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 kleine Speisen (Fast Food und 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 werden kann, 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. Die Verarbeitung der Anfragen und Daten erfolgt über eine API, die Endpunkte für die

benötigten Funktionen bieten, welche von Clients frei verwendet werden können. Ein Client ist jedoch nicht Teil des Erwartungshorizonts. Das Projekt muss im Zeitraum vom 27.12.2022 bis 13.03.2023 fertiggestellt werden. Eine finanzielle Begrenzung gibt es nicht.

2 Technologien

Dieses Projekt verwenden gängige Technologien, welches die Entwicklung vereinfachen und die Integration mit anderen Systemen ermöglichen. Im folgenden werden die wesentlichen T. erklärt.

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.[**?**]

Die Einsatzbereiche dieser Sprache sind nahezu Grenzenlos. Schon seit Veröffentlichung entstanden Projekte in den Bereichen Webseiten, Entwicklerwerkzeuge und Kompilierer. Durch diese einfach, moderne und flexible Sprache qualifiziert sie sich auch für dieses Projekt. Der Zusammenschluss von C und C++ brachte diese Eigenschaften herbei und macht es somit selbst Java Entwicklern einfach, die Sprache zu verwenden.[**?**]

2.2 HTTP

HTTP steht fur „Hypertext Transfer Protocol“ und ist das 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.[**?**]

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.[?]

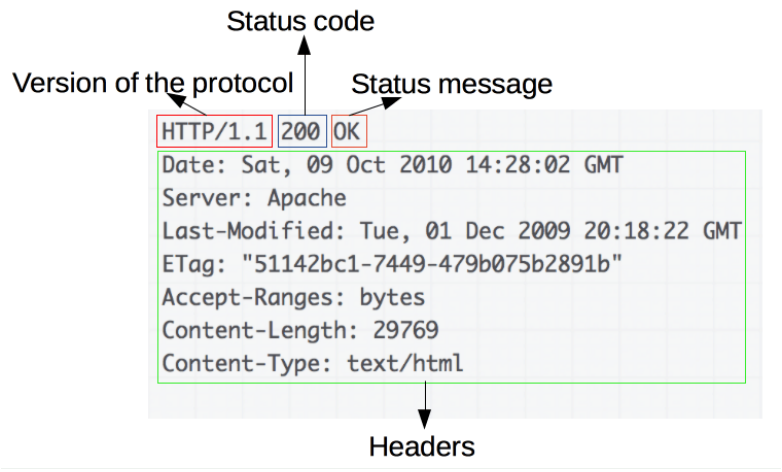

Abbildung 2.1: Beispiel einer HTTP-Antwort

2.3 IDE

Fur eine effiziente Arbeitsweise benötige ich Programme, welche mir beim Schreiben des Quelltextes helfen und diesen fur mich kompilieren, damit am Ende eine ausfuhrbare Anwendung bereitsteht. Auch hier greife ich wieder auf eine mir bereits bekannte Lösung von Microsoft zuruck: Microsoft Visual Studio 2022. Das Programm bietet hilfreiche Funktionen wie das automatische Installieren von Paketen, das farbige Markieren von Quelltexten fur eine bessere Lesbarkeit und einen Kompilierer, der C-Sharp versteht. Die IDE gibt mir außerdem Möglichkeiten in meine Anwendung während der Laufzeit hereinzuschauen, um Fehler schneller finden zu können sowie die Korrektheit der Vorgänge zu uberprufen.

2.4 Git

Git ist ein ausgereiftes, aktiv gepflegtes Open-Source-Projekt, das ursprünglich 2005 von Linus Torvalds, entwickelt wurde. Eine erstaunliche Anzahl von Softwareprojekten verlässt sich auf Git für die Versionskontrolle, einschließlich kommerzieller Projekte sowie Open Source. Im Gegensatz anderer Versionskontrollsoftware

lässt sich Git bei der Versionshistorie des Dateibaums nicht von den Namen der Dateien täuschen, stattdessen konzentriert sich Git auf den Dateiinhalt selbst. Somit gehört es mit zu den Leistungsstärksten seiner Art. Die Integriergerät des Quellcodes war bei der Entwicklung die höchste Priorität. Der SHA1 Algorithmus wird für die Sicherung der Verzeichnissen verwendet. So wird der Verlauf vor Änderungen beschützt.

