

---

*TAU INF202 Software Engineering*  
*Individuelles Projekt*  
***Pflichtenheft***

Projektdokumentation

Version: 2023.v1.0

Status: In Review

Price Tracker

Verantwortliche/r:

Metin Kerem ÖZTÜRK, e200503057@stud.tau.edu.tr

Yavuz Selim Tunçer, e200503001@stud.tau.edu.tr

Stakeholder: DI. Ömer Karacan, omer.karacan@tau.edu.tr

---

## Dokumentenverwaltung

### Dokument-Historie

Version	Status *)	Datum	Verantwortlicher	Änderungsgrund
v1.0	In Review	16.04.2023	K. Öztürk Y. Tunçer	Alle anderen Abschnitte wurden hinzugefügt.

### Dokument wurde mit folgenden Tools erstellt:

Microsoft Office Word

Figma

Lucidchart

## Inhaltsverzeichnis

1. Einleitung.....	4
2. Ausgangssituation und Ziele .....	5
3. Gesamtarchitektur .....	6
4. Funktionale Anforderungen .....	8
5. Nichtfunktionale Anforderungen .....	12
6. Abnahmekriterien.....	13
7. Projekt Meilensteine .....	14
8. Referenzen.....	15

## 1. Einleitung

Das vorliegende Pflichtenheft definiert die Anforderungen an das Software-Engineering-Projekt zur Entwicklung eines Preisverfolgungssystems für E-Commerce-Plattformen. Dieses Dokument stellt sicher, dass alle Anforderungen an das System aus der Sicht der Auftraggeber, Kunden, Anwender und Stakeholder vollständig und konsistent beschrieben werden. Die Bedeutung dieses Dokuments liegt in der klaren Kommunikation der Anforderungen, um eine erfolgreiche Produktrealisierung zu gewährleisten.

Das Pflichtenheft gliedert sich in die folgenden Kapitel:

1. **Einleitung:** Dieses Kapitel stellt die Bedeutung des Pflichtenhefts und einen Überblick über die einzelnen Kapitel vor.
2. **Ausgangssituation** und Ziele: In diesem Kapitel wird der aktuelle Kontext des Projekts erläutert, und die Projektziele werden festgelegt.
3. **Gesamtarchitektur:** Dieses Kapitel beschreibt die Architektur des Systems, um die Anforderungen zu erfüllen.
4. **Funktionale Anforderungen:** In diesem Kapitel werden die funktionalen Anforderungen an das System definiert und ausführlich beschrieben.
5. **Nichtfunktionale Anforderungen:** sind die funktionalen Anforderungen durch diejenigen Anforderungen erweitert, die keine funktionalen Anforderungen sind.
6. **Abnahmekriterien:** In diesem Kapitel werden die Kriterien festgelegt, die erfüllt sein müssen, um das Projekt als erfolgreich abgeschlossen zu betrachten.
7. **Projekt Meilensteine:** Hier werden die wichtigsten Meilensteine im Projektplan dargelegt.
8. **Referenzen:** In diesem Kapitel werden relevante Dokumente und Informationsquellen aufgeführt, die im Zusammenhang mit dem Projekt stehen.

Die Hauptfunktionen des zu entwickelnden Systems umfassen die Benutzerkontoerstellung und -authentifizierung, das Hinzufügen von Produkten zur Verfolgungsliste über URLs von beliebten E-Commerce-Plattformen, die Anzeige von Preisentwicklungen in grafischer Form und die Benachrichtigung der Benutzer bei festgelegten Preisbedingungen. Das System soll regelmäßig die Preise der verfolgten Produkte überprüfen und mit vorherigen Preisen vergleichen.

Mit diesem Pflichtenheft erhalten alle Projektbeteiligten einen klaren Überblick über die Anforderungen und Ziele des Projekts, um eine erfolgreiche Produktrealisierung zu ermöglichen.

## 2. Ausgangssituation und Ziele

In diesem Kapitel werden die Ausgangssituation und der Grund zur Wahl des Projektthemas anschaulich dargestellt. Es wird eine Einleitung zum Thema gegeben, gefolgt von einer Problemstellung, einer Beschreibung der beteiligten Stakeholder, dem Systemumfeld, den Rahmenbedingungen und schließlich den angestrebten Zielen.

### Einleitung

Dieses Kapitel beschäftigt sich mit der Ausgangssituation und den Zielen des geplanten Projekts, das eine Software zur Verfolgung von Produkt-Preishistorie und zum Empfang von E-Mail-Alarmen bei Preisänderungen entwickeln möchte. Es bietet eine Übersicht über das Problem und die Funktionalitäten, die durch das neue System gelöst bzw. erreicht werden sollen.

