

Pflichtenheft und technische Spezifikation im Programmierprojekt

Projekt: Restverwerter

Mitarbeitende: Ariol Jospin Tsadjeu (567222)
Domenico Reynaldo (574417)
Leon Wagner (575232)

Inhaltsverzeichnis

1	Visionen und Ziele	1
2	Anforderungen an Ihr System	2
2.1	Use-Cases	2
2.2	Risiken	3
2.3	GUI	4
3	Realisierung	9
3.1	Allgemeines	10
3.2	Interne Schnittstellen	13
3.3	Visual-Studio-Projektsetup	14
3.4	Externe Schnittstellen	14
4	Test und Implementierungsphase	15
5	Planung	15
6	Lizenz	16

1. Visionen und Ziele

ID	Priorität	Beschreibung des Features	Aufwand
Zutaten Eingabe	Sehr wichtiges Feature	Eine Person gibt im Programm an, welche Lebensmittel sie gerade vorrätig hat. Das Programm sammelt intern die eingegebenen Lebensmittel bzw. Zutaten und speichert sie in Datenbank 1, sodass zu den Zutaten passende Rezepte gesucht werden aus Datenbank 2. Die passenden Rezepte werden nach dem Eintragen der Zutaten angezeigt und vorgeschlagen. Die Person kann dann entscheiden für welches Rezept zum Kochen Er/Sie sich nun entscheidet.	Ca. 6 PW
Präferenzen	Optionales Feature	Die Person kann gleich beim Öffnen des Programms angeben, welche allgemeinen Präferenzen vorliegen, wie zum Beispiel, ob die Person sich vegetarisch ernährt oder eine Allergie hat.	Ca. 1 PW
Eingabe der Basis	Optionales Feature	Die Person kann nach der Eingabe der Zutaten, bevor sie die Rezepte angezeigt bekommt, eingeben welche der Zutaten sie auf jeden Fall verwenden will. Praktisch eine Präferenz für das diesmalige Kochen.	Ca. 1 PW
Login	Optionales Feature	Damit die Person Ihre Präferenzen, wie Allergien oder die vegane Diät nicht jedes Mal aufs Neue eingeben muss, kann Sie sich mit dem Login Feature einloggen, sodass die Angaben und Präferenzen gespeichert und geladen werden.	Ca. 3 PW

PW: Personen Woche (5 Wochentage je 8 Arbeitsstunden)

