

GymTracker

Dein Sportshelfer



Flávio Correia Marta und Tino Joseph

Inhaltsverzeichnis

Einleitung.....	5
Ursprung der Idee	6
Werkzeuge.....	7
Datenquelle	8
Programmkompatibilität.....	8
Entwicklungsprozess.....	9
Aufgabenverteilung	10
Unit-Tests.....	10
Klassendiagramm	11
Quellenangaben.....	12

Einleitung

Am Anfang des zweiten Lehrjahres unserer Ausbildung zum Fachinformatiker in der Fachrichtung Anwendungsentwicklung haben wir von Herrn Patett in dem Lernfeld 8 – Daten systemübergreifend bereitstellen den Auftrag bekommen, bis Mitte des Jahres einen Partnerprojekt zu entwickeln und durchzuführen.

Zusammen haben wir uns gefragt, über welches Thema wir ein Programm entwickeln könnten. Zuerst dachten wir an einem kleinen Tool, das die Wetter-, Luft- und UV-Daten vergleicht. Obwohl es interessant klingt, fanden wir das Thema nicht anziehend genug, was dazu führte, dass wir das Projekt hin- und hergeschoben haben. Wir starteten denn wieder von Anfang an, wir waren wieder in der Analyse und nicht mehr in der Planung des Projektes. Da wir das Thema freiwillig auswählen durften, haben wir erstmal geschaut, wo unser Interesse liegt. Am Ende entschieden wir uns für die Entwicklung eines Tools, das Sport und Rostock verbindet. So entwickelten wir GymTracker.

In der Planung und Entwicklung des Projekts sollten wir an folgenden Punkten denken:

- Daten aus zwei verschiedenen Datenquellen
- Datensätze mit mindestens zehn Attributen
- Abfrage der Daten
- Auswertung der Daten
- Zusammenführung der Daten
- Visuelle Darstellung
- Erstellung eines Klassendiagramm
- Auswahl eines geeigneten Datenmodels
- Verfügbarkeit einer grafischen Oberfläche
- Entwicklung von Unittests

Das Abgabedatum für alle Projekte ist der 4. April 2022. Abschließend dürfen wir dieses Projekt vorstellen und verteidigen. Die Verteidigung startet am 30. Mai 2022.

Ursprung der Idee

Wie schon erwähnt haben wir uns ursprünglich für ein kleines Tool entschieden, die Wetterdaten mit ähnlichen Daten vergleicht. Auf diese Idee sind wir gekommen, als wir im September 2021 in der Schule waren und uns fragten, ob man nicht einfach etwas schlichtes programmieren kann, um eine einfache und aktuelle Darstellung der Temperatur, Luftfeuchtigkeit und UV-Strahlung zu sehen. Im Laufe des ersten halben Jahres haben wir das Thema analysiert und geplant, sind aber leider zu dem Entschluss gekommen, dass das gewählte Ziel nicht mehr unseren Interessen entsprach.

Damit fingen wir wieder von vorne an und landeten wieder in der Analyse. Wir haben uns die Zeit genommen, stellten uns die Frage „Was gefällt uns?“ und haben dementsprechend mehrere Stichpunkte aufgeschrieben. Einen gemeinsamen Nenner hatten wir erst nach dem Ideenaustausch und so fanden wir erstmal das Thema: **Sport**.

Hat man allerdings ein Thema, ist das Ganze nicht erledigt. Wir müssten noch das Thema spezifizieren, das Thema Sport ist einfach zu allgemein. Deshalb suchten wir erstmal Daten, die mit Sport zu tun hatten. Zuerst sammelten wir eine gewisse Anzahl an Datenquellen und danach schauten wir uns sie genau an. Nebenbei waren wir beide tatsächlich auch auf der Suche nach einem neuen Fitnessstudio. Und so kamen wir auf die Idee, GymTracker zu entwickeln. Mit GymTracker wollten wir eine einfache Lösung anbieten, die die nächstliegenden Fitnessstudios anzeigt. Wir wollen auf die Überforderung der vielfältigen Auswahl verzichten und auf das Wesentliche konzentrieren, ohne zu vergessen, dass eine stetige Weiterentwicklung immer möglich ist.

Werkzeuge

Für die Entwicklung von GymTracker wurden verschiedenen Werkzeuge benutzt.

Auflistung der Werkzeuge:

Name	Funktion
GitHub	Versionsverwaltung
JetBrains PyCharm	Programmierung
Microsoft Word	Erstellung der Dokumentation
GIMP	Bearbeitung des Logos
Microsoft Teams	Kommunikation
WhatsApp	Kommunikation
Python	Programmiersprache

GitHub:

Für die Versionsverwaltung unseren Codes dürften wir zwischen GitLab und GitHub entscheiden. Die Entscheidung fiel bei GitHub. In GitHub haben wir unseren Quellencode, Dokumentation, Logo sowie, zur Absicherung, die Dateien mit den Datensätzen hochgeladen. Das Projekt ist derzeit noch privat, wird aber in nähere Zukunft öffentlich sein.

