12. März 2023

Tom Mohr

WireDev ERP

2024

Inhaltsverzeichnis

1 Anforderungsdefinition

Dieses Projekt entsteht im Zusammenhang mit einer Schularbeit. Der Fokus liegt primär auf dem Lerninhalt und der Umsetzung. Die Eignung und Verwendbarkeit des Resultats ist sekundär.

1.1 Projektinhalt

Ziel ist es, für ein kleines fiktives Unternehmen, welches Speisen wie Fast Food und ähnliche Snacks verkauft, eine Kassenverwaltungssoftware zu implementieren, die den Verkauf analysiert und das Lager verwaltet. Bisher wurden die Arbeiten ohne Verwaltungssystem durchgeführt und die Kassierung erfolge manuell. Dadurch herrscht ein höherer Aufwand für Kassenbuch, Steuerberater und die Verwaltung der liquiden Mittel. Bei Einnahmezählungen kann nicht festgestellt werden, welche Produkte sich am besten Verkauft haben und ob es sich Einnahmen mit Ausgaben decken.

Besonderen Wert soll darauf gelegt werden, dass die Software ohne großen Einarbeitungsaufwand bedient und durch die Verwendung einen modernen Technologiestacks in Zukunft einfach gewartet und verwendet werden kann. Da es sich um ein Kassensystem handelt, sind die Anforderungen an Datensicherheit und -integrität besonders hoch.

1.2 Erwartungshorizont

Die bereits vorhandene Infrastruktur basiert auf Windows 10 Computern der x64-Architektur. Daher muss die Anwendung mit mindestens Windows 10 oder neuer kompatibel sein. Eine Netzwerkfähigkeit muss gewährleistet sein, um den Datenaustausch über das interne Netzwerk zu ermöglichen. Im Verwaltungsbereich sollen Mitarbeiter Einnahmen und Produkte verwalten können. Genannte Produkte sollen sinnvolle Eigenschaften wie Bezeichnung, Einzelpreis, Verfügbarkeit im Lager besitzen.

Die getätigten Transaktionen sollen als Statistik gespeichert und dargestellt werden, um dem Benutzer einen Überblick über die vergangenen Verkäufe zu geben. Dabei soll die Darstellung der Daten vorerst nicht erfolgen, d. h., die Entwicklung

eines Clients (etwa in Form einer GUI) ist nicht Teil des Projekts. Die Verarbeitung der Anfragen und Daten erfolgt über eine API, die Endpunkte für die benötigten Funktionen bieten, welche von Clients frei verwendet werden können. Das Projekt muss im Zeitraum vom 27.12.2022 bis 13.03.2023 fertiggestellt werden.

Eine Begrenzung der finanziellen Mittel und somit der Arbeitsstunden gibt es nicht.

2 Technologien

Dieser Abschnitt erläutert die im Projekt verwendeten Technologien. Es wurde Wert darauf gelegt, auf gängige Programme und Technologien zurückzugreifen, um auch zu künftige Entwicklung zu vereinfachen und die Integration mit anderen Systemen ermöglichen.

2.1 C#

C# (gesprochen „C-Sharp“) ist eine objektorientierte Programmiersprache, die im Auftrag von Microsoft entwickelt und 2000 veröffentlicht wurde. Die Sprache ist grundsätzlich plattformunabhängig und wurde fur die Softwareplattform .NET entwickelt. Auf GitHub gehört sie zu den fünf am meisten verwendeten Programmiersprachen.[Mir]

Die Einsatzbereiche dieser Sprache sind vielfältig. Schon seit Veröffentlichung im Jahre 2001 entstanden Projekte in den Bereichen Webseiten, Entwicklerwerkzeuge und Kompilierer. Durch diese einfach, moderne und flexible Sprache qualifiziert sie sich auch für dieses Projekt. Die aktuelle Versions 11 wurde im November 2022 veröffentlicht, was zeigt, dass die Sprache noch aktiv gepflegt wird. Einflüsse durch ältere Programmiersprachen wie C, C++ und Java führeen dazu, dass die Einstiegshürde auch für Entwickler anderer Sprachen gering ist. [Jon]

2.2 HTTP

HTTP steht fur „Hypertext Transfer Protocol“ und ist das Standard-Kommunikationsprotokoll im World Wide Web (WWW). Es ist fur das Abrufen von statischen Inhalten, wie HTML-Dokumenten, vorgesehen und bildet somit die Grundlage für jeden Datenaustausch im Web als Client-Server-Protokoll. In der Regel werden Anfragen an den Server über einen Browser gestellt, der ebenso die Antwort des Servers empfängt und darstellt.