2. Anforderungen an Ihr System

2.1 Use-Case

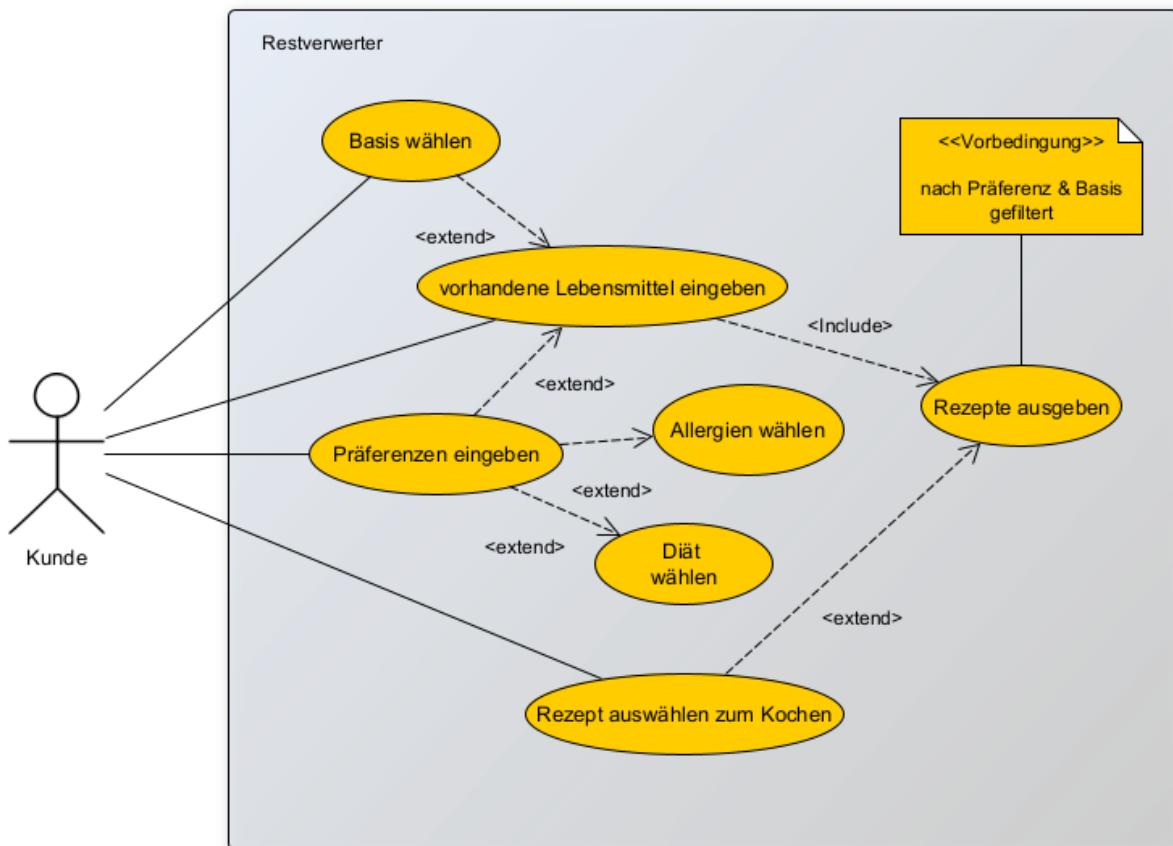


Abb. 1: Use-Case-Diagramm

2.2 Risiken

Risiken in der Implementierung des Restverwerter-Programms, und was man tun kann, um ihnen entgegenzuwirken.

Möglicherweise gibt es keine externen herunterladbaren Datenbanken von Rezepten und Zutaten, die frei zugänglich sind.

- *Frühzeitig herausfinden, ob und wo man die externen Datenbanken herunterladen kann und ob sie frei zugänglich sind.*

Geringe Programmiererfahrung.

- *Nach kleinen „Programmier-Blöcken“ testen, ob sie funktionieren. Dadurch Zeit für die Fehlererkennung reduzieren. (Scrum-Strategie)*

Zusammenarbeit als ein Team mit gerechter Arbeitsteilung.

- *Gute Kommunikation, keine „Solo-Anläufe“ ohne die anderen Programmierer zu informieren und Git und Branches verwenden.*

Geringe Erfahrung mit Git

- *Den Umgang mit Git erlernen und üben. Stets Push und Pulls durchführen und in Branches arbeiten zur Protokollierung und Datensicherung.*

Fehlendes Wissen über die Einbindung externer Datenbanken.

- *Frühzeitig informieren und analysieren, welche Möglichkeiten es gibt, externe Datenbanken im Programm einzufügen.*

2.3 GUI

In diesem Abschnitt wird die Benutzeroberfläche als MockUp im Detail beschrieben.

Achtung: Manche im MockUp enthaltenen Features sind optional und manche dienen nur zur Veranschaulichung. Features wie die "Rezepte dieser Saison" und andere mit "Achtung" gekennzeichneten Features sind nicht geplant zu implementieren.

