Florian Fritz

Lego-Set-Verwaltungssystem

Pflichten und Lastenheft

Florian Emilio Fritz
22.1.2025

Inhalt

1. Einführung

Das Projekt wird im Rahmen eines internen Entwicklungsprojekts durchgeführt.
Das Projekt soll die Verwaltung von Lego-Sets vereinfachen. Das Programm ist grundsätzlich für jeden Lego-Fan mit besonderer Zielgruppe von Sammlern, die ihre Sammlung/Inventar verwalten wollen, um eine besseren Übersicht zu haben. Durch eine zusätzliche Datenbank lassen sich Daten perfekt abspeichern.

2. Ist-Situation

Als großer Lego-Fan mit einem größeren Inventar kann es schnell passieren das man den überblick seiner Sets verliert. Hier kommt das Projekt zum Einsatz, es soll dem Nutzer die Verwaltung seiner Sammlung vereinfachen. Vorher haben viele es mit einer Tabelle oder auf Papier von Hand zu Fuß gemacht, was bei größeren Sammlungen schnell zu Fehlern führen kann. Dies soll das Programm erleichtern.

3. Soll-Situation

3.1 Soll-Zustand

Nach Abschluss des Projekts wird eine einfache Software bereitstehen, mit der Lego-Sammler ihre Sets verwalten können. Die Software soll den Wunsch erfüllen, alle Informationen zu einer Sammlung an einem zentralen Ort zu speichern und leicht abrufbar zu sein.
Mit der Anwendung können Nutzer ihre Sets erfassen, Informationen wie Name, Nummer, Thema und Preis(UVP) speichern sowie den Gesamtwert der Sammlung automatisch berechnen lassen. Durch die Anbindung an eine externe Datenbank (Rebrickable) wird es möglich sein, Sets direkt zu suchen und hinzuzufügen, was Zeit spart, und die manuelle Eingabe reduziert.
Die Vorteile der Software liegen vor allem darin, dass Sammler ihre Sammlung besser im Blick haben und schneller Änderungen vornehmen können.
Insgesamt bietet die Software eine sinnvolle Lösung, um die Verwaltung einer Lego-Sammlung zu erleichtern und übersichtlicher zu gestalten.

3.2 FunktionaleAnforderungen

Funktion	*Beschreibung*	*Aufwand (Stunden)*
Set hinzufügen	Nutzer können neue Lego-Sets manuell anlegen oder über eine API-Suche hinzufügen.	12
Set löschen	Sets können aus der Sammlung entfernt werden.	8
Sets durchsuchen	Sets per Nummer suchen	12
Inventarwert berechnen	Gesamtwert des Inventars Berechnen	8
Integration einer API	Integration von API(Rebrickable) für Daten abruf	20
Datenbankanbindung	Verbindung zur Datenbank zum Speichern und Abrufen von Sets.	15

3.3 Nicht-Funktionale Anforderungen

Funktion	*Beschreibung*	*Aufwand (Stunden)*
GUI erstellen	Eine Intuitive GUI erstellen	25

3.4 Schnittstellen

-Rebrickable API: Sucht Lego-Sets anhand eines Namens oder einer Nummer.

3.5 Risiken

Risiko	Verantwortlicher	Alternative Lösung
API nicht verfügbar	Entwickler	Manuelles Hinzufügen von Sets als Alternative anbieten. API Daten in der Datenbank speichern
Datenbankzugriffsprobleme	Entwickler	Lokale Backups erstellen

4. Abnahmekriterien

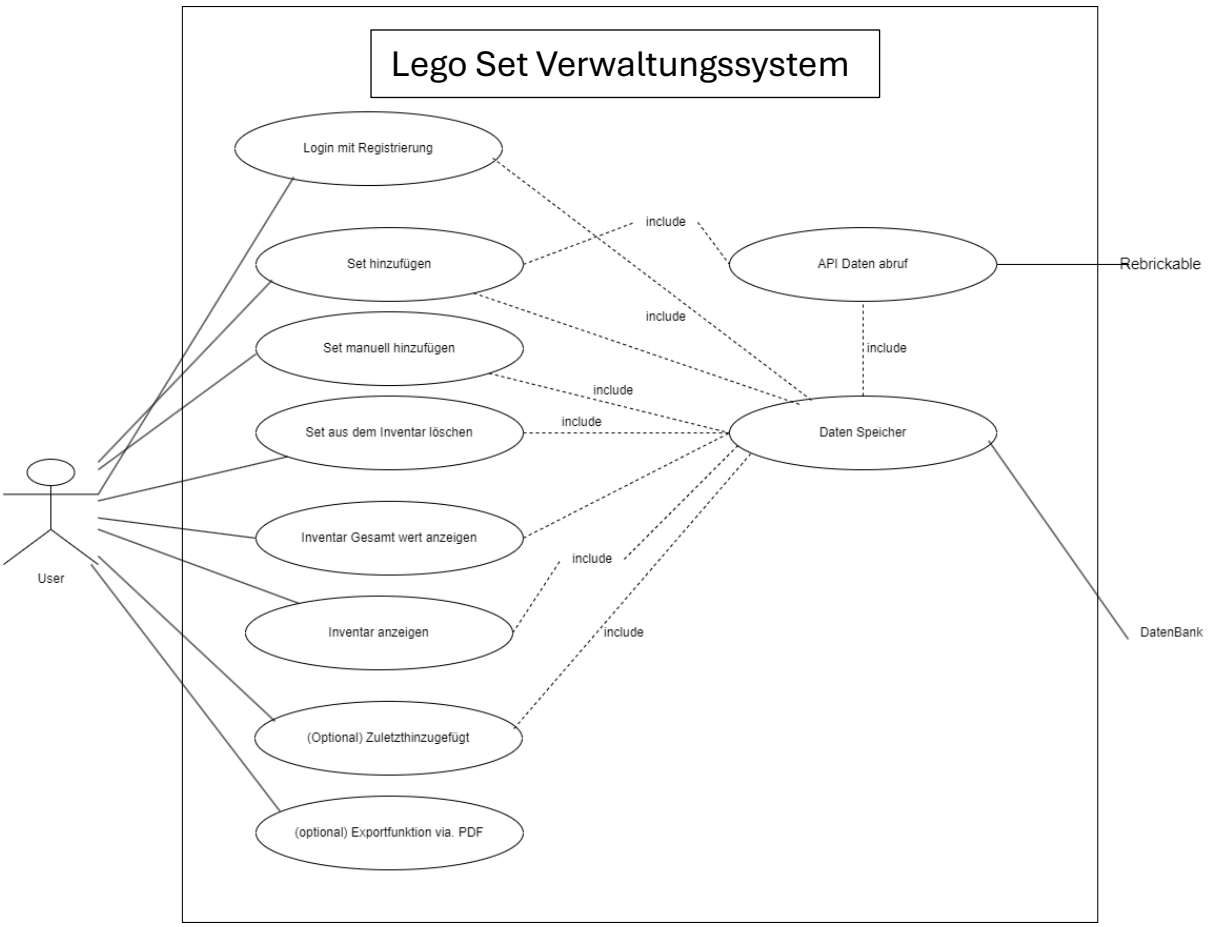
Muss-Kriterien

1. **Sets verwalten**
    - Sets können hinzugefügt und gelöscht werden.
2. **Inventarwert anzeigen**
    - Der Gesamtwert der Sammlung wird korrekt berechnet.
3. **Filterfunktion nach Nummer**
    - Sets können gezielt anhand ihrer Set-Nummer gefiltert werden.
4. **Benutzeroberfläche**
    - Die Software verfügt über eine funktionsfähige und intuitive grafische Benutzeroberfläche.
5. **Login Service**
    - Man kann sich registrieren und Anmelden

Kann-Kriterien

1. **Exportfunktion**
    - Die Sammlung kann als PDF exportiert werden.
2. **Zulätzt hinzugefügt**
    - Eine Fenster das die Zuletzt hinzugefügten Sets anzeigt
3. **Erweiterte Filterfunktionen**
    - Weitere Filtermöglichkeiten wie nach Namen, Thema oder Jahr.
4. **Dynamische GUI**
    - Die GUI passt sich unterschiedlichen Bildschirmgrößen an.
5. **Passwort zurücksetzen**
    - Das Passwort via. Email zurück setzen

5. Use-Case-Diagramme



Tabellarische Beschreibung der Use-Cases

Use Case	Beschreibung
Set hinzufügen	Der Benutzer fügt ein Lego-Set über die API-Suche hinzu. Hierfür werden die Daten über den Use-Case „API-Daten abrufen“ abgerufen und gespeichert.
Set manuell hinzufügen	Der Benutzer gibt die Informationen zu einem Set manuell ein, ohne die API zu nutzen. Die eingegebenen Daten werden über „Daten speichern“ gespeichert.
Set aus dem Inventar löschen	Der Benutzer kann ein Set aus seinem Inventar entfernen. Dabei werden die Änderungen über den Use-Case „Daten speichern“ in der Datenbank aktualisiert.
Inventar Gesamtwert anzeigen	Der Benutzer kann den Gesamtwert seines Inventars basierend auf den gespeicherten Sets anzeigen lassen.
API-Daten abrufen	Für „Set hinzufügen“ werden die Daten über die API-Schnittstelle abgerufen, um

	automatisch Informationen wie Name, Nummer und Preis einzutragen.
Daten speichern	Die zentralen Datenänderungen (Hinzufügen, Löschen, Bearbeiten) werden in der Datenbank gespeichert.
Login mit Registrierung	Man kann sich anmelden und Registrieren
(optional) zuletzt hinzugefügt	Man kann die letzten gespeicherten Einträge anschauen
Inventar Anzeigen	Man kann sich das Inventar in einer Tabelle anzeigen
(Optional) Exportfunktion via PDF	Der Benutzer kann seine Sammlung in einem PDF-Format exportieren.

Beziehungen zwischen Use-Cases

Beziehung	Beschreibung
Set hinzufügen => API-Daten abrufen	Set hinzufügen ruft immer die API-Daten auf, um die benötigten Set-Informationen zu laden.
API-Daten abrufen => Daten Speicher	API-Daten abrufen ruft Daten Speicher auf um die API daten langfristig zu speichern das bewirkt keine langen API-Abfragen so wie ein kleines Backup falls die API mal nicht verfügbar ist
Set hinzufügen => Daten speichern	Nach dem Hinzufügen eines Sets werden die Daten in der Datenbank gespeichert.
Set manuell hinzufügen => Daten speichern	Set manuell hinzufügen ruft Daten speichern auf, um die eingegebenen Daten dauerhaft zu speichern.
Set aus dem Inventar löschen => Daten speichern	Beim Löschen eines Sets aus dem Inventar wird Daten speichern verwendet, um die Änderungen zu aktualisieren.
Login mit Registrierung => Daten speichern	Login und Registrierung rufen Daten Speicher auf um bei Registrierung Daten zu speichern und beim Login Daten Vergleichen
Inventar anzeigen => Daten speichern	Um zu schauen welche Sets in seinem Inventar sind
Inventar Gesamtwert anzeigen => Daten speichern	Der Gesamtwert wird anhand der gespeicherten Daten in der Datenbank berechnet.

6. Projektplan

Gantt Diagramm Für Lego-Set-Verwaltungssystem																	
Projektplanung																	
Implementierung GUI																	
Implementierung Basisfunktionen																	
Implementierung Datenbank																	
Implementierung API																	
Paralleles Testen währen der Implementierung																	
Testen und Debugging																	
Exportfunktion (optional)																	
Puffer																	
Schreiben der Abschlussarbeit																	
	Woche 1	Woche 2	Woche 3	Woche 4	Woche 5	Woche 6	Woche 7	Woche 8	Woche 9	Woche 10	Woche 11	Woche 12	Woche 13	Woche 14	Woche 15	Woche 16	
			22.01.2025							02.04.2025	11.04.2025				05.05.2025	15.05.2025	
			Abgabe Pflichten Lastenheft							Erste Abschlussarbeit	Letzte Abschlussarbeit				Vorab-Abgabtermin Hausarbeit	Abgabe der Hausarbeit	

7. Produktumgebung

- **C#**:

  Das Projekt wird in C# programiert

- **WPF**:

  WPF wird benutzt, um eine benutzerfreundliche GUI zu schaffen

  Datenbank

- **SQLite**:

  Wird verwendet, um die Daten dauerhaft zu speichern.

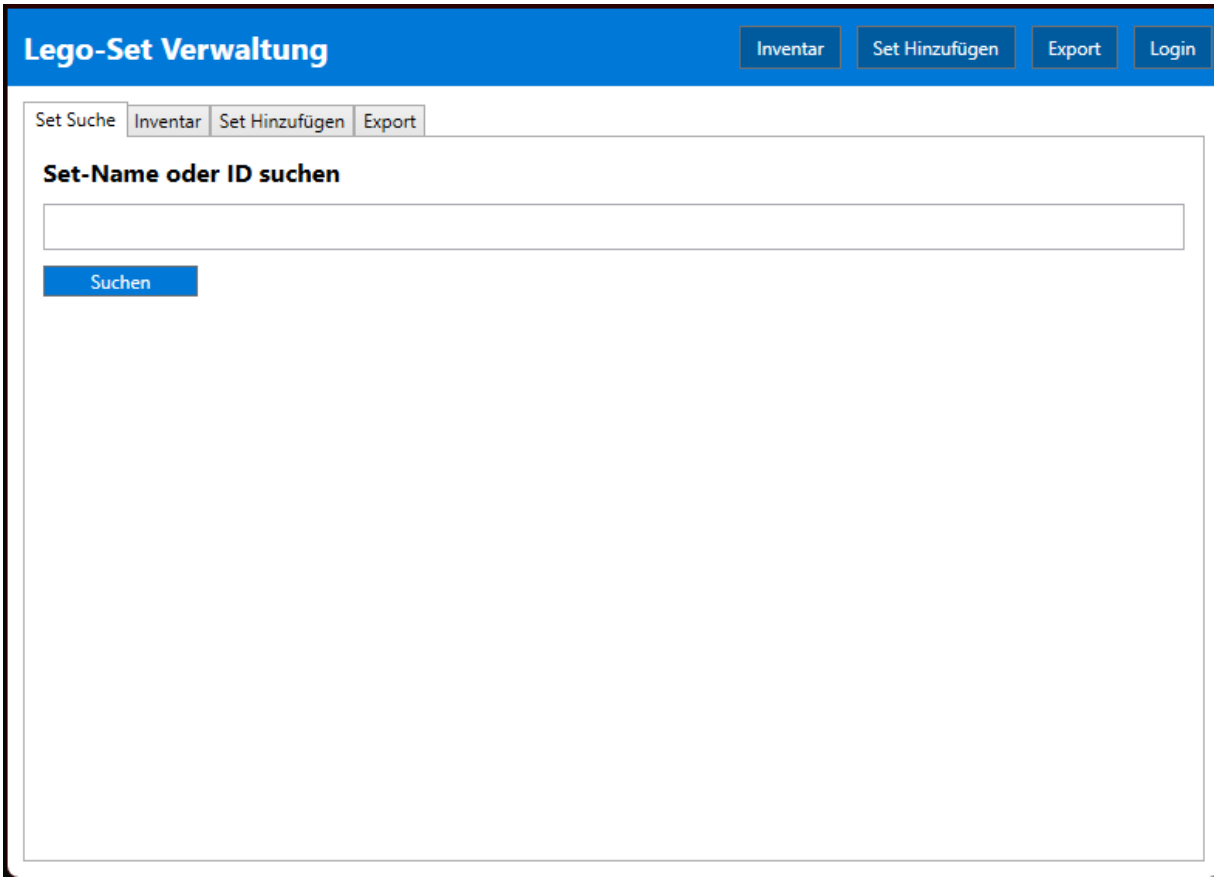
- **Rebrickable API**:

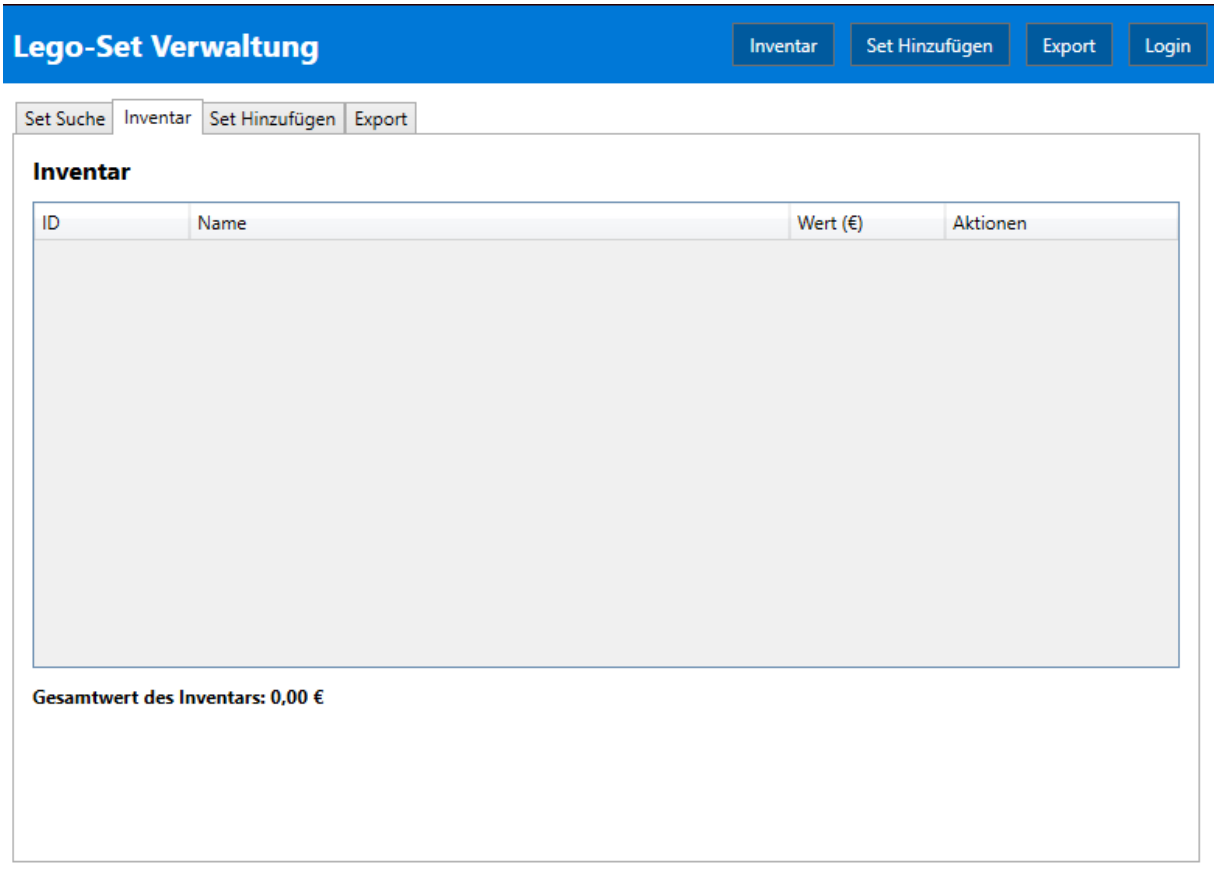
  Eine externe Schnittstelle, die genutzt wird, um Daten zu Lego-Sets wie Name, Nummer, Jahr und Thema abzurufen. Dies erleichtert das Hinzufügen von Sets und reduziert manuelle Eingaben.

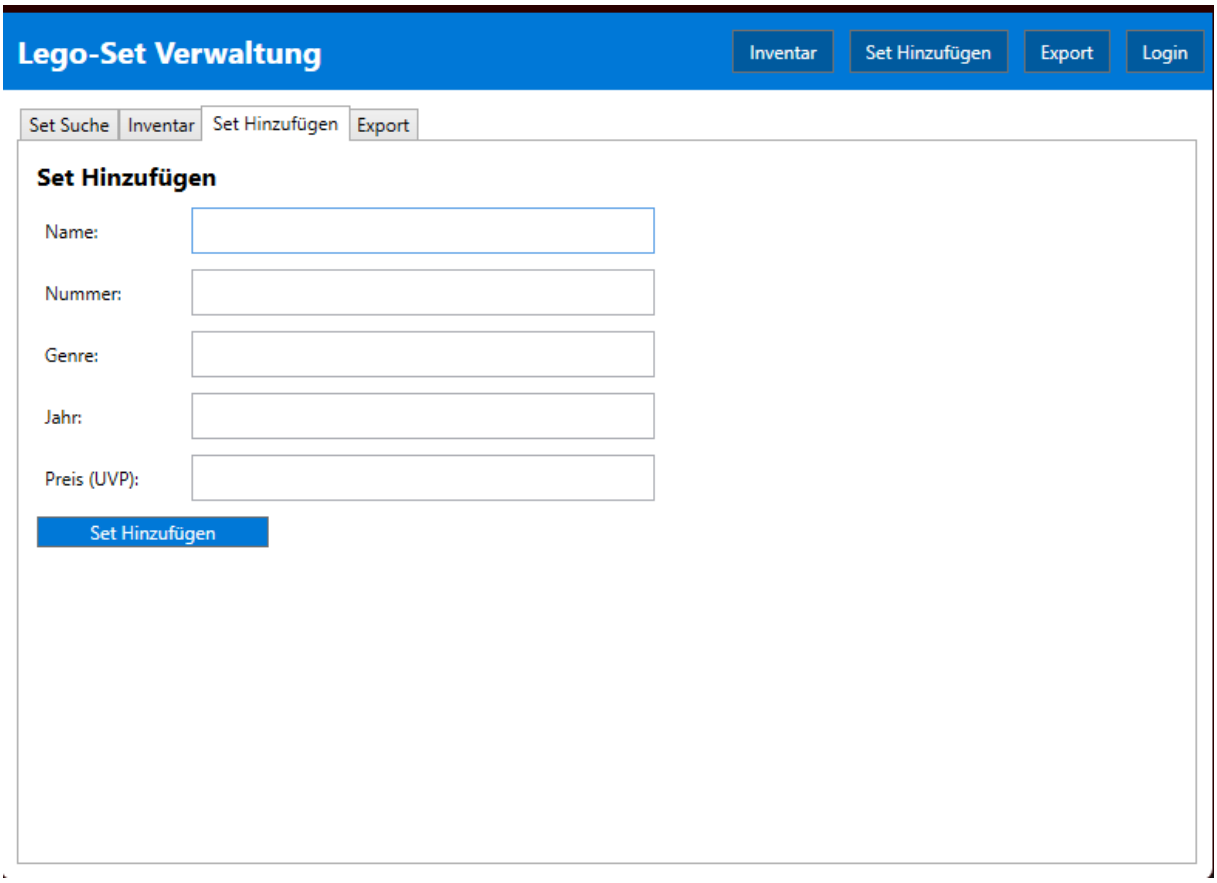
- **Visual Studio**:
  - Das ganze Projekt wird in Visual Studio geschrieben, getestet und gedebugget
- **Git/GitHub**:
  - Zur Versionierung des Projekts wird Git/GitHub benutzt. Das Projekt wird in einem GitHub-Repository gehostet.

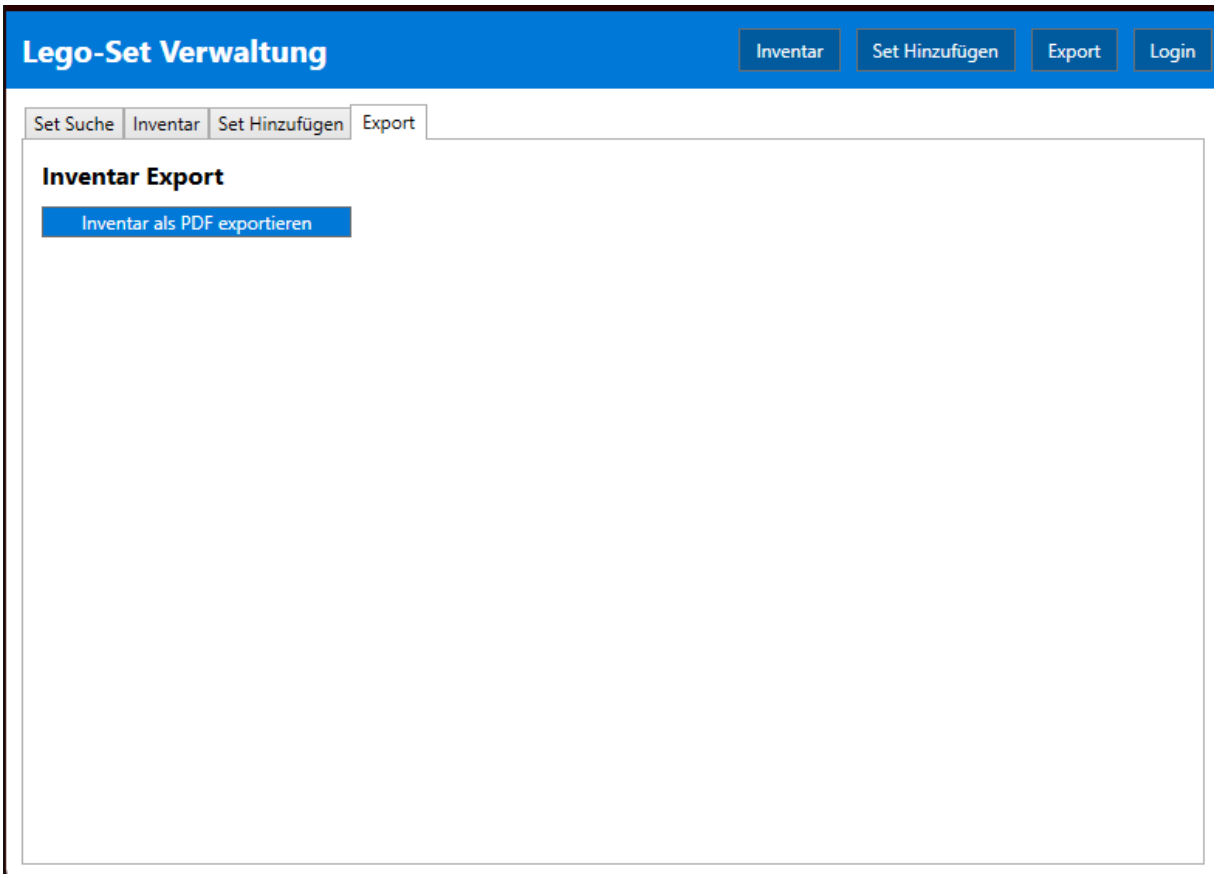
    .

8. Skizze von GUI oder Webseite

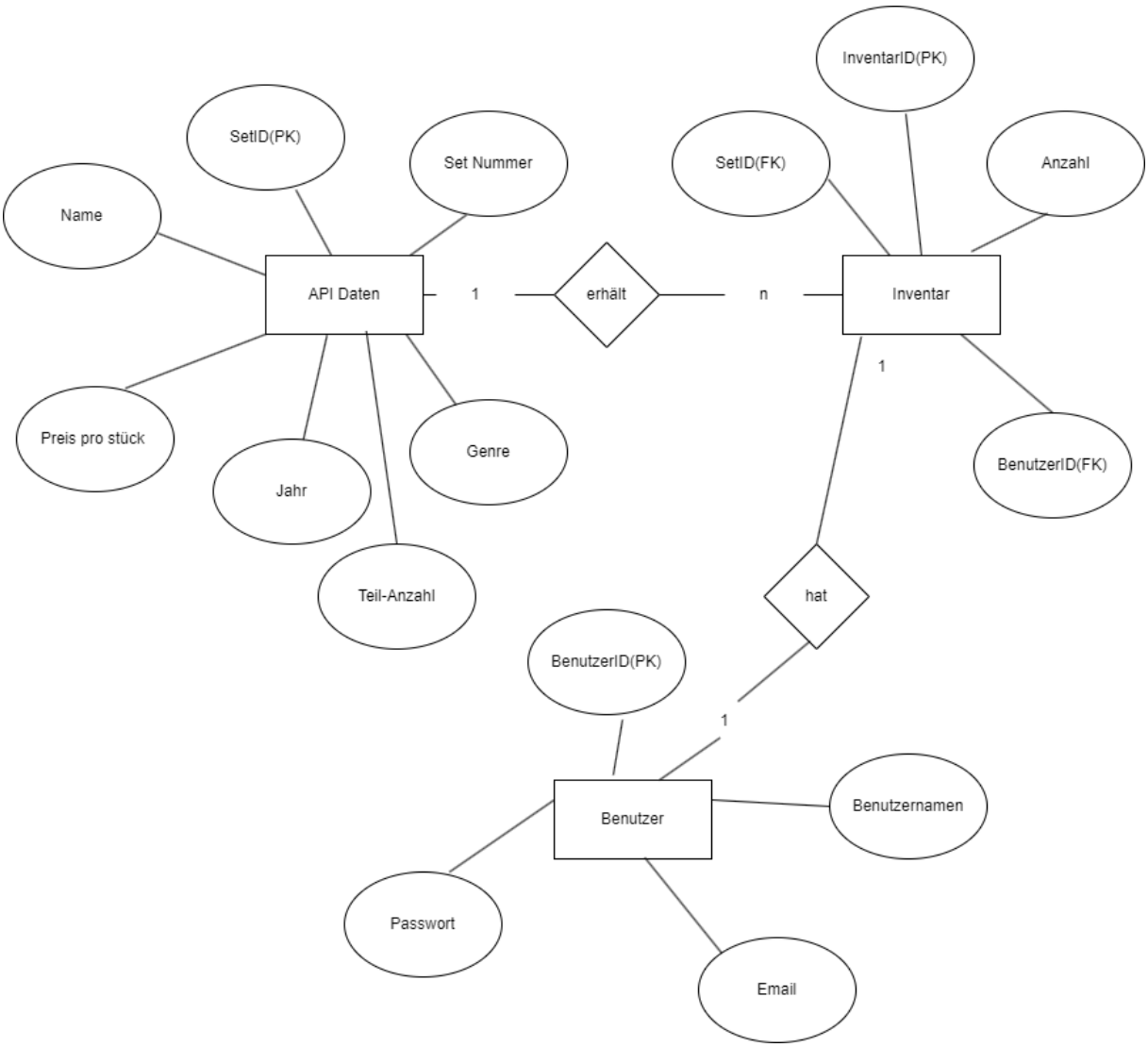





Lego-Set Verwaltung
Inventar
Set Hinzufügen
Export
Login
Set Suche
Inventar
Set Hinzufügen
Export
Set Hinzufügen
Name:
Nummer:
Genre:
Jahr:
Preis (UVP):
Set Hinzufügen


Lego-Set Verwaltung
Inventar
Set Hinzufügen
Export
Login
Set Suche
Inventar
Set Hinzufügen
Export
Inventar Export
Inventar als PDF exportieren

9. DB-Entwurf



.

10. Link zu einem gehosteten Git-Repository

https://github.com/FloFritz/Lego-Set-Verwaltungssystem

11. Testplan

ID	T01	☐
Beschreibung	Ein Benutzer registriert sich mit Benutzername, E-Mail und Passwort.	
Vorbedingung	Das Registrierungsformular ist geöffnet.	
Test-Schritte	1. Benutzer gibt gültigen Benutzernamen, E-Mail und Passwort ein.	
	2. Benutzer klickt auf „Registrieren“.	
Erwartetes Resultat	Der Benutzer wird erfolgreich in der Datenbank gespeichert und erhält eine Bestätigung.	

ID	T02	☐
Beschreibung	Ein Benutzer versucht, sich mit einem bereits registrierten Benutzernamen zu registrieren.	
Vorbedingung	Der Benutzername existiert bereits in der Datenbank.	
Test-Schritte	1. Benutzer gibt existierenden Benutzernamen ein.	
	2. Benutzer klickt auf „Registrieren“.	
Erwartetes Resultat	Das System gibt eine Fehlermeldung zurück: „Benutzername bereits vergeben.“	

ID	T03	☐
Beschreibung	Ein registrierter Benutzer loggt sich ein.	
Vorbedingung	Der Benutzer existiert in der Datenbank und ist registriert.	
Test-Schritte	1. Benutzer gibt Benutzername und Passwort ein.	
	2. Benutzer klickt auf „Login“.	
Erwartetes Resultat	Der Benutzer wird eingeloggt und das Inventar wird angezeigt.	

ID	T04	☐
Beschreibung	Ein Benutzer versucht, sich mit einem falschen Passwort einzuloggen.	
Vorbedingung	Der Benutzername existiert in der Datenbank, aber das Passwort ist falsch.	
Test-Schritte	1. Benutzer gibt Benutzernamen und falsches Passwort ein.	
	2. Benutzer klickt auf „Login“.	
Erwartetes Resultat	Das System gibt eine Fehlermeldung zurück: „Falsches Passwort.“	

ID	T05	☐
Beschreibung	Der Benutzer fügt ein Set manuell hinzu.	
Vorbedingung	Der Benutzer ist eingeloggt.	
Test-Schritte	1. Benutzer gibt Name, Nummer, Thema, Jahr und Preis ein.	
	2. Benutzer klickt auf „Hinzufügen“.	
Erwartetes Resultat	Das Set wird erfolgreich zum Inventar hinzugefügt.	

ID	T06	☐
Beschreibung	Ein Set wird aus dem Inventar gelöscht.	
Vorbedingung	Das Set existiert im Inventar.	
Test-Schritte	1. Benutzer wählt das Set aus.	
	2. Benutzer klickt auf „Löschen“.	
Erwartetes Resultat	Das Set wird aus der Datenbank entfernt und nicht mehr angezeigt	

ID	T07	☐
Beschreibung	Der Gesamtwert des Inventars wird berechnet und angezeigt.	
Vorbedingung	Es existieren mindestens zwei Sets mit Preis und Anzahl im Inventar.	
Test-Schritte	1. Benutzer öffnet die Inventarseite.	
	2. Benutzer überprüft den Gesamtwert.	
Erwartetes Resultat	Der Gesamtwert entspricht der Summe der Einzelwerte (Preis * Anzahl).	

ID	T08	☐
Beschreibung	Ein Set wird über die API hinzugefügt, einschließlich der Bild-URL.	
Vorbedingung	Die API liefert gültige Daten für das Set.	
Test-Schritte	1. Benutzer gibt die Set-Nummer ein.	
	2. System ruft Daten von der API ab.	
	3. Benutzer bestätigt das Hinzufügen.	
Erwartetes Resultat	Das Set, einschließlich der Bild-URL, wird in der Datenbank gespeichert.	

ID	T09	☐
Beschreibung	Benutzer sieht die letzten 5 hinzugefügten Sets auf der Startseite.	
Vorbedingung	Es existieren mindestens 5 Sets in der Datenbank mit Bild-URLs.	
Test-Schritte	1. Benutzer öffnet die Anwendung.	
	2. Benutzer prüft die Startseite.	
	3. Bilder und Namen der Sets werden angezeigt.	
Erwartetes Resultat	Die letzten 5 Sets werden korrekt angezeigt (Name und Bild).	

ID	T10 (optional)	☐
Beschreibung	Ein Benutzer setzt sein Passwort zurück.	
Vorbedingung	Der Benutzer existiert in der Datenbank.	
Test-Schritte	1. Benutzer klickt auf „Passwort vergessen“.	
	2. System sendet eine E-Mail mit einem Link.	
	3. Benutzer öffnet den Link und gibt ein neues Passwort ein.	
Erwartetes Resultat	Das Passwort wird aktualisiert und der Benutzer kann sich mit dem neuen Passwort einloggen.	

ID	T11 (optional)	☐
Beschreibung	Der Benutzer exportiert das aktuelle Inventar in eine PDF-Datei.	
Vorbedingung	- Der Benutzer ist eingeloggt.	
	- Es befinden sich Sets im Inventar.	
Test-Schritte	1. Der Benutzer öffnet die Inventarseite.	
	2. Der Benutzer klickt auf den Button „Exportieren als PDF“.	
	3. Das System generiert eine PDF-Datei mit den Details der Sets (z. B. Name, Anzahl, Preis).	
	4. Der Benutzer prüft, ob die PDF-Datei korrekt erstellt wurde und alle Informationen enthält.	
Erwartetes Resultat	Eine PDF-Datei wird erfolgreich erstellt und gespeichert. Die Datei enthält die Inventardetails in tabellarischer Form.	

ID	T12(optional)	☐
Beschreibung	Der Benutzer versucht, ein leeres Inventar als PDF zu exportieren.	
Vorbedingung	- Der Benutzer ist eingeloggt.	
	- Es befinden sich keine Sets im Inventar.	
Test-Schritte	1. Der Benutzer öffnet die Inventarseite.	
	2. Der Benutzer klickt auf den Button „Exportieren als PDF“.	
Erwartetes Resultat	Das System zeigt eine Meldung an: „Kein Inventar zum Exportieren vorhanden.“ Keine PDF-Datei wird erstellt.	

UnitTests

1. Registrierung

- **Ziel**: Sicherstellen, dass die Registrierung fehlerfrei funktioniert.
- **Testfälle**:
    - Erfolgreiche Registrierung eines neuen Benutzers.
    - Fehler bei der Registrierung mit bereits existierendem Benutzernamen oder E-Mail.

2. Login

- **Ziel**: Prüfen, ob Benutzer erfolgreich eingeloggt werden und Fehler korrekt behandelt werden.
- **Testfälle**:
    - Erfolgreicher Login mit gültigen Benutzerdaten.
    - Fehler bei falschem Passwort.
    - Fehler bei nicht existierendem Benutzer.

3. Inventarverwaltung

- **Ziel**: Sicherstellen, dass Sets korrekt hinzugefügt, gelöscht und angezeigt werden.
- **Testfälle**:
    - Hinzufügen eines neuen Sets, sowohl über die API als auch manuell.
    - Verhindern von doppelten Einträgen im Inventar.
    - Löschen eines Sets aus dem Inventar.

4. Berechnung des Inventarwerts

- **Ziel**: Überprüfen, ob der Gesamtwert des Inventars korrekt berechnet wird.
- **Testfälle**:
    - Korrekte Berechnung des Wertes bei mehreren Sets.

- Fehlerfreies Verhalten bei leerem Inventar.

5. (Optional) Anzeige der zuletzt hinzugefügten Sets

- **Ziel**: Sicherstellen, dass die letzten 5 hinzugefügten Sets korrekt angezeigt werden.
- **Testfälle**:
  - Anzeige der Sets in der korrekten Reihenfolge (nach Hinzufügedatum).
  - Korrekte Anzeige bei weniger als 5 vorhandenen Sets.

6. (Optional) Passwort zurücksetzen

- **Ziel**: Prüfen, ob Benutzer ihr Passwort sicher zurücksetzen können.
- **Testfälle**:
  - Senden einer E-Mail mit einem Reset-Link.