Das in den 1990er Entwickelte Protokoll ist erweiterbar und kann mittels TLS Verschlüsselt werden. Es lässt sich inder Anwendungsschicht im ISO/OSI-Referenzmodell einordnen. HTTP ist zustandslos, da es keine Verbindung zwischen zwei Anfragen gibt, die nacheinander auf der selben Verbindung ausgeführt werden, kann aber mittels HTTP-Cookies für zustandshafte Sitzungen Verwendung finden.[conc]

HTTP-Antworten enthalten die Version des Protokolls, Header, den Statuscode der angibt, ob die Anfrage erfolgreich war (wie in Abbildung 2.1 gezeigt), eine Status-Nachricht und den Körper der Nachricht mit dem angeforderten Inhalt. HTTP-Header ermöglichen Client und Server den Austausch zusätzlicher Informationen, so z.B. für die Authentifizierung des Clients am Server, für das verwendete Caching (Pufferspeichern) oder gespeicherte Cookies.[conb] Das Protokoll wird stetig weiterentwickelt; die aktuelle Version (per Juni 2022) ist HTTP/3.

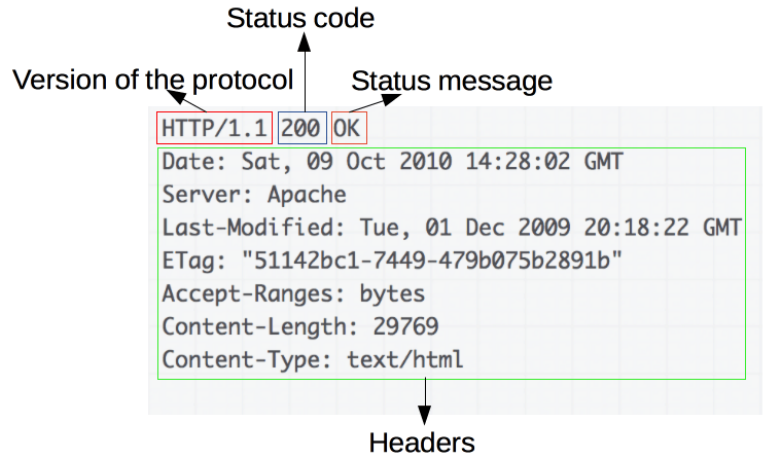

Abbildung 2.1: Beispiel einer HTTP-Antwort

2.3 IDE

Fur eine effiziente Arbeitsweise benötigt man Programme, die beim Schreiben und Kompilieren des Quelltextes unterstützen, indem sie viele Schritte automatischieren. Das Ergebnis ist eine ausfuhrbare Anwendung in Form einer `.exe`-Datei. Da bereits Erfahrung mit der Software Microsoft Visual Studio 2022 existiert und diese für die Entwicklung mit C# optimiert ist, wird diese Software auch für dieses Projekt verwendet. Das Programm bietet hilfreiche Funktionen wie das automatische Installieren von Paketen, das farbige Markieren von Quelltexten fur eine bessere Lesbarkeit. Die IDE bietet außerdem die Möglichkeit, während der Laufzeit in die Anwendung zu schauen, um Fehler schneller finden zu können sowie die Korrektheit der Vorgänge zu uberprufen (Debugging).

2.4 Git

Git ist ein ausgereiftes, aktiv gepflegtes Open-Source-Projekt zur Versionsverwaltung von Software, das ursprünglich 2005 von Linus Torvalds entwickelt wurde. Git ist der Standard für fast alle Softwareprojekte für die Versionskontrolle, sowohl bei kommerziellen Projekten als auch in Open-Source-Software. Git verwendet dabei einen Algorithmus, der sowohl sehr schnell als auch speichersparend arbeitet. Somit gehört es mit zu den leistungsstärksten Versionsverwaltungstools. Die Integriergerät des Quellcodes war bei der Entwicklung die höchste Priorität. Der SHA1 Algorithmus wird für die Sicherung der Verzeichnissen verwendet. So können nachträgliche Veränderungen des Verlaufs erkannt werden.