### Problemstellung (Funktionalität)

Aktuelle Lösungen ermöglichen es Verbrauchern möglicherweise nicht, die Preishistorie von Produkten auf verschiedenen Websites effizient zu verfolgen und zeitnahe Benachrichtigungen über Preisänderungen zu erhalten. Das neue System wird dieses Problem lösen, indem es registrierten Nutzern erlaubt, Produkte von verschiedenen Websites zu überwachen und bei Preisänderungen informiert zu werden.

### Stakeholder (Anwender):

Die Hauptzielgruppe für dieses System sind Verbraucher, die ihre Ausgaben optimieren und von Preisänderungen bei ihren bevorzugten Produkten profitieren möchten. Durch die Verwendung des neuen Systems können sie die Preisentwicklung von Produkten leicht verfolgen und schneller auf günstige Angebote reagieren.

### Systemumfeld (Einsatzumgebung)

Das neue System wird auf Desktop-Computern innerhalb eines Server-Client-Modells betrieben. Es wird mit Technologien wie Java, JavaFx, Spring Boot und MySQL entwickelt. Die Anwendung ist für den Einsatz in Windows konzipiert.

### Rahmenbedingung (Einschränkungen)

Das Projekt muss innerhalb der technischen Rahmenbedingungen und der vorgegebenen Systemarchitektur umgesetzt werden. Es ist möglich, dass die Unterstützung für bestimmte E-Commerce-Websites (Hepsiburada.com, trendyol.com, amazon.com.tr) eingeschränkt oder durch API-Nutzungsbedingungen und die Implementierung von Gegenmaßnahmen gegen Web-Scraping begrenzt ist. Das Projekt muss den Datenschutz und die informationelle Selbstbestimmung der Benutzer gewährleisten, etwa durch die Verarbeitung personenbezogener Daten gemäß den geltenden gesetzlichen Normen.

### Ziele (Lösung)

Das Hauptziel des Projekts ist die Entwicklung einer Verbrauchersoftware, die Benutzern ermöglicht, die Preishistorie von Produkten auf verschiedenen Websites zu verfolgen und bei Preisänderungen benachrichtigt zu werden. Qualitative Ziele sind die Benutzerfreundlichkeit, die Anpassungsfähigkeit an Windows Betriebssystem und die einfache Handhabung durch die Stakeholder. Quantitative Ziele umfassen die Genauigkeit der Preisüberwachung und die Schnelligkeit der Benachrichtigung bei Preisänderungen. Durch die Umsetzung dieser Ziele soll das System den Nutzern dabei helfen, informierte Kaufentscheidungen zu treffen und von Preisänderungen zu profitieren.

### 3. Gesamtarchitektur

#### Einleitung

In diesem Kapitel sind die Systemgrenzen definiert, die mit adäquaten Abbildungen präzisiert. Unter Berücksichtigung der Systemgrenzen sind per Anwender die externen Schnittstellen definiert (graphische Schnittstellen und API). Die notwendigen Systemkomponenten und Datenstrukturen sind definiert und modelliert. Dieses Kapitel führt zu dem besseren Verständnis des angeforderten Systems und dadurch genaue Definition der funktionalen und nicht-funktionalen Anforderungen.

#### Gesamtarchitektur

Die konzeptionelle Architektur besteht aus folgenden Hauptkomponenten:

- Benutzerschnittstelle (UI)
- Server (Backend)
- Datenbank

Die Benutzerschnittstelle (UI) besteht aus Login-, Registrierungs- und Hauptseite mit allgemeinen Funktionen. Der Server (Backend) ist für die Verarbeitung von Benutzeranfragen und die Kommunikation mit der Datenbank verantwortlich. Die Datenbank speichert Benutzerinformationen, verfolgte Produkte und Preishistorien.