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.[?] 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.

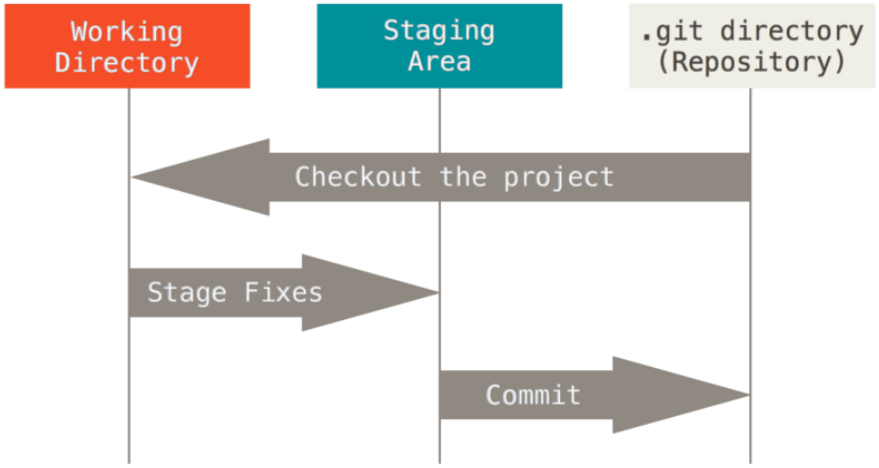

Abbildung 2.2: Arbeitsverzeichnis, Entwicklungsbereich, Git Verzeichnis

2.5 SQLite

Das relationale Datenbank-Management-System (DBMS) ist eine leichtgewichtige Lösung, die ohne Server auskommt, für eine schnelle Datenbankanbindung. 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. 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 Funtkionsumfang ist gegenüber MySQL sehr 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 TODO, In Progress, Done als einfachste Form. Ein Vorteil dieser Arbeitsweise ist, einen direkten überblicke über getane und geplante Arbeit zu erhalten. Du durch einzelne 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. [**?**]

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 zeigt. Sind die Verläufe der Bereiche nahezu parallel, wurde der Zeitplan eingehalten.

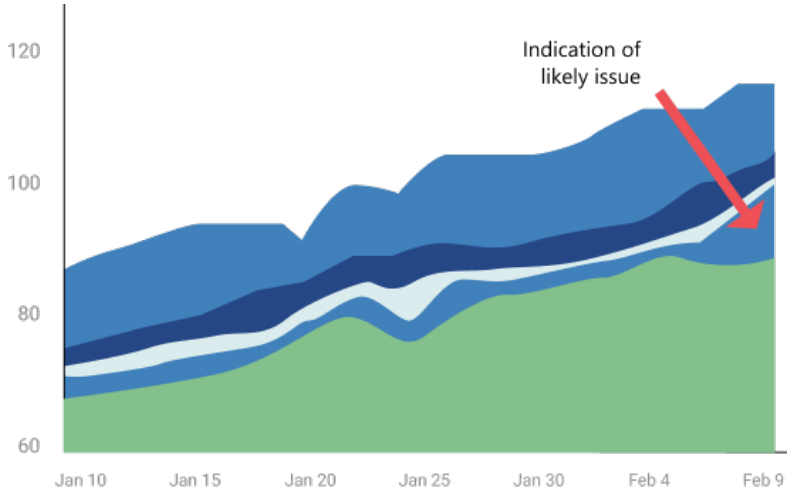

Abbildung 2.3: Beispiel eines CFD mit Verschiebung im Zeitplan

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, das Tests keine Fehlerfreiheit garantieren können, jedoch insbesondere die Software gegen Fehler durch Erweiterungen robust machen.[**?**]