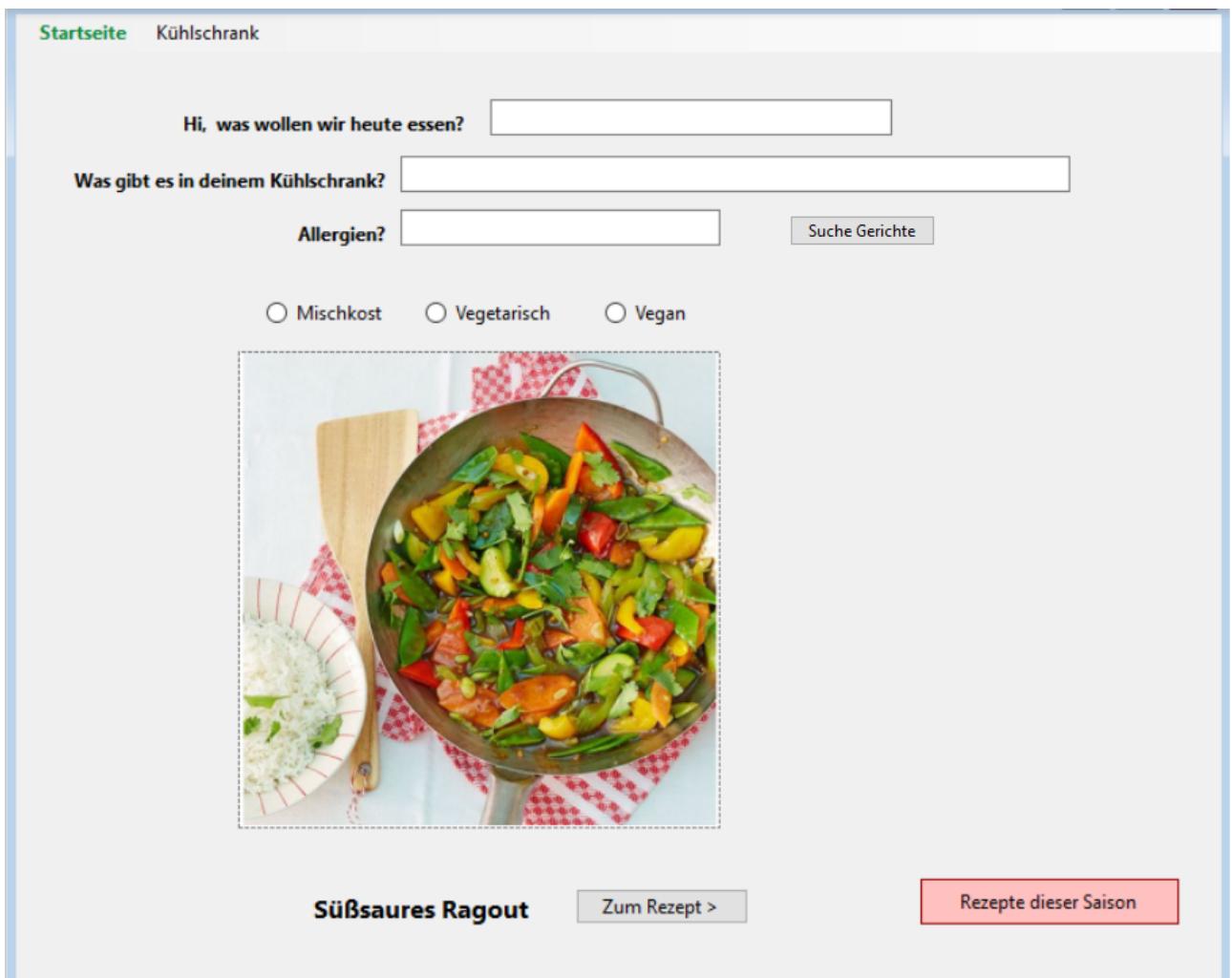


Abb. 2: Startseite

Hier ist unsere Startseite. Der Benutzer oder die Benutzerin kann Lebensmittel angeben, die zu Hause oder wo auch immer vorrätig sind. Durch die Angabe der Zutaten, sucht das Programm nach passenden Rezepten, die möglichst gut zu den vorrätigen Zutaten passen. Dabei wird es optional die Features geben, Präferenzen anzugeben (z.B. Allergien, Diät), Basiszutaten anzugeben und eventuell ein Login zur Speicherung der Präferenzen.

MAGISCHRAN

Startseite Kühlenschrank

?

Hi 'Users Name'!! was wollen wir heute essen ??

Was gibt es in deinem Kühlschrank ?? Tomaten, Zwiebel, Nudel, Salat

Italienischer Nudelsalat... 30 Min

Mediterraner Nudel... 25 Min

Süß-saure Garnele.... 25 Min

Paprika-Nud... 25 Min

Nudelsalat auf... 25 Min

Der beste Italien... 20 Min

Sommerlicher Nudel... 20 Min

Italienischer Nudel... 20 Min

Rezepte dieser Saison

Abb. 3: Was gibt es in deinem Kühlschrank?

Hier kann der Benutzer oder die Benutzerin die Lebensmittel, die im Kühlschrank oder andernorts vorhandenen sind, eingeben. Danach wird eine Liste von Rezepten angeboten, die genau diese Zutaten beinhalten & ggfs. Zutaten, die zusätzlich gekauft werden müssen. Der/die Benutzer*in kann nun das Rezept auswählen, welches ihm/ihr gefällt.

MAGISCHANK

?

Startseite Kühlschrank

Hi "Users Name"!! was wollen wir heute essen ??

Was gibt es in deinem Kühlschrank ??

 Gefüllte Eier 30 Min

 Traditionell span.... 15 Min
Traditionell spanisch Tortilla

 Shakshuka 15 Min

 Rührei á la Slonski 1Min

 Rührei Mit Sahne 10 Min

 Eierstich selbst.. 10 Min

 Omelett mit To... 20 Min

Abb. 4: Was wollen wir heute essen?