Git zeichnet sich ebenso durch seine Flexibilität aus: es unterstützt verschiedene Arten von Entwicklerworkflows, die nichtlinear sind. Es eignet sich somit für alle Größen von Projekten.[cona] Durch die Einteilung von Git in drei Arbeitsbereiche können Änderungen flexibel angebracht werden.

Im Arbeitsverzeichnis (Working Directory) liegen alle Projektdateien zur Bearbeitung oder Benutzung. Änderungen an diesen Dateien werden im Entwicklungsbereich (Staging Area) erfasst und aufbewahrt. Um diese Änderungen zu veröffentlichen, muss ein Commit (Beitrag) erzeugt werden, welcher die Informationen der Änderungen trägt. Diese können dann im Versionsverlauf des Git Verzeichnisses gespeichert werden. Abbildung 2.2 stellt diese Beziehungen dar.

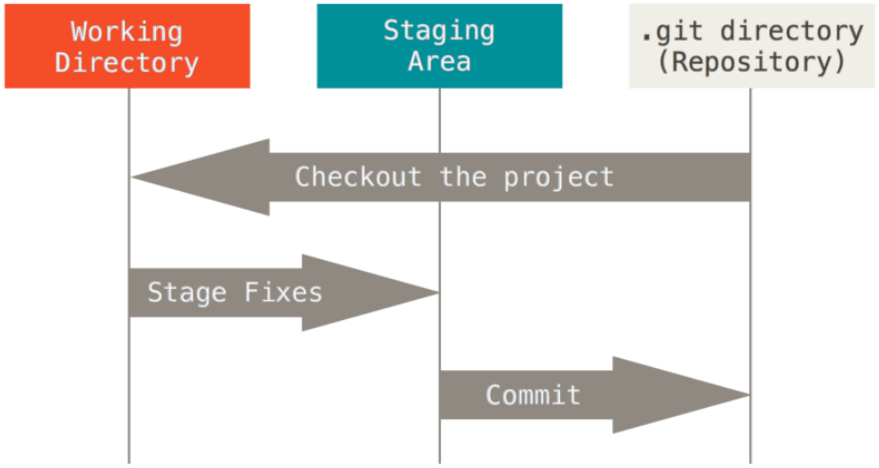

Abbildung 2.2: Arbeitsverzeichnis, Entwicklungsbereich, Git Verzeichnis

2.5 SQLite

SQLite ist eine im Funtkionsumfang reduzierte Alternative zu SQL-Imlementierungen wie MySQL. Die Open-Source-Software kann in Anwendungen integriert werden, um Speicherung von Daten ohne separaten Datenbankserver zu ermöglichen. SQLite wurde in C geschrieben und ist mit vielen Betriebssystem wie Windows, Linux, MacOS, Android und iOS kompatibel. Unterstützt werden grundlegende SQL-Operationen wie `INSERT`, `UPDATE SELECT` und `DELETE`. Abfragen mit Aggregatfunktionen, Unterabfragen und Joins sind ebenso möglich. Durch die ACID-Konformität ist Zuverlässigkeit und Robustheit gewährleistet. Der Funktionsumfang ist gegenüber MySQL jedoch eingeschränkt. Diese Einschränkung haben jedoch keinen negativen Einfluss auf das Projekt.

2.6 Kanban

Kandban ist ein Arbeitsmodus, indem ein Projekt in kleine Arbeitspakete zerlegt wird und diese mithilfe eines Boards geplant werden können. Das Board besteht aus Spalten wie z. B. *To Do*, *In Progress*, *Done* als einfachste Form. Ein Vorteil dieser Arbeitsweise ist, einen direkten überblicke über getane und geplante Arbeit zu erhalten. Innerhalb der einzelnen Spalten könne zusätzlich priorisierte Gruppen erstellt werden. Beim Kollaborativen Arbeiten werden die Tickets, die in Bearbeitung sind, einer Verantwortlichen Person zugewiesen. Um den Erfolg einer Arbeit in Kanban zu messen, verwendet man zwei wesentliche Metriken: die Zykluszeit, welche die benötige Zeit für eine Aufgabe misst und den Durchsatz, der die Anzahl der abgeschlossenen Daten für eine bestimmte Zeiteinheit zurückgibt. [kan]

Eine weitere Darstellungsform des Erfolgs ist das kumulatives Flussdiagramm (CFD). Es veranschaulicht die Anzahl der Elemente (Vertikale Achse) im Laufe der Zeit (Horizontale Achse). Die Zustände der Bereiche werden mir Farben gekennzeichnet, wie Abbildung 2.3 auf der nächsten Seite zeigt. Sind die Verläufe der Bereiche nahezu parallel, wurde der Zeitplan eingehalten.

