

Verkaufsprojekt

Software (Engineering)

Oliver Schlüter

20. Dezember 2021

Inhaltsverzeichnis

1	Konzept	3
1.1	Idee	3
1.2	Spezifikation	3
2	Planung	5
2.1	Entity-Relation-Diagramm	5
2.2	UML-Klassen-Diagramm	6
2.3	Programmablaufplan	7
2.4	Datenbank Skript	8

Einleitung

Aufgabe

Die Aufgabe war es ein Anwendung zu realisieren, welche eine Simulation von einen Ver- oder Kaufunternehmen darstellt. Es sollte dabei ein besonderer Schwerpunkt auf die Planung und Dokumentation gelegt werden. Es ist gefordert folgende Diagramme zu erstellen:

- Programmablaufplan
- Entity-Relationship-Modell
- Klassendiagramm (UML)

Auch soll ein Accountmanagement-System vorhanden sein, d.h. Benutzer können sich registrieren, anmelden und abmelden. Eine Art von Mitarbeiteransicht soll ebenfalls vorhanden sein, wo man Produkte und Benutzer verwalten kann.

Die Umsetzung soll mit der Programmiersprache C# mit Windows-Forms erfolgen.

Kapitel 1

Konzept

1.1 Idee

Meine Idee war es eine Anwendung zu erschaffen, wo Software-Produkte angeboten werden, welche von Autoren geschrieben und veröffentlicht werden. Diese Software-Produkte kann alles mögliche sein, von Spiel-Quellcode bis zu komplexen Cloud-Systemen. Voraussetzung ist nur, dass man es runter laden kann. Auch sollen Kunden die Möglichkeit bekommen Produkte und Autoren zu bewerten und ein Kommentar zu hinterlegen. Da dabei auch viel Unsinn bei auftreten könnte, braucht man Mitarbeiter, welche anstößige Kommentare blockieren und ggf. den Benutzer sperren können. Wenn es schon Mitarbeiter gibt, kann man auch eine einführen, dass sie Produkte unter die Lupe nehmen können, d.h. dass es sie den Anforderungen entsprechen und keine Schadsoftware enthalten.

1.2 Spezifikation

Um spezieller zu werden habe ich mir folgende Eigenschaften der Bestandteile erarbeitet, welche auch den Grundstein des UML-Diagramms sein soll.

Jedes Produkt soll eine eindeutige ID haben, welche fünf Zeichen lang ist, außerdem einen Name und eine ausführliche Beschreibung. Fehlen dürfen nicht die Autoren, welche die Software geschrieben haben. Produkte können umsonst angeboten werden, allerdings ist in der Regel ein Festpreis festgelegt. Kunden, welche ein Produkt gekauft haben, können dieses Bewerten (1-5 Sterne, wobei 5 Sterne besonders gut ist) und kommentieren und ihre Bewertung dadurch begründen.

Es gibt drei Arten von Benutzer: Kunden, Autoren und Mitarbeiter. Sie alle besitzen eine eindeutige ID, welche fünf Zeichen lang ist, einen Vor- und Nachname, eine Email-Adresse, ihr Geburtsdatum, das Erstelldatum des Kontos und ein Passwort.

Zusätzlich werden Autoren mit einen Entwicklerstatus versehen, welches angibt, wie Erfahren und ggf. professionell sie sind. Auch können sie aus angeben, mit welchen Programmier- und Skriptsprachen sie ihre Software erzeugen.

Kunden haben eine Liste von bereits gekauften Produkten und Produkten, welche sie beobachten und sich ggf. in Zukunft kaufen wollen. Die Anwendung basiert auf das Prepaid-Prinzip, sodass jeder Kunde ein Guthaben haben, welches man jederzeit aufladen kann.

Jeder Mitarbeiter werden ein oder mehrere Aufgabenbereiche zugeteilt, wofür sie Zuständig sind und eine Liste von Produkten, welche sie zurzeit Prüfen.

Nun noch die Formulare. Beim Start der Anwendung landet man auf der Startseite, wo man die Möglichkeit hat sich anzumelden, sich zu registrieren, sich abzumelden. Falls man angemeldet ist, hat man die Möglichkeit zu einen weiteren Formular zu wechseln, wo man alle Produkte sieht und

mit ihnen weiter interagieren kann. Auch kann man sich seine bereits gekauften Produkte in einen weiteren Formular anzeigen lassen. Betrachten wir jedoch diese Formular genauer.