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.[**?**]

Allgemein sollen Tests Sicherheit, Produktqualität und Kostenersparnisse sichern. Wenn ein defekt in der Software nicht Zeitnah erkannt wird, steigt der Suchaufwand nach des 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.[**?**]

3 Authentifizierung

3.1 HTTP-Authentifizierung

Die Authentifizierung für HTTP soll sicherstellen, das die Anfragen eines Benutzers nur bearbeitet werden, wenn dieser dazu berechtigt ist, diese Anfragen zu stellen. Durch die Feststellung der Identität kann die Sicherheit im Netzwerk gewährleistet werden. Es gibt mehrere Arten von Methoden zur Authentifizierung. Im wesentlichen bestehen diese darin, 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 Basic, 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.

OAuth2 Authentication. Der hier verwendete Token wird von einem Drittanbieter bereitgestellt, der sein eigenes Rechtesystem verwaltet. Dieser teilt dem Server bei einer Anfrage mit, ob der Benutzer berechtigt ist, auf diese Ressourcen zuzugreifen.

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 die Signaturmethode enthält, dem Payload, welcher Informationen über den Benutzer (z.B. E-Mail Adresse) beinhaltet und die Signatur, welche die Richtigkeit des Tokens bezeugen soll.[?]

Ein Beispiel Token mir Inhalt kann der Abbildung 3.1 entnommen werden.

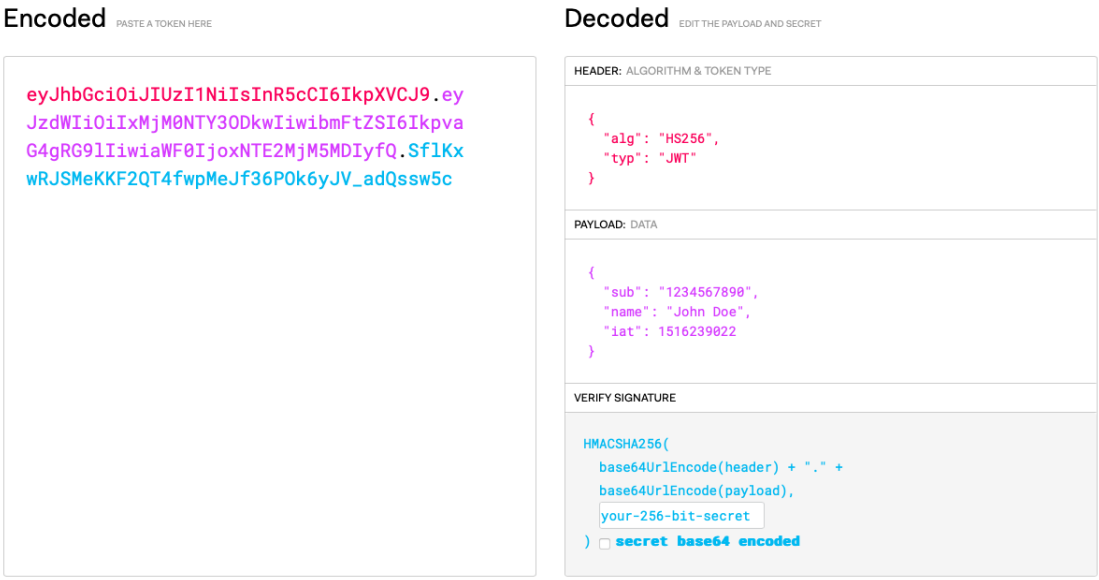

Abbildung 3.1: Vergleich von enkodiertem und dekodiertem JWT

Dieser Standard überträgt Authentifizierungs- und Autorisierungsinformationen zwischen zwei Teilnehmern eines Netzwerks in Form einer Zeichenkette, dem Token. Diese Methode soll einen einfachen Informationsaustausch für Anmeldungen ermöglichen, ohne das eine aktive Verbindung aufrechterhalten werden muss.