2.7 Testing

Um sicherzustellen, das die Software den gestellten Anforderungen entspricht, müssen zentrale Bestandteile dieser durch Tests abgesichert werden. Es gibt verschiedene Arten von Test, von denen die wichtigsten Unit-Test sind. Hierbei handelt es sich um de Prüfung isolierter Softwarebestandteile, die gegen vordefinierte Daten geprüft werden. Zu beachten ist, dass Tests keine Fehlerfreiheit garantieren können, jedoch insbesondere die Software gegen Fehler durch Erweiterungen robust machen.[cond]

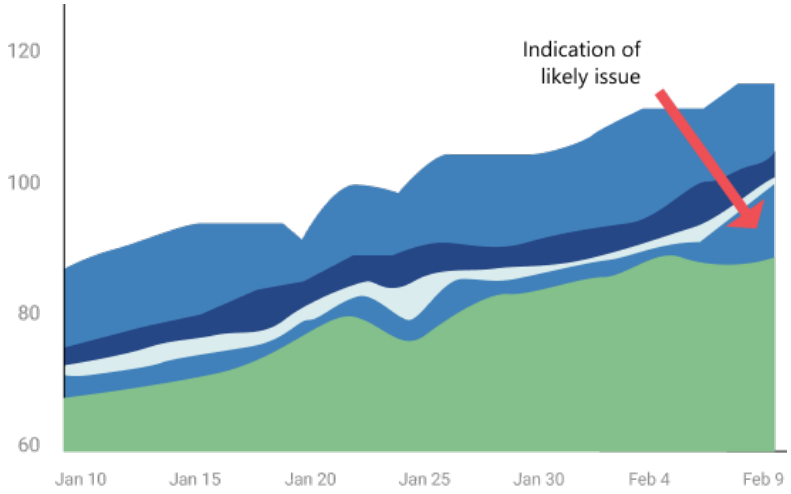

Abbildung 2.3: Beispiel eines CFD mit Verschiebung im Zeitplan

Andere Arten von Softwaretest sind Integration- und Smoke-Test, welche als Blackboxtest zusammengefasst werden können. Integrationstests überprüfen die einzelnen Module einer Software auf ihre Zusammenarbeit mit anderen. So können z.B. Datenbankanbindungen und Microservices auf ihre Funktionalität geprüft werden.[Pit]

Allgemein sollen Tests Sicherheit, Produktqualität und Kostenersparnisse sichern. Wenn ein Defekt in der Software nicht zeitnah erkannt wird, steigt der Suchaufwand nach dem Auslösers in komplexen Projekten enorm. Werden Fehler in der frühen Entwicklungsphase behoben, werden so die Kosten für die weiteren Entwicklung niedriger gehalten. Sicherheitslücken die noch vor der Veröffentlichung einer Version geschlossen werden, bieten Angreifern keine Möglichkeit eine Software zu missbrauchen. Nebenbei sei auch erwähnt, dass eine fehlerfreie Software mehr Kundenzufriedenheit bedeutet.[Raj]

3 Authentifizierung

Dieser Abschnitt erklärt die Arbeitsweise der implementierten Authentifizierung.

3.1 HTTP-Authentifizierung

Die Authentifizierung für HTTP soll sicherstellen, das die Anfragen eines Benutzers nur bearbeitet werden, wenn dieser dazu berechtigt ist. Durch die Feststellung der Identität wird die Sicherheit des Systemen gewährleistet. Es gibt mehrere Authentifizierungsverfahren. Die einfachste ist, dass der Client Benutzername und Passwort an den Server schickt, welcher diese anschließend validiert.