Login Formular:

Registrations Formular:

Produkt Formular:

Gekaufte Produkte Formular:

Kapitel 2

Planung

Ein essentieller schritt in der Softwareentwicklung ist die Planung. Auch für dieses Projekt habe ich mich ausführlich mit der Planung auseinander gesetzt. Erste habe ich mir ein generelles Konzept erarbeitet, folgend diese Ideen spezifiziert und anschließend in anschaulichen visualisiert.

2.1 Entity-Relation-Diagramm

Zu Beginn erstellte ich das Entity-Relation-Diagramm, weil dort die Beziehungen zwischen Entitätstypen genauer beschrieben werden.

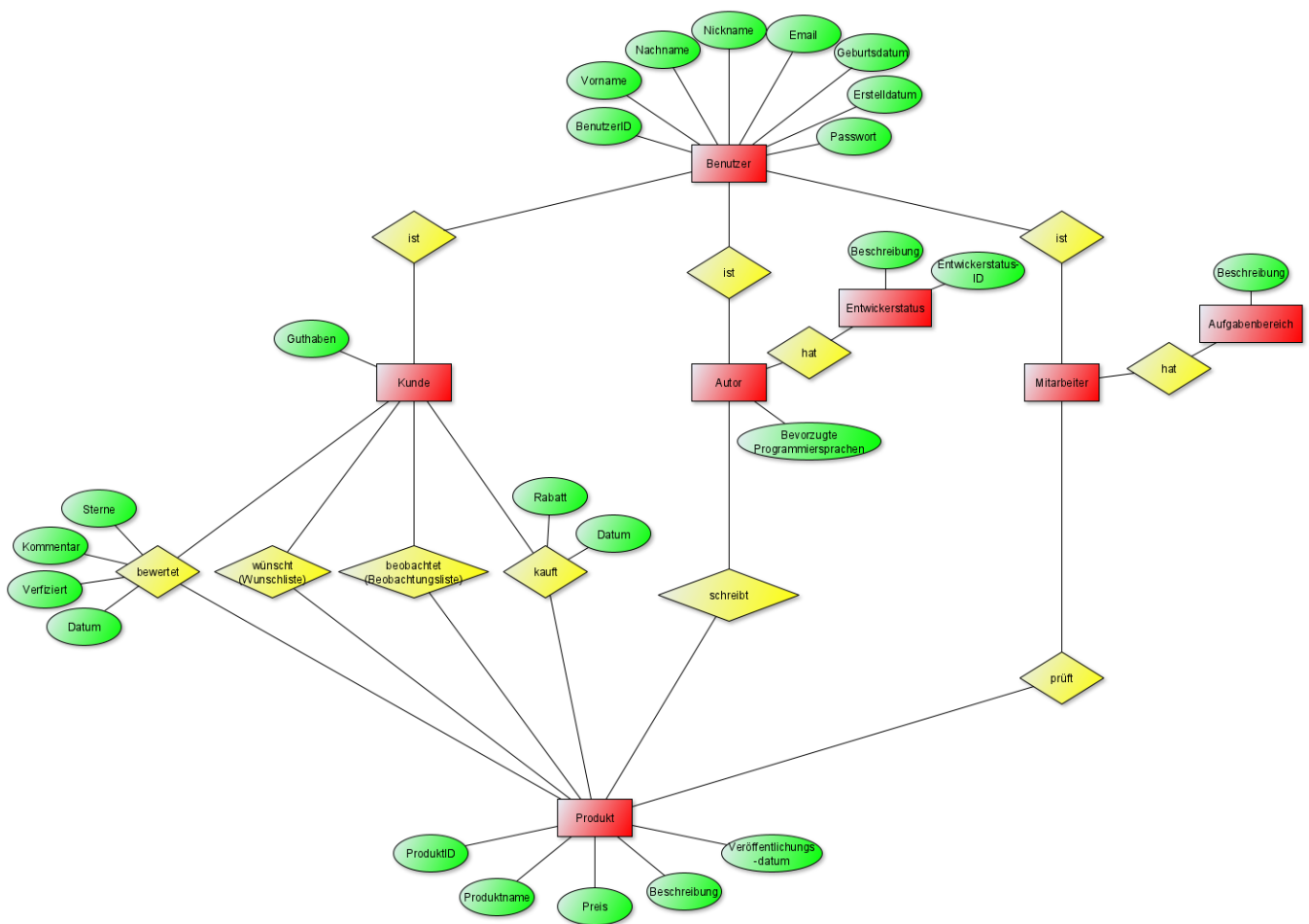


Abbildung 2.1: Entity-Relation-Diagramm

2.2 UML-Klassen-Diagramm

Nachdem ich das erste ER-Diagramm hatte, erstellte ich ein UML-Klassen-Diagramm, wo man eine Übersicht darüber bekommt, welche Klassen und Datentypen notwendig sind. Zu bemerken ist, dass die Inhalte der Enumerationen stetig geändert werden.

