

LEGO GCT

Caner Kara, Dennis Behrendt, Sven Wolf, Tim Sibum, Hannes Scherer, Tim Turowski

7. Mai 2023

1 Änderungshistorie

Version	Datum	Autor	Änderungen
Version 1.0	23.04.2023	Tim Turowski	Git-Repository angelegt und Pflichtenheft Text-Dokument
Version 1.1	26.04.2023	Tim Silbum	ERM und UML Diagramm zur Datenstruktur eingefügt
Version 1.2	26.04.2023	Hannes Scherer	Frontend Prototypen entwickelt
Version 1.3	27.04.2023	Tim Silbum	Sequenzdiagramm, Zustandsdiagramm und GUI Skizze
Version 1.4	28.04.2022	Tim Turowski	Bilder ins Pflichtenheft eingefügt und Tabellen angepasst
Version 1.5	02.05.2023	Dennis Behrendt	Bezogen auf das erste Feedback: Inhalte korrigiert und Ausformulierungen angepasst
Version 1.6	03.05.2023	Hannes Scherer	Bezogen auf das erste Feedback: Frontend Prototypen angepasst
Version 1.7	06.05.2023	Tim Turowski	Nach Problemen beim mergen im Repository: Pflichtenheft ist eigenständiges Dokument
Version 1.8	07.05.2023	Tim Turowski	Inhalte der Projektmitglieder eingefügt für morgiges Meeting

Inhaltsverzeichnis

1	Änderungshistorie	2
2	Zielbestimmung	4
2.1	Musskriterien	4
2.2	Kannkriterien	4
2.3	Wunschkriterien	4
2.4	Abgrenzungskriterien	5
3	Produkteinsatz	5
4	Produktfunktionen	5
4.1	Benutzersicht	5
4.2	Produktfunktionen Backend	6
5	Produktdaten	7
5.1	System	7
5.2	Benutzeraccount	7
6	Produktleistungen	7
7	Benutzungsoberfläche	8
8	Nichtfunktionale Anforderungen	8
9	Technische Produktumgebung	9
10	Gliederung in Teilprodukte	9
10.1	Crawler	9
10.2	TeileFilter	10
10.3	Anleitungsparser	10
10.4	Anleitungslader	10
10.5	Datenzugriffsobjekt	10
10.6	Teilevergleich	10
11	Globale Testfälle	10

2 Zielbestimmung

Heutzutage günstig Lego-Sets zu erwerben kann auf mehreren Ebenen ein wohlüberlegtes Unterfangen sein. Welche der Lego-Händler bieten die Sets günstig an? Ist es günstiger sich die Einzelteile des Sets einzeln zu bestellen? Was ist wenn bestimmte Einzelteile nicht mehr verfügbar sind?

Das Ziel des Projekts ist genau diese Fragestellungen, durch die Entwicklung eines Tools zur Ermittlung der günstigsten Anbieter, zu beantworten. Das Tool soll dabei auch in der Lage sein, abzuwägen, ob es günstiger ist, das Set oder Einzelteile zu kaufen und zu kombinieren.

2.1 Musskriterien

1. Es soll eine Datenbank mit Stücklisten und Bauteilen geben
2. Die PDF-Bauanleitungen der Lego-Sets sollen mit OCR auslesbar sein
3. Ein Crawler bezieht die Preise der unterschiedlichen Händler zur Laufzeit der Suche
4. Ein Preisvergleich unter den Händlern, im Bezug auf Kauf eines Sets oder der jeweiligen Einzelteile, findet statt
5. Darstellung des Tools als graphische Oberfläche. (Startseite, Benuterverwaltung, Preisvergleich, Warenkorb)
6. Es soll möglich sein Benutzeraccounts anzulegen und diese zu verwalten
7. Drei feste Händler sollen beim Vergleich berücksichtigt werden
8. Auf nicht verfügbare Einzelteile sollte hinreichend hingewiesen werden.
9. Die Lieferkosten sollen beim Preisvergleich berücksichtigt werden.
10. Bei der Ausgabe des Vergleichs soll eine Verlinkung zum Produkt, sowie eine Liste der Bauteile, angezeigt werden.

2.2 Kannkriterien

1. Plattform sollte auch auf Mobilgeräten gut dargestellt sein
2. Mehr Händler sollten noch mit aufnehmbar sein

2.3 Wunschkriterien

1. Einzelteile die besonders selten/teuer sind sollten gesondert aufgelistet werden
2. Filterfunktion um zum Beispiel Figuren rauszufiltern
3. Sticker berücksichtigen
4. Eigene Bauanleitung auslesen lassen
5. Aktuelle Angebote hervorheben
6. Dashboard: Statistiken zum Suchverhalten unserer Benutzer / Wie erfolgreich war unser Programm?
7. Es sollte eine Funktion geben um den Warenkorb beim entsprechenden Händler automatisch befüllen zu lassen

2.4 Abgrenzungskriterien

1. Unser Produkt wird keine Verkaufsplattform haben
2. Keine Bevorzugung von Händlern
3. Wir berücksichtigen keine nicht offiziellen Klemmbausteine
4. Website ist nur für deutschsprachige Benutzer
5. Alte Sets ohne Stückliste in der Anleitung können nicht berücksichtigt werden (2006)

3 Produkteinsatz

Das Produkt soll Benutzer bei Kaufentscheidung unterstützen, außerdem unterstützt es die Benutzer Sets zu bauen die aus irgendwelchen Gründen nicht mehr verfügbar sind.

Das Tool ist für Lego-Enthusiasten und Sammler gedacht, die nach dem günstigsten Angebot suchen. Das Tool kann auch von Einzelpersonen oder Unternehmen genutzt werden, die große Mengen an Lego-Sets kaufen möchten.

4 Produktfunktionen

4.1 Benutzersicht

/F10/

Geschäftsprozess: Über eine Suchmaske können Lego Setnummern eingegeben

Vorbedingung: User befindet auf der Website des LegoGCT

Nachbedingung: Nach der Eingabe wird die Datenbank auf die Existenz des Lego Sets geprüft

Fehlerfall: Die Eingegebene Setnummer stimmt nicht mit einer in der Datenbank vorhandenen Setnummer überein. Oder Syntaxfehler führt zur falschen Eingabe

Anwender: Akteur im Kontext der Webapplikation

Beschreibung: Über die Suchfunktion kann der Benutzer eine Setnummer eingeben.

/F20/

Geschäftsprozess: Anzeigen der Historie

Vorbedingung: Benutzer sollte auf der Webplattform angemeldet sein

Nachbedingung: Benutzer kann sich seine persönliche Historie anzeigen lassen

Fehlerfall: Benutzer hat bisher noch keinen Suchvorgang gestartet

Anwender: Akteur im Kontext der Webapplikation

Beschreibung: Für Angemeldete Nutzer steht die Historie vergangener Suchen zur Verfügung. Diese unterstützt den Benutzer bereits gesuchte Sets wiederzufinden.

/F30/

Geschäftsprozess: Registrierung auf der Webplattform

Vorbedingung: User befindet auf der Website des LegoGCT und hat den Registrieren Button gedrückt

Nachbedingung: User konnte sich Erfolgreich registrieren lassen sein Benutzer Account wurde in einer Datenbank abgelegt

Fehlerfall: E-Mail ist bereits vergeben, Datenbank ist nicht ansprechbar, Passwortsicherheit zu gering

Anwender: Akteur im Kontext der Webapplikation

Beschreibung: Um die vollen Funktionsumfang der Webapplikation zu nutzen. Muss der Benutzer eine

Registrierung durchführen.

/F40/

Geschäftsprozess: Anmeldung auf der Webplattform

Vorbedingung: Benutzer befindet auf der Website des LegoGCT und ist ein Registrierter Nutzer

Nachbedingung: Benutzer konnte sich erfolgreich an Webplattform anmelden

Fehlerfall: Eine Anmeldung war nicht erfolgreich, weil Passwort und Benutzername nicht übereinstimmen oder der Benutzer noch keine Registrierung vorgenommen hat.

Anwender: Akteur im Kontext der Webapplikation

Beschreibung: Um die volle Funktion der Webapplikation nutzen zu können, kann sich der Benutzer an der Webplattform anmelden

/F50/

Geschäftsprozess: Darstellung der Stücklisten

Vorbedingung: Benutzer hat nach einer Setnummer gesucht, welche in der Datenbank vorhanden ist.

Nachbedingung: Benutzer bekommt eine Stückliste mit Einzelteilen angezeigt

Fehlerfall: Stückliste konnte nicht dargestellt werden, da zu viele Einzelteile bei Händler nicht zur Verfügung stehen.

Anwender: Akteur im Kontext der Webapplikation

Beschreibung: Dem Benutzer wird eine Stückliste des Legosets ausgegeben nach dem der Benutzer gesucht hat. Die Stückliste enthält die Einzelteile mit folgenden Attributen Einzelteilnummer, Anzahl, Preis, URL

/F60/

Geschäftsprozess: Minimieren der Stücklisten

Vorbedingung: Benutzer bekommt eine Stückliste mit Einzelteilen angezeigt

Nachbedingung: Benutzer bekommt nur noch den Endpreis angezeigt

Fehlerfall: keinen

Anwender: Akteur im Kontext der Webapplikation

Beschreibung: Der Benutzer möchte eine übersichtlichere Anzeige haben und minimiert deshalb die Stückliste, um sich den Endpreis übersichtlicher darstellen zu lassen.

/F70/

Geschäftsprozess: Anzeigen des Endpreises

Vorbedingung: Benutzer

4.2 Produktfunktionen Backend

PDFs mit Crawler ansprechen können

PDFs mittels OCR auslesen

Einzelteile in Datenbank zwischenspeichern

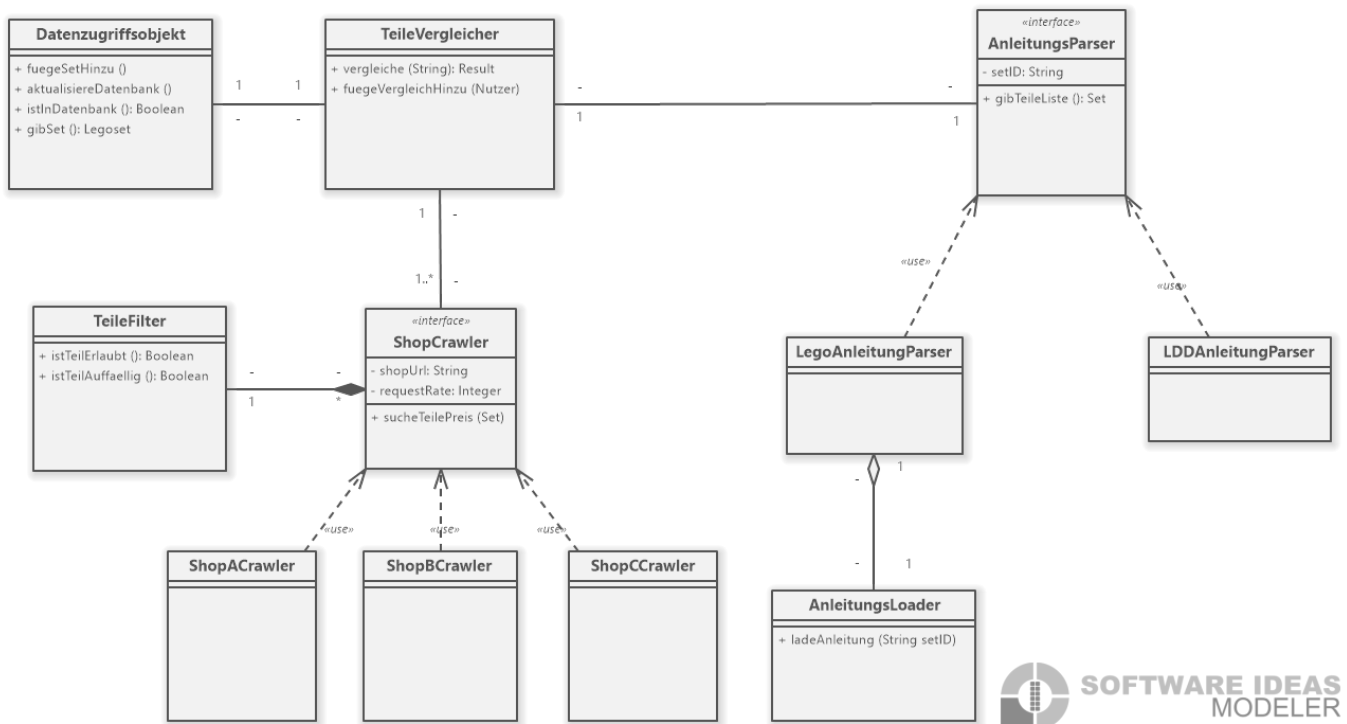
Informationen bei Händlern abfragen können

Preisvergleich zwischen Sets und Einzelteilen

Darstellung der Endergebnisse

5 Produktdaten

5.1 System



5.2 Benutzeraccount

Benutzername
Passwort (verschlüsselt)
List: Suchhistorie

6 Produktleistungen

Bestandsdatenbank soll aufgebaut werden mit Informationen aus den Anleitungen

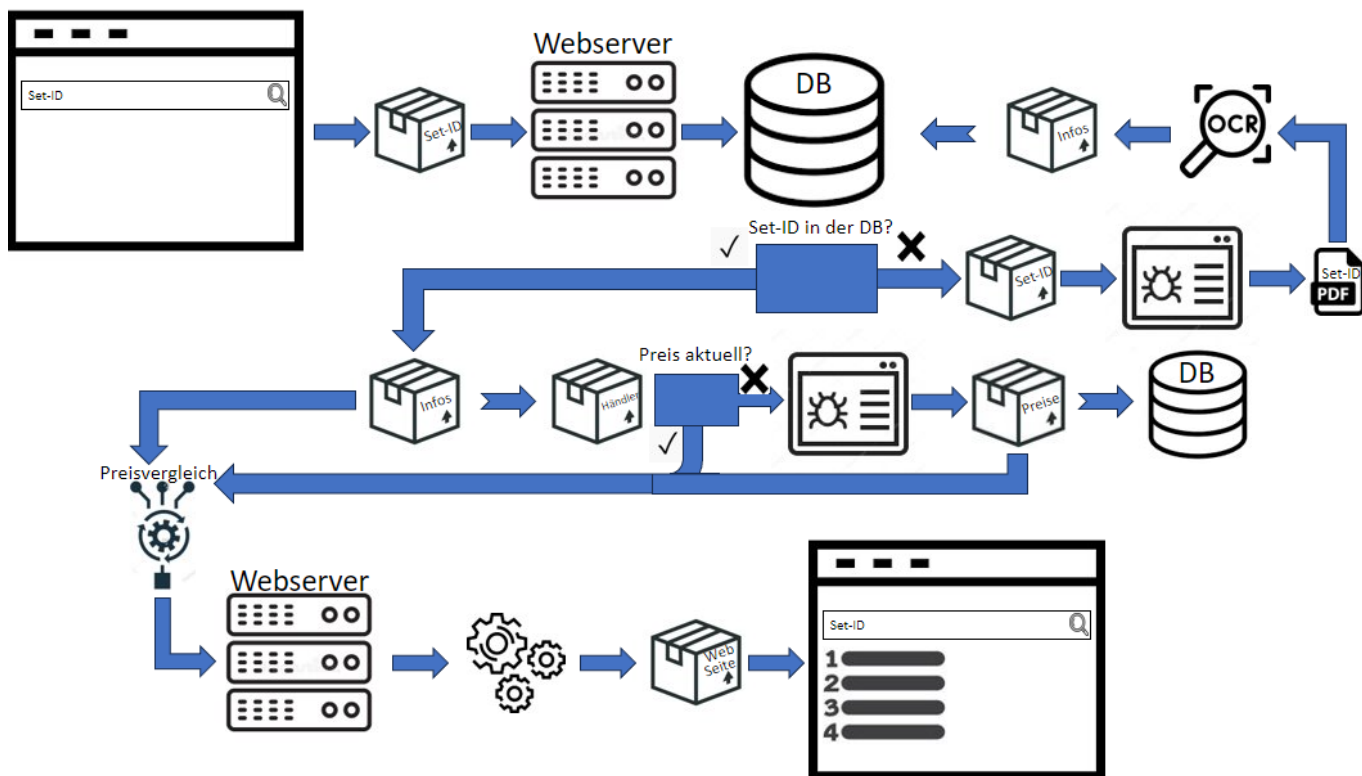
Preise sollen bei der Suche gecrawlt werden

Die verwendete Datenbank muss in der Lage sein große Datenmengen zu speichern und zu verarbeiten
Der Crawler soll in der Lage sein schnell Preisdaten bei den Anbietern zu crawlen, außerdem soll ein Algorithmus regelmäßig die neuen Anleitungen erkennen und dem Parser übergeben

7 Benutzungsoberfläche



8 Technische Produktumgebung



Eine Datenbank wird mithilfe von SQLAlchemy in Python erstellt und aufgesetzt. Dies läuft auf einem Server mit xB Speicher, damit die ganzen Daten gespeichert werden können
Es wird 1/2 Crawler erstellt in Python, die auf dem Server laufen. Diese dienen zur Suche von Preisen, so wie zur Suche der richtigen PDF

Der Webserver ist jederzeit ansprechbar, damit Benutzer auf unser Tool jederzeit zugreifen können OCR wird zum Auslesen der PDF benutzt, damit die jeweilige Stückzahl und ID der Einzelteile herausgefunden werden kann

Angular dient als Frameworkerstellungstool für eine interaktive Front End

HTML und CSS zur Aufbesserung des Framework

Python als primäre Programmiersprache, für alle Bereiche, wo programmiert werden soll

Tex zur gemeinsamen Dokumentation

Git zum einfachen Austausch von Fortschritt und jegliche andere Projekt relevante Erzeugnisse

9 Gliederung in Teilprodukte

9.1 Crawler

Die Klassen, welche das ShopCrawler Interface implementieren, besitzen die Logik einen bestimmten Shop zu crawlen. Für jeden berücksichtigten Shop wird eine Klasse benötigt, welche das Interface ShopCrawler implementiert. Durch die Verwendung des Interfaces ist es möglich von unterschiedlichen Anbietern Webseiten die gecrawlten Ergebnisse in der gleichen Struktur zurückzugeben. Es ist möglich einen ShopCrawler ein Objekt der Klasse Filter zu übergeben, welcher es ermöglicht, beim Crawlen nach bestimmten Eigenschaften die Daten zu filtern.

9.2 TeileFilter

Der Teile Filter verfügt über Funktionen, welche auf die gecrawlten Daten angewendet werden und prüft, ob die Daten auf die Filtereigenschaften zutreffen.

9.3 Anleitungsparser

Das Interface Anleitungsparser liefert eine Struktur eines Parsers für Bauanleitungen. Dadurch ist gewährleistet, dass die Klassen, die das Interface implementieren die geparschte Anleitung in der richtigen Struktur zurückgeben. Die jeweiligen Klassen der Parser müssen zusätzlich eine OCR-Bibliothek einbinden, welche das Auslesen von PDF Dateien ermöglicht.

9.4 Anleitungslader

Für das Parsen einer Anleitung muss die Anleitung aus dem Internet heruntergeladen werden, dies ermöglicht der Anleitungslader. Mit einem Web-Crawler findet er die PDF-Datei zu einer angegebenen SetID und lädt diese herunter und stellt sie einer Anleitungsparser Implementierung zur Verfügung.

9.5 Datenzugriffsobjekt

Das Datenzugriffsobjekt ermöglicht den Zugriff auf die Datenbank. Es besitzt die nötigen Funktionen die notwendigen Objekte auf der Datenbank zu persistieren oder auf der Datenbank persistente Daten abzurufen.

9.6 Teilevergleich

Teilevergleich ist die Zentrale Klasse im System. Die Klasse besitzt die Funktion eine Suche zu starten und sie bündelt alle anderen Objekte im System. Sie bestimmt den logischen Ablauf der Ausführung der Suche.

10 Globale Testfälle

/T010/

Set suchen: Der Nutzer gibt in der Suchleiste die Lego-Setnummer 75355 ein.

/T020/

Historie anzeigen: Der Nutzer klickt auf "Historie" um sich seine persönliche Such-Historie anzeigen zu lassen.

/T030/

Registrieren: Der Nutzer klickt auf "Registrieren" und registriert sich mit den Daten

- Name: Max Mustermann

- E-Mail: mustermann@gmx.de

- Passwort: password

/T040/

Anmelden: Der Nutzer klickt auf "Anmelden" und meldet sich mit den in /T30/ benannten Daten an.

/T050/

Stückliste anzeigen: Der Nutzer lässt sich die Stückliste anzeigen ?

/T060/

Kommt aufs Design der Nutzeroberfläche an ?

/T100/

PDF crawlen: Der Nutzer gibt die Set-Nummer 75192 in die Suchleiste ein, welche noch nicht in der Datenbank liegt. Der Web-Crawler sucht die jeweilige PDF Anleitung des Lego Sets.

/T110/

PDF auslesen: Nach /T100/ wird das entsprechende PDF-Dokument mittels OCR ausgelesen.

/T120/

Einzelteile in Datenbank speichern: Nach /T110/ werden die Nummern der Einzelteile korrekt in die Datenbank gespeichert.

/T130/

Preise crawlen: Nach /T120/ werden die Preise auf den Händlerseiten je nach Set und Einzelteilen abgefragt und gespeichert.

/T140/

Preise vergleichen: Nach /T130/ findet ein Preisvergleich zwischen den Sets und Einzelteilen ab. Dies erfolgt für alle erfassten Händler.

/T150/

Darstellung des Ergebnis-Fensters: Nach der Funktion /F100/ wird das nächste Fenster mit den entspre-

chenden Ergebnissen nach /F110/ - /F140/ angezeigt.

/T160/

Benutzer verwalten: Der Admin löscht den Nutzer Max Mustermann aus /T030/.

/T170/

Statistiken anzeigen: Der Admin Maxine Mustermann klickt auf die Einstellung "Dashboardüm sich Statistiken zu den Suchanfragen anzeigen zu lassen.