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 JWT alle Informationen über die Berechtigungen enthält. Eine Manipulation ist nicht möglich, da es sonst zu Fehlern bei einer Validierung der Signatur kommt.[?]

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.

4 REST-API

Erklären, Aufbau, Endpoints

Der Architekturstil von 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, wird das Prinzip der Nutzung von HTTP verwendet: GET, POST, PUT, DELETE. Zur Adressierung werden URI (Uniform Resource Identifier) und URL (Uniform Resource Locator) verwendet.

Anfragen an die API erfolgen gängigerweise im JSON- oder XML-Format, um die angeforderten oder übertragenen Daten zu gliedern. Zur Manipulation werden 4 Anfragetypen verwendet:

- 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-Controller und Funktionen voneinander getrennt. Es erfolgt lediglich eine Verweisung von Request-URL auf eine Funktion. Somit können Endpunkte einfach verschoben und Integration Tests einfacher durchgeführt werden. Der Verweis erfolgt mit einem Attribut (HttpGet, HttpPost, etc.) über der Definition einer Funktion. 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 Fehler.

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, ohne HTTP, durchgeführt werden.

5 Back-End

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

Das bereits im Framework eingebaute 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 Wahl ist in der Funktionsweise der Anbindung begründet. 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 bei einer Änderung der Datenstruktur die Benutzerdaten 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 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 kann in Abbildung 5.2 auf Seite 13 betrachtet werden.