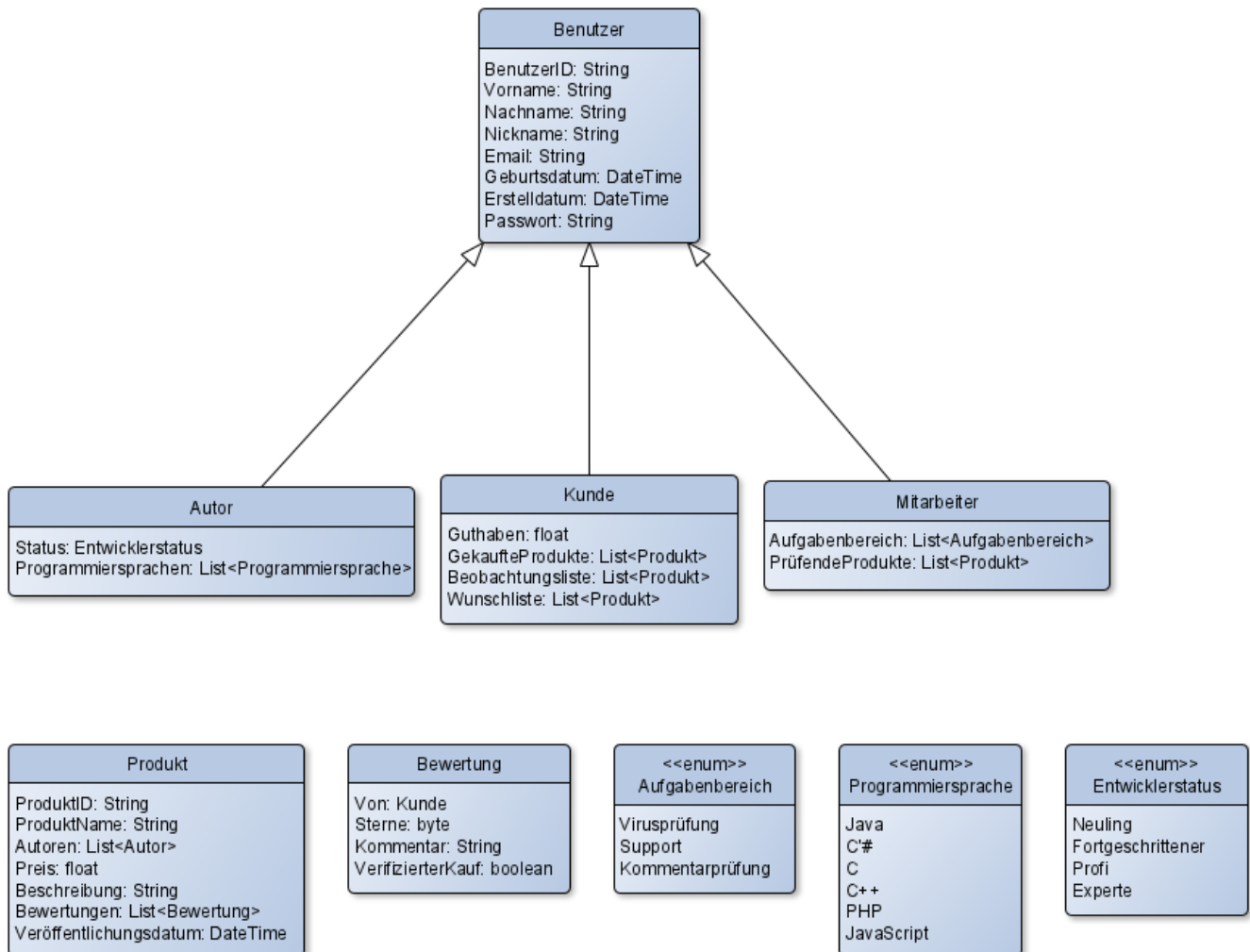


Abbildung 2.2: UML-Klassen-Diagramm

2.3 Programmablaufplan

Für einen übersichtlichen Überblick der Funktionen der Applikation ist ein Programmablaufplan sinnvoll. Vor allem zum Testen der Applikation ist er gut geeignet, da man alle möglichen Zweige einzeln testen kann.