JetBrains PyCharm:

Als IDE haben wir uns für PyCharm entschieden. PyCharm half uns unser Tool zu entwickeln sowie es zu testen. Die Verbindung mit GitHub klappte auch nahezu reibungslos. Sowohl die Community als auch die Professional Edition sind im Einsatz gewesen.

Microsoft Word:

Das Office-Paket des Hauses Microsoft ist die meistverbreitete Office Lösung und auch von uns wurde Word benutzt, um diese Dokumentation zu erstellen.

GymTracker
von Flávio Correia Marta und Tino Joseph

GIMP:

Der Einsatz von GIMP als Ersatz für Adobe Produkte ist bekannt und wurde auch von uns benutzt. Mithilfe von GIMP haben wir unser Logo entwickelt.

Microsoft Teams:

Kommunikationstools gibt es zahlreiche. Für längere Gespräche und Pair-Coding wurde Microsoft Teams benutzt.

WhatsApp

Das Hauptkommunikationstool der Firma Meta wurde für kleine Rückfragen und Absprachen benutzt.

Datenquelle

Transparenz und freier Zugang waren für uns ein wichtiger Punkt. Als wir uns für das Projekt entschieden haben, haben wir nach offenen Datenquellen gesucht. Mit offenen Daten wird Innovation vorangetrieben. Deshalb haben wir mit OpenData.HRO die richtige Webseite gefunden.

OpenData.HRO „dient als zentrale Plattform zur Veröffentlichung offener Daten der Hanse- und Universitätsstadt Rostock“^[1].

Programmkompatibilität

Die Kompatibilität mit Windows 10 wurde als Mindestanforderung festgesetzt. Da Python plattformübergreifend und sogar Teil von einigen Linux Distributionen ist, ist davon auszugehen, dass der Einsatz von GymTracker mit wenigen Problemen zu rechnen. Obwohl der Computer, an dem wir die Präsentation durchführen werden, nicht vorher getestet wird, wurde GymTracker mit fünf verschiedenen Computern getestet und konnte problemlos laufen. Die Ausstattung der Computer war unterschiedlich, getestet wurden Laptops sowie Desktop-PCs, die als Betriebssystem Windows 10, Windows 11 oder KDE Neon (Linux) im Einsatz hatten. Da keiner von uns tagtäglich mit Python arbeitet, hatten wir mit der Kompilierung des Endprogramms Schwierigkeiten. Für Windows wurde ein Executable erstellt, zu finden unter dem Ordner „Main“.

Entwicklungsprozess

Die Entwicklung von GymTracker wurde als Projekt eingestuft. Mit dem Gedanken starteten wir mit der groben Vorbereitung, wie erwähnt sowohl in der Einleitung und Ursprung der Idee. In dieser Phase war die größte Herausforderung die zeitliche Planung.

In Planung stolperten wir über einige Probleme, wie zum Beispiel Zeit finden. Wir schrieben uns dann einige Projektdaten auf, besprachen organisatorische Fragen, definierten die Beschreibung. Mit der Planung entschieden wir uns für dieses Projekt. Wir teilten uns die Aufgaben grob auf sowie die Verantwortungsbereiche.

Während der Durchführung war uns unsere Deadline immer wichtig, die Risiken des Projekts wurden nie vergessen und die Ressource (Zeit als Wichtigste) effizient genutzt. Der Dialog zwischen uns war stetig, offen und gut. Die endgültige Aufgabenverteilung wurde hier durchgezogen und mit der Programmierung angefangen.

Die Überwachung des Projekts erfolgte auch problemlos, obwohl uns bewusst geworden ist, dass die geplante Zeit etwas mehr der tatsächlichen Zeit abgewichen hat. Es zeigte sich, dass diese Aufgabe eine gute Vorbereitung fürs tägliche Geschäft ist.

Der Abschluss wird ab dem 4. März 2022 durchgeführt.

Aufgabenverteilung

Die Verteilung der Aufgaben erfolgte reibungslos und mit offener Absprache, persönlich aber auch mit Einsatz auf andere Kommunikationswege, wie oben schon erwähnt. Die Verteilung der Aufgaben wird in der untenstehenden Tabelle dargestellt:

Aufgabe	Flávio	Tino
Dokumentation	X	
Klassendiagramm	X	
Programmierung – Back-End	X	X
Programmierung – Front-End		X
Programmierung – Unit Tests	X	X

Bemerkung: die Tabelle zeigt nicht die genaue Prozentzahl der Aufgabenverteilung und dient nur als Kurzübersicht.