Basic Authentication. Als am einfachsten zu Implementieren gilt das Senden und Überprüfen von Benutzername und Passwort im Klartext. Dies ist jedoch nur mit der Verwendung von HTTPS zu empfehlen, da die Sicherheit sonst nicht gewährleistet werden kann.

Digest Authentication. Anders als bei der Basic Authentication wird das Passwort hier nicht im Klartext übertragen. Ein Hash des Passworts erhöht hier die Sicherheit und gibt somit nicht das Geheimnis preis.

Token-Based Authentication. Statt eines Passworts kommt hier ein eindeutiges Token zum Einsatz, welches bei jeder Aufforderung mitgesendet wird. Der Server validiert dieses jedes mal, um die Berechtigung des Benutzers zu prüfen. Ein Vorteil ist, dass hierbei Berechtigungen mit dem Token verknüpft sein können.

OAuth2 Authentication. Dieser Standard überträgt Authentifizierungs- und Autorisierungsinformationen zwischen zwei Teilnehmern eines Netzwerks in Form eines Token. Diese Methode soll einen einfachen Informationsaustausch für Anmeldungen ermöglichen, ohne das eine aktive Verbindung aufrechterhalten werden muss. Der hier verwendete Token kann von dem eigenen oder sogar einem Drittanbieter bereitgestellt werden, der sein eigenes Rechtesystem verwaltet. Dieser teilt einem anderen Server bei einer Anfrage mit, ob der Benutzer berechtigt ist, auf diese Ressourcen zuzugreifen. Dieses Verfahren hat den Vorteil, dass dass Passwort

nur dem Server, der das Token ausstellt, bekannt sein muss, nicht jedoch dem anfragenden Server. Diese Eigenschaften machen JWT portabel und skalierbar. Die Übertragung ist plattformübergreifend möglich und das Format ist kompakt. Der Server muss hier keine Sitzungsinformationen speichern, da ein JWT alle Informationen über die Berechtigungen enthält. Eine Manipulation kann sehr einfach erkannt werden, da es bei Veränderunen des Inhalts zu Fehlern bei einer Validierung der Signatur kommt.[ION]

JWTs wirken bei Single-Sign-On (SSO) unterstützend, da der Benutzer nach einer Authentifizierung einen JWT erhält, welchen er für die Anmeldung bei Anfragen an einen Server verwenden kann.

3.2 JSON-Web-Token

Eine mögliche Implementierung eines OAuth2 Token kann über JWT realisiert werden (bei der Erweiterung OpenIdConnect ist das immer der Fall). Das JSON-Web-Token (JWT) setzt sich aus drei Teilen zusammen:

- dem **Header** mit Token-Typ und der Signaturmethode (base64-encoded),
- der **Payload**, welcher Informationen über den Benutzer (z. B. die E-Mail-Adresse) beinhaltet (base64-encoded) und
- der **Signatur**, welche die Richtigkeit des Tokens bezeugen soll.[aut]

Ein Beispiel-Token mit decodiertem (Klartext-)Inhalt kann der Abbildung 3.1 entnommen werden.