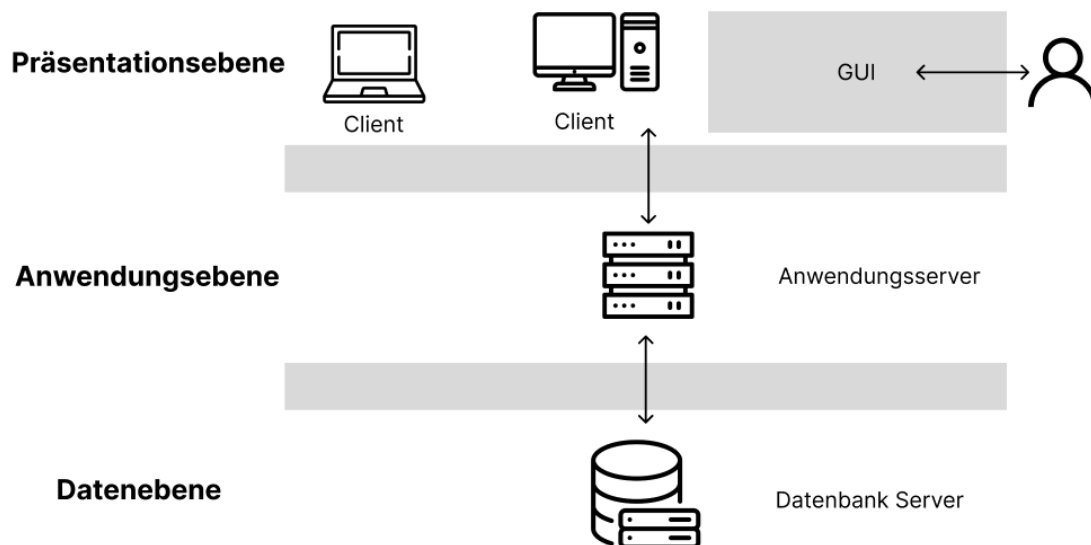


Figure 1 Gesamtarchitektur

#### Komponenten

##### Benutzerschnittstelle (UI)

Die Benutzerschnittstelle besteht aus den folgenden Teilen:

- Login-Seite: Ermöglicht Benutzern, sich mit ihrer E-Mail-Adresse und ihrem Passwort anzumelden.
- Registrierungsseite: Ermöglicht Benutzern, ein Konto mit Vor- und Nachnamen, E-Mail-Adresse und Passwort zu erstellen.
- Hauptseite: Zeigt die verfolgten Produkte, Preishistorie und ermöglicht Benutzern, Benachrichtigungseinstellungen festzulegen.

## Server (Backend)

Der Server ist für die Verarbeitung von Benutzeranfragen, die Kommunikation mit externen E-Commerce-Websites und die Datenbank verantwortlich. Er besteht aus den folgenden Hauptmodulen:

- **Kontoverwaltung:** Verwaltet Benutzerkonten, Anmeldung und Registrierung.
- **Produktverfolgung:** Verwaltet die verfolgten Produkte und Preisinformationen.
- **Benachrichtigungssystem:** Sendet Benachrichtigungen an Benutzer, wenn die festgelegten Bedingungen erfüllt sind.

## Datenbank

Die Datenbank speichert Benutzerinformationen, verfolgte Produkte und Preishistorien. Sie besteht aus den folgenden Haupttabellen:

- **Benutzer:** Speichert Benutzerinformationen wie Vor- und Nachname, E-Mail-Adresse und Passwort.
- **Produkte:** Speichert Informationen über verfolgte Produkte, einschließlich URLs und Benutzerzuordnungen.
- **Preishistorie:** Speichert Preisinformationen für jedes verfolgte Produkt im Laufe der Zeit.