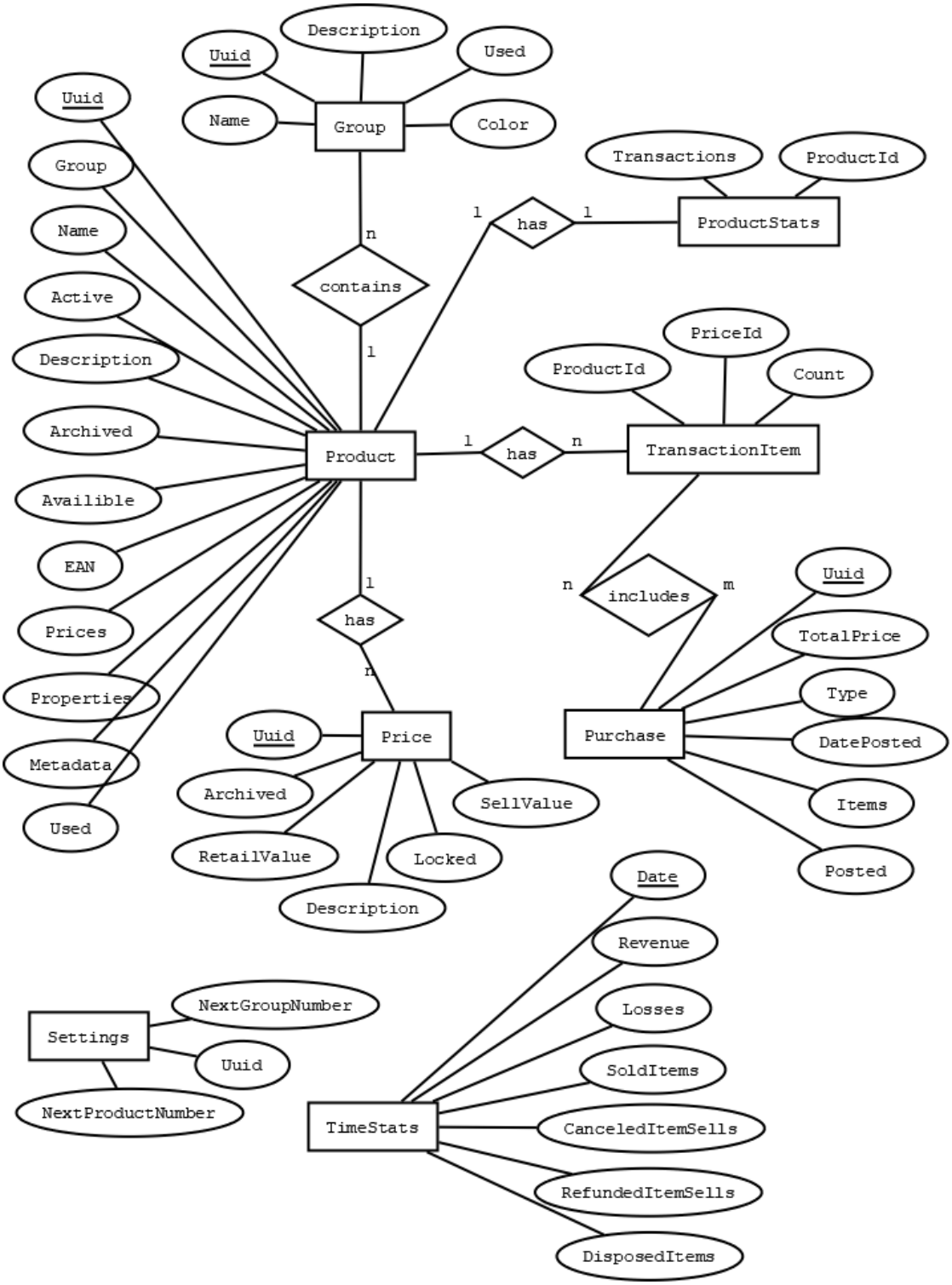

Abbildung 5.1: ERM der Anwendungsdaten

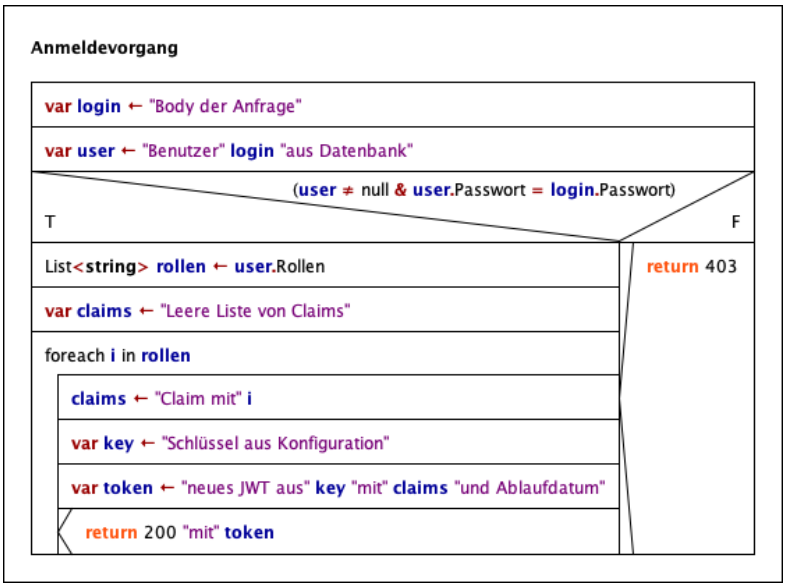

Abbildung 5.2: Ablauf der Anmeldung

6 Reflexion

Auswertung des Endprodukts

7 Einführung

Huhn
Fett
Unterstrichen
Das ist ein g e s p e r r t e r Text.

