

Dokumentation

Shared-Shopping-List-System

Wirtschaftsinformatik und digitale Medien

Betreuende Professoren: Prof. Dr. Thies, Prof. Dr.-Ing. Christoph Kunz

Sommersemester 2019

Hochschule der Medien, Stuttgart

Scart



Team 03

Student	Matrikelnummer
Tom Hager	35353
Julian Hofer	32387
Bastian Tilk	35200
Patrick Lehle	34332
Marco Dell'Oso	32615
Van Duy Ho	33046



Inhaltsverzeichnis:

<i>Einleitung</i>	3
Abbildungsverzeichnis	4
Tabellenverzeichnis	5
Abkürzungsverzeichnis	6
Anforderungsanalyse	6
Aufgabenstellung des Projekts	6
Anforderungen an das System	9
Umfang	9
Mussfunktionen	9
Wunschk funktionen	11
Abgrenzungsfunktionen	11
Begriffslexikon	12
Entwurf	13
Use-Case Diagramme	13
Präsentationsschicht	26
Mockups	26
Klassendiagramme	28
Applikationsschicht	29
Klassendiagramme	29
Datenbankschicht	30
Entity-Relationship-Model	30
Klassendiagramme	32
Implementierung	32
Google App Engine	32
Google Cloud SQL	33
Git - Source Code Management	33
Package-Struktur	33
Präsentationsschicht	34
Klassendiagramme	34
Applikationsschicht	35
Klassendiagramme	35
Datenbankschicht	46
Entity-Relationship-Model	46
Klassendiagramme	47
Test	49
Präsentationsschicht	49
Applikationsschicht	49
Datenbankschicht	49
Ratgeberliteratur	49



Benutzungshandbuch	49
Login	50
Editor:	52
Profil	53
Gruppe	54
Einkaufsliste	56
Report	58
Installationshandbuch	63
Anhang	63
Quellcode inkl. JavaDoc	63
Literaturverzeichnis	63

Einleitung

Die folgende Dokumentation, basiert auf dem Modul "Software-Praktikum" im Sommersemester 2019, an der Hochschule der Medien. Das Projekt wurde unter der Betreuung von Prof. Dr. Peter Thies und Prof. Dr.-Ing. Christoph Kunz begleitet. Das Modul dient zur praxisorientierten Erkenntnisvermittlung, die sich aus den erlernten Grundlagen aus den vorherigen Semestern zusammensetzt. Das Projekt wurde in 5 vordefinierten Meilensteine aufgegliedert.

Die Anforderungen für die Umsetzung des Zielsystems wird in einem Anforderungskatalog der Professoren ausführlich beschrieben. Die Entwicklungsprozesse basieren auf dem IEEE Standards. Im weiteren Vorgehen werden Anwendungsfalldiagramme und Klassendiagramme erstellt, welche das System modellhaft abbilden. Durch ein Entity-Relationship-Model wird die Datenbankstruktur verdeutlicht. Das Abgebildete System basiert auf der 3-Tier-Architektur. Diese wären:

- Präsentationsschicht
- Applikationsschicht
- Datenbankschicht.

Die Präsentationsschicht besteht dabei aus zwei Clients. Den Editor und den Report-Generator. Die Entwicklung des Zielsystems erfolgt mit der objektorientierten Programmiersprache Java. Eclipse wird hier als Entwicklungsumgebung verwendet.

Mit der Plattform GitHub wird der Code-Entwicklungsprozess koordiniert. Die **Grafische Oberfläche** wird ausschließlich mit dem WebFramework **Google Web Toolkit** realisiert.

Das Projekt wird in ein **Google App Engine Projekt** konvertiert und anschließend mittels **Google Cloud Service** für die Online Nutzung deployed. Ziel dieses Projektes ist, ein funktionsfähiges **Shared-Shopping-List-System** mit allen Anforderungen bereitzustellen. Die entwickelte Software ist durch eine URL abrufbar. Zur Hilfestellung wird ein Installationshandbuch sowie ein Benutzerhandbuch bereitgestellt.



Abbildungsverzeichnis

Abbildung 1: Scart - Struktur des Zielsystems	7
Abbildung 2: Scart - Struktur des Zielsystems	7
Abbildung 3: Use-Case : Ein und Ausloggen	13
Abbildung 4: Scart - Use-Case: Einkaufsliste verwalten	15
Abbildung 5: Scart. - Use-Case: Report-Generator	20
Abbildung 6: Scart - Use-Case: Gruppe verwalten.....	22
Abbildung 7: Scart - Use-Case: Durchsuchen der Einkaufsliste.....	25
Abbildung 8: Editor: Nutzer dem System hinzufügen.....	26
Abbildung 9: Editor: Homepage von Scart	27
Abbildung 10: Editor: Gruppe mit deren Einkaufslisten und Usern	27
Abbildung 11: Report: Report-Generator.....	28
Abbildung 12: Klassendiagramm Applikationslogik - Server Package	29
Abbildung 13: Klassendiagramm Applikationslogik - Shared Package	30
Abbildung 14: Entity-Relationship-Model	31
Abbildung 15: Klassendiagramm - Entwurf Mapper Klassen	32
Abbildung 16: Package Struktur.....	33
Abbildung 17: Klassendiagramm - GUI und Client Package	34
Abbildung 18: shared/server Zusammenhang	35
Abbildung 19: Business Object Klassen: shared.bo	38
Abbildung 20: Business Object Klassen: GroceryListArticle	39
Abbildung 21: Business Object Klassen: GroupUser	39
Abbildung 22: Klassendiagramm - server.....	44
Abbildung 23: Klassendiagramm - shared	45
Abbildung 24: Klassendiagramm - shared.report	45
Abbildung 25: Entity-Relationship-Model	46
Abbildung 26: Klassendiagramm - Datenbank-Mapper Serverseitige Klassen	47
Abbildung 27: Scart - Login.....	50
Abbildung 28: Scart - Google Login.....	51
Abbildung 29: Scart - Registrierungsformular	51
Abbildung 30: Scart - Homepage	52
Abbildung 31: Scart - Menüleiste.....	52
Abbildung 32: Scart - Nutzerprofil	53
Abbildung 33: Scart - Profiländerung	53
Abbildung 34: Scart – deine Gruppen	54
Abbildung 35: Scart – Gruppenmenü.....	54
Abbildung 36: Scart – Gruppe verlassen	55
Abbildung 37: Scart – Einkaufsliste von Scart.....	56
Abbildung 38: Scart – Einkaufsliste mit Favoriten und gekauftem Artikel	57
Abbildung 39: Scart – Löschen der Einkaufsliste	58
Abbildung 40: Scart - Report Startbildschirm	58
Abbildung 41: Scart - ReportFilterForm	59
Abbildung 42: Scart - Deaktivierter Filter.....	59
Abbildung 43: Scart - Aktivierter Filter.....	59
Abbildung 44: Scart - Report nach Gruppe gefiltert	60
Abbildung 45: Scart - Report nach Einzelhändler gefiltert.....	60



Abbildung 46: Scart - Report nach Zeitraum gefiltert.....	61
Abbildung 47: Scart - Report nach allen Möglichkeiten gefiltert	62
Abbildung 48: Scart - Editor	62

Tabellenverzeichnis

Tabelle 1: Abkürzungsverzeichnis	6
Tabelle 2: Begriffslexikon.....	12
Tabelle 3: Beschreibung: Im System einloggen	14
Tabelle 4: Beschreibung: Im System ausloggen.....	14
Tabelle 5: Beschreibung: Einkaufsliste verwalten	15
Tabelle 6: Beschreibung: Einkaufsliste bearbeiten.....	16
Tabelle 7: Beschreibung: Einkaufsliste anzeigen	16
Tabelle 8: Beschreibung: Einkaufsliste löschen	17
Tabelle 9: Beschreibung: Einkaufsliste archivieren.....	17
Tabelle 10: Beschreibung: Einkaufsliste auswählen	17
Tabelle 11: Beschreibung: Einkaufsliste anlegen.....	18
Tabelle 12: Beschreibung: Artikel hinterlegen	18
Tabelle 13: Beschreibung: Artikel markieren	19
Tabelle 14: Beschreibung: Menge hinterlegen.....	19
Tabelle 15: Beschreibung: Einzelhändler hinterlegen.....	20
Tabelle 16: Beschreibung: häufig gekaufte Artikel aufrufen.....	21
Tabelle 17: Beschreibung: Zeitraum aufrufen	21
Tabelle 18: Beschreibung: Einzelhändler aufrufen	21
Tabelle 19: Beschreibung: Gruppe verwalten	22
Tabelle 20: Beschreibung: Gruppenmitglieder anzeigen	23
Tabelle 21: Beschreibung: Person entfernen	23
Tabelle 22: Beschreibung: Person hinzufügen	24
Tabelle 23: Beschreibung: Einkaufsliste teilen	24
Tabelle 24: Beschreibung: Einkaufsliste durchsuchen.....	25
Tabelle 25: Beschreibung: Einkaufsliste filtern.....	26
Tabelle 26: Beschreibung: EditorServiceImpl	37
Tabelle 27: Beschreibung: GroupUser	40
Tabelle 28: Beschreibung: Retailer	41
Tabelle 29: Beschreibung: GroceryList	42
Tabelle 30: Beschreibung: Article	43
Tabelle 31: Beschreibung:GroceryListArticle.....	43
Tabelle 32: Beschreibung: LoginInfo	44
Tabelle 33: Beschreibung: GroupUserMapper	48
Tabelle 34: Beschreibung: GroceryListArticleMapper	48
Tabelle 35: Beschreibung: RetailerMapper	49
Tabelle 36: Beschreibung: EntryMapper	49



Abkürzungsverzeichnis

HTML	Hypertext Markup Language
CSS	Cascading Stylesheet
GWT	Google Web Toolkit
SDK	Software Development Kit
JDBC	Java Database Connectivity
RPC	Remote Procedure Call
SQL	Structured Query Language
UC	Use Case
BO	Business Object
JRE	Java Runtime Environment
GUI	Graphical User Interface
UML	Unified Modelling Language

Tabelle 1: Abkürzungsverzeichnis

Anforderungsanalyse

Aufgabenstellung des Projekts

Der Inhalt stellt einen Ausschnitt der Aufgabenstellung dar, der die Anforderungen des Zielsystems beschreibt. Ziel ist es, „[...] ein verteiltes **Shared-Shopping-List-System** zu realisieren, [...] auf das von mehreren Benutzern simultan zugriffen werden kann. Im Zuge dessen soll eine dreischichtige Architektur realisiert werden (vgl. Abbildung 1). [...]“.

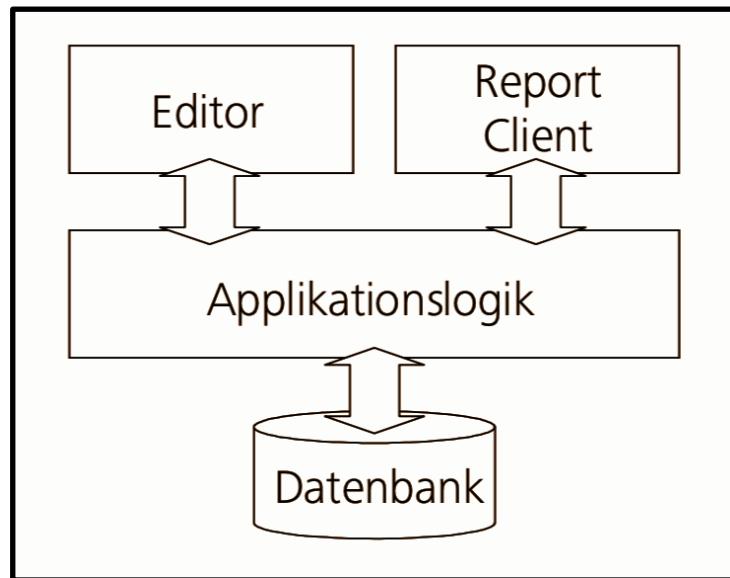


Abbildung 1: Scart - Struktur des Zielsystems

(Quelle: Prof. Dr. Peter Thies und Prof. Dr.-Ing. Christoph Kunz, 2019)

Die Abbildung stellt folgende Funktionen dar. „[...] Die unterste Schicht wird die Datenbankschicht sein. Diese Schicht sorgt für die so genannte Persistenz der mit dem System gehandhabten Daten. Hier soll mit Hilfe von Google Cloud SQL bzw. MySQL eine relationale Datenbank entworfen und implementiert werden, die sämtliche Daten des Systems aufnehmen kann.

Die oberste Schicht wird die Präsentationsschicht sein. Sie wird durch zwei getrennte Clients mit graphischer Benutzungsschnittstelle [...] realisiert, die einzeln als Webseite in einem Internetbrowser aufgerufen werden können. Der erste Client dient zur Erstellung und Verwaltung der Einkaufslisten, sowie der Abarbeitung der persönlichen Einkaufslisten. Der zweite Client ermöglicht die Anzeige einer Einkaufsstatistik nach Zeitraum. Die Clients werden nachfolgend als Editor [...] und als Report Client [...] bezeichnet.

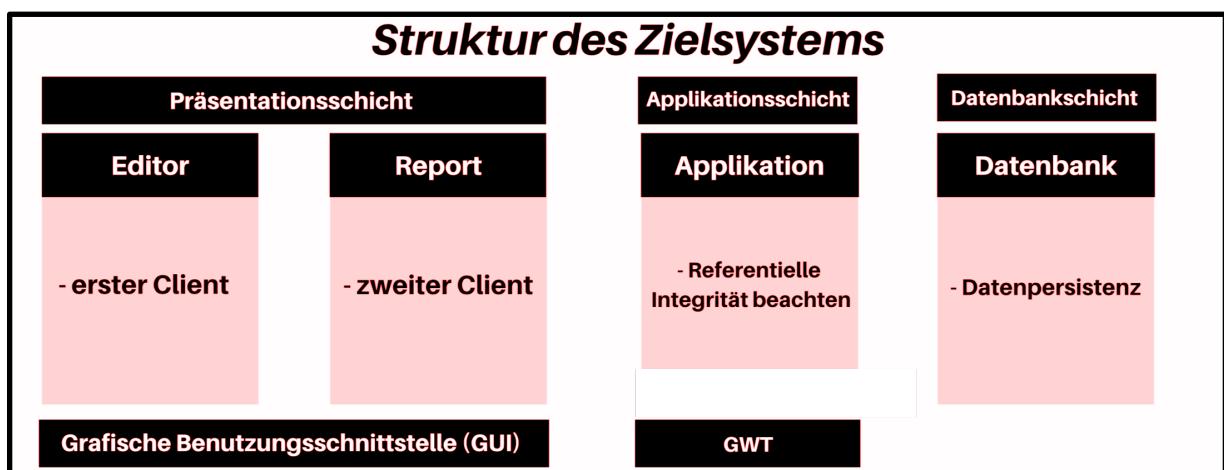


Abbildung 2: Scart - Struktur des Zielsystems



Es ist darauf zu achten, dass die Clients eine klare Aufteilung von Funktionsbereichen besitzen. Zur Navigation in einer größeren Datenmenge [...] bietet sich z.B. die Verwendung einer scrollbaren, listenartigen Darstellung an.

Zum Erzeugen von Objekten sollte für jede Klasse [...] eine spezifische Interaktionsform existieren, mit deren Hilfe die jeweiligen Objekte erzeugt, manipuliert oder gelöscht werden können. Die Interaktion wird durch Auswahl von Schaltflächen oder Selektion aus Listendarstellungen initiiert. Die Zuordnung von z.B. Artikeln zu Personen ergibt sich aus der Platzierung und Auswahl der jeweiligen Interaktionselemente. Die Einkaufslisten sollen in HTML-Form nach Responsive Design ausgegeben werden, d.h. sie sind auf Desktop und Mobilgeräten einsehbar. Jeder Artikel kann abgehakt werden, wenn er eingekauft wurde.

Die mittlere Schicht nimmt die Applikationslogik [...] auf. Sie enthält die Algorithmen, Regeln und Strukturen, die notwendig sind, um die Elemente [...] und Funktionen [...] einer Anwendung beschreiben zu können. Damit erfolgt eine Trennung von den anderen Funktionen des Systems, die sich z.B. mit der äußeren Darstellung oder der internen Abspeicherung der Daten befassen. Für eine Applikationslogik lassen sich somit mehrere Präsentationslogiken und mehrere Speicherlogiken definieren. Die mittlere Schicht stellt den sog. Applikationsserver dar. Er soll der Präsentationsschicht mindestens folgende Dienste anbieten:

- 1) Registrieren von Anwendern, Anlegen, Editieren und Löschen von Familien und Personen, Artikeln, Einkaufslisten, Einzelhändlern, etc.
- 2) Verknüpfen von Personen, Artikeln, Einkaufslisten mit Einzelhändlern bzw. Modifizieren und Aufheben (Storno) von solchen Verknüpfungen.
- 3) Filtern und Ausgeben von Einkaufslisten, Abhaken von Einträgen.
- 4) Kenntlichmachung der letzten Änderung einer Einkaufsliste.
- 5) Ausgeben einer Statistik über häufig eingekaufte Artikel bzgl. Zeitraum und Händler zur Optimierung des Einkaufsverhaltens. Zeitraum und Händler sollen jeweils optional angebbar sein.

1-4 werden von dem Editor, 5 durch den Report- Client genutzt. Weitere Funktionen der Applikationslogik können bei Bedarf hinzugefügt werden. Es ist in dem gesamten System auf die Einhaltung der referentiellen Integrität zu achten. Die oben genannte Funktionalität erfordert, dass sich die Applikationslogik in Kontakt mit der Datenbank befindet, um die nötigen Daten aus der Datenbank auslesen bzw. neue Daten in der Datenbank ablegen zu können. Die Datenbankschicht wird demzufolge mit dem Datenbank-Server sowie seiner Anbindung an den Applikationsserver [...] realisiert, der den Zugriff auf die abgespeicherten Daten als Dienst zur Verfügung stellt. Als Datenbank-Server ist Google Cloud SQL zu verwenden. Hierzu müssen die Gruppen mindestens ein Benutzerkonto bei diesem Dienst anlegen. Applikationsserver und Datenbankserver müssen bei der Präsentation vollständig in der Cloud laufen. [...] Ein Teil der Datenbankschicht soll relationale Strukturen [...] in Objekte und vice versa transformieren. [...] Zudem soll die Applikationsschicht mit der



Präsentationsschicht mittels des GWT Remote Procedure Call [...] kommunizieren. Die Applikationslogik ist auf Basis von Google App Engine zu realisieren. Die Kommunikation mit dem Datenbankmanagementsystem erfolgt gemäß den Vorgaben von Google Cloud SQL mittels der Java Database Connectivity [...].

(Prof. Dr. Peter Thies, Prof. Dr.-Ing. Christoph Kunz, S. 1-5, 2019)

Anforderungen an das System

Das Projekt beinhaltet die Erstellung eines verteilten Systems für die Verwaltung und Nutzung von geteilten Einkaufslisten. Die Anforderung besteht aus der Verwaltung und Bearbeitung solcher Listen, die es zum Beispiel ermöglichen Familien und WG's den Einkaufs Alltag zu erleichtern. Daraus ergibt sich ein Einkaufslistenverwaltungssystem, das Einkaufslisten mit Einzelhändlern und deren Mitglieder in Gruppen beinhaltet (Shared Shopping-List-System). Der Nutzer hat so die Möglichkeit Gruppen zu erstellen, löschen und nach belieben zu modifizieren. In diesen Gruppen können neue Mitglieder eingeladen werden mit denen man die gemeinsamen Einkaufslisten teilt und verwaltet. „[...] Diese Einkaufslisten können Einträge über zu beschaffende Artikel, deren Menge [...] und den bevorzugt verkaufenden Einzelhändler enthalten. Jeder Eintrag soll einer Person einer Familie zugeordnet werden können, welche den Artikel zu besorgen hat. Die jeweilige Person kann dann in einer Einkaufssituation ihre persönlichen Einkaufsliste einsehen und gekauft Artikel abhaken. Die Artikel sollen nach Einzelhändler gruppiert werden können, so dass die Einkaufstour geplant werden kann. Beim Erstellen von neuen Einkaufslisten sollen regelmäßige Standardartikel zur Bequemlichkeit schon enthalten sein. Für eine solche Funktionsweise müssen dem System die Begriffe Familie, Familienmitglied, Anwender, Einkaufsliste, [...] Artikel, Einzelhändler und deren Verknüpfung zu einem Listeneintrag bekannt sein. Objekte der vorgenannten Begriffe [...] sollen in dem Zielsystem jederzeit angelegt, editiert und gelöscht werden können. Für jedes Element kann eine Reihe von Attributen festgelegt werden. Dies sind für alle: Erstellungszeitpunkt, ID und Name [...]; Für Artikel zusätzlich Menge in der jeweiligen Einheit [...]. Gemeinsame Attribute werden dabei vorteilhaft in einer gemeinsamen Basisklasse definiert. Zur Erstellung der jeweiligen Elemente müssen diese in dem System in geeigneter Weise repräsentiert und miteinander in Beziehung gesetzt werden können. Für die Beziehung zwischen Artikel, Einzelhändler und einkaufende Person bietet sich dabei die Definition einer eigenen Klasse an [...].

(Prof Dr. Peter Thies, Prof. Dr.-Ing. Christoph Kunz, S. 1-2, 2019)

Umfang

Mussfunktionen

F001 Login

Der Nutzer meldet sich mit seinem Google-Account im Shared-Shopping-List-System über den “Login”-Button an. Nach der Anmeldung sieht der Nutzer sein eigenes Profil mit den entsprechenden Gruppen und Einkaufslisten, sofern diese bereits erstellt worden sind.



F002 Registrierung

Bei der ersten Nutzung des Systems kann der Nutzer mit seinem Google-Account, Name und E-Mail-Adresse über den “Registrieren”-Button sich registrieren.

F003 Gruppe erstellen

Der Nutzer kann über das Textfeld “Gruppe hinzufügen” eine neue Gruppe mit einem Namen für diese Gruppe erstellen.

F004 Gruppe bearbeiten

Über den “Änderungen speichern”-Button kann der Nutzer die Änderungen an einer Gruppe speichern.

F005 Gruppe löschen

Durch das Austreten einer Gruppe wird die Verknüpfung zwischen Nutzer und Gruppe gelöscht.

F006 Nutzer in eine Gruppe hinzufügen

Über den “add”-Button wird der Nutzer in einer Gruppe hinzugefügt.

F007 Nutzer aus einer Gruppe entfernen

Über den “Aus Gruppe austreten”-Button wird der Nutzer von einer Gruppe entfernt.

F008 Einkaufsliste erstellen

Der Nutzer kann über das Textfeld “Einkaufsliste hinzufügen” eine neue Einkaufsliste mit einem Namen für diese Einkaufsliste erstellen.

F009 Einkaufsliste bearbeiten

Über den “Laden einfügen”-Button kann der Nutzer einen Händler zu seiner Einkaufsliste hinzufügen. Mit dem “+”-Button kann der Nutzer Artikel zu diesem Händler hinzufügen. Andersherum kann der Nutzer mit dem “-“-Button Artikel zu diesem Händler entfernen.

F010 Einkaufsliste löschen

Der Nutzer kann mit dem “Mülleimer”-Icon die Einkaufsliste löschen.



F011 Profil bearbeiten

Dem Nutzer soll es möglich sein, sein eigenes Profil zu bearbeiten und im nachhinein abzuspeichern. Der Nutzer kann seinen Namen und seine E-Mail-Adresse ändern.

F012 Einzelhändler erstellen

Es soll dem Nutzer des Zielsystems möglich sein, eigene Einzelhändler anzulegen und diese mit den jeweiligen Einkaufslisten und Mitgliedern der jeweiligen Gruppe zu verknüpfen.

F013 Artikel der Einkaufsliste hinzufügen

Es soll möglich sein, verschiedene Artikel der Einkaufslisten hinzuzufügen und Usern mit Zustimmung diese dann zuzuweisen. Somit kann der Einkauf geplant werden, wer was wo zu besorgen hat.

F014 Artikel der Einkaufslisten entfernen

Es sollte auch möglich sein solche Verknüpfungen mit einem User, Einkaufsliste etc. wieder zu stornieren. Damit keine Abhängigkeit der einzelnen Komponenten mehr besteht.

F015 Artikel Archivieren

Es sollen Artikel die gekauft wurden im System archiviert werden um auf spätere Statistiken im Report zurück zu greifen.

F016 Profil löschen

Der Nutzer soll frei entscheiden können ob er ein Profil auf unserem Zielsystem besitzen möchte oder entfernen möchte.

Wunschkosten

WF001 Profilbild

Im eigenen Profil kann der Nutzer ein Profilbild hinterlegen.

Abgrenzungsfunktionen

Wir als Gruppe haben festgelegt, dass wir keine Abgrenzungsfunktionen definieren. Der Fokus liegt bei diesem Projekt darauf, dass alle Mussfunktionen implementiert und im funktionsfähigen Umfang vorhanden sind.



Begriffslexikon

Begriffsbezeichnung	Synonyme	Beschreibung des Begriffs
Nutzer	User, Anwender	Eine Natürliche Person, die mit der Shared-Shopping-List interagiert.
Editor	Client 1	Der Editor dient zur Interaktion mit dem Zielsystem.
Report	Client 2	Der Report dient zur Erzeugung von übersichtlichen Berichten innerhalb des Zielsystems.
Artikel	Article, Gegenstand, Gut	Ein Artikel im Zielsystem beschreibt einen Gegenstand der innerhalb einer Gruppe angelegt wird und besorgt werden sollte.
Einkaufsliste	GroceryList	Die Einkaufsliste enthält sämtliche Artikel einer Gruppe die besorgt werden müssen.
Einzelhändler	Retailer	Ein Einzelhändler ist mit einem Artikel verknüpft und dient zur Identifikation, wo der Artikel gekauft werden soll.
Profil	Account	Ein Profil lässt sich durch den Nutzer selbst editieren oder entfernen.
Eigenschaft	Attribut	Eine Eigenschaft kann durch verschiedene Ausprägungen beschrieben werden.
Gruppe	Group	Eine Gruppe beschreibt eine Gemeinschaft von Menschen/ Nutzer im Zielsystem.
Startseite	Homepage, Dashboard	Die erste Seite, die der User/ Nutzer beim Systemstart zu Gesicht bekommt.
Favorisiert	Beliebt, Lieblings-	Ein Favorit beschreibt einen beliebten Artikel innerhalb einer Gruppe im Zielsystem.

Tabelle 2: Begriffslexikon



Entwurf

Use-Case Diagramme

Use-Case-Diagramme verschaffen einen groben Überblick über Funktionen innerhalb des Zielsystems. Zu den jeweiligen erstellten Use-Cases werden zur Verdeutlichung textuelle Beschreibungen hinterlegt. Sie erläutern den essenziellen Ablauf und geben Informationen über Sonderfälle, die eventuell auftreten können. Die hinterlegten Texte beziehen sich immer auf einzelne Ellipsen in den Use-Cases. Die zwei Arten von Assoziationen werden durch *include* oder *exclude* kenntlich gemacht. Die erstellten Diagramme stellen keine tatsächliche Reaktion des Systems dar. Diese werden ausschliesslich aus der Sicht des Anwenders formuliert.

Die *Include* Assoziation bezeichnet einen Anwendungsfall der einen anderen Anwendungsfall einbindet. Diese Beziehung stellt eine zwingende Beziehung dar und wird deshalb auch oft "Mussbeziehung" genannt.

Die *Exclude* Assoziation hingegen beschreibt einen Anwendungsfall der einen anderen darauffolgenden Anwendungsfall erweitert. Diese Beziehung stellt eine optionale Verknüpfung dar und wird auch "Kannbeziehung" genannt.

Nachfolgend werden die Use-Cases mit ihrer textuellen Beschreibung eingebunden und ausführlich erklärt. Die Tabellen unter den jeweiligen Use-Cases dienen zur textuellen veranschaulichung der jeweiligen Uses-Cases.

Use-Case Diagramm: Ein und Ausloggen

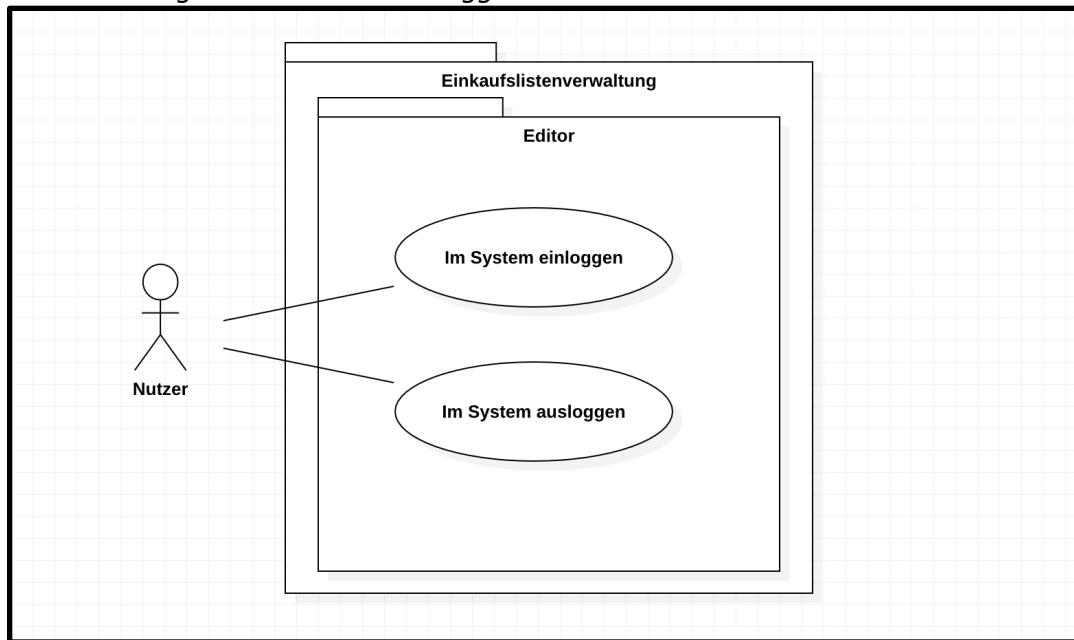


Abbildung 3: Use-Case : Ein und Ausloggen

Im ersten dargestellten Use-Case: dem **Ein und Ausloggen**, wird die erste Interaktion zwischen Nutzer und dem System veranschaulicht. Dieses stellt dar, dass sich der Nutzer



zuerst einloggen muss, um das Zielsystem verwenden zu dürfen. Um das Zielsystem wieder zu verlassen, muss der Nutzer sich am Ende auch wieder ausloggen.

Name	Im System einloggen
Kurzbeschreibung	Der Nutzer loggt sich im System ein.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die Benutzerauthentifizierung ist sichergestellt.
Eingehende Daten	Google-ID und Passwort
Vorbedingung	Nutzer muss bereits registriert sein + Der Nutzer ist noch nicht authentifiziert.
Nachbedingung	Nutzer ist im System eingeloggt und hat vollen Zugriff auf alle Funktionen vom Einkaufsverwaltungssystems.
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.
Sonderfall	<ol style="list-style-type: none">1. Die Nutzerauthentifizierung schlägt fehl durch falsch eingegebene Nutzerdaten.2. Die Nutzerauthentifizierung schlägt fehl durch fehlende Nutzerdaten.
Sonstiges. Anmerkung	-

Tabelle 3: Beschreibung: Im System einloggen

Name	Im System ausloggen
Kurzbeschreibung	Der Nutzer loggt sich im System aus.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Der Nutzer hat sich abgemeldet und kann nicht mehr auf das System zugreifen.
Eingehende Daten	-
Vorbedingung	Der Nutzer muss authentifiziert sein.
Nachbedingung	Das Nutzer Konto gilt als Abgemeldet.
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt die Funktion zum ausloggen aus.
Sonderfall	-
Sonstiges. Anmerkung	-

Tabelle 4: Beschreibung: Im System ausloggen



Use-Case-Diagramm: Einkaufsliste verwalten

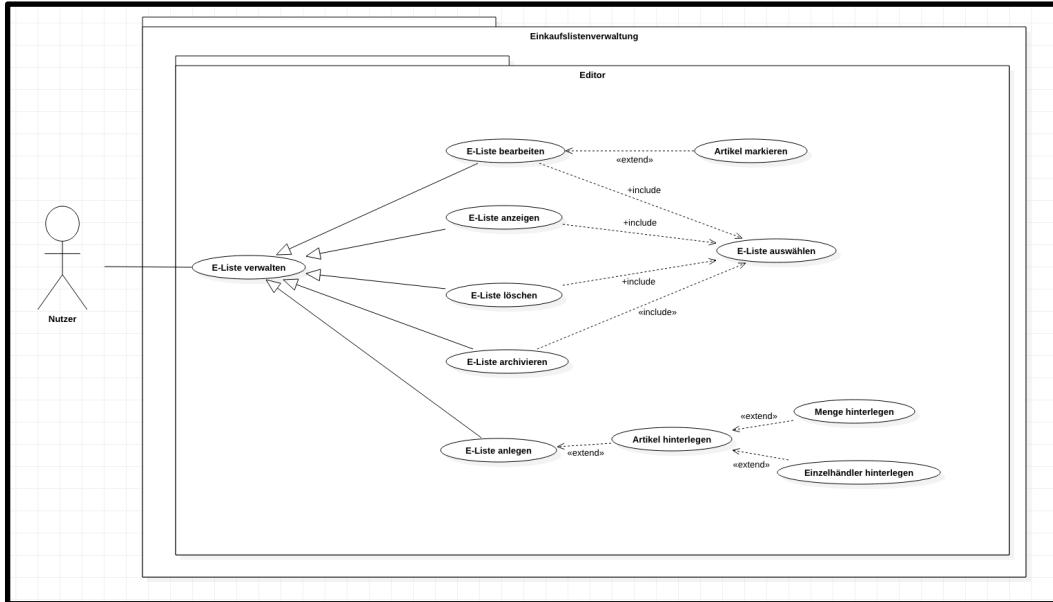


Abbildung 4: Scart - Use-Case: Einkaufsliste verwalten

Das zweite Use-Case-Diagramm: **Einkaufsliste verwalten**, beschreibt die funktionelle Notwendigkeit Einkaufslisten im Zielsystem zu verwalten. Die Ellipse "E-Liste verwalten" generalisiert alle untergeordneten Eigenschaften im Zielsystem. Dies beinhaltet die Funktionen Einkaufslisten zu bearbeiten, anzeigen, löschen, archivieren, neue Listen anzulegen und Artikel in der Einkaufsliste zu hinterlegen.

Name	E-Liste verwalten
Kurzbeschreibung	Der Nutzer will Änderungen an seiner/seinen E-Liste(n) vornehmen.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die Verwaltung der E-Listen, das bearbeiten, anzeigen, löschen, archivieren und anlegen der E-Liste einer Gruppe.
Eingehende Daten	-
Vorbedingung	Angemeldet + Gruppe ausgewählt
Nachbedingung	Der Nutzer befindet sich in der Einkaufslistenverwaltung und kann die Systemfunktionen zur weiteren Anwendung verwenden.
Essentielle Schritte	<ol style="list-style-type: none"> Der Nutzer startet die App Der Nutzer Authentifiziert sich via des Google Dienstes. Der Nutzer wählt die Funktion zum Verwalten der E-Listen einer Gruppe aus.
Sonderfall	-
Sonstiges. Anmerkung	-

Tabelle 5: Beschreibung: Einkaufsliste verwalten

Name	E-Liste bearbeiten
Kurzbeschreibung	Der Nutzer will Änderungen an seiner/seinen E-Liste(n) vornehmen.



Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	E-Liste wurde bearbeitet
Eingehende Daten	Gruppe + E-Liste
Vorbedingung	Angemeldet + Gruppe ausgewählt + E-Liste ausgewählt
Nachbedingung	<ol style="list-style-type: none"> 1. E-Liste wurde bearbeitet 2. Die E-Liste kann zur weiteren Anwendung der Systemfunktion verwendet werden
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Der Nutzer startet die App 2. Der Nutzer Authentifiziert sich via des Google Dienstes. 3. Der Nutzer wählt die Funktion zum bearbeiten einer E-Liste einer Gruppe aus. 4. Der Nutzer speichert die bearbeitete E-Liste ab.
Sonderfall	Bearbeitung nicht möglich, da keine E-Liste vorhanden ist.
Sonstiges. Anmerkung	-

Tabelle 6: Beschreibung: Einkaufsliste bearbeiten

Name	E-Liste anzeigen
Kurzbeschreibung	Der Nutzer möchte sich seine E-Liste anzeigen lassen.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die E-Liste wird dem Nutzer im System angezeigt
Eingehende Daten	Gruppe + E-Liste
Vorbedingung	Angemeldet + Gruppe ausgewählt + E-Liste ausgewählt
Nachbedingung	E-Liste wird dem Nutzer angezeigt. Die E-Liste kann zur weiteren Anwendung der Systemfunktion verwendet werden.
Essenzielle Schritte	<ol style="list-style-type: none"> 5. Der Nutzer startet die App 6. Der Nutzer Authentifiziert sich via des Google Dienstes. 7. Der Nutzer wählt die Funktion zum Anzeigen einer E-Liste einer Gruppe aus.
Sonderfall	E-Liste wird nicht korrekt angezeigt.
Sonstiges. Anmerkung	-

Tabelle 7: Beschreibung: Einkaufsliste anzeigen

Name	E-Liste löschen
Kurzbeschreibung	Der Nutzer möchte eine E-Liste löschen.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	E-Liste ist für die Gruppe im System gelöscht.
Eingehende Daten	Gruppe + E-Liste
Vorbedingung	Angemeldet + Gruppe ausgewählt + E-Liste ausgewählt
Nachbedingung	Die E-Liste ist im System nicht mehr vorhanden.
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Der Nutzer startet die App 2. Der Nutzer Authentifiziert sich via des Google Dienstes.



	<ol style="list-style-type: none">3. Der Nutzer wählt die zu löschen E-Liste einer Gruppe aus.4. Der Nutzer wählt die Funktion zum Löschen einer E-Liste der Gruppe aus.
Sonderfall	Es sind keine E-Listen in der Gruppe vorhanden.
Sonstiges. Anmerkung	-

Tabelle 8: Beschreibung: Einkaufsliste löschen

Name	E-Liste archivieren
Kurzbeschreibung	Der Nutzer möchte eine E-Liste archivieren.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	E-Liste wird für die Gruppe im System archiviert.
Eingehende Daten	Gruppe + E-Liste
Vorbedingung	Angemeldet + Gruppe ausgewählt + E-Liste ausgewählt
Nachbedingung	E-Liste ist im System veraltet (nicht mehr gelistet).
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt die Funktion zum Archivieren einer E-Liste einer Gruppe aus.
Sonderfall	E-Liste wurde nicht aufbewahrt.
Sonstiges. Anmerkung	Wie lange soll eine E-Liste aufbewahrt werden?

Tabelle 9: Beschreibung: Einkaufsliste archivieren

Name	E-Liste auswählen
Kurzbeschreibung	E-Liste wird für das bearbeiten, anzeigen, löschen oder archivieren ausgewählt.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die E-Liste wurde zum bearbeiten, anzeigen, löschen oder archivieren ausgewählt.
Eingehende Daten	Gruppe + E-Liste
Vorbedingung	Angemeldet + Gruppe ausgewählt
Nachbedingung	Der Nutzer hat eine E-Liste ausgewählt. Die E-Liste kann zur weiteren Anwendung der Systemfunktion verwendet werden.
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt eine E-Liste einer Gruppe aus.4. Der Nutzer verwendet die Funktionen zum bearbeiten, anzeigen, löschen oder archivieren der E-Liste.
Sonderfall	Es ist keine E-Liste vorhanden, in der ausgewählten Gruppe.
Sonstiges. Anmerkung	-

Tabelle 10: Beschreibung: Einkaufsliste auswählen



Name	E-Liste anlegen
Kurzbeschreibung	Der Nutzer möchte eine E-Liste erstellen.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Eine neue E-Liste wurde angelegt.
Eingehende Daten	Gruppe + E-Liste
Vorbedingung	Angemeldet + Gruppe ausgewählt
Nachbedingung	Eine neue E-Liste wurde einer Gruppe hinzugefügt.
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Der Nutzer startet die App 2. Der Nutzer Authentifiziert sich via des Google Dienstes. 3. Der Nutzer wählt die Funktion zum anlegen einer E-Liste für eine Gruppe aus.
Sonderfall	Es kann keine E-Liste angelegt werden, da keine Gruppen vorhanden ist.
Sonstiges. Anmerkung	Häufig eingekaufte Artikel werden als Standard sofort bei E-Liste anlegen in die Liste hinzugefügt ohne das der Nutzer diese extra hinzufügen muss

Tabelle 11: Beschreibung: Einkaufsliste anlegen

Name	Artikel hinterlegen
Kurzbeschreibung	Der Nutzer möchte einen Einkaufszettel nach ausgewählten Filterkriterien filtern.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Artikel, auf die der Filter zutrifft
Eingehende Daten	Ausgewählter Filter
Vorbedingung	Angemeldet + Gruppe ausgewählt + Liste ausgewählt
Nachbedingung	<ul style="list-style-type: none"> - Alle Artikel die den Filterkriterien entsprechen, werden angezeigt.
Essenzielle Schritte	<ol style="list-style-type: none"> 1. Der Nutzer startet die App 2. Der Nutzer Authentifiziert sich via des Google Dienstes. 3. Es wird die Funktion zum erstellen einer Gruppe ausgewählt. 4. Der Nutzer konfiguriert die geteilten E-Listen sowie die dazugehörigen Gruppenmitglieder. 5. Der Nutzer teilt die E-Liste durch Bestätigung.
Sonderfall	<ol style="list-style-type: none"> 1. Keine Artikel vorhanden 2. Keine Artikel entsprechen den Filterkriterien
Nachbedingung im Sonderfall	<ol style="list-style-type: none"> 1. Kein Artikel wird angezeigt 2. Kein Artikel wird angezeigt

Tabelle 12: Beschreibung: Artikel hinterlegen

Name	Artikel markieren
Kurzbeschreibung	Der Nutzer markiert einen Artikel als gekauft.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Der markierte Artikel verschiebt sich in eine Liste von bereits gekauften



	Artikeln für die Gruppe
Eingehende Daten	Artikel + Einzelhändler + Menge
Vorbedingung	Angemeldet + Gruppe ausgewählt + E-Liste ausgewählt + Artikel ausgewählt
Nachbedingung	Eine neue Liste wird unter der Liste mit den Artikeln die zu kaufen sind kreiert.
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt die Funktion zum markieren des Artikels als "gekauft" auf, für seine Gruppe.
Sonderfall	Alle Artikel sind bereits gekauft. Es gibt keine Artikel auf der E-Liste.
Sonstiges. Anmerkung	Falls falscherweise ein Artikel als gekauft markiert wurde, sollte eine Undo Funktion verfügbar sein.

Tabelle 13: Beschreibung: Artikel markieren

Name	Menge hinterlegen
Kurzbeschreibung	Menge für einen Artikel wird hinterlegt.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Eine Menge mit Mengeneinheit wird einem Artikel auf einer E-Liste zugewiesen.
Eingehende Daten	Artikel
Vorbedingung	Angemeldet + Gruppe ausgewählt + E-Liste ausgewählt + Artikel ausgewählt
Nachbedingung	Ausgewählter Artikel hat eine Mengenangabe mit Mengeneinheit hinterlegt.
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt die Funktion zum bearbeiten der Mengenangabe mit Mengeneinheit eines Artikels aus.
Sonderfall	Mengeneinheit nicht vorhanden.
Sonstiges. Anmerkung	Custom Mengeneinheit wird für Gruppen als Standard gespeichert.

Tabelle 14: Beschreibung: Menge hinterlegen

Name	Einzelhändler hinterlegen
Kurzbeschreibung	Der Nutzer legt einen Einzelhändler im System an.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Der Einzelhändler wurde im System angelegt.
Eingehende Daten	Einzelhändler
Vorbedingung	Angemeldet + Gruppe ausgewählt + E-Liste ausgewählt
Nachbedingung	<ol style="list-style-type: none">1. Der Nutzer hat einen neuen Einzelhändler angelegt.2. Der Einzelhändler kann zur weiteren Anwendung der Systemfunktion verwendet werden.
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.



	<p>3. Der Nutzer wählt die Funktion zum Hinterlegen eines Einzelhändlers für einer Gruppe aus.</p> <p>4. Der Nutzer wählt die Funktion zum Speichern aus.</p>
Sonderfall	-
Sonstiges. Anmerkung	-

Tabelle 15: Beschreibung: Einzelhändler hinterlegen

Use-Case-Diagramm: Report-Generator

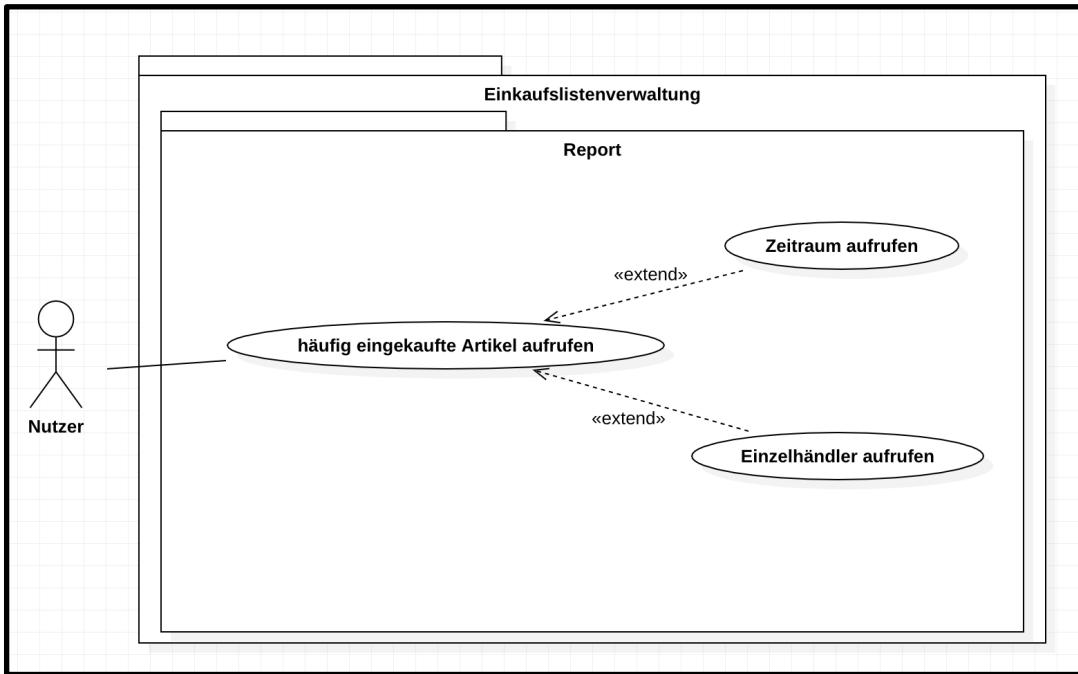


Abbildung 5: Scart. - Use-Case: Report-Generator

Der dritte Use-Case beschreibt das generieren des Reports im Zielsystem. Der Report erstellt eine statische Abfrage einer speziellen Ausgabe. Dieser soll über die Funktion verfügen, häufig gekaufte Artikel innerhalb einer Gruppe mit deren Einkaufslisten zu erstellen. Der Zeitraum sowie der Einzelhändler sollen optional filterbar sein.

Name	häufig eingekaufte Artikel aufrufen
Kurzbeschreibung	Der Nutzer möchte häufig eingekaufte Artikel aufrufen.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Eine Übersicht über häufig gekaufte Artikel mit optional Zeitraum und Händler wird angezeigt.
Eingehende Daten	Häufig eingekaufte Artikel
Vorbedingung	Angemeldet + Gruppe ausgewählt + Reportgenerator ausgewählt
Nachbedingung	Der Nutzer erhält einen Report, welcher häufig eingekaufte Artikel sowie optional Zeitraum und Händler anzeigt.
Essentielle Schritte	<ol style="list-style-type: none"> Der Nutzer startet die App Der Nutzer Authentifiziert sich via des Google Dienstes. Der Nutzer wählt Reportgenerator aus.



	4. Der Nutzer wählt die Funktion zur Ausgabe häufig gekaufter Artikel sowie optional Zeitraum und Händler auf.
Sonderfall	Keine Artikel vorhanden.
Sonstiges. Anmerkung	Die Ausgabe ist leer wenn keine E-Liste/Artikel angelegt wurden.

Tabelle 16: Beschreibung: häufig gekaufte Artikel aufrufen

Name	Zeitraum aufrufen
Kurzbeschreibung	Auflisten der Artikel und deren Häufigkeit die in einem gewählten Zeitraum eingekauft wurden
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Eine Übersicht über eingekauft Artikel im gewählten Zeitraum
Eingehende Daten	Artikel + Ausgewählter Zeitraum
Vorbedingung	Angemeldet + Gruppe ausgewählt + Reportgenerator ausgewählt
Nachbedingung	Der Nutzer erhält einen Report, welcher häufig eingekauft Artikel, sowie Zeitraum der Artikel anzeigen.
Essentielle Schritte	<ol style="list-style-type: none">Der Nutzer startet die AppDer Nutzer Authentifiziert sich via des Google Dienstes.Der Nutzer wählt Reportgenerator aus.Der Nutzer wählt die Funktion zur Ausgabe häufig gekaufter Artikel sowie gewünschtem Zeitraum aus.
Sonderfall	Keine Artikel im gewählten Zeitraum vorhanden
Sonstiges. Anmerkung	-

Tabelle 17: Beschreibung: Zeitraum aufrufen

Name	Einzelhändler aufrufen
Kurzbeschreibung	Der Nutzer möchte alle Artikel und deren Häufigkeit bei einem/ mehreren gewählten Einzelhändler aufrufen.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Übersicht über alle eingekauften Artikel bei dem gewählten Einzelhändler
Eingehende Daten	Artikel + Ausgewählter Einzelhändler
Vorbedingung	Angemeldet + Gruppe ausgewählt + Reportgenerator ausgewählt
Nachbedingung	Der Nutzer erhält einen Report, welcher häufig eingekauft Artikel, sowie deren Einzelhändler der Artikel anzeigen.
Essentielle Schritte	<ol style="list-style-type: none">Der Nutzer startet die AppDer Nutzer Authentifiziert sich via des Google Dienstes.Der Nutzer wählt Reportgenerator aus.Der Nutzer wählt die Funktion zur Ausgabe häufig gekaufter Artikel sowie gewünschtem Einzelhändler aus.
Sonderfall	Kein Händler vorhanden; kein Artikel vorhanden
Sonstiges. Anmerkung	-

Tabelle 18: Beschreibung: Einzelhändler aufrufen



Use-Case-Diagramm: Gruppe verwalten:

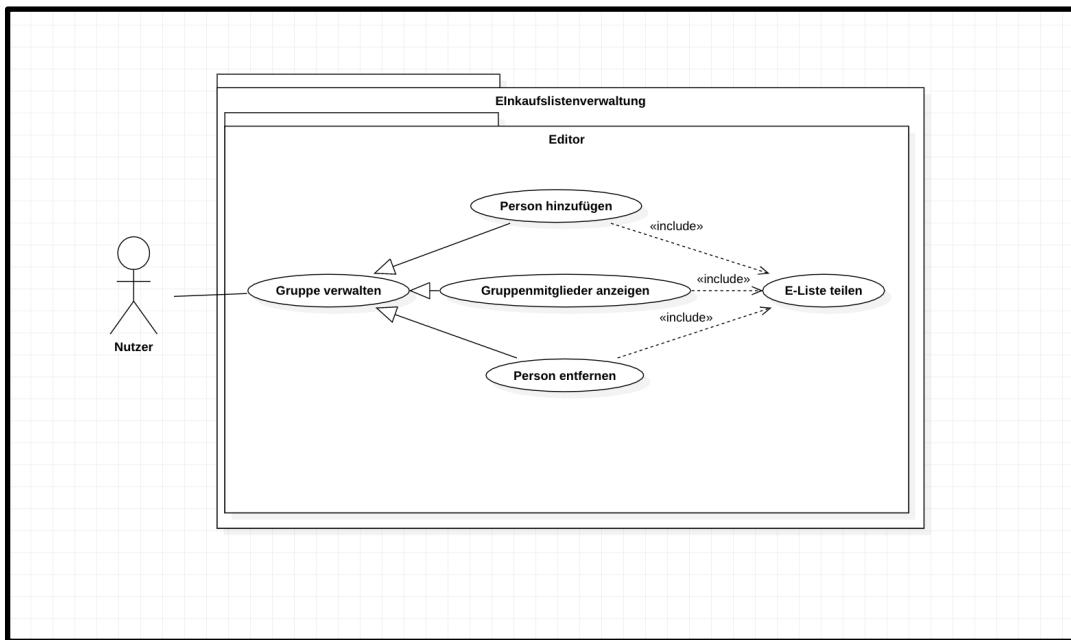


Abbildung 6: Scart - Use-Case: Gruppe verwalten

Der vierte Use-Case beschreibt das verwalten der Gruppe innerhalb des Zielsystems. Die Ellipse "**Gruppe verwalten**" generalisiert alle untergeordneten Anwendungsfälle. Es sollten bekannte Funktionen wie neue Mitglieder hinzufügen, Mitglieder entfernen und Mitglieder innerhalb der Gruppe anzeigen, möglich sein. All diese Funktionen inkludieren die Funktion eine Einkaufsliste zu teilen mit ein.

Name	Gruppe verwalten
Kurzbeschreibung	Der Nutzer möchte seine Gruppe verwalten.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die Verwaltung der Gruppe; anzeigen, hinzufügen sowie entfernen einer Person in einer Gruppe
Eingehende Daten	-
Vorbedingung	Angemeldet + Gruppe ausgewählt
Nachbedingung	Der Nutzer befindet sich in der Einkaufslistenverwaltung und kann die Systemfunktionen zur weiteren Anwendung verwenden
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt Reportgenerator aus.4. Der Nutzer wählt die Funktion zur Gruppenverwaltung aus.
Sonderfall	-
Sonstiges. Anmerkung	-

Tabelle 19: Beschreibung: Gruppe verwalten



Name	Gruppenmitglieder anzeigen
Kurzbeschreibung	Der Nutzer möchte sich die Gruppenmitglieder einer Gruppe anzeigen lassen.
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die Gruppenmitglieder werden dem Nutzer im System angezeigt
Eingehende Daten	Gruppenmitglieder
Vorbedingung	Angemeldet + Gruppe ausgewählt
Nachbedingung	<ol style="list-style-type: none">1. Die Gruppenmitglieder werden dem Nutzer angezeigt2. Die Gruppenmitglieder können zur weiteren Anwendung der Systemfunktion verwendet werden
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt die Funktion zum Anzeigen der Gruppenmitglieder einer Gruppe aus
Sonderfall	Die Gruppenmitglieder können nicht angezeigt werden, da keine Gruppe angelegt ist
Sonstiges. Anmerkung	-

Tabelle 20: Beschreibung: Gruppenmitglieder anzeigen

Name	Person entfernen
Kurzbeschreibung	Der Nutzer möchte eine Person aus dem System/ Gruppe entfernen
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die Person wurde aus dem System/ Gruppe entfernt
Eingehende Daten	Gruppenmitglieder
Vorbedingung	Angemeldet + Gruppe ausgewählt
Nachbedingung	Die Person ist im System/ Gruppe nicht mehr vorhanden
Essentielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer Authentifiziert sich via des Google Dienstes.3. Der Nutzer wählt die Funktion zum Anzeigen der Gruppenmitglieder einer Gruppe aus4. Der Nutzer wählt die zu entfernende Person aus der Gruppe aus.5. Der Nutzer wählt die Funktion zum Löschen einer Kontaktliste aus
Sonderfall	Es sind keine Personen in einer Gruppe vorhanden
Sonstiges. Anmerkung	-

Tabelle 21: Beschreibung: Person entfernen

Name	Person hinzufügen
Kurzbeschreibung	Der Nutzer möchte eine Person einer Gruppe hinzufügen
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die Person wurde der Gruppe als Gruppenmitglied hinzugefügt
Eingehende Daten	Gruppe + Person



Vorbedingung	Angemeldet + Gruppe ausgewählt + Person ausgewählt
Nachbedingung	Der Gruppe wurde eine Person hinzugefügt
Essenzielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer authentifiziert sich via des Google Dienstes.3. eine Gruppe aus4. Der Nutzer zeigt sich die Gruppenmitglieder an5. Der Nutzer verwendet die Funktionen zum Hinzufügen einer Person zur Gruppe
Sonderfall	<ol style="list-style-type: none">1. Die Person ist nicht vorhanden.2. Die Person ist bereits Teil der Gruppe.
Sonstiges, Anmerkung	-

Tabelle 22: Beschreibung: Person hinzufügen

Name	E-Liste teilen
Kurzbeschreibung	Der Nutzer teilt seine erstellten Listen automatisch mit seinen Gruppenmitglieder der zugehörigen Gruppe
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Die E-Liste des Nutzers kann von Gruppenmitglieder bearbeitet, angezeigt, gelöscht, archiviert oder eine neue E-Liste erstellt werden
Eingehende Daten	Gruppenmitglieder + Inhalt der Liste
Vorbedingung	Angemeldet + Gruppe ausgewählt
Nachbedingung	<ul style="list-style-type: none">- E-Liste kann von anderen Gruppenmitgliedern eingesehen und bearbeitet werden.
Essenzielle Schritte	<ol style="list-style-type: none">1. Der Nutzer startet die App2. Der Nutzer authentifiziert sich via des Google Dienstes.3. Es wird die Funktion zum Erstellen einer Gruppe ausgewählt.4. E-Liste wird automatisch mit der ausgewählter Gruppe geteilt
Sonderfall	Der Nutzer ist kein Mitglied des E-Listenverwaltungssystem.
Sonstiges, Anmerkung	Das bearbeiten, archivieren oder löschen einer E-Liste kann durch alle Mitglieder der Gruppe durchgeführt werden.

Tabelle 23: Beschreibung: Einkaufsliste teilen



Use-Case-Diagramm: Durchsuchen der Einkaufsliste

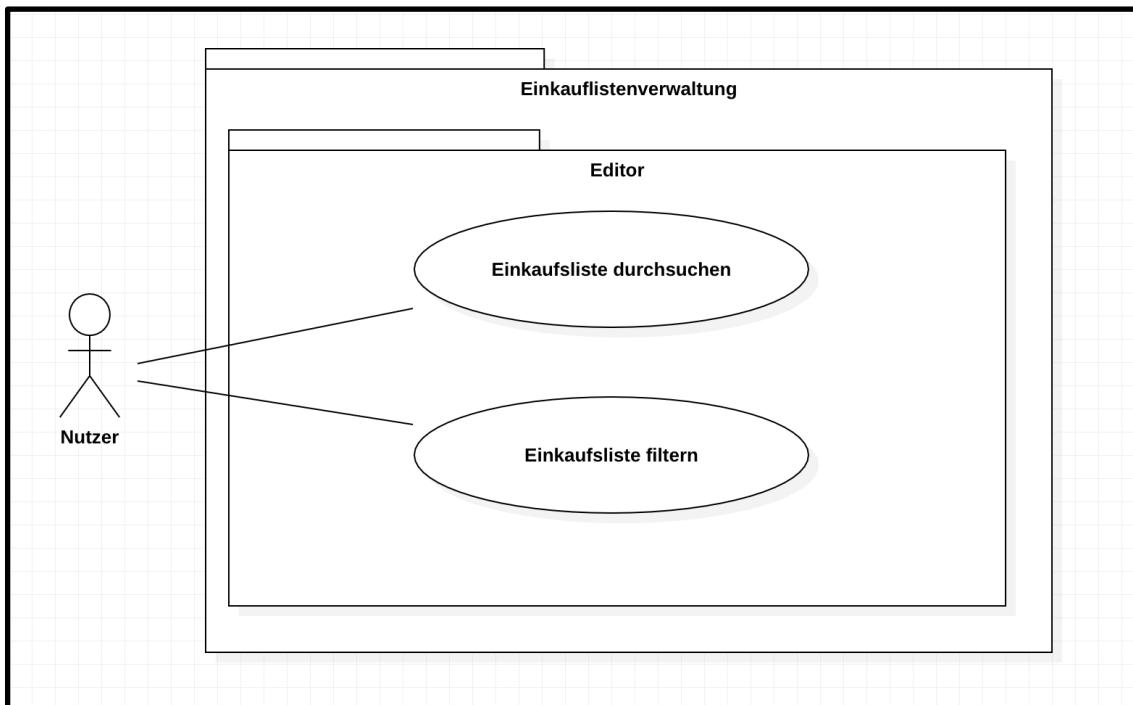


Abbildung 7: Scart - Use-Case: Durchsuchen der Einkaufsliste

Im letzten Use-Case-Diagramm: **Durchsuchen der Einkaufsliste**, wird beschrieben das eine Einkaufsliste mindestens die Funktionen Einkaufsliste durchsuchen und Einkaufsliste filtern, aufweisen muss. Der Nutzer bestimmt hier selbst nach welchen Kriterien er die Einkaufsliste filtert.

Name	E-Liste durchsuchen
Kurzbeschreibung	Der Nutzer möchte eine Einkaufsliste nach einem/ mehrerer Artikel durchsuchen
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Gesuchte/r Artikel
Eingehende Daten	Artikel, Zeitraum, Einzelhändler
Vorbedingung	Angemeldet
Nachbedingung	Dem Nutzer werden Artikel angezeigt, welche den gesuchten Kriterien entsprechen
Essentielle Schritte	<ol style="list-style-type: none">Der Nutzer startet die AppDer Nutzer Authentifiziert sich via des Google Dienstes.Es wird die Funktion zum durchsuchen einer E-Liste ausgewählt.Der Nutzer entscheidet nach Artikel in einem gewissen Zeitraum oder Einzelhändler.
Sonderfall	Artikel wird nicht gefunden
Sonstiges, Anmerkung	Die Ausgabe ist leer, wenn keine Gruppen oder Artikel angelegt wurden.

Tabelle 24: Beschreibung: Einkaufsliste durchsuchen



Name	E-Liste filtern
Kurzbeschreibung	Der Nutzer möchte eine E-Liste nach ausgewählten Filterkriterien aufrufen
Akteure	Nutzer
Auslöser	Nutzer
Ergebnisse	Artikel, auf die der Filter zutrifft
Eingehende Daten	Ausgewählter Filter die auf Artikel zutreffen
Vorbedingung	Angemeldet + Gruppe ausgewählt + Liste ausgewählt
Nachbedingung	Der Nutzer erhält alle Artikel die den Filterkriterien entsprechen
Essenzielle Schritte	<ol style="list-style-type: none">6. Der Nutzer startet die App7. Der Nutzer Authentifiziert sich via des Google Dienstes.8. Es wird die Funktion zum durchsuchen einer E-Liste ausgewählt.9. Der Nutzer entscheidet nach Artikel in einem gewissen Zeitraum oder Einzelhändler durch gesetzte Filter.
Sonderfall	<ol style="list-style-type: none">1. Keine Artikel vorhanden2. Keine Artikel entsprechen den Filterkriterien
Sonstiges, Anmerkung	Die Ausgabe ist leer, wenn keine Gruppen oder Artikel angelegt wurden

Tabelle 25: Beschreibung: Einkaufsliste filtern

Präsentationsschicht

Mockups

Wir haben uns am Anfang mit der Frage auseinandergesetzt, wie unser Zielsystem denn aussehen soll. Durch Mockups findet eine Visualisierung der Applikation statt, die bei der Entwicklung beträchtlich helfen kann. Dadurch war es möglich frühzeitig festzulegen, welche Formen/ Klassen benötigt werden und wie diese in Beziehung zueinander stehen. Im folgenden werden Mockups des Zielsystem im Anfangsstadium präsentiert. Diese weichen Teilweise von der jetzigen Visualisierung ab.

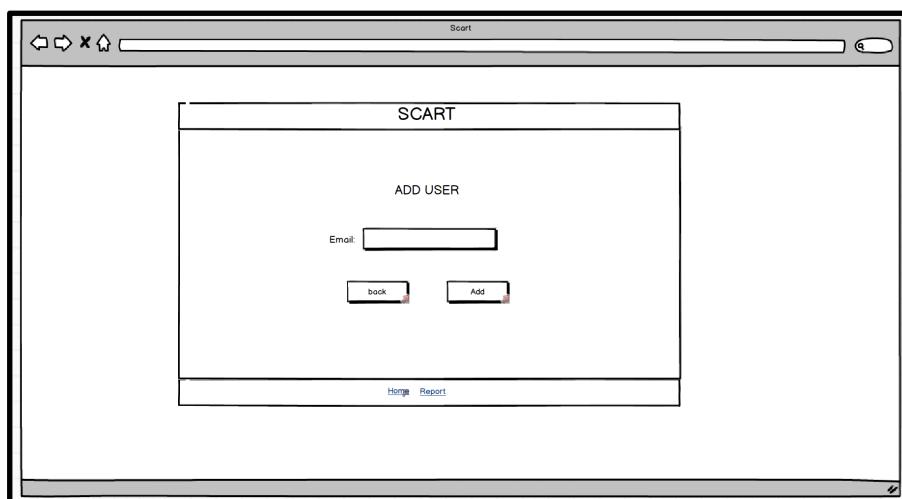


Abbildung 8: Editor: Nutzer dem System hinzufügen



In der Abbildung # werden Nutzer im Zielsystem erstellt. Zu jeder Form wurde damals solch eine Form erstellt. Bei Gruppen, Einkaufslisten, Einzelhändler und Artikeln wurde genau dieselbe Form im Mockup verwendet.

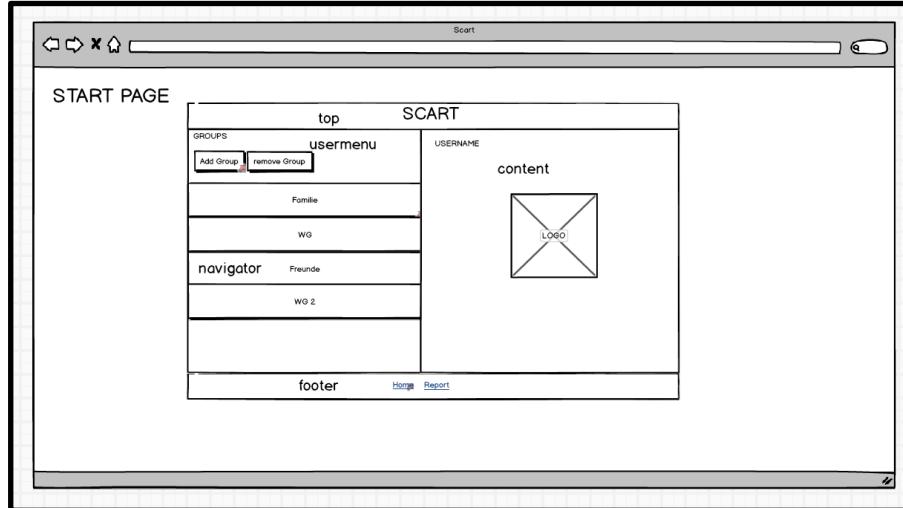


Abbildung 9: Editor: Homepage von Scart

In der Abbildung # wurde die Homepage visualisiert wie sie anfangs aussehen sollte. Auf der linken Seite im Navigator befinden sich die Gruppen des eingeloggten Nutzers. Im Footer konnte man jeder Zeit zur Homepage zurück springen. Zum Report Generator konnte ebenfalls via Footer gesprungen werden.

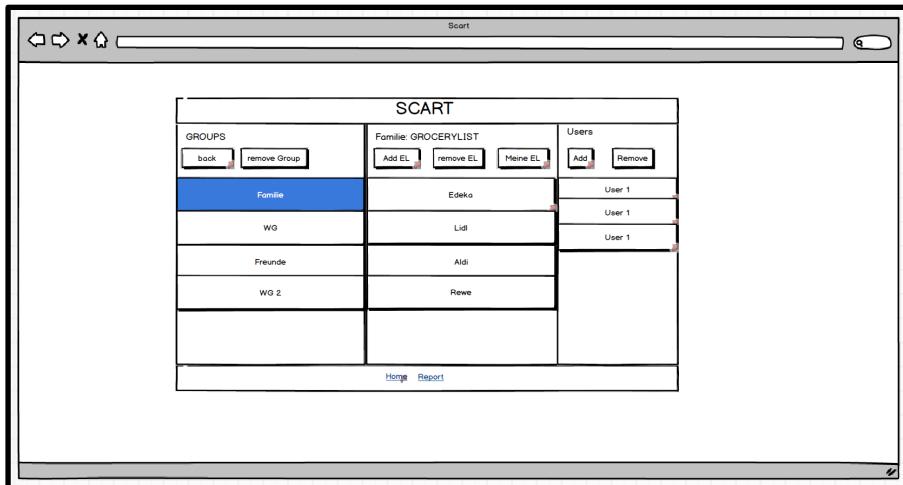


Abbildung 10: Editor: Gruppe mit deren Einkaufslisten und Usern

In der Abbildung # sind Gruppen mit deren Einkaufslisten und deren Mitglieder visualisiert. Durch das Anklicken der jeweiligen Gruppe öffnet sich in der mittleren Sektion alle angelegten Einkaufslisten der Gruppe. Durch das Anklicken der Einkaufsliste öffnet sich eine weiter Sektion mit allen Artikeln der angeklickten Einkaufsliste mit dem jeweiligen Einzelhändler.

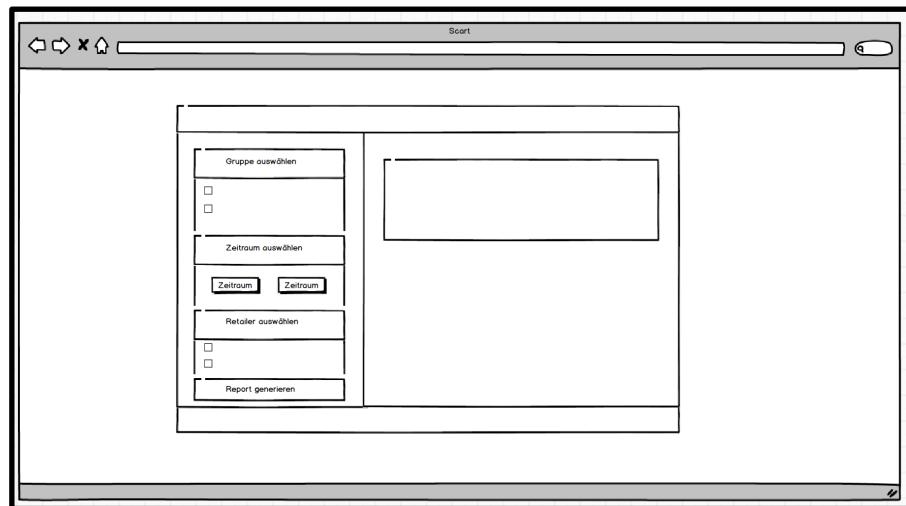


Abbildung 11: Report: Report-Generator

In der Abbildung # wird der Report visualisiert. Auf der linken Seite können Gruppen, Zeitraum und Retailer unabhängig ausgewählt werden. Dadurch wird auf klicken des "Report generieren" Buttons auf der rechten Seite eine statische Ausgabe ausgegeben die eine Statistik der häufig gekauften Artikel einer gewählten Gruppe des Nutzers generiert.

Klassendiagramme

Klassendiagramme sind ein weiteres Tool der Unified Modeling Language, die zur Visualisierung der einzelnen Klassen und deren Zusammenhänge dient. Die Klassendiagramme dienen Primär der Planung und Konzeptionsphase am Anfang eines Projektes. Sie sind eine Art Hilfestellung den Überblick zu behalten und etwaige Probleme vorher zu erkennen. Zu beachten ist dennoch, dass es sich hier nicht um ein vollständiges Entwurfsmuster handelt, sondern lediglich zur Hilfestellung dient. Da sich große Teilbereiche während der Entwicklung ergeben und die Klassendiagramme im Laufe der Entwicklung stets ändern, sollte darauf geachtet werden, die Diagramme immer aktuell zu halten, um Verwirrungen bei der Ausführung zu vermeiden.

Die Klassendiagramme werden zunächst mit Klassennamen benannt und mit Methoden einer Klasse erstellt. Diese Methoden innerhalb der erstellten Klasse erhalten einen Namen, notwendige Übergabeparameter mit deren festgelegten Datentypen und eventuell einem Rückgabewert. Die Attribute innerhalb der Klasse, stellen die Inhalte der Klasse dar.

Während der Konzeptionsphase wurden keine Klassendiagramme für die Grafische Oberfläche erstellt, da noch nicht der Umfang der Klassen und Formen bekannt war. Im Abschnitt Implementierung befinden sich sämtliche Formen der GUI.



Applikationsschicht

Klassendiagramme

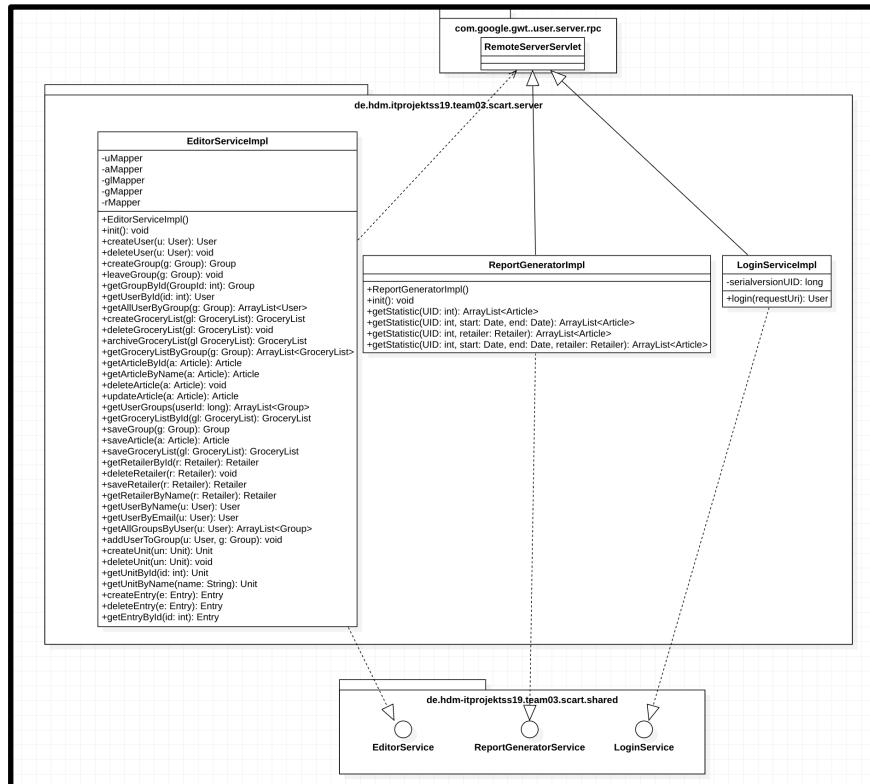


Abbildung 12: Klassendiagramm Applikationslogik - Server Package

In der Abbildung 12 sieht man das Klassendiagramm für das Server Package. Bei der Applikationslogik hat sich zumindest bei dem Umfang der Klassen wenig geändert. Nur die Klasse für die ServersideSettings wurde hinzugefügt. Größere Änderungen fanden innerhalb der Klassen EditorServiceImpl und ReportGeneratorImpl statt. Eine Vielzahl von neuen Methoden wurde hinzugefügt und sämtliche Abfragen an die Datenbank verarbeiten zu können.

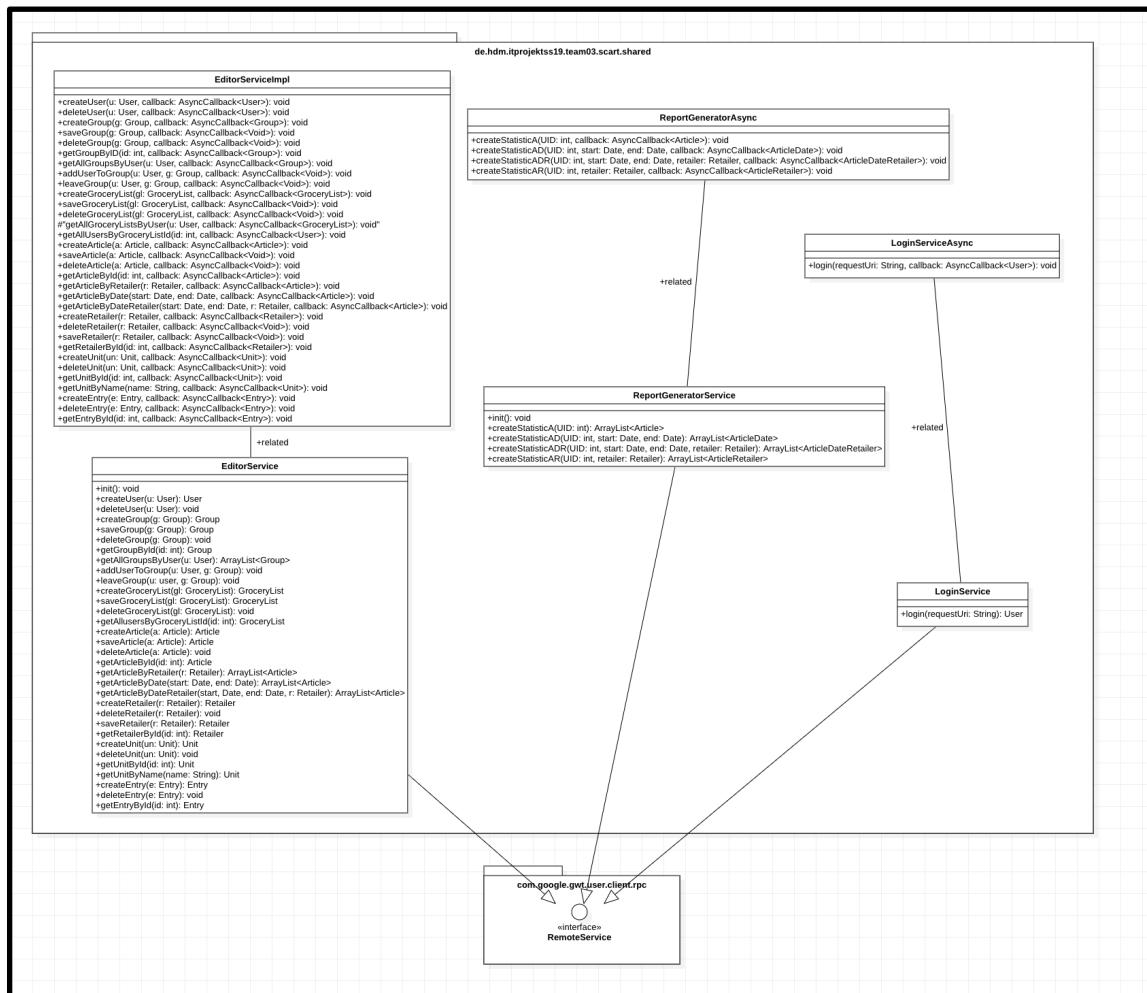


Abbildung 13: Klassendiagramm Applikationslogik - Shared Package

In der Abbildung #13 sieht man das shared Package. Dort wurde während der Entwicklung nur noch die Database Exception und die CommonSettings Klasse hinzugefügt. Da die EditorService/-Async und die ReportGenerator-/Async mit der mit den Klassen EditorserviceImpl/ReportGeneratorImpl zusammenhängt, sind auch hier eine Vielzahl von neuen Methoden hinzugekommen. Dies war unbedingt nötig um Abfragen für die Verbindungstabellen und neuen Business Objects zu bedienen.

Datenbankschicht

Es wurde eine relationale Datenbank mit SQL verwendet. Als Schnittstelle zur Datenbank wird hier der Java Database Connectivity Treiber (JDBC) verwendet. Durch Mamp und phpMyAdmin konnten Tests durchgeführt werden um die Richtigkeit der Mapper zu gewährleisten.

Entity-Relationship-Model

Ein Entity-Relationship-Model hilft bei der logischen Abbildung von Beziehungen zwischen Tabellen. Dieses Modell ist ebenfalls ein weiteres Hilfsmittel beim Entwurf. Mithilfe von diesem Datenmodell kann eine Typisierung von Objekten und deren zugehörigen Attributen stattfinden. Dieses Modell dient zur Analyse und dem Design von Datenstrukturen und ist



unabhängig von einem bestimmten Datenbanksystem. Die Beziehungen untereinander werden mithilfe der Chen - Notation gekennzeichnet. Zur Vollständigkeit gehört zu jedem Diagramm eine ausführliche Beschreibung des logischen Datenmodells.

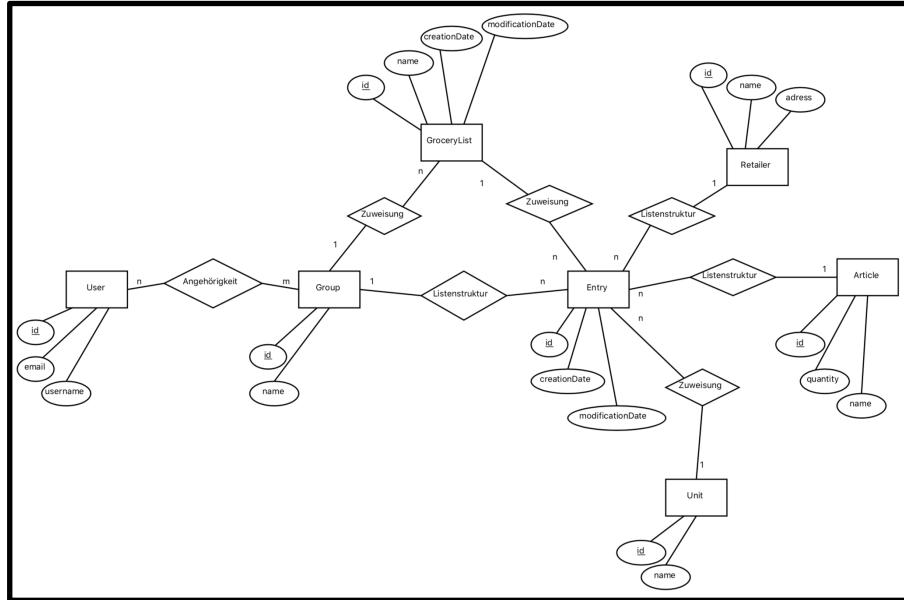


Abbildung 14: Entity-Relationship-Model

- Ein User ist Mitglied von ein oder mehreren Gruppen. Eine Gruppe hat ein oder mehrere User als Mitglieder.
- Eine Gruppe besitzt eine oder mehrere Grocery Lists. Eine Grocery List ist in genau einer Gruppe.
- Eine Grocery List besitzt ein oder mehrere Entry's. Ein Entry ist in genau einer Grocery List.
- Ein Entry gehört zu genau einem Retailer. Ein Retailer hat ein oder mehrere Entry's.
- Ein Entry hat genau einen Article. Ein Article hat ein oder mehrere Entry's.
- Ein Entry besitzt genau eine Unit. Eine Unit besitzt ein oder mehrere Entry's.

Das Entity-Relationship-Model in Abbildung 14 beschreibt die allererste Version bei der Entwurfsphase. Laufe der Entwicklung wurden immer mehr Attribute hinzugefügt. Zwischentabellen kamen ebenfalls dazu. Da etwaige n-m Beziehungen aufgetreten sind. Diese werden durch sogenannte Zwischentabellen aufgelöst um eine genau Zuordnung von Datensätzen zu gewährleisten. Unter dem Abschnitt Implementierung wird das erweiterte ERM(siehe Abbildung 25) verdeutlicht und beschrieben. Jede Tabelle besitzt eine Id. Diese wird benötigt, um einen Datensatz eindeutig zu kennzeichnen.



Klassendiagramme

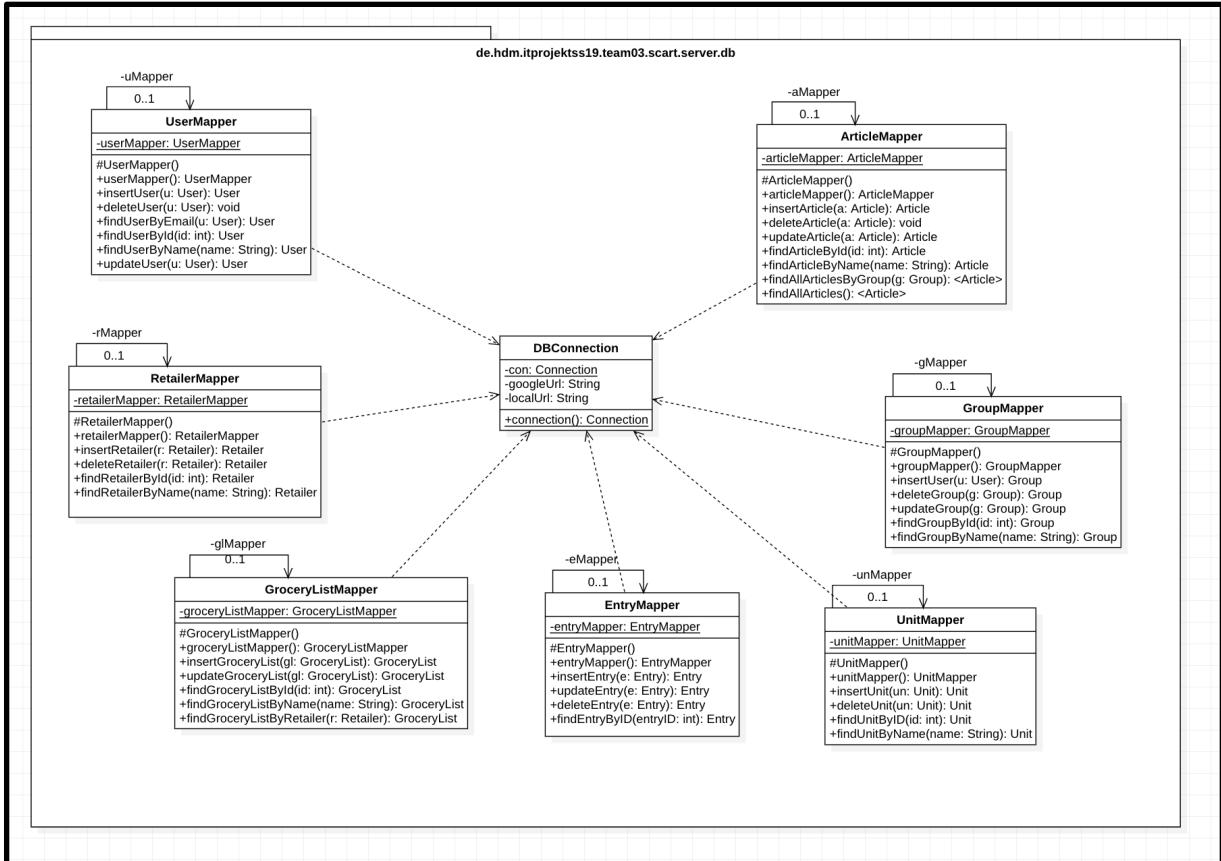


Abbildung 15: Klassendiagramm - Entwurf Mapper Klassen

Das Klassendiagramm in der Abbildung 15 beschreibt den Entwurf für die Datenbankanbindung. Ursprünglich war für jedes Business Object ein eigener Mapper geplant. Während der Entwicklung wurde klar das für Verbindungstabellen und neue Business Objects zusätzliche Mapper nötig sind. Zudem wurden Mapper für Unit- und Entry-BOs nicht mehr benötigt.

Implementierung

Die Implementierung wird nach der Code-Konvention, die von Oracle bereitgestellt wird, durchgeführt. Diese Konventionen dienen der Vorbeugung, unnötige Kosten an Wartung zu verlieren. Ca. 70-80% der Gesamtkosten einer Software geht an Wartung drauf. Darum ist es umso wichtiger eine einheitliche Struktur der unterschiedlichen Code-Abschnitte zu gewährleisten. Ebenso ist eine einheitliche Formatierung des Codes wichtig um die Lesbarkeit zu verbessern.

Google App Engine

Das Zielsystem wird mithilfe der Google App Engine auf einem entfernten Google Server gehostet und ausgeführt. Hier wird die Google Infrastruktur bezüglich Skalierung, Clustering übernommen.



Google Cloud SQL

Die Datenbank auf die das System zugreift läuft über einen Server der Google Cloud SQL. Diese Plattform bietet sämtliche Hosting- Services an, die auf einer Google- Hardware laufen. Damit der Dienst genutzt werden kann, wird das Abbild der Datenbank in einer SQL-Formdatei in die Google Cloud importiert. Zusätzlich müssen Änderungen in der DBConnection Klasse, die die URL und den Login beinhalten, stattfinden.

"Replikationen und Sicherungen zum Schutz Ihrer Daten sind ganz einfach konfigurierbar. Durch die zusätzliche Aktivierung von automatischem Failover wird Ihre Datenbank hochverfügbar. Ihre Daten werden automatisch verschlüsselt."

(Quelle: "Google", <https://cloud.google.com/sql/?hl=de>, 06.07.2019)

Git - Source Code Management

Zur Verwaltung des Sourcecodes und zur Zusammenarbeit wurde Git und das Eclipse Plugin Egit benutzt. Der Code wurde auf GitHub gespeichert. In der Regel arbeitet nur eine oder zwei Personen auf einem Branch. Branches werden für einzelne Funktionsbereiche("feature branch") erstellt. Wenn die Funktionsfähigkeit eines feature branch gewährleistet ist, wird in den Dev-Branch gemerged. Ist der Dev-Branch funktionsfähig, wird dieser periodisch in den Master-branch gepusht.

Package-Struktur

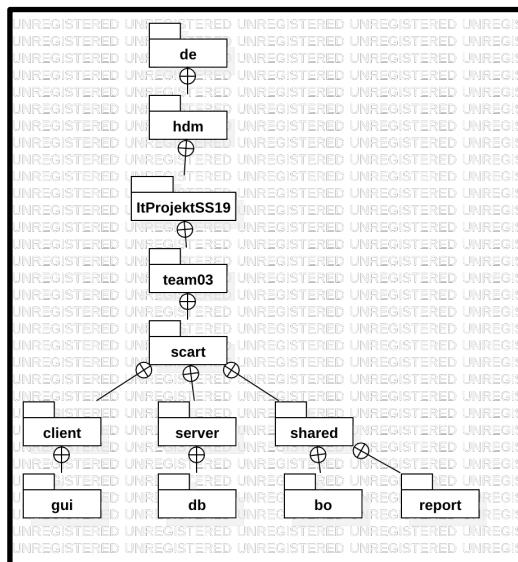


Abbildung 16: Package Struktur

Die Package-Struktur in der Abbildung 16, wurde anhand der Anforderungen und Vorgaben des Bank-Projektes der Begleitvorlesung erstellt. Die abgebildete Struktur stellt die Pakete des Projektes hierarchisch und bildlich dar. Das Paket **de.hdm.itprojektss19.team03.scart.client** ordnet die Klassen des Zielsystems ein, die für die Grafische Benutzeroberfläche zuständig sind. Diese Klassen werden hauptsächlich beim Aufruf in JavaScript kompiliert. Das Paket



de.hdm.itprojektss19.team03.scart.server ordnet die Klassen, die ausschließlich auf serverseitiger Ausführung zum Einsatz kommen. Diese beinhalten die Applikationslogik sowie den Editor als auch den Report. Ergänzend steht das Package `server.db` für die Datenbankverbindung und Kommunikation zur Verfügung. Das Package **de.hdm.itprojektss19.team03.scart.shared** steht sowohl für alle Klassen seitig des Servers als auch die des Clients zur Verfügung.

(Prof. Dr. Peter Thies, 2019)

Präsentationsschicht

Wir haben uns dafür entschieden, dass jede grafische Einheit eine eigene Form bekommt. Somit wird die Komplexität einzelner Klassen reduziert. Beispiel wäre der Report. Dieser wurde in `ReportFilterForm` und `ReportForm` aufgeteilt. Diese Aufteilung steigert die Übersichtlichkeit beträchtlich. Dieses Vorgehen unterstützt auch die Aufteilung unter den Teammitgliedern, wer für welche Form zuständig ist. Somit konnte die Produktivität in der Gruppe gesteigert werden und Merge Konflikte konnten so meist umgangen werden. Des Weiteren hilft dieses Vorgehen auch, dass die jeweilige Form eindeutig dem Business Objekt zugewiesen werden konnte.

Klassendiagramme

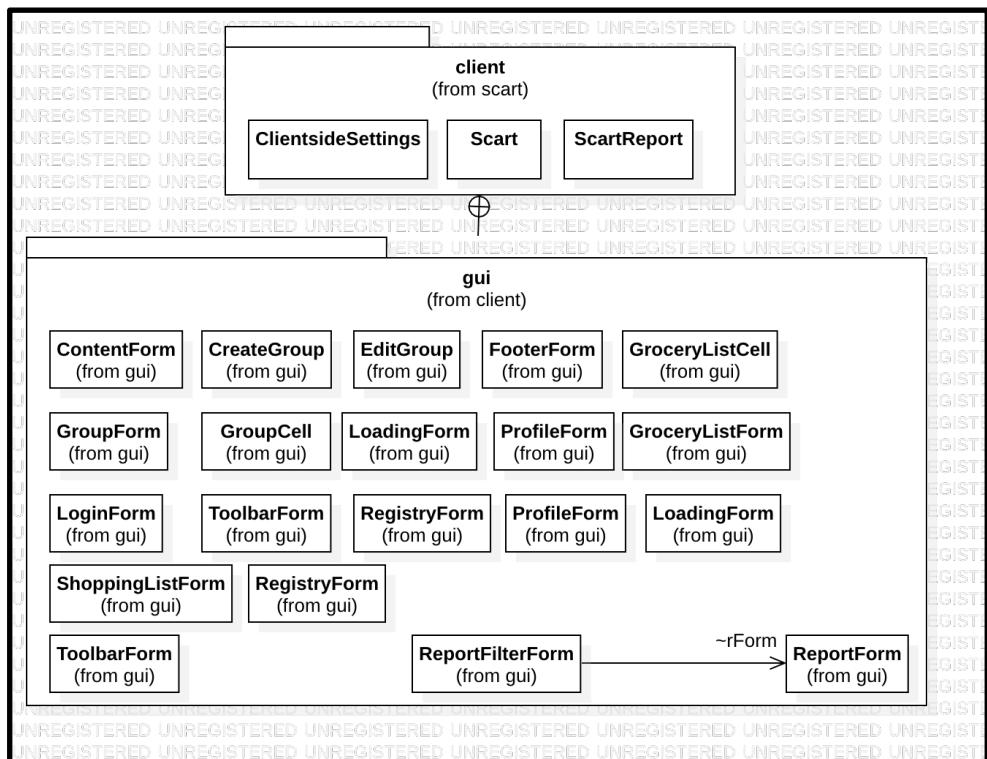


Abbildung 17: Klassendiagramm - GUI und Client Package

Wie auch in der Abbildung 17 zu sehen gab es vor allem bei der Präsentationsschicht im GUI und Client-Package große Änderungen. Sämtliche Seiten und Ansichten der Software wurden hier hinzugefügt. Auch für die Präsentation des ReportGenerators mussten neue Klassen erstellt werden.



Eine eigene Seite wurde immer auch in einer eigenen Klasse angeordnet um so eine klar strukturierte Trennung zu erreichen. Die nächste Seite wird bei der Anwendung dann durch einen Konstruktoraufruf mit den benötigten Parametern aufgerufen.

Applikationsschicht

Klassendiagramme

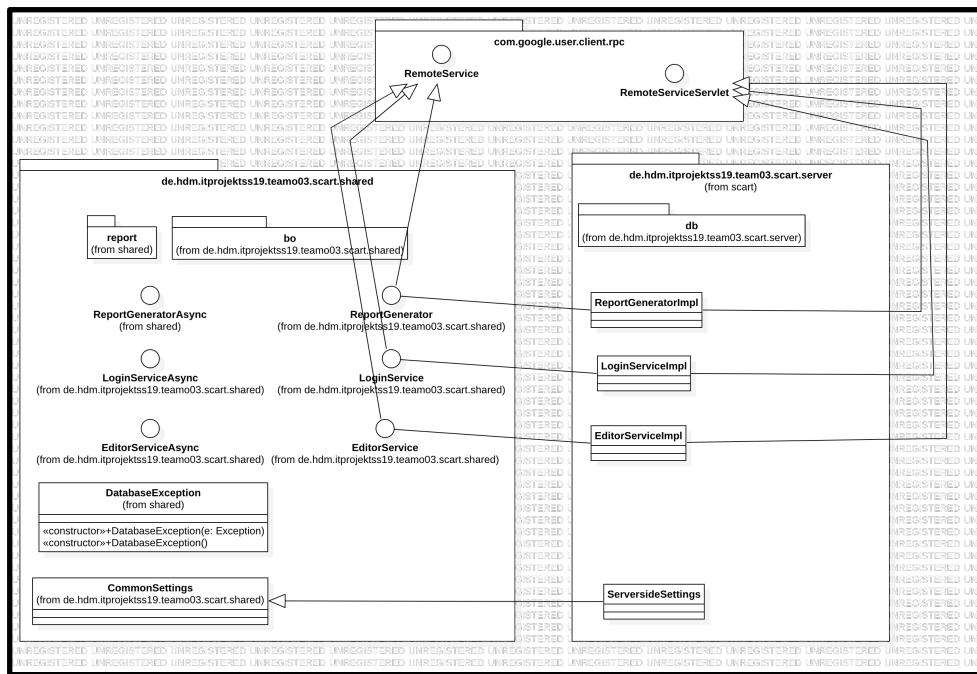


Abbildung 18: shared/server Zusammenhang

Die Klasse **EditorServiceImpl** stellt alle Methoden zur Verfügung, die Zugriff zum Server haben. In der Klasse sind sämtliche Funktionen der Applikationslogik gesammelt. In jeder Methode werden Daten aus den Parametern verarbeitet und mit Instanzen der Mapper-Klassen wird so ein Datenbank Zugriff ermöglicht.

EditorServiceImpl erbt aus der Klasse `RemoteServiceServlet` in dem Google-Package `com.google.user.client.rpc`. Zudem werden Methoden aus der `EditorService` Klasse implementiert.

Bei der Klasse **ReportGeneratorImpl** handelt es sich um das Gegenstück der `EditorServiceImpl`. Hier findet die Applikationslogik des Report-Generators statt. Auch diese Klasse erbt von `RemoteServiceServlet` und hat eine Klasse die alle Methoden zur Verfügung stellt, die `ReportGenerator`-Klasse.



Methoden	Beschreibung
+generateldenticons()	wurde hinzugefügt um die eindeutigen Icons von jedem User zu generieren
+encodeToString()	wurde hinzugefügt um die Erstellung der jeweiligen Icon zu ermöglichen
+getOwnProfile()	wurde hinzugefügt um das vollständige User-Objekt des aktuellen Users aus der Datenbank zu bekommen
+getAllGroupsByUser()	wurde hinzugefügt um alle Gruppen in der sich ein User befindet in Form eines Vektors aus der Datenbank zu bekommen
+addUserToGroup()	wurde hinzugefügt um einen User mit einer Gruppe über die GroupUser-Verbindungstabelle zu verknüpfen
+findAllGroups()	wurde hinzugefügt um für den ReportGenerator alle Gruppen die in der Datenbank gespeichert sind auszulesen
+findAllGroupsById()	wurde hinzugefügt um alle Gruppen in denen sich der User befindet zu bekommen. Die Abfrage läuft über die GroupUser-Tabelle.
+getAllUserByGroupId()	wurde hinzugefügt um alle User einer bestimmten Gruppe zu bekommen
+removeUserFromGroup()	wurde hinzugefügt um aus einer Gruppe auszutreten bzw. einen User aus der Gruppe zu entfernen
+getGroceryListByOwner()	wurde hinzugefügt um den in das GroceryList-Objekt des Owners zu finden
+findAllGroceryLists()	wurde hinzugefügt um alle GroceryLists aus der Datenbank für den ReportGenerator auszulesen
+getGroupByGroceryList()	wurde hinzugefügt um die Gruppe in der sich eine GroceryList befindet auszulesen
+getAllGroceryListsByGroupVector()	wurde hinzugefügt um alle GroceryLists von den Gruppen im Parameter-Vektor zurückzugeben



+findAllGroceryListByUserId()	wurde hinzugefügt um alle GroceryLists des Users zu bekommen
+findAllArticleByGroceryListId()	wurde hinzugefügt um alle Artikel die sich in einer GroceryList befinden zu erhalten
+getAllArticleByDateRetailer()	wurde hinzugefügt um alle Artikel eines Retailers in einem bestimmten Zeitfenster auszulesen
+findAllArticle()	wurde hinzugefügt um alle Artikel die angelegt wurden für den ReportGenerator auszulesen
+findAllArticleByGroupIdReport()	wurde hinzugefügt um alle Artikel die innerhalb einer Gruppe erstellt wurden aus der DB auszulesen
+findAllArticleByOwnerId()	wurde hinzugefügt um alle Artikel mit der OwnerId des Parameters zu bekommen
+getAllArticleByName()	wurde hinzugefügt um alle Artikel mit dem Namen des Parameters innerhalb einer Gruppe zu erhalten
+findAllArticleByFavouriteTRUE()	wurde hinzugefügt um alle Favoriten-Artikel der Gruppen im Parametervektor zu bekommen
+findAllArticleByRetailerFavouriteTRUE()	wurde hinzugefügt um alle Favoriten-Artikel eines bestimmten Retailers des Gruppen-Vektors zu bekommen
+findAllArticleByDateFavouriteTRUE()	wurde hinzugefügt um alle Favoriten-Artikel des Gruppen-Vektors innerhalb eines Zeitfensters auszulesen
+findAllArticleByDateRetailerFavouriteTRUE()	wurde hinzugefügt um alle Favoriten-Artikel eines Retailer- und Gruppen-Vektors innerhalb des Zeitfensters zu erhalten
+getAllArticleByDate()	wurde hinzugefügt um alle Artikel innerhalb des Zeitfensters im Parameter auszulesen
+findAllRetailer()	wurde hinzugefügt um alle angelegten retailer aus der Datenbank auszulesen
+deleteArticleFromAllLists()	wurde hinzugefügt um einen Artikel global aus allen GroceryLists zu entfernen

Tabelle 26: Beschreibung: EditorServiceImpl

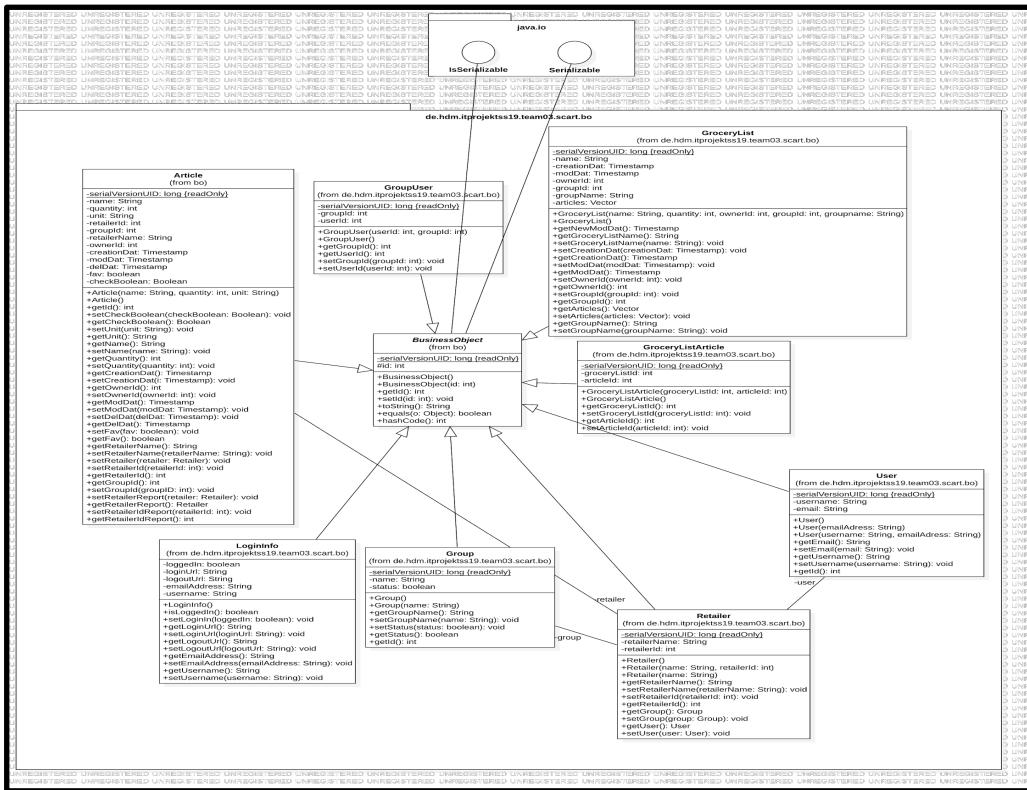


Abbildung 19: Business Object Klassen: `shared.bo`

In der Abbildung # sind alle Klassen zu sehen die sich in dem `shared.bo`-Package befinden und deren Zusammenhänge. Es ist erkennbar, dass jede Klasse von der Super-Klasse `BusinessObject` grundlegende Methoden erben. Zudem sind während des Entwicklungsverlaufs neue BusinessObjects hinzugekommen. Zum Beispiel `GroupUser` und `GroceryListArticle` wurden hinzugefügt. Diese sind wichtig um den Zusammenhang zwischen Usern und der Gruppe bzw. den Artikel und der `GroceryList` darzustellen.

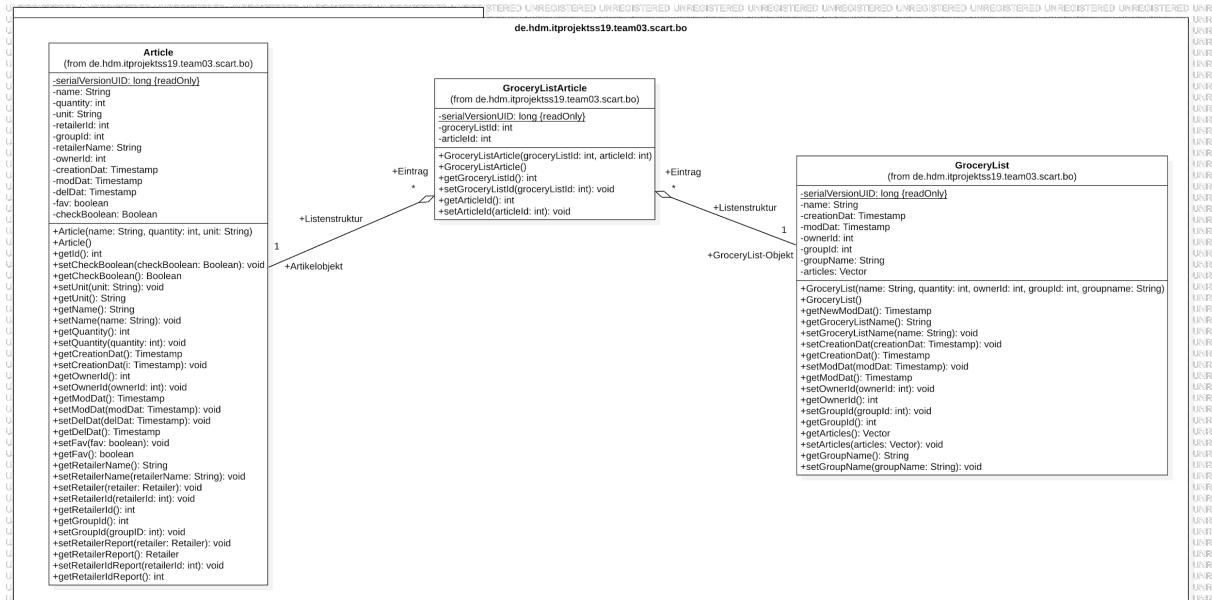


Abbildung 20: Business Object Klassen: *GroceryListArticle*

Die Abbildung 20 zeigt das Klassendiagramm des GroceryListArticle-BusinessObjects. Die Klasse GroceryListArticle beschreibt die Struktur und die Einträge die in einer GroceryList enthalten sind. Sie stellt die Verbindung zwischen dem Artikel und der GroceryList her. So sind in jeder GroceryList beliebig viele GroceryListArticle-Objekte und jedes GroceryListArticle-Objekt ist mit einem Artikel-Objekt verknüpft.

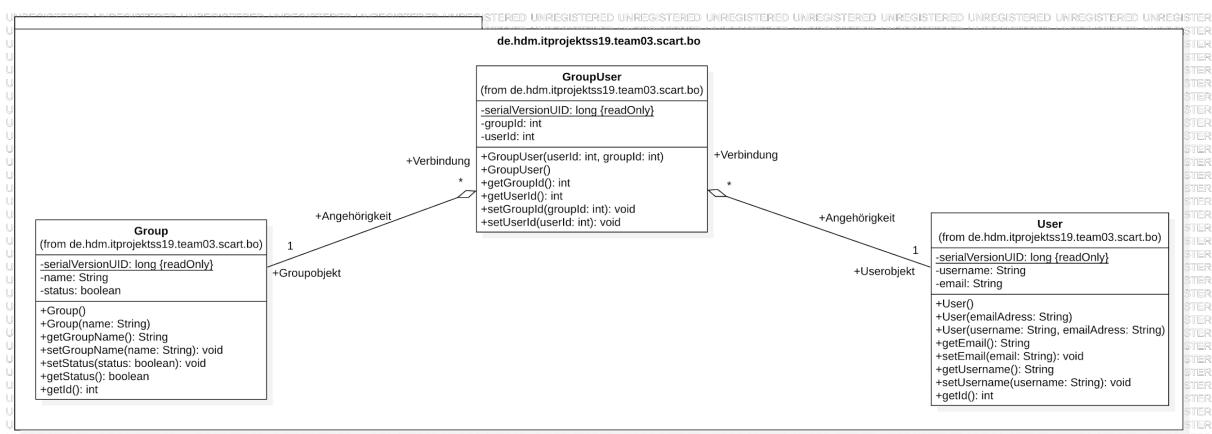


Abbildung 21: Business Object Klassen: Gruppen / User

In dem Klassendiagramm in der Abbildung 21 ist das Klassendiagramm des GroupUser-BusinessObject dargestellt. Dieses BusinessObject wurde hinzugefügt um eine Struktur zwischen einer Gruppe und deren Mitgliedern sicherzustellen. Über eine solche Verbindung in der Datenbank ist klar ersichtlich welche Mitglieder sich in einer Gruppe befinden. Für Abfragen bei der Präsentationsschicht oder dem ReportGenerator kommt dieses BusinessObject bzw. dessen Verbindungstabelle oft zur Verwendung.



GroupUser:

Die GroupUser Klasse wurde mit folgenden Methoden komplett neu implementiert um bei der Datenbankabfrage die Komplexität zu reduzieren. Folgende Methoden wurden implementiert:

Methoden	Beschreibung
+setGroupId	wurde hinzugefügt um die Id einer Gruppe zu setzen die in Verbindung mit einer UserId steht. Somit entsteht in der Datenbank ein zusammengesetzter Primärschlüssel zur eindeutigen Identifizierung einzelner Datensätze.
+getGroupId	wurde hinzugefügt um die Id einer Gruppe zu setzen die in Verbindung mit einer UserId steht. Somit entsteht in der Datenbank ein zusammengesetzter Primärschlüssel zur eindeutigen Identifizierung einzelner Datensätze.
+setUserId	wurde hinzugefügt um die Id einer Gruppe zu setzen die in Verbindung mit einer UserId steht. Somit entsteht in der Datenbank ein zusammengesetzter Primärschlüssel zur eindeutigen Identifizierung einzelner Datensätze.
+getUserId	wurde hinzugefügt um die Id einer Gruppe zu setzen die in Verbindung mit einer UserId steht. Somit entsteht in der Datenbank ein zusammengesetzter Primärschlüssel zur eindeutigen Identifizierung einzelner Datensätze.

Tabelle 27: Beschreibung: GroupUser

Retailer:

Die Retailer Klasse implementiert folgende neue Methoden (siehe Tabelle 28). Diese Methoden helfen um einem Retailer einen Owner zuzuweisen. Somit können später User und Einkaufslisten mit dem Retailer und einer Gruppe verknüpft werden.



Methoden	Beschreibung
+setGroup	wurde hinzugefügt um die Verknüpfung zwischen Retailer und einer Gruppe zu definieren
+getGroup	wurde hinzugefügt um die Verknüpfung zwischen Retailer und einer Gruppe zu definieren
+setUser	wurde hinzugefügt um die Verknüpfung zwischen Retailer und einem User zu definieren
+getUser	wurde hinzugefügt um die Verknüpfung zwischen Retailer und einem User zu definieren

Tabelle 28: Beschreibung: Retailer

GroceryList:

Die GroceryList Klasse implementiert nach der Konzeptionsphase einige neue Methoden. In der Konzeptionsphase wurden nur allgemeine Methoden hinzugefügt die für das Zielsystem nicht ausgereicht haben. Darum sind folgende neue Methoden hinzugefügt worden:

Methoden	Beschreibung
+getGroceryListByName	wurde hinzugefügt um Einkaufslisten via Name zu suchen. (weiter option der Filterung)
+setOwnerId	wurde hinzugefügt um einer Einkaufsliste einen Besitzer zuzuweisen, wer denn die Einkaufsliste in der Gruppe die Article zu besorgen hat
+getOwnerId	wurde hinzugefügt um einer Einkaufsliste einen Besitzer zuzuweisen, wer denn die Einkaufsliste in der Gruppe die Article zu besorgen hat
+setGroupId	wurde hinzugefügt um einer Liste einer Gruppe zuzuweisen.
+getGroupId	wurde hinzugefügt um einer Liste einer Gruppe zuzuweisen.
+setGroupName	wurde hinzugefügt um den Namen der



	Gruppe einer EInkaufsliste zu setzen.
+getGroupName	wurde hinzugefügt um den Namen der Gruppe einer EInkaufsliste zu setzen.
+setModDat	wurde hinzugefügt um ein Modifikationsdatum zu setzen.
+getModDat	wurde hinzugefügt um ein Modifikationsdatum zu holen.
+setCreationDat	wurde hinzugefügt um ein Erstellungsdatum festzulegen.
+getCreationDat	wurde hinzugefügt um ein Erstellungsdatum festzulegen.

Tabelle 29: Beschreibung: *GroceryList*

Article:

Folgende Methoden wurden implementiert um die Synergie zwischen drei verschiedenen Klassen zu wahren. Im Folgenden werden nicht alle Methoden die neu implementiert wurden aufgelistet. Das würde den Rahmen der Beschreibung sprengen und wäre hier nicht Zielführend. Die implementierten Methoden lauten wie folgt:

Methoden	Beschreibung
+setUnit	wurde hinzugefügt da die Klasse Unit entfernt wurde aber der Article eine Unit benötigt.
+getUnit	wurde hinzugefügt da die Klasse Unit entfernt wurde aber der Article eine Unit benötigt.
+setDelDat	wurde hinzugefügt um zu gewährleisten, dass ein Article archiviert wird und nicht vollständig aus der Datenbank gelöscht wird.
+getDelDat	wurde hinzugefügt um zu gewährleisten, dass ein Article archiviert wird und nicht vollständig aus der Datenbank gelöscht wird.
+setFav	wurde hinzugefügt um einen Article als Favorit zu markieren. Wird Beim Report benötigt für die Filterung.



+getFav	wurde hinzugefügt um einen Article als Favorit zu markieren. Wird beim Report benötigt für die Filterung.
+setOwner	wurde hinzugefügt um einen Owner zuzuweisen der den Article besorgt.
+getOwner	wurde hinzugefügt um einen Owner zuzuweisen der den Article besorgt.

Tabelle 30: Beschreibung: Article

GroceryListArticle:

Wie auch schon bei **GroupUser** wurde diese Klasse implementiert um die Komplexität zweier Tabellen zu reduzieren. Folgende Methoden wurden implementiert:

Methoden	Beschreibung
+setGorceryListId	wurde hinzugefügt für die Verknüpfung zwischen Article und GorceryList
+getGorceryListId	wurde hinzugefügt für die Verknüpfung zwischen Article und GorceryList
+setArticleId	wurde hinzugefügt für die Verknüpfung zwischen Article und GorceryList
+getArticleId	wurde hinzugefügt für die Verknüpfung zwischen Article und GorceryList

Tabelle 31: Beschreibung: GroceryListArticle

LoginInfo:

Das Business Objekt **LoginInfo** wurde hinzugefügt um festzustellen, ob der Nutzer der das Zielsystem verwendet, eingeloggt ist oder die Berechtigung besitzt sich auf das Zielsystem einzuloggen und zusätzlich die Login URL zu erhalten.

Methoden	Beschreibung
+isLoggedIn	wurde hinzugefügt um zu prüfen ob der Nutzer eingeloggt ist oder nicht.
+set/getLoginIn	wurde hinzugefügt um den Login info zuholen/setzen
+set/getLoginURL	wurde hinzugefügt um die Login URL zu



	setzen/ holen
+get/setEmailAddress	wurde hinzugefügt um die Email Adresse des Nutzers zu setzen/ holen.
+set/getUsername	wurde hinzugefügt um den Usernamen des Nutzer zu setzen/ holen.

Tabelle 32: Beschreibung: LoginInfo

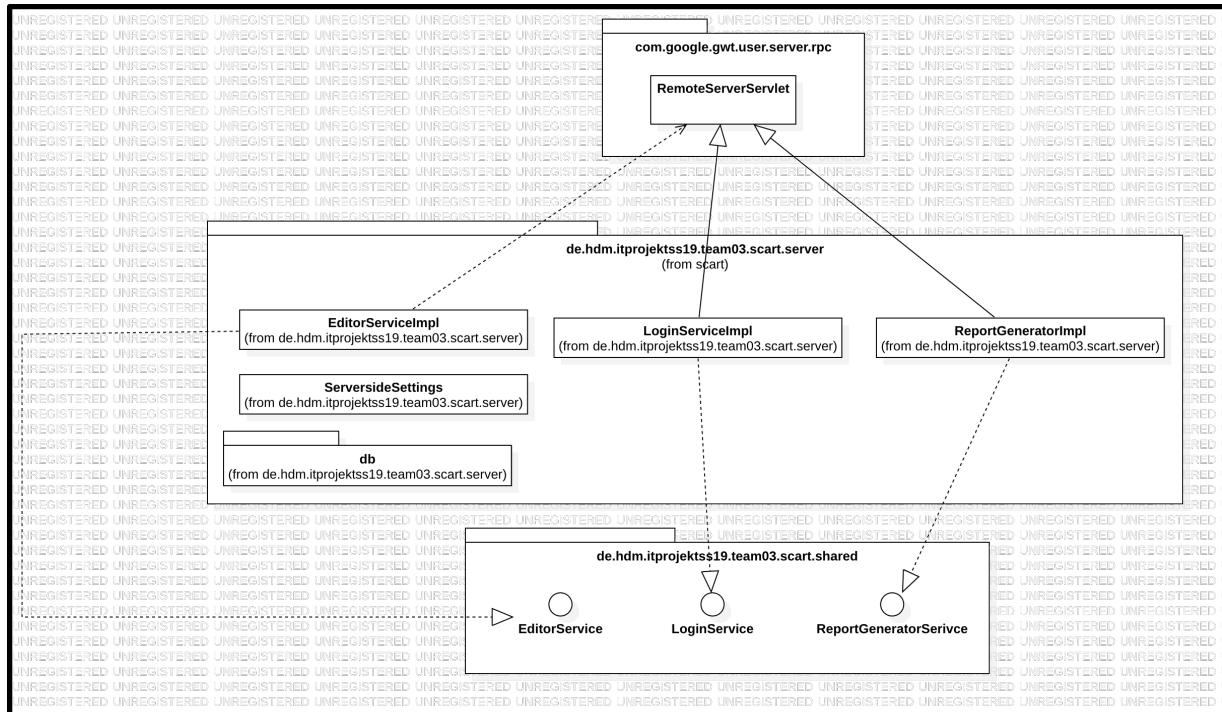


Abbildung 22: Klassendiagramm - server

In der Abbildung 22 ist das Klassendiagramm für die Klassen im Server Package zu sehen. Während der Entwicklungsphase wurden die Klassen EditorServiceimpl, LoginServiceimpl und ReportGeneratorimpl immer wieder ergänzt um die Funktionsfähigkeit der Software mit immer neuen Methoden zu gewährleisten.

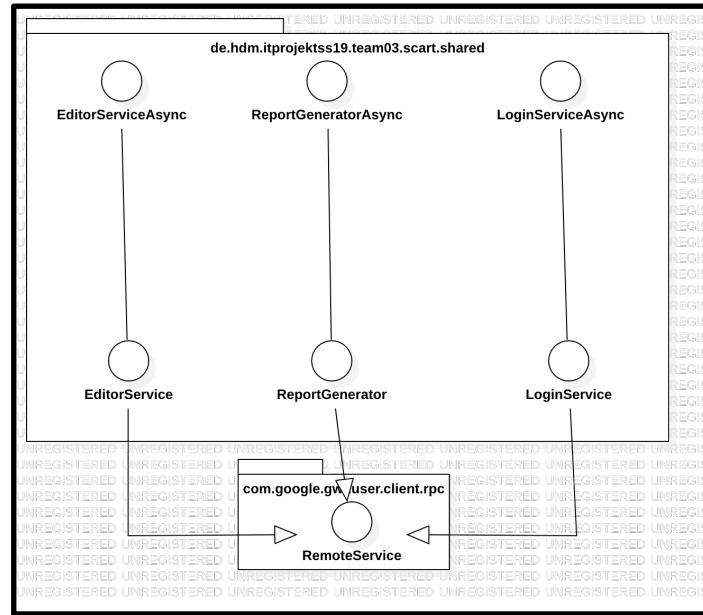


Abbildung 23: Klassendiagramm - shared

In der Abbildung 23 ist das endgültige Klassendiagramm der Klassen des shared-Packages. Im Vergleich zum Entwurf hat sich der generelle Aufbau hier nicht wirklich verändert. Inhaltlich wurden aber in jeder Klasse mehrere Methoden hinzugefügt um ständig neuen Anforderungen nachzukommen. Vor allem für den ReportGenerator wurden neue Methoden hinzugefügt um Daten aus der gesamten Datenbank auszulesen.

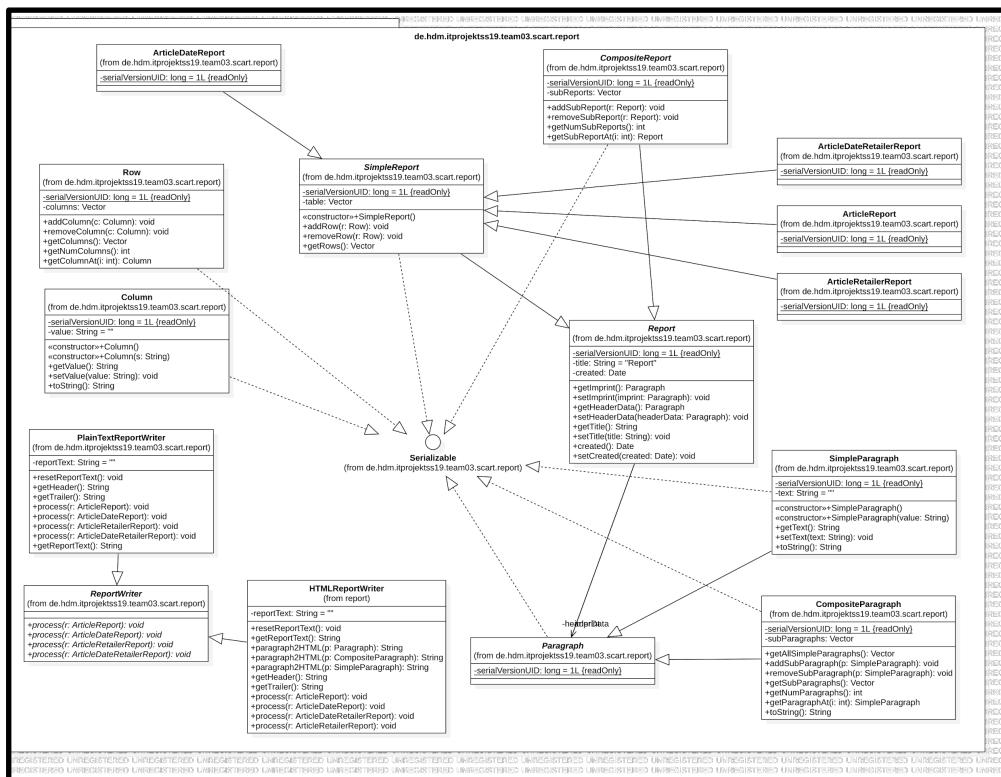


Abbildung 24: Klassendiagramm - shared.report



Das Klassendiagramm in der Abbildung # zeigt das Package shared.report und stellt die Klassen dar, die für den Aufbau des Reports notwendig sind. Das Klassendiagramm zeigt die Klassen, die aufgerufen werden, die für die statische Ausgabe eines Reports benötigt werden.

In der Konzeptionsphase war es nur eine Klasse namens Article, die die CompositeReport Klasse erweitert. Um strikt die Filtereigenschaften zu trennen und die Komplexität so gering wie möglich zu gestalten, haben wir uns in der Implementierung dafür entschieden, für jede Filterart eine neue Klasse im Report zu hinterlegen. Somit wurde der Report um 4 neue Klassen erweitert. Die prozessierten Klassen ArticleReport, ArticleDateReport, ArticleRetailerReport und ArticleDateRetailerReport erweitern die Klasse SimpleReport, welche wiederum die Klasse Report erweitert.

Die Klasse SimpleReport verbindet die Klasse Row und diese wiederum die Klasse Column, die für die Darstellung der einzelnen Tabellen im Report mit deren Inhalt verantwortlich sind. SimpleParagraph und CompositeParagraph erweitern die Klasse Paragraph, die für die textuelle Ausgabe verantwortlich sind.

Der CompositeReport steht in jedem Zusammenhang für eine Sammlung vieler Teilklassen, die über diese Klassen gebündelt aufgerufen werden können. Die Klassen HTMLReportWriter und PlainTextReportWriter implementieren die abstrakten Methoden zur Erstellung des Reports. Alle weiteren Klassen, die über das Netzwerk an den Client kommuniziert werden, implementieren das Interface Serializable, welches die Klassen als serialisierbar markieren.

Datenbankschicht

Entity-Relationship-Model

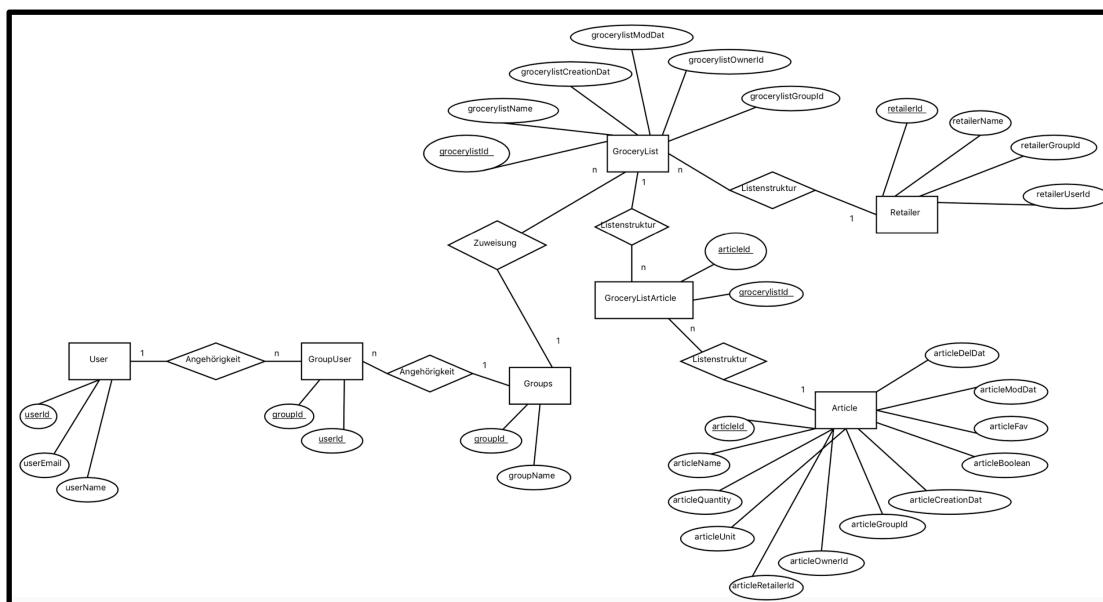


Abbildung 25: Entity-Relationship-Model



Das Finale Entity-Relationship-Model in Abbildung # wurde in 7 Tabellen aufgeteilt. Die Tabellen **GroceryListArticle** und **GroupUser** besitzen einen zusammengesetzten Schlüssel, da diese erstellt wurden um eine vorhandene n zu m Beziehung aufzulösen. N zu m Beziehungen sind sehr komplex und in dieser Form kaum zu verwalten. Durch Fremdschlüssel Beziehungen sind alle Klassen (Tabellen) miteinander verbunden und können so untereinander agieren. Der Primärschlüssel (`id`) wird via `AUTO_INCREMENT` automatisch hochgezählt um die Eindeutigkeit zu wahren.

Klassendiagramme

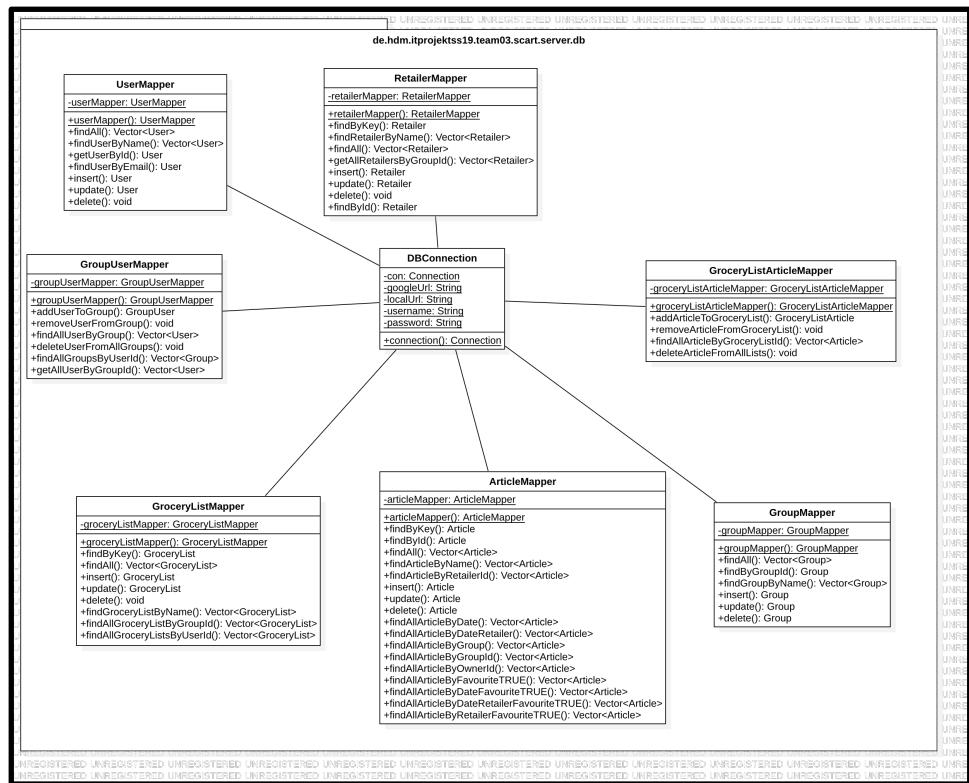


Abbildung 26: Klassendiagramm - Datenbank-Mapper Serverseitige Klassen

Die Mapper-Klassen werden genutzt um SQL-Befehle zur Datenbank zu senden. Die **DBConnection** Klasse beinhaltet die genauen Attribute um eine Verbindung zur Datenbank herzustellen. Jede Mapper Klasse ist mit der **DBConnection** verbunden um über die `connection()`-Methode Befehle an die Datenbank zu senden.

Folgende Mapper-Klassen wurden im Laufe der Implementierung hinzugefügt/ entfernt:

- **GroceryListArticleMapper** und **GroupUserMapper** wurden wegen Komplexitätsreduktion zu den Mappern hinzugefügt.
- **EntryMapper** und **UnitMapper** wurden wegen unnötiger Implementierung entfernt. Unit wurde ein Attribut des BO's Article.



GroupUserMapper:

Methoden	Beschreibung
+addUserToGroup	wurde hinzugefügt um einen Nutzer einer Gruppe hinzuzufügen und eine Verknüpfung zwischen den Tabellen Group und User herzustellen.
+removeUserFromGroup	wurde hinzugefügt um einen Nutzer von einer Gruppe zu entfernen und die Verknüpfung zwischen den Tabellen User und Group aufzulösen.
+findUserFromGroup	wurde hinzugefügt um einen Nutzer einer Gruppe zu finden.
+findAllUserFromAllGroups	wurde hinzugefügt um Alle Nutzer einer Gruppe auszugeben.
+findAllGroupsByUserId	wurde hinzugefügt um alle Gruppen eines Nutzers auszugeben.
+getAllUserByGroupId	wurde hinzugefügt um alle Nutzer einer Gruppe via eindeutiger Id auszugeben.

Tabelle 33: Beschreibung: GroupUserMapper

GroceryListArticleMapper:

Methoden	Beschreibung
+addArticleToGroceryList	wurde hinzugefügt um es zu erlauben einen Article mit einer GroceryList zu verknüpfen.
+findAllArticleByGroceryListId	wurde hinzugefügt um alle Article einer gesuchten GroceryList auszugeben.
+removeArticleFromGroceryList	wurde hinzugefügt um eine Verknüpfung zwischen Article und GroceryList aufzulösen.

Tabelle 34: Beschreibung: GroceryListArticleMapper



RetailerMapper:

Methoden	Beschreibung
+findAllRetailerByGroupId	wurde hinzugefügt um alle Retailer innerhalb einer Gruppe auszugeben.

Tabelle 35: Beschreibung: RetailerMapper

EntryMapper:

Beschreibung
Dieser Mapper wurde entfernt, da dieser Mapper nicht benötigt wurde. Das Problem mit der Eindeutigkeit wurde durch eine andere Zwischentabelle gelöst.

Tabelle 36: Beschreibung: EntryMapper

Test

Präsentationsschicht

Es wurde ein Dummy-Projekt in Eclipse erstellt um einzelne Klassen sowie einzelne Funktionen von der Applikation abzugrenzen. Somit konnten einzelne Elemente wie ClickHandler und Buttons auf ihre Funktion getestet werden. Wenn Fehler auftraten, konnte der Fehler schnell lokalisiert werden. Die Darstellung von Editor und Report wurden jeweils getrennt voneinander getestet. Aber auch das Zusammenspiel zwischen Editor und Report wurden getestet.

Applikationsschicht

Der korrekte Programmablauf von Methoden und Klassen wurde mit Hilfe von der Eclipse Console oder Window alerts auf Richtigkeit geprüft.

Datenbankschicht

Die Datenbank wurde mithilfe einer Test.class im Projekt selbst mehrfach überprüft. Durch Mamp und phpMyAdmin konnten die Mapper unabhängig voneinander überprüft werden. Ergebnisse wurden via Eclipse Console evaluiert.

Ratgeberliteratur

Benutzungshandbuch

Das Benutzungshandbuch dient zur Visualisierung des Shared-Shopping-List-Systems "**Scart**". Mit Hilfe von Scart können Sie Ihre Einkaufstouren präzise Planen. Scart ist übersichtlich, einfach zu bedienen und organisatorisch eine der besten Applikationen auf dem Markt. Es



können Gruppen erstellt, editiert und gelöscht werden. Es können beliebig viele Nutzer via E Mail zur Gruppe hinzugefügt werden. In diesen Gruppen können Sie ihre Einkaufstouren organisieren. Sie können Einkaufslisten mit spezifischen Einzelhändlern erstellen. Sie können Einzelne Einkaufslisten Gruppenmitglieder zuweisen um zu bestimmen, wer für welche Einkaufsliste zuständig ist. Am Ende können Sie eine Übersicht der häufig gekauften Artikel über den Report tätigen und so stets ihre Statistik im Auge behalten.

Die folgende Beschreibung des Zielsystems soll den Aufbau der Applikation verdeutlichen, sowie den gebotenen Leistungsumfang darstellen.

Login

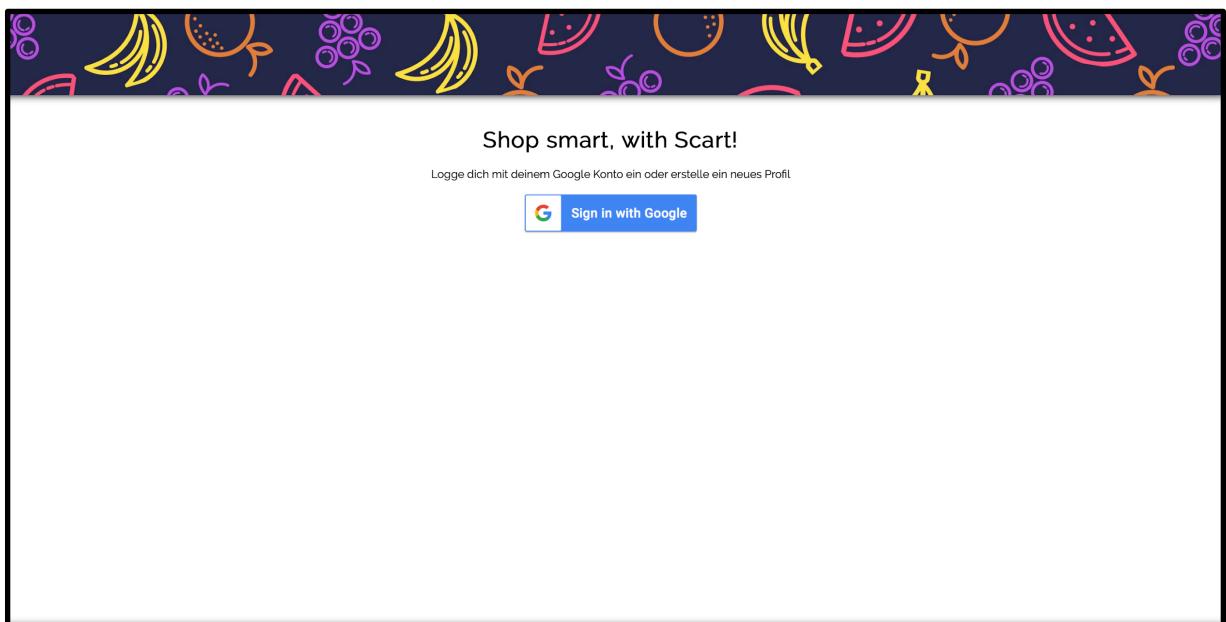


Abbildung 27: Scart - Login

In der Abbildung 27 werden Sie mit dem Login von Scart begrüßt. Dort werden Sie gebeten sich mit ihrer Google Mail Adresse anzumelden. Mit Klick auf den Button "**Sign in with Google**" werden Sie zum Google Login weitergeleitet.



The screenshot shows the Google sign-in interface for the Scart system. At the top is the Google logo. Below it is the heading "Anmeldung" (Login) and the sub-instruction "Mit Google-Konto anmelden." (Log in with Google account). There is a text input field labeled "E-Mail oder Telefonnummer" (Email or phone number). Below the input field is a blue link "E-Mail-Adresse vergessen?" (Forgot email address?). A note in small text says "Nicht Ihr Computer? Dann können Sie für die Anmeldung ein Fenster zum privaten Surfen öffnen." (Not your computer? Then you can open a window for private browsing for logging in.) followed by a blue link "Weitere Informationen" (More information). At the bottom of the form are two buttons: "Konto erstellen" (Create account) and a larger blue "Weiter" (Continue) button. Below the form are language links "Deutsch ▾", "Hilfe", "Datenschutz", and "Nutzungsbedingungen".

Abbildung 28: Scart - Google Login

Nach der Weiterleitung in das Sign - In - Formular, muss die bei Google hinterlegte Google Mail eingetragen werden. Nach der Eintragung der Mail und dem Passwort werden Sie zu Scart weitergeleitet.

Warnung! Es muss sich um eine Google Mail Adresse handeln, die bei Google schon existiert. Wenn Sie bei Google noch nicht Registriert sind, müssen Sie ein neues Konto auf Google selbst zuerst erstellen, bevor Sie das **Shared-Shopping-List-System Scart** verwenden können.

The screenshot shows the Scart registration form. At the top is a decorative banner with various fruit icons like bananas, watermelons, and grapes. Below the banner, a message reads "Wie es scheint bist du neu hier bei SCart, herzlich Willkommen!" (It seems you are new here at SCart, welcome!). The registration form has two input fields: "Wähle einen Nutzernamen" (Choose a username) and "patrick@hotmail.de" (an example email address). At the bottom are two buttons: "Zurück zum Login" (Back to login) and "Registrieren" (Register).

Abbildung 29: Scart - Registrierungsformular



Nach dem Login im System, wird im Hintergrund geprüft, ob Sie bereits ein Konto auf Scart besitzen und ob ihre Mail Adresse unserem System bekannt ist. Wenn Sie sich zum ersten Mal anmelden, werden Sie zum Registrierungsformular weitergeleitet. Dort müssen Sie nur noch ihren Namen eintragen und auf den Button "**Registrieren**" klicken. Wenn Sie aber bereits registriert sind, werden Sie automatisch zur Homepage weitergeleitet.

Editor:

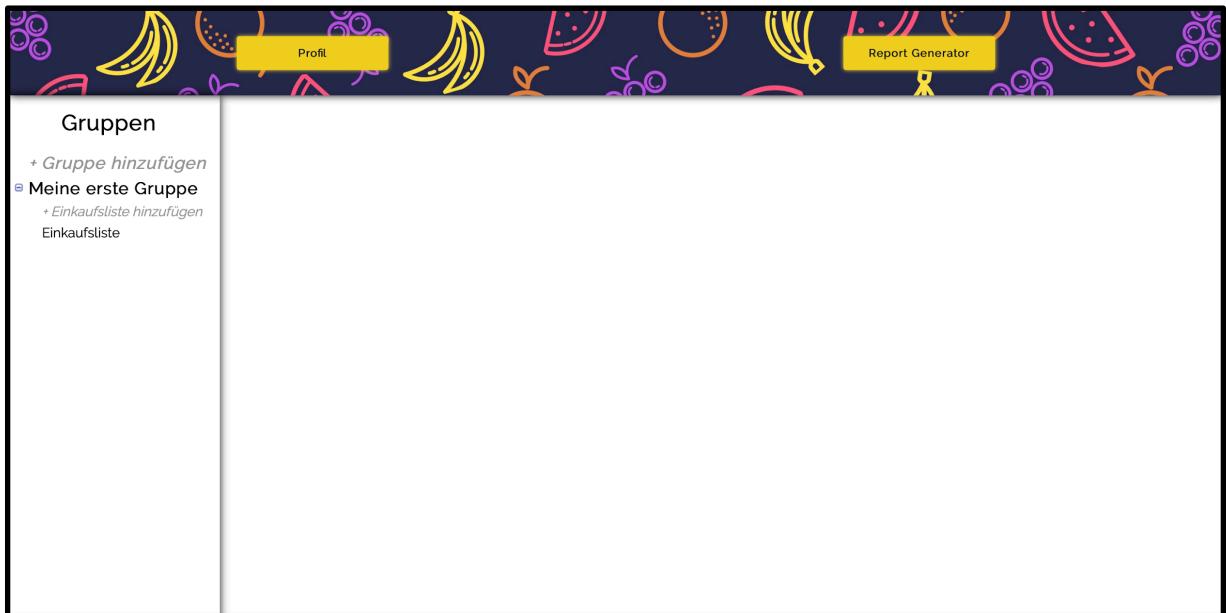


Abbildung 30: Scart - Homepage

Nach der erfolgreichen Registrierung auf dem **Shared-Shopping-List-System Scart** erscheint die Homepage. Zunächst können Sie sich mit den Funktionen von Scart vertraut machen, die in den weiteren Abschnitten beschrieben sind.



Abbildung 31: Scart - Menüleiste

Im oberen Bereich gelangen Sie mit einem Klick zu Ihrem "**Profil**" oder zum "**Report Generator**".



Profil

The screenshot shows a user profile interface. At the top, there's a decorative banner with fruit illustrations (bananas, watermelons, grapes) and two yellow buttons: 'Profil' on the left and 'Report Generator' on the right. Below the banner, on the left, is a sidebar titled 'Gruppen' with options like '+ Gruppe hinzufügen' and 'Meine erste Gruppe'. The main area is titled 'Dein Profil' and displays the user's name ('Name: Patrick') and email ('E-Mail: patrick@hotmail.de'). Below this are two buttons: 'Profil bearbeiten' (highlighted with a yellow box) and 'Profil löschen'.

Abbildung 32: Scart - Nutzerprofil

In der Abbildung 32 wird Ihnen Ihr ganzes Profil angezeigt. Dort können Sie Ihr Profil durch klicken von "Profil löschen" vom System löschen oder durch klicken auf "Profil bearbeiten" Ihr Profil modifizieren. Es lassen sich sowohl Name als auch E-Mail Adresse nachträglich ändern.

This screenshot shows the profile editing process. The 'Profil bearbeiten' button from the previous screen has been clicked, transitioning the interface into edit mode. The user's name ('Name: Patrick') and email ('E-Mail: patrick@hotmail.de') are now displayed in input fields. Below these fields are two buttons: 'Änderung speichern' (highlighted with a yellow box) and 'Abbrechen'. A small confirmation dialog box is overlaid on the bottom right, asking 'Möchten Sie die Änderungen speichern?' with 'Ja' and 'Nein' buttons.

Abbildung 33: Scart - Profiländerung

Wenn Sie auf den Button "Profil bearbeiten" klicken, wird in den Bearbeitungsmodus gewechselt. Von dort können Sie Ihren Namen und/oder Email Adresse nachträglich



bearbeiten. Nach der Bearbeitung Ihres Profils erscheint ein Popup Fenster. Dort müssen Sie Ihre Änderungen mit "Ja" oder "Nein" bestätigen.

Wenn Sie nun mit "Ja" oder "Nein" bestätigen, kommen Sie wieder zu Ihrem modifizierten Profil zurück.

Gruppe



Abbildung 34: Scart – deine Gruppen

Ihre Gruppen werden auf der linken Seite angezeigt. Mit Hilfe von „**+Gruppe hinzufügen**“, können Sie weiter Gruppen anlegen und neue Nutzer nach belieben hinzufügen. Des weiteren sehen Sie die in der Gruppe enthaltenen Einkaufslisten der Gruppe.

The screenshot shows the group details for "Meine erste Gruppe". It includes fields for "Gruppenname" (set to "Meine erste Gruppe"), "Benutzername" (set to "Patrick"), and "Email" (set to "patrick@hotmail.de"). Below these fields are two input fields, one for "Name" and one for "Email", followed by a yellow "+" button. At the bottom are two buttons: "Aus Gruppe austreten" (Logout from group) and "Änderungen speichern" (Save changes).

Abbildung 35: Scart – Gruppenmenü

In der folgenden Abbildung ist eine Übersicht des Gruppenmenüs zu sehen. Dort können Sie neue Mitglieder der Gruppe hinzufügen oder entfernen. Durch klicken von „

Änderungen speichern

“, können Sie ihre Änderungen abspeichern. Mit „



Aus Gruppe austreten

“ können Sie aus ihrer jetzigen Gruppe austreten und somit aus Ihrer Liste entfernen.

The screenshot shows a user profile page with a yellow header bar containing the text "Aus Gruppe austreten". Below this, there is a section for "Gruppenname" (Group Name) set to "Meine erste Gruppe". Underneath, there are fields for "Benutzername" (Username) and "Email", both filled with "Patrick" and "patrick@hotmail.de" respectively. A yellow "+" button is located to the right of these fields. At the bottom of the main section are two buttons: "Aus Gruppe austreten" and "Änderungen speichern" (Save changes). A modal dialog box is overlaid on the page, titled "Gruppe verlassen" (Leave group). It contains the question "Möchten Sie die Gruppe wirklich verlassen?" (Do you really want to leave the group?) and two buttons: "Ja" (Yes) and "Nein" (No).

Abbildung 36: Scart – Gruppe verlassen

Hier müssen Sie nochmals mit „Ja“ oder „Nein“ Ihre Entscheidung bestätigten. Durch bestätigen kommen Sie zurück zur aktualisierten Gruppe.

Warnung: Ihre Liste wird vollständig entfernt, wenn Sie sich entscheiden diese zu löschen.



Einkaufsliste

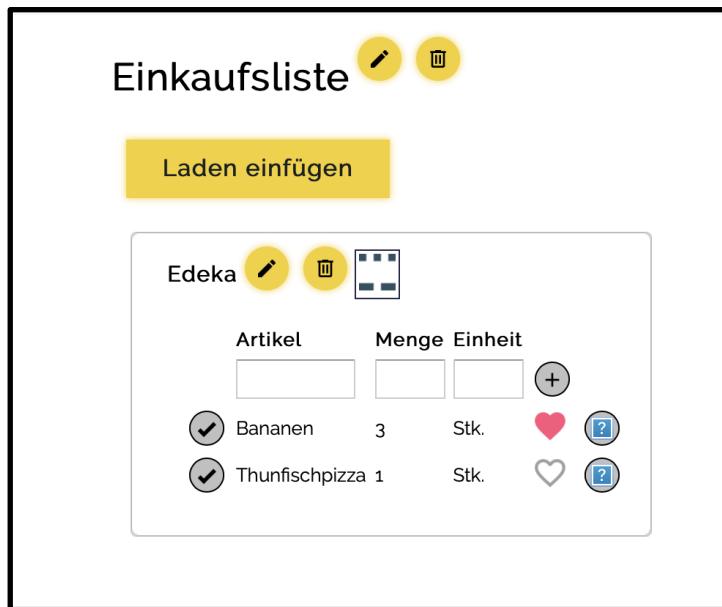


Abbildung 37: Scart – Einkaufsliste von Scart

Abbildung 37 wird die Einkaufsliste von Scart dargestellt. Dort befinden sich alle Artikel eines

Einzelhändlers. Mit den Buttons „“ können Sie Ihre Einkaufsliste bearbeiten oder

löschen. Durch klicken auf das Symbol neben dem , können Sie einzelne Artikel in der

Liste zusätzlich modifizieren. Dadurch wird dieses Fenster eingeblendet ->



Abbildung 38: Scart – Einkaufsliste mit Favoriten und gekauftem Artikel

Sie können auch Zuständigkeiten innerhalb der Gruppe zuweisen. Durch klicken aufs Profilbild



öffnet sich ein Dropdown Menü mit allen Nutzern innerhalb der Gruppe. Somit können Sie bestimmen, wer für welche Liste, in welchem Laden, zuständig ist. Artikel werden nach unten verschoben sobald Sie abgehakt werden. Das abhaken symbolisiert, dass ein Artikel eingekauft wurde.



Abbildung 39: Scart – Löschen der Einkaufsliste

In der überliegenden Einkaufsliste in Abbildung #, müssen Sie mit „Ja“ oder „Nein“ bestätigen, um Ihre Einkaufsliste zu löschen.

Warnung: Durch Bestätigung löschen Sie Ihre Einkaufsliste komplett und können nicht mehr darauf zurückgreifen.

Report

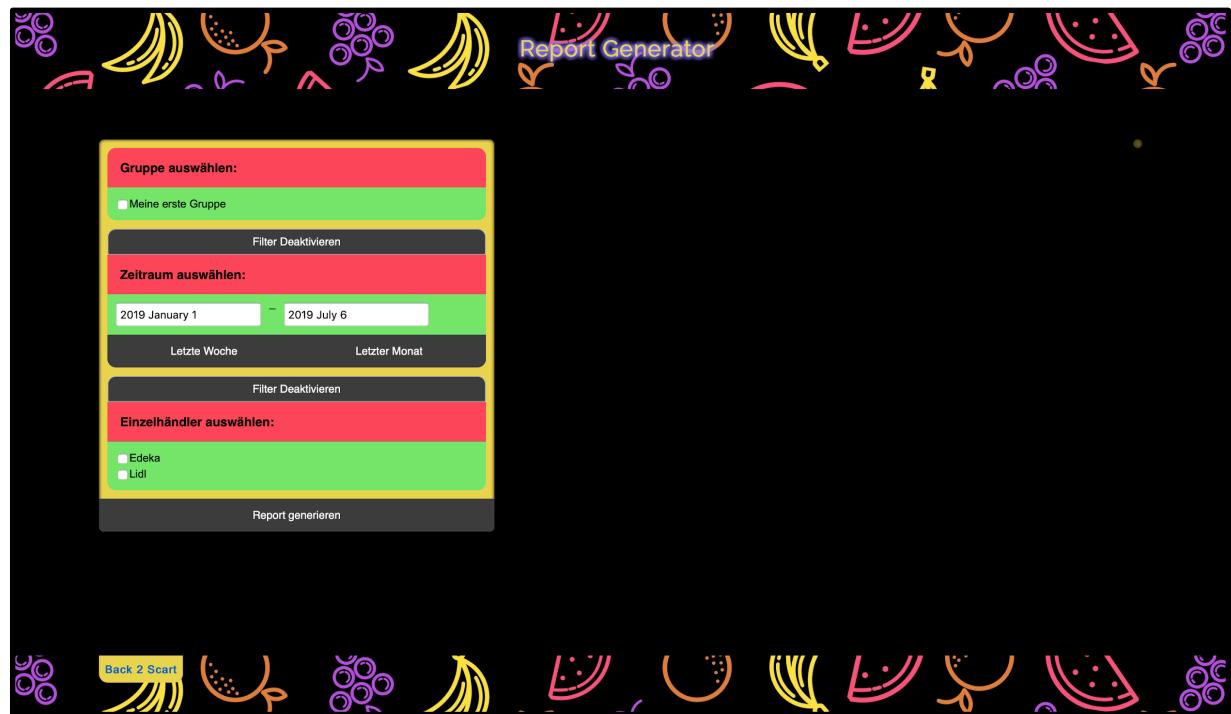


Abbildung 40: Scart - Report Startbildschirm



Für Abfragen, Auswertungen oder eine generelle Übersicht Ihrer Artikel, die Sie häufig in Ihren Einkaufslisten haben, können Sie im Report Generator verschiedene Berichte erstellen.

The screenshot shows the 'ReportFilterForm' interface. It has three main sections: 'Gruppe auswählen:' (Group selected: Meine erste Gruppe), 'Zeitraum auswählen:' (Time period selected: 2019 January 1 to 2019 July 7), and 'Einzelhändler auswählen:' (Retailer selected: Edeka). Each section has a 'Filter Deaktivieren' (Deactivate filter) button above it. A 'Report generieren' (Generate report) button is at the bottom.

Abbildung 41: Scart - ReportFilterForm

In der Abbildung 41 wird visualisiert, welche Optionen Ihnen beim Report erstellen zur Verfügung stehen. Sie können zwischen verschiedenen Filtern wählen.

The screenshot shows the 'Scart - Deaktivierter Filter' interface. The 'Zeitraum auswählen:' section is highlighted in orange, indicating that the filter for the time period is deactivated. The date range '2019 January 1' to '2019 July 7' is visible.

Abbildung 42: Scart - Deaktivierter Filter

The screenshot shows the 'Scart - Aktivierter Filter' interface. The 'Zeitraum auswählen:' section is highlighted in green, indicating that the filter for the time period is activated. The date range '2019 January 1' to '2019 July 7' is visible.

Abbildung 43: Scart - Aktivierter Filter

Die Abbildungen 42 und 43 zeigen deutlich, welcher Filter Aktiviert oder Deaktiviert ist. Wenn ein Filter deaktiviert ist, wird dieses Suchkriterium beim erstellen des Reports nicht weiter beachtet. Dies sehen Sie in Abbildung #, wenn das Suchkriterium durchsichtig hinterlegt ist. Ist ein Filter aber aktiviert, wird dieser beim erstellen berücksichtigt und fließt mit in die Erstellung des Reports. In Abbildung # ist das Suchkriterium farblich erkenntlich und nicht durchsichtig hinterlegt.



The screenshot shows the Scart application interface. On the left, there is a sidebar with various icons representing different categories like fruits and vegetables. The main area has a yellow header bar with the text "Report Generator". Below this, there are three sections for filtering: "Gruppe auswählen" (with "Meine erste Gruppe" selected), "Zeitraum auswählen" (with date range from "2019 January 1" to "2019 July 6"), and "Einzelhändler auswählen" (with "Edeka" and "Lidl" selected). At the bottom of the sidebar is a "Report generieren" button. To the right, a yellow box displays the generated report: "Alle häufig gekauften Artikel von: patrick@hotmail.de" followed by a table with columns: Artikel-Name, Einzelhändler, Erstellungsdatum, Modifikationsdatum, and Archivierungsdatum. The table contains two rows: Steinpilze (Edeka) with date 2019-07-06 23:23:35.0, and Thunfisch (Lidl) with date 2019-07-06 23:24:40.0.

Artikel-Name	Einzelhändler	Erstellungsdatum	Modifikationsdatum	Archivierungsdatum
Steinpilze	Edeka	2019-07-06 23:23:35.0	2019-07-06 23:23:35.0	2019-01-09 00:00:00.0
Thunfisch	Lidl	2019-07-06 23:24:40.0	2019-07-06 23:24:40.0	2019-01-09 00:00:00.0

Abbildung 44: Scart - Report nach Gruppe gefiltert

Durch klicken von "**Report generieren**", in Abbildung 44, wird ein Report nur von der ausgewählten Gruppe erstellt. Auf der rechten Seite wird eine statische Ausgabe erstellt mit dem Artikel-Name, Einzelhändler, Erstellungsdatum, Modifikationsdatum und dem Archivierungsdatum der aktivierte Filter.

This screenshot is similar to Abbildung 44 but shows a different filter selection. In the "Einzelhändler auswählen" section, only "Lidl" is checked. The generated report table shows the same data as in Abbildung 44, but the "Modifikationsdatum" column is now populated with the date "2019-07-06 23:24:40.0" for both entries.

Artikel-Name	Einzelhändler	Erstellungsdatum	Modifikationsdatum	Archivierungsdatum
Steinpilze	Edeka	2019-07-06 23:23:35.0	2019-07-06 23:23:35.0	2019-01-09 00:00:00.0
Thunfisch	Lidl	2019-07-06 23:24:40.0	2019-07-06 23:24:40.0	2019-01-09 00:00:00.0

Abbildung 45: Scart - Report nach Einzelhändler gefiltert



In dieser Abbildung 45, wird nach dem Einzelhändler und der Gruppe gefiltert. Auf der rechten Seite wird eine statische Ausgabe erstellt, die Artikel-Name, Einzelhändler, Erstellungsdatum, Modifikationsdatum und Archivierungsdatum der aktivierte Filter ausgibt.

The screenshot shows the Scart application interface. On the left, there is a sidebar with various filters:

- Gruppe auswählen:**
Meine erste Gruppe (checkbox checked)
- Zeitraum auswählen:**
2019 January 1 - 2019 January 2
Letzte Woche | Letzter Monat
- Filter Aktivieren**
- Einzelhändler auswählen:**
Edeka | Lidl

At the bottom of the sidebar is a "Report generieren" button. To the right, a yellow box displays the generated report:

Alle häufig gekauften Artikel von: patrick@hotmail.de

Erstellungsdatum des Reports | Sun Jul 07 01:10:44 GMT+200 2019
Artikel-Name Einzelhändler Erstellungsdatum Modifikationsdatum Archivierungsdatum

Abbildung 46: Scart - Report nach Zeitraum gefiltert

Hier wird nach Zeitraum via Datepicker für die ausgewählte Gruppe gefiltert. Sie haben auch schnell Zugriff auf die "Letzte Woche" und den "Letzten Monat". Auf der rechten Seite wird eine statische Ausgabe erstellt, die Artikel-Name, Einzelhändler, Erstellungsdatum, Modifikationsdatum und Archivierungsdatum der aktivierte Filter ausgibt.



The screenshot shows the Scart Report Generator interface. On the left, there is a sidebar with three main sections: 'Gruppe auswählen:' (with 'Meine erste Gruppe' checked), 'Zeitraum auswählen:' (with date range from '2019 January 1' to '2019 July 6'), and 'Einzelhändler auswählen:' (with 'Edeka' checked). Below these is a 'Report generieren' button. On the right, a yellow box displays a report titled 'Alle häufig gekauften Artikel von: patrick@hotmail.de'. It includes the creation date ('Erstellungsdatum des Reports | Sat Jul 06 23:48:26 GMT+200 2019') and a table with columns: Artikel-Name, Einzelhändler, Erstellungsdatum, Modifikationsdatum, and Archivierungsdatum. The table contains one row for 'Steinpilze' from 'Edeka' at '2019-07-06 23:23:35.0'.

Abbildung 47: Scart - Report nach allen Möglichkeiten gefiltert

In Abbildung 47, werden alle Filter aktiviert und nach allen Suchkriterien gefiltert. Auf der rechten Seite wird eine statische Ausgabe erstellt, die Artikel-Name, Einzelhändler, Erstellungsdatum, Modifikationsdatum und Archivierungsdatum der aktivierte Filter ausgibt.

Es muss dazu gesagt werden, dass eine Multi Selektion nicht möglich ist. Wenn ein Filter aktiviert ist, muss auch mindestens eine Auswahl getroffen werden. Alle Artikel die nicht der gesuchten Kriterien entsprechen, werden nicht ausgegeben.

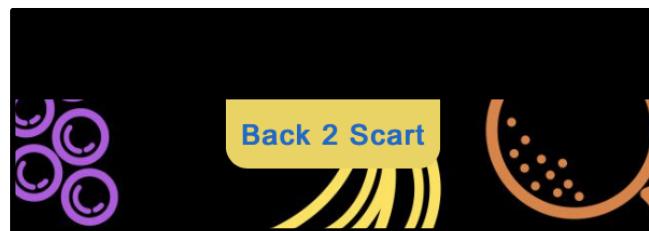


Abbildung 48: Scart - Editor

Wenn Sie Ihren individuellen Report erstellt haben und zu Scart zurückkehren möchten, klicken Sie auf den unten links gesetzten Button "**Back 2 Scart**" um zum Editor zurückzukehren.



Installationshandbuch

Mit **Scart, dem Shared-Shopping-List-System** können Sie sich mit anderen Nutzern ihre Einkaufstour präzise Planen. Somit steht dem nächsten Shopping-Spree nichts mehr im Wege.

Das Zielsystem besteht aus zwei Komponenten die unabhängig voneinander agieren. Dem Editor und dem Report.

Mit folgendem Link können Sie Scart betreten und anfangen Ihren Einkauf zu Planen. Worauf warten Sie denn noch?

Scart: <https://norse-decoder-240009.appspot.com/>

Bevor Sie aber anfangen Ihren Einkauf zu planen, sorgen Sie dafür, dass Sie die richtige Software zur Ausführung auf Ihrem Device zur Verfügung steht. Es reicht schon, wenn Sie einen Browser installiert haben. Wenn nicht, keine Sorge. In den folgenden Schritten wird ausführlich erklärt, wie Sie Scart auf Ihrem Gerät zum Laufen bekommen. Damit Sie Smart zum nächsten Shopping-Spree aufbrechen können.

Auf jedem Gerät ist bereits ein Browser vorinstalliert. (Apple - Safari, Windows - Internet Explorer). Wenn Sie aber nicht mit dem jetzigen Browser zufrieden sind:

- Öffnen Sie ihren Default-Browser
- Laden Sie Ihren präferenzierten Browser herunter
- Führen Sie den Installationsprozess durch mithilfe der Installationssoftware
- Öffnen Sie den neu installierten Browser durch Doppelklick auf das Icon
- Fügen Sie den Link: <https://norse-decoder-240009.appspot.com/> mit Hilfe von einem Rechtsklick in das Suchfeld des Browsers ein und bestätigen Sie die Eingabe mit der Enter-Taste.
- Scart wird nun geladen und ist bereit für die Nutzung

Anhang

Quellcode inkl. JavaDoc

Literaturverzeichnis

Google, (07.07.2019): Google Cloud SQL <https://cloud.google.com/sql/docs/> (Stand: 07.07.19)

Oestereich, B. (2006). [Analyse und Design mit der UML 2.1 - Objektorientierte Softwareentwicklung](#). 8. Auflage. München: Oldenbourg. (Stand: 07.07.2019)

Prof. Dr. Peter Thies, P.D.C., 2019, Aufgabenstellung, Stuttgart: Hochschule der Medien