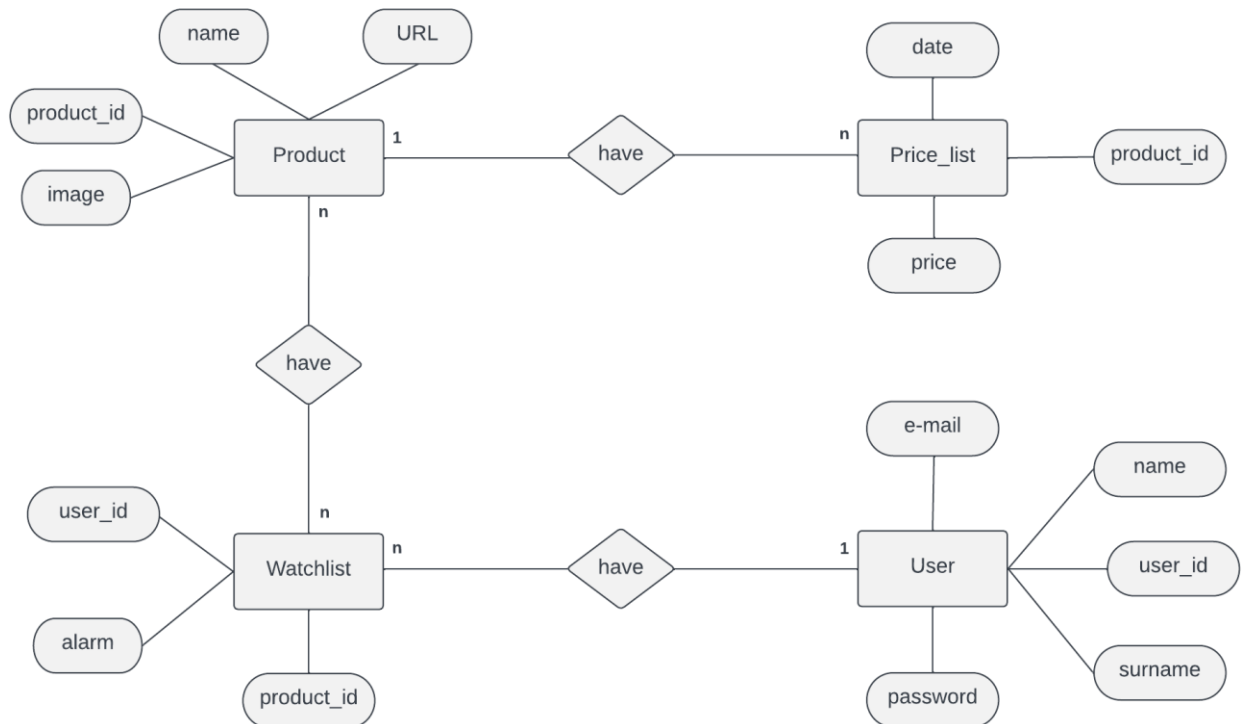


Figure 2 E/R Model für Datenbank

## Externe Schnittstellen

Die Anwendung kommuniziert mit externen E-Commerce-Websites wie Hepsiburada.com, trendyol.com und amazon.com.tr, um Produktdaten und Preisinformationen abzurufen.

## 4. Funktionale Anforderungen

### Einleitung

Die funktionalen Anforderungen definieren die Funktionalität des Systems aus Sicht der Nutzer. Sie werden in Form von Use Cases beschrieben, die die Interaktion zwischen Nutzern und dem System darstellen. Zusätzlich enthalten die funktionalen Anforderungen technische und fachliche Anforderungen sowie ein Datenmodell.

### UI Use Cases

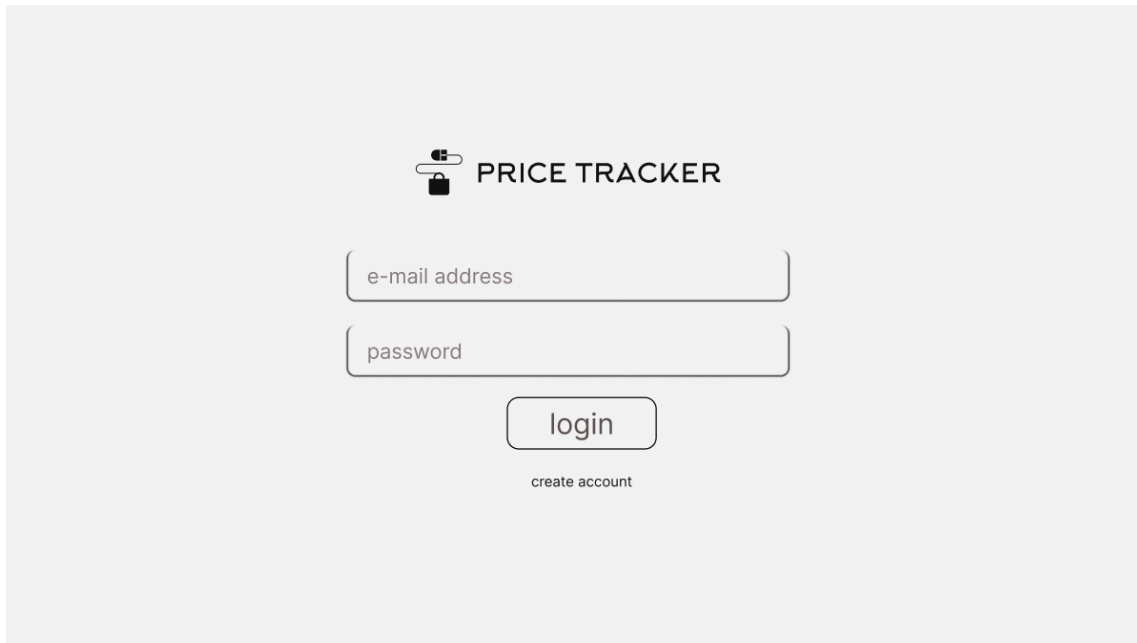


Figure 3 GUI Form 1

1. Wenn der Benutzer ein Konto auf dieser Seite hat, wird er durch die Eingabe seiner Informationen zum 3. Formular geleitet. Wenn kein Konto vorhanden ist, wird es mit der Schaltfläche Konto erstellen zum 2. Formular geleitet.