Die Graphik € 12,34 **??** auf Seite ?? zeigt ein ERM [**?**] k € 12,34 **??** auf Seite ?? zeigt k € 12,34 7.1 auf der nächsten Seite zeigt k € 12,34 **??** auf Seite ?? zeigt
dfg
d Huhn

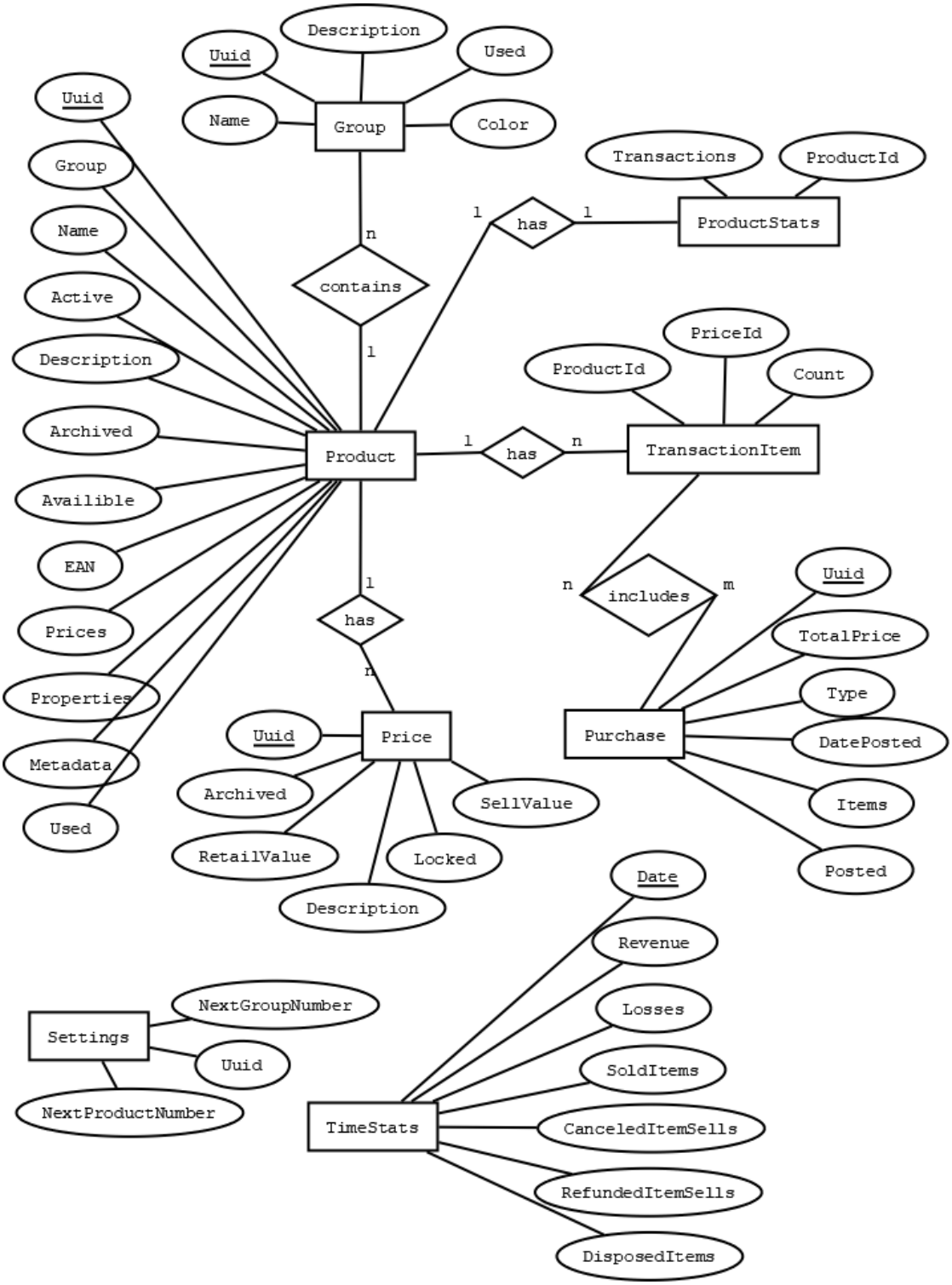

Abbildung 7.1: ERM