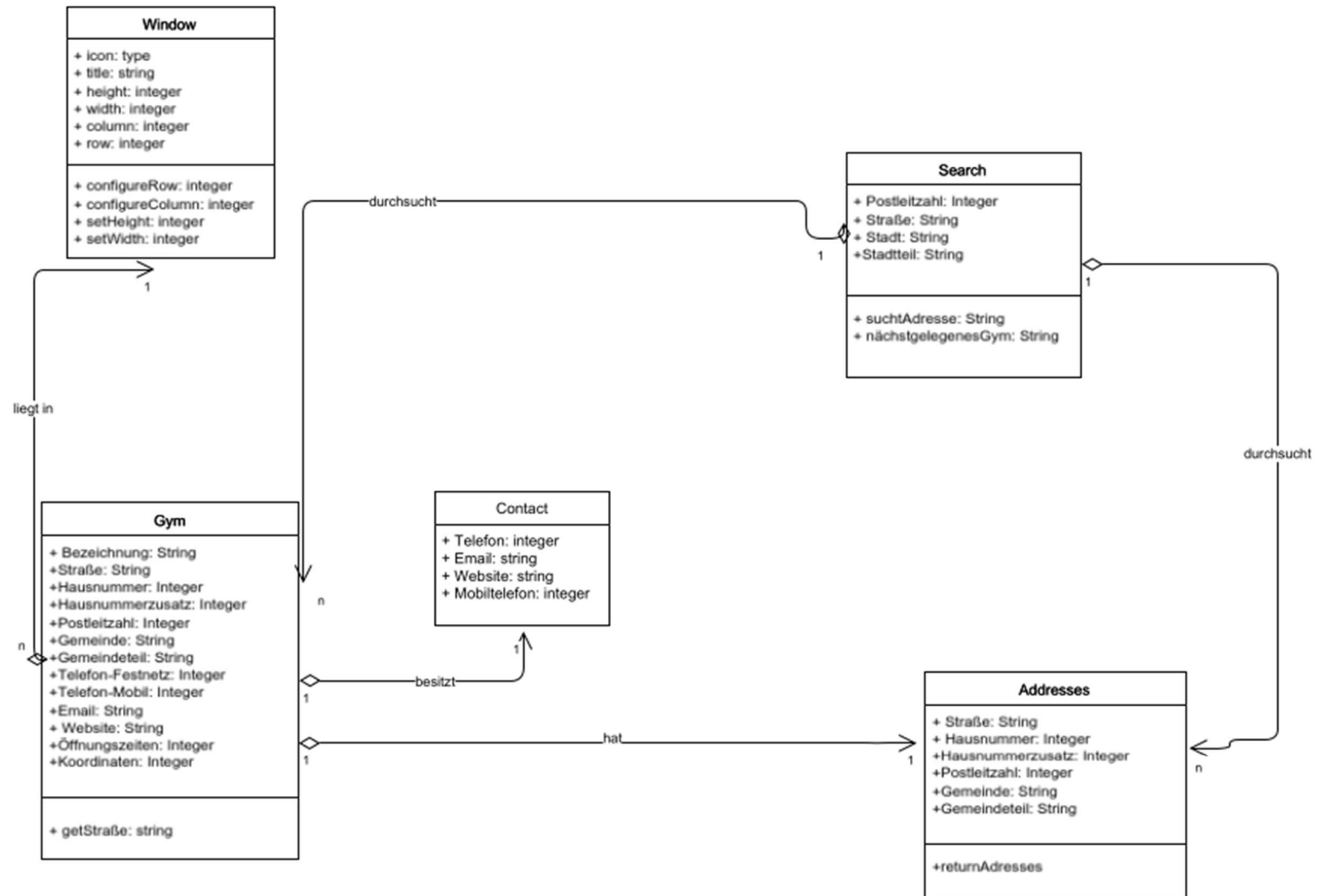
Unit-Tests

Tests sind eine Selbstverständlichkeit in der Programmierwelt. Sie gehören zum A&O jedes Projektes in der Anwendungsentwicklung. Mit den Tests werden sowohl Kleinigkeiten als auch größere Probleme, die in der Praxis auftauchen können, erkannt und dienen als Prophylaxe späteren Schmerzens.

Die Tests würden für alle Klassen, die nicht grafische Elemente enthalten, geschrieben. Diese gehören der Kategorie der White-Box-Tests an. Wir haben uns für die White-Box-Tests entschieden, da man damit sehr gut in Erscheinung getretene Fehler lokalisieren kann, also die fehlerverursachende Komponente zu finden und als Regressionstest ein Wiederauftreten des Fehlers zu vermeiden. Des Weiteren wird bei einem White-Box-Test unnötiger Code entfernt, sie sind außerdem Zeilenüberdeckend, Anweisungsüberdeckend, Bedingungsüberdeckend und Pfadüberdeckend.

White-Box-Tests sind Tests, die geschrieben werden, wenn der Quellcode bewusst ist und dieser auch geprüft wird.

Klassendiagramm



Quellenangaben

[1] – *OpenData.HRO*. (Besucht am 01.04.2022). <https://www.opendata-hro.de/>