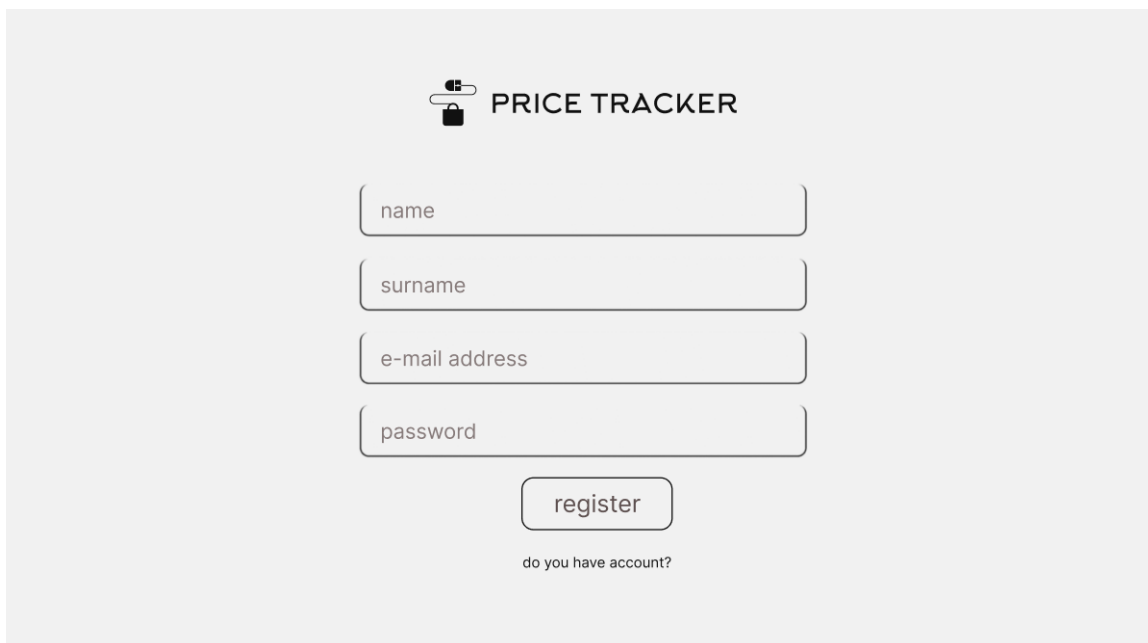


Figure 4 GUI Form 2



2. Der Benutzer erstellt ein neues Konto, indem er die erforderlichen Informationen auf dieser Seite ausfüllt. Er wird dann zum ersten Formular weitergeleitet, um sich anzumelden, und erhält eine Rückmeldung, dass sein Konto erstellt wurde.

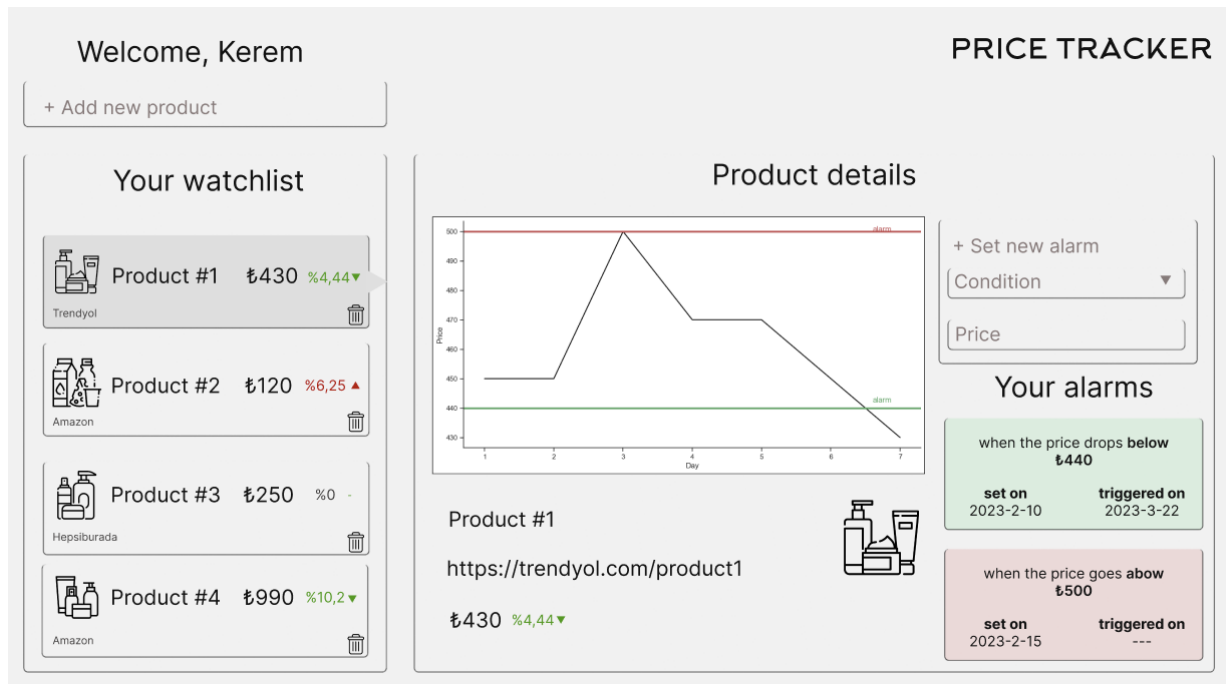


Figure 5 GUI Form 3

3. Auf dieser Seite können im linken Frame die Watchlist des Benutzerkontos, falls vorhanden, und die Informationen zu den darin enthaltenen Produkten angezeigt werden. Ansonsten werden ihm in diesem Abschnitt die Produkte mit Chancen angezeigt. Im Rahmen über diesem Teil befindet sich ein Eingabefeld zum Hinzufügen eines neuen Produkts. Der linke Rahmen besteht aus zwei Teilen. Die erste zeigt das Preisänderungsdiagramm und Produktdetails. Der zweite Teil zeigt die Alarmer und den Bereich zum Hinzufügen von Alarmen. Der Benutzer erhält visuelles Feedback für alle Löschungen und Hinzufügungen.

## API Use Cases

**Funktion:** Registrierung

**Eingaben:** Vorname, Nachname, E-Mail-Adresse, Passwort

**Verarbeitungsschritte:**

- Prüfe, ob die E-Mail-Adresse bereits in der Datenbank vorhanden ist.
- Wenn nicht, wird ein neuer Benutzer mit den eingegebenen Daten in der Datenbank erstellt.
- Der Benutzer wird zur Login-Seite weitergeleitet.
- Falls die E-Mail-Adresse bereits vorhanden ist, wird eine Fehlermeldung ausgegeben.

**Ausgaben:** Der Benutzer wird zur Login-Seite weitergeleitet oder eine Fehlermeldung wird angezeigt.

**Funktion:** Login

**Eingaben:** E-Mail-Adresse, Passwort

**Verarbeitungsschritte:**

- Prüfe, ob die E-Mail-Adresse und das Passwort in der Datenbank vorhanden sind.
- Wenn ja, wird der Benutzer zur Hauptseite weitergeleitet.
- Falls die E-Mail-Adresse oder das Passwort falsch sind, wird eine Fehlermeldung ausgegeben.

**Ausgaben:** Der Benutzer wird zur Hauptseite weitergeleitet oder eine Fehlermeldung wird angezeigt.

**Funktion:** Produkt hinzufügen

**Eingaben:** URL einer beliebigen Einkaufsseite

**Verarbeitungsschritte:**

- Prüfe, ob die URL bereits in der Datenbank vorhanden ist.
- Wenn nicht, wird ein neues Produkt mit den Daten aus der URL in der Datenbank erstellt.
- Das Produkt wird zur Watchlist des Benutzers hinzugefügt.

**Ausgaben:** Das Produkt wird zur Watchlist des Benutzers hinzugefügt.

**Funktion:** Preisverlauf anzeigen

**Eingaben:** Produkt aus der Watchlist

**Verarbeitungsschritte:**

- Wenn der Benutzer keine Watchlist hat, werden die in der Datenbank registrierten Opportunity-Produkte angezeigt.
- Wenn der Benutzer eine Watchlist hat Preisdaten des ausgewählten Produkts werden aus der Datenbank abgerufen.
- Die Preisentwicklung wird in einem grafischen Format angezeigt.

**Ausgaben:** Der Preisverlauf des ausgewählten Produkts wird angezeigt.

**Funktion:** Alarm festlegen

**Eingabe:** Benutzer-ID, Produkt-ID, Preislimit

**Verarbeitungsschritte:**

- Der Alarm wird für das ausgewählte Produkt mit dem angegebenen Preislimit in der Datenbank gespeichert.
- Das System überprüft regelmäßig den Preis des Produkts und sendet eine E-Mail an den Benutzer, wenn der Preis unter das Limit fällt.

**Ausgaben:** Der Alarm wird erfolgreich festgelegt und der Benutzer wird per E-Mail benachrichtigt, wenn das Preislimit unterschritten wird.

**Funktion:** Erstelle Alarm

**Eingaben:** Benutzer-ID, Produkt-ID, Bedingung und Ziel-Preis

**Verarbeitungsschritte:**

- Überprüfe, ob der Benutzer bereits einen Alarm für das Produkt und die Bedingung erstellt hat
- Füge einen neuen Alarm hinzu, falls keine Übereinstimmungen gefunden wurden, und speichere die Alarmdaten in der Datenbank

**Ausgaben:** Bestätigungsmeldung, dass der Alarm erfolgreich erstellt wurde, oder Fehlermeldung, wenn der Benutzer bereits einen Alarm für das Produkt und die Bedingung hat oder der Benutzer oder das Produkt nicht existieren.

**Funktion:** Entferne Produkt aus der Watchlist

**Eingaben:** Benutzer-ID und Produkt-ID

**Verarbeitungsschritte:**

- Überprüfe, ob der Benutzer existiert und das Produkt auf seiner Watchlist hat
- Entferne das Produkt aus der Watchlist des Benutzers

**Ausgaben:** Bestätigungsmeldung, dass das Produkt erfolgreich aus der Watchlist entfernt wurde, oder Fehlermeldung, wenn der Benutzer das Produkt nicht auf seiner Watchlist hat oder der Benutzer oder das Produkt nicht existieren.

### **Technischen und fachliche Anforderungen**

1. Das System muss auf einer Desktop-Umgebung laufen.
2. Die Anwendung muss als Server-Client-Modell realisiert werden.
3. Die Anwendung soll in Java und JavaFx entwickelt werden.

4. Die Backend-Logik soll mit Spring Boot umgesetzt werden.
5. Die Datenbank soll MySQL sein.

### Datenmodell

- User-Entität mit den Attributen name, surname, email und password
- Watchlist-Entität mit den Attributen Products (als Liste von Produkt-Entitäten), active\_alarms (als Liste von Alarm-Entitäten) und passive\_alarms (als Liste von Alarm-Entitäten)
- Alarm-Entität mit den Attributen product (als Produkt-Entität), date\_created, condition und target\_Price
- Produkt-Entität mit den Attributen Name, URL, Image und Prices (als Liste von Preis-Entitäten)

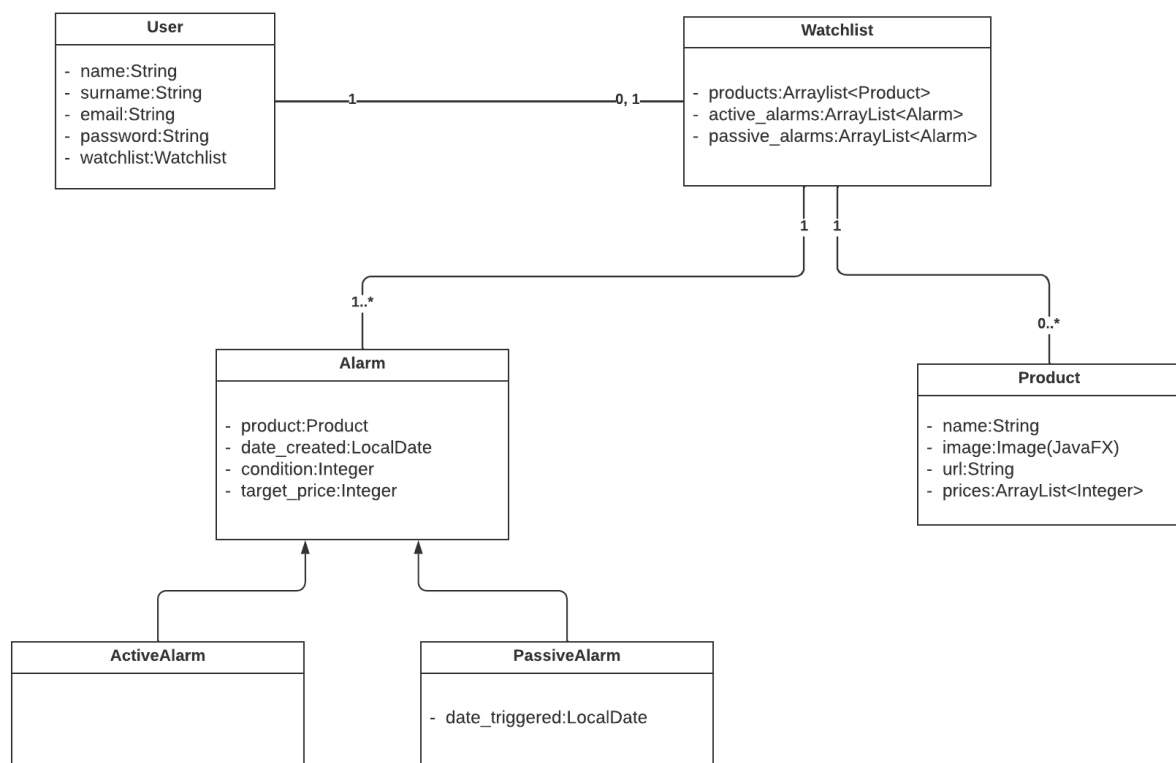


Figure 6 UML Diagram

## 5. Nichtfunktionale Anforderungen

### Einleitung

Dieser Abschnitt enthält die nicht-funktionalen Anforderungen für das Preisvergleichsprojekt. Nicht-funktionale Anforderungen beschreiben Qualitätsmerkmale des Systems wie Leistung, Sicherheit, Verfügbarkeit und Wartbarkeit.

### Nicht-funktionale Anforderungen an die Systemarchitektur (Architekturmuster, Deployment)

- Das System muss als Server-Client-Modell entwickelt werden, um eine einfache Bereitstellung und Wartung zu ermöglichen.
- Das System muss in der Java VM 1.8 laufen.
- Die Softwarearchitektur muss modular aufgebaut sein, um eine einfache Erweiterung und Anpassung an neue Anforderungen zu ermöglichen.

### Nicht-funktionale Anforderungen an die Entwicklungsumgebung:

- Die Entwicklungsumgebung sollte git für die Versionskontrolle unterstützen.
- Die Entwicklungsumgebung sollte eine automatisierte Build-Pipeline ermöglichen, um sicherzustellen, dass das System jederzeit korrekt kompiliert und getestet wird.
- Die Entwicklungsumgebung soll die Zusammenarbeit zwischen den Entwicklern erleichtern und eine effektive Versionskontrolle unterstützen.

### Nicht-funktionale Anforderungen an die Entwicklungswerkzeuge (Sprache, IDE, Frameworks)

- Die Entwicklungswerkzeuge sollen die Verwendung von Java, JavaFx, Spring Boot und MySQL ermöglichen.
- Die Entwicklungswerkzeuge sollen einfach zu konfigurieren und zu verwenden sein, um eine effiziente Entwicklung zu gewährleisten.
- Die Entwicklungswerkzeuge sollen eine effektive Debugging-Unterstützung bieten, um Fehler in 10 Minuten zu identifizieren und zu beheben.

### Nicht-funktionale Anforderungen an die Teststrategie (Qualitätssicherung)

- Es sollten sowohl Unit-Tests als auch Integrationstests implementiert werden, um eine hohe Codequalität sicherzustellen.
- Das Projekt sollte eine Testabdeckung von mindestens 90% aufweisen.

### Zusätzliche nichtfunktionale Anforderungen:

- **Ergonomie:** Die Benutzeroberfläche des Systems muss in deutscher Sprache gehalten sein und intuitiv bedienbar sein.
- **Zuverlässigkeit:** Die Verfügbarkeit des Systems muss bei 99,999% pro Jahr liegen.
- Das System muss innerhalb von 30 Sekunden auf Benutzeranfragen (z. B. für Preisaktualisierungen) reagieren können.
- Der Speicherbedarf des Systems im laufenden Betrieb darf 1024 MB nicht überschreiten.

## 6. Abnahmekriterien

- Benutzer können erfolgreich ein Konto erstellen und sich anmelden.
- Benutzer können Produkte hinzufügen, die sie verfolgen möchten, und der Preisverlauf wird korrekt angezeigt.
- Benachrichtigungseinstellungen funktionieren wie erwartet, und Benutzer erhalten Benachrichtigungen, wenn die festgelegten Bedingungen erfüllt sind.
- Das System überprüft erfolgreich die Preise der verfolgten Produkte und aktualisiert die Preisinformationen.

Das Projekt wird mit den folgenden Artefakten asbgegeben:

- Dokumentation:
  - Pflichtenheft: *INF202-Price-Tracker-Pflichtenheft-2023.v2.0.docx*
- Software
  - Link zu GitHub Projekt: <https://github.com/orgs/Gruppe1-Fulya/teams/keremundyavuz>

Anm.: Die Abgabetermine der Projetartefakts werden durch den Stakeholder festgelegt!

## 7. Projekt Meilensteine

**Folgende Meilensteine sind verbindlich definiert:**

- Meilenstein #1: Das Lastenheft ist fertiggestellt und mit dem Stakeholder abgestimmt.
- Meilenstein #2: Das Pflichtenheft ist fertiggestellt und mit dem Stakeholder abgestimmt.
- Meilenstein #3: (An diesem Meilenstein ist die Architektur im Vordergrund.)
  - Komponenten-basierte Architektur ist entworfen und komponentenweise implementiert.
  - Die externen Schnittstellen (Webservices) wurden entworfen/implementiert.
  - Die GUI-Komponente ist ansatzweise fertig.
- Meilenstein #4: (An diesem Meilenstein ist das Testen im Vordergrund.)
  - Die Test-Cases wurden aus den Use Cases abgeleitet und ansatzweise beschrieben.
  - Die Testumgebung ist vorbereitet und ein Smoke Test ist durchgeführt.
- Meilenstein #5
  - Das Projekt ist per Vereinbarung abgegeben.

## 8. Referenzen

Hepsiburada.com - <https://www.hepsiburada.com>  
Trendyol.com - <https://www.trendyol.com>  
Amazon.com.tr - <https://www.amazon.com.tr>  
Java - <https://www.java.com>  
JavaFX - <https://openjfx.io>  
Spring Boot - <https://spring.io/projects/spring-boot>  
MySQL - <https://www.mysql.com>