Der Programmablaufplan befindet sich auf GitHub: <https://github.com/OliverSchlueter/Verkaufsproj/blob/master/Planning/PAP%20v1.png>

2.4 Datenbank Skript

Relativ schnell nach dem Erstellen der Diagramme, schrieb ich das Skript zur Erzeugung der Datenbank, wobei vorher die Überlegung der Wahl des DBMS zu treffen war. In meinem Projekt benutze ich eine MySQL-Datenbank für die Speicherung der Benutzer- sowie Produktdaten, um auf diese zuzugreifen ist diese mit einer lokalen Microsoft Access-Datenbank synchronisiert.

Das Skript finden sie hier: <https://github.com/OliverSchlueter/Verkaufsprojekt/blob/master/Planning/DB-Script.sql>

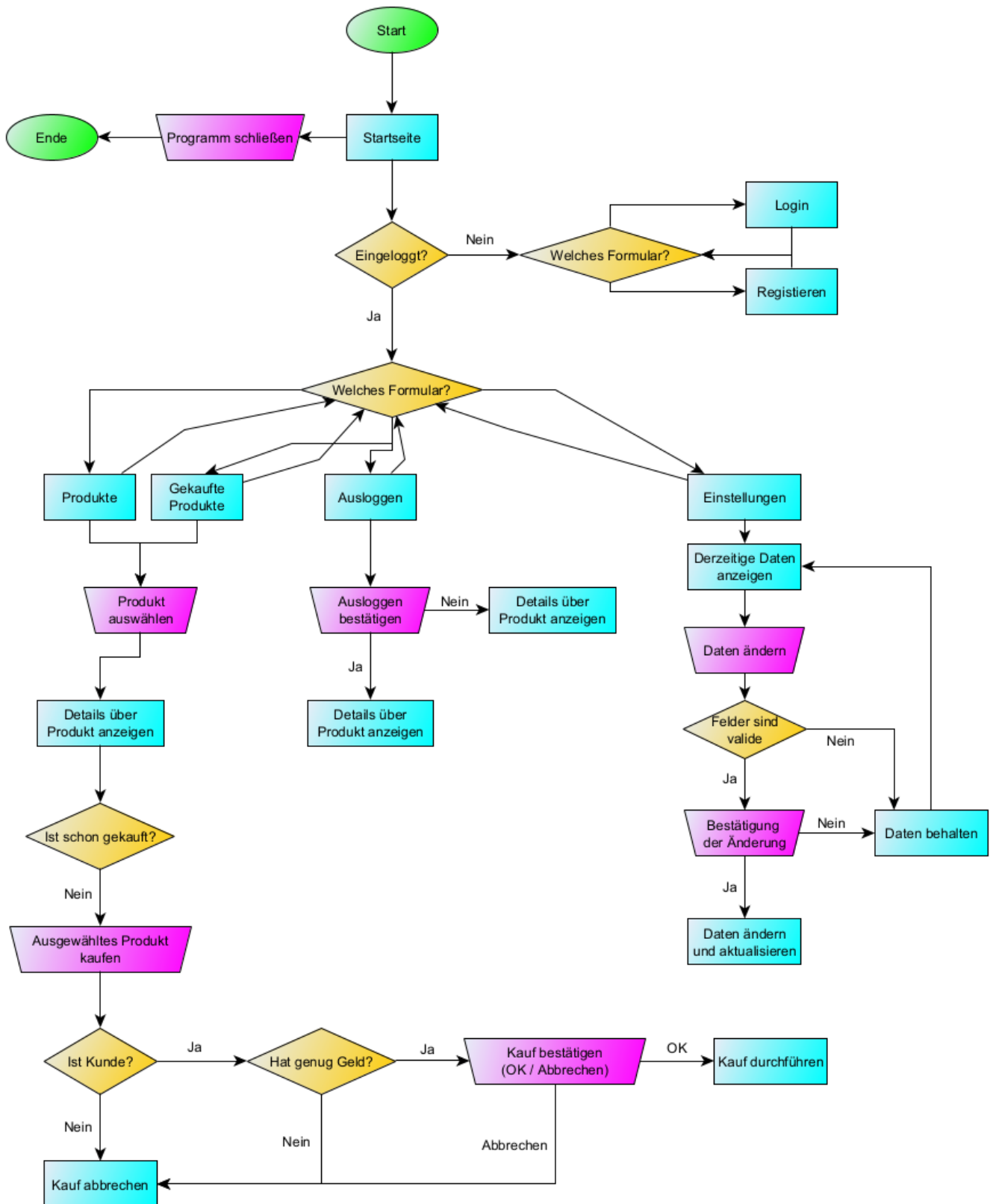


Abbildung 2.3: Programmablaufplan