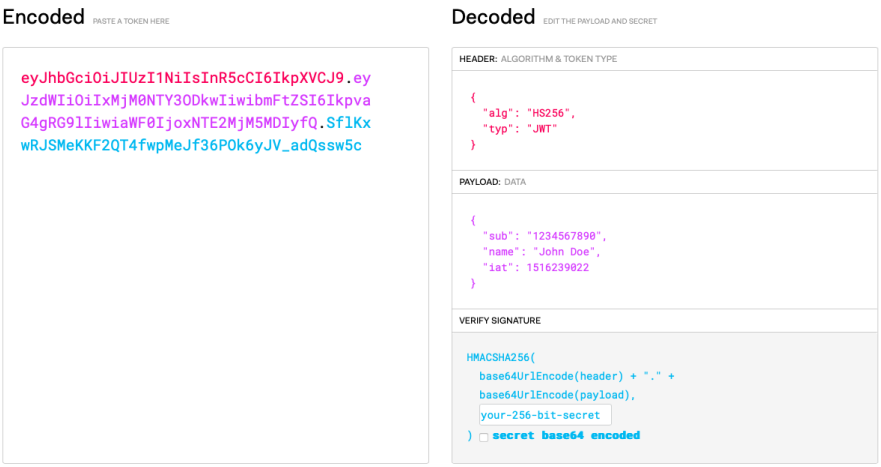

Abbildung 3.1: Vergleich von enkodiertem und dekodiertem JWT

4 REST-API

Der Architekturstil des *Representational State Transfer Application Programming Interface* (REST-API) wurde für Webanwendungen entwickelt, welche eine Schnittstelle zwischen verschiedenen Endpunkten implementieren. Diese dienen dem Austausch von Daten und Ressourcen. Um Zugriff auf diese zu erhalten, stellt das HTTP-Protokoll verschiedene Methoden (*HTTP-Verben*) zuer Verfügung. Die wichtigsten sind `GET`, `POST`, `PUT` und `DELETE`. Zur Adressierung werden URLs (Uniform Resource Locator), eine spezielle Form von URIs (Uniform Resource Identifier) verwendet.

Anfragen an die API erfolgen üblicherweise im JSON- oder XML-Format, um die angeforderten oder übertragenen Daten zu gliedern. Die oben genannten Verben haben folgende Bedeutungen:

- `GET`: Abrufen von Daten aus einer Ressource
- `POST`: Erstellen von Daten auf der Ressource
- `PUT`: Aktualisieren von Daten auf der Ressource
- `DELETE`: Löschen von Daten von der Ressource

In diesem Projekt werden API-Endpunkte (gekapselt in sog. *Controllern*) und andere Funktionen wie z. B. Datenbankoperationen voneinander getrennt. Das verwendete HTTP-Framework bindet den HTTP-Request an eine Funktion, wodurch eine einfache und saubere Implementierung ermöglicht wird. Somit können Endpunkte einfach verschoben und Integration Tests einfacher durchgeführt werden. Annotationen mit den HTTP-Verben definieren dabei die Art der Anfrage. Das ermöglicht, die Zugehörigkeit direkt im Quelltext abzulesen.

Die in der URL übertragenen Parameter werden direkt die der Funktion zugeordnet. Dadurch wird sichergestellt, das die Daten aus dem Body der Anfrage dem Datentyp entsprechen, den die Funktion verarbeiten kann. Sollte dies nicht der Fall sein, beantwortet das Framework, welches in Abschnitt 5 genauer erklärt wird, die Anfrage mit einem HTTP-Fehlercode.

Funktion geben immer ein `IActionResult` zurück, welches die Klasse `ObjectResult` enthält. Die hier am Wichtigsten enthaltenen Eigenschaften sind der Statuscode und der Body. Das Framework kann somit eine HTTP-Antwort zurückgeben und

gleichzeitig können Integration Tests intern, d. h. ohne HTTP, durchgeführt werden.

5 Back-End

Die API basiert hauptsächlich auf dem Framework `Microsoft.EntityFrameworkCore`. Ein Objekt vom Typ `IHost` (Host) mit einer `IConfiguration` (Konfiguration) schafft die Grundlage der API. Welche API-Controller eingebunden sind, die Anbindung der Datenbank sowie die Definition der Zugriffsrechte sind Informationen, die in der Konfiguration enthalten sind.

Das bereits im Framework eingebaute Werkzeug *SwaggerUI* kann verwendet werden, um den Aufbau der API grafisch darzustellen und Anfragen direkt über den Browser stellen zu können. Hier werden auch Definitionen, Kommentare und Datentypen beschrieben und wie sie in der API verwendet werden können.

Das verwendete Datenbanksystem ist SQLite. Benutzer und Anwendungsdaten werden in zwei getrennten .db-Dateien gespeichert. Der Grund für diese Trennung liegt in der Funktionsweise der Anbindung. Während für die Datenbank mit den Anwendungsdaten ein Klasse vom Typ `DbContext` verwendet wird, benötigt die Verarbeitung der Benutzer und Rollen einen `IdentityDbContext`.

Damit es hier nicht zu Migrationsproblemen kommt und die Benutzerdaten bei einer Änderung der Datenstruktur nicht erneut generiert werden müssen, wurden die beiden Kontext-Klassen nicht zusammengelegt. Außerdem macht diese Methode es möglich, Anwendungsdaten und Benutzerdaten an getrennten Orten aufzubewahren, was im Sinne der Datensicherheit und des Datenschutzes ist.

Die Klassen für Benutzer- und Rollendaten werden von den beiden Frameworks `Microsoft.AspNetCore.Identity` und `Microsoft.AspNetCore.Identity.EntityFrameworkCore` bereitgestellt. Die Eigenschaften und Beziehungen der Anwendungsdaten können Abbildung 5.1 auf der nächsten Seite entnommen werden.

Damit Endpunkte verwendet werden können, muss sich der Benutzer dafür authentifizieren. Dieser Vorgang wird in Abbildung 5.2 auf Seite 14 beschrieben.

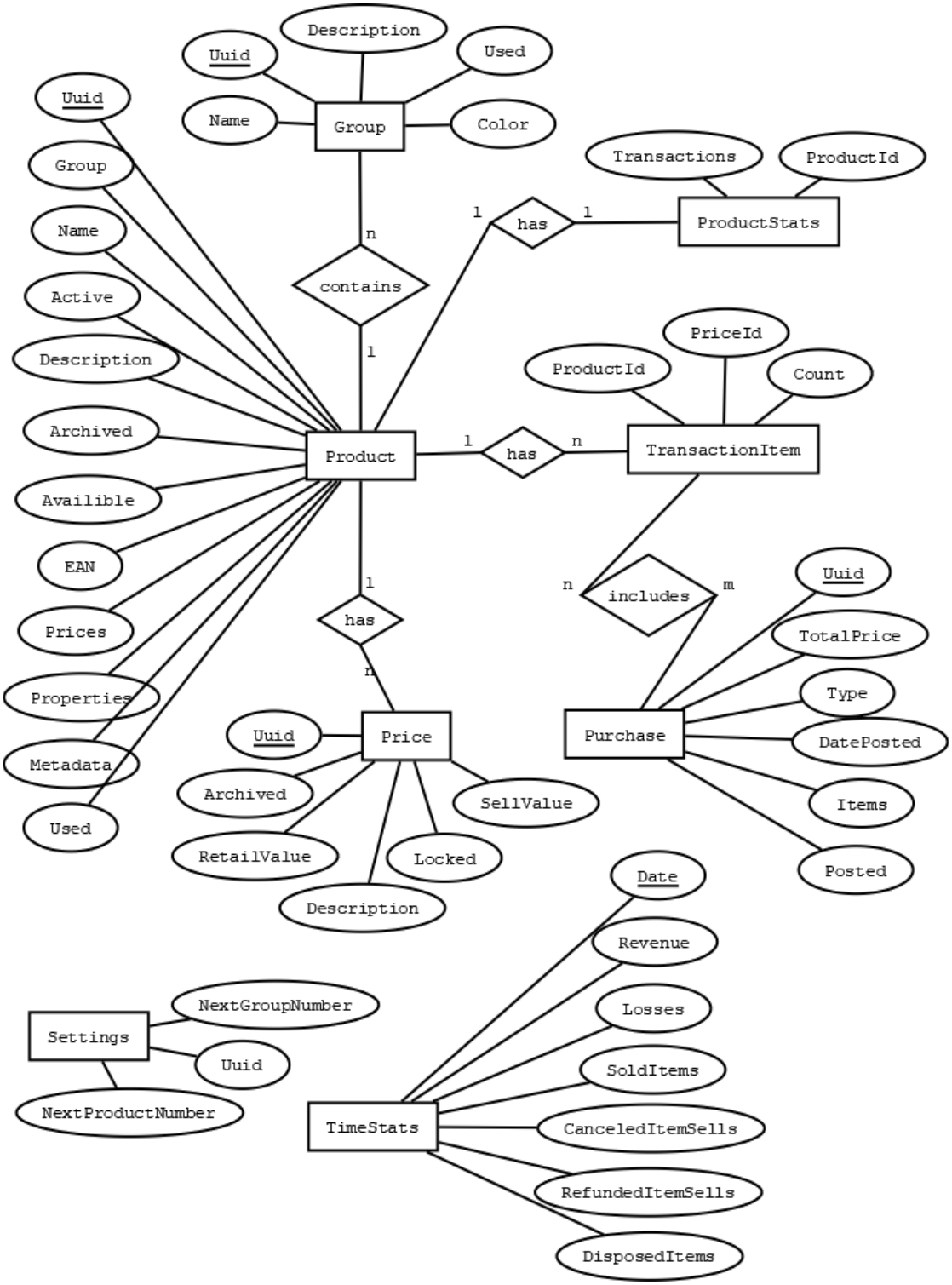

Abbildung 5.1: ERM der Anwendungsdaten

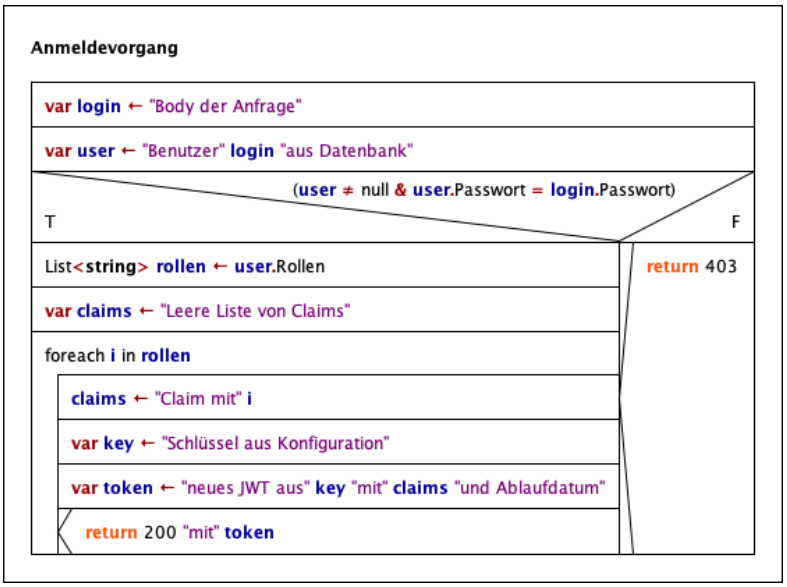

Abbildung 5.2: Ablauf der Anmeldung

6 Reflexion

Auswertung des Endprodukts

Literaturverzeichnis

[aut] AUTH0: *Introduction to JSON Web Tokens.* Online. Abgerufen am 11.03.2023. https://jwt.io/introduction

[Bit] BITBUCKET, Atlassian: *What is Git.* Online. Abgerufen am 11.03.2023. https://www.atlassian.com/git/tutorials/what-is-git

[cona] CONTRIBUTORS, Git: *1.3 Getting Started - What is Git?* Online. Abgerufen am 11.03.2023. https://git-scm.com/book/en/v2/Getting-Started-What-is-Git

[conb] CONTRIBUTORS, MDN: *HTTP headers.* Online. Abgerufen am 11.03.2023. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

[conc] CONTRIBUTORS, MDN: *An overview of HTTP.* Online. Abgerufen am 11.03.2023. https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

[cond] CONTRIBUTORS, Mircosoft: *Testing in .NET.* Online. Abgerufen am 11.03.2023. https://learn.microsoft.com/en-us/dotnet/core/testing/

[cone] CONTRIBUTORS, SQLite: *About SQLite.* Online. Abgerufen am 11.03.2023. https://www.sqlite.org/about.html

[ION] IONOS: *JSON Web Token (JWT): an introduction.* Online. Abgerufen am 11.03.2023. https://www.ionos.com/digitalguide/websites/web-development/json-web-token-jwt/

[Jon] JONES, Brad: *What is C?* Online. Abgerufen am 11.03.2023. https://www.developer.com/guides/what-is-c/

[kan] KANBANIZE: *Kanban 101: The Complete Kanban Guide.* Online. Abgerufen am 11.03.2023. https://kanbanize.com/kanban-resources

[mij] MIJACOBS, v-thepet E. alexbuckgit: *What is Kanban?* Online. Abgerufen am 11.03.2023. https://learn.microsoft.com/en-us/devops/plan/what-is-kanban

[Mir] MIRCOSOFT: *C.* Online. Abgerufen am 11.03.2023. `https://dotnet.microsoft.com/en-us/languages/csharp`

[Pit] PITTET, Sten: *The different types of software testing.* Online. Abgerufen am 11.03.2023. `https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing`

[Raj] RAJKUMAR: *What Is Software Testing | Everything You Should Know.* Online. Abgerufen am 11.03.2023. `https://www.softwaretestingmaterial.com/software-testing`