Nachdem der Benutzer oder die Benutzerin eine/mehrere Basiszutaten angegeben hat, die Sie/Er auf jeden Fall im Gericht enthalten haben möchte, wird ihm/ihr eine Liste von Gerichten basierend auf der eingegebenen Zutat angeboten.

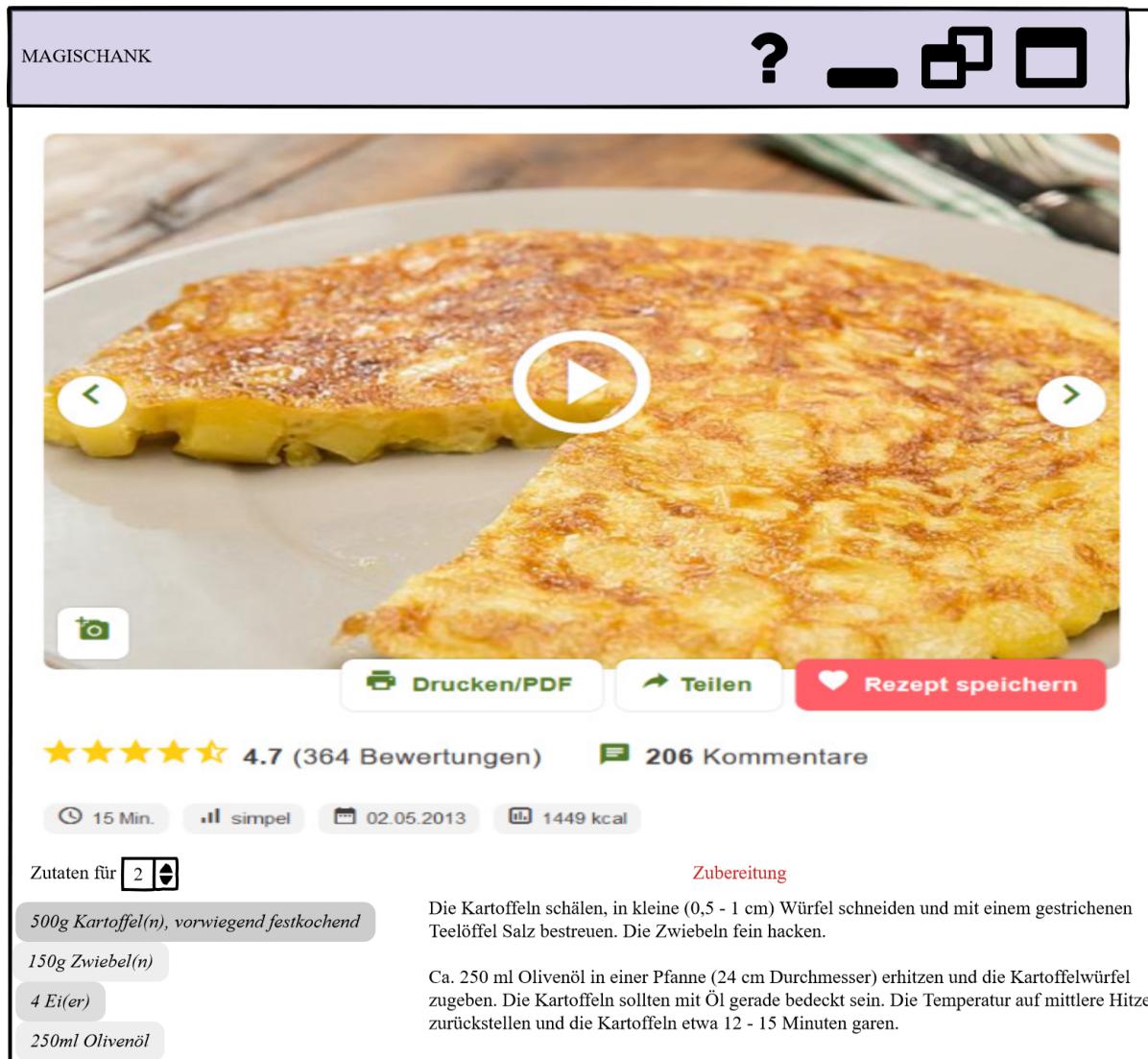


Abb. 5: Rezept

Nach der Auswahl des gewünschten Gerichts hat der Benutzer Zugriff auf Bilder und alle Informationen, die die Zubereitung des Rezepts betreffen.

Achtung:

Ein Video sowie die Möglichkeit zur Bewertung, zum Teilen, zum Drucken oder zum Speichern der Rezepte ist nicht Teil unserer Implementierung. Das MockUp dient lediglich zur Veranschaulichung.

