet. Dieses zeigt die verschiedenen Beziehungen zwischen den einzelnen Komponenten an.

8 Größen und Leistung (Size and Performance)

Es können sich mehrere Nutzer gleichzeitig anmelden. Auch alle weiteren Funktionen der Applikation können von mehreren Nutzern ausgeführt werden. Die MySQL-Datenbank kann unendlich viele Einträge speichern, weswegen es kein maximal-Limit an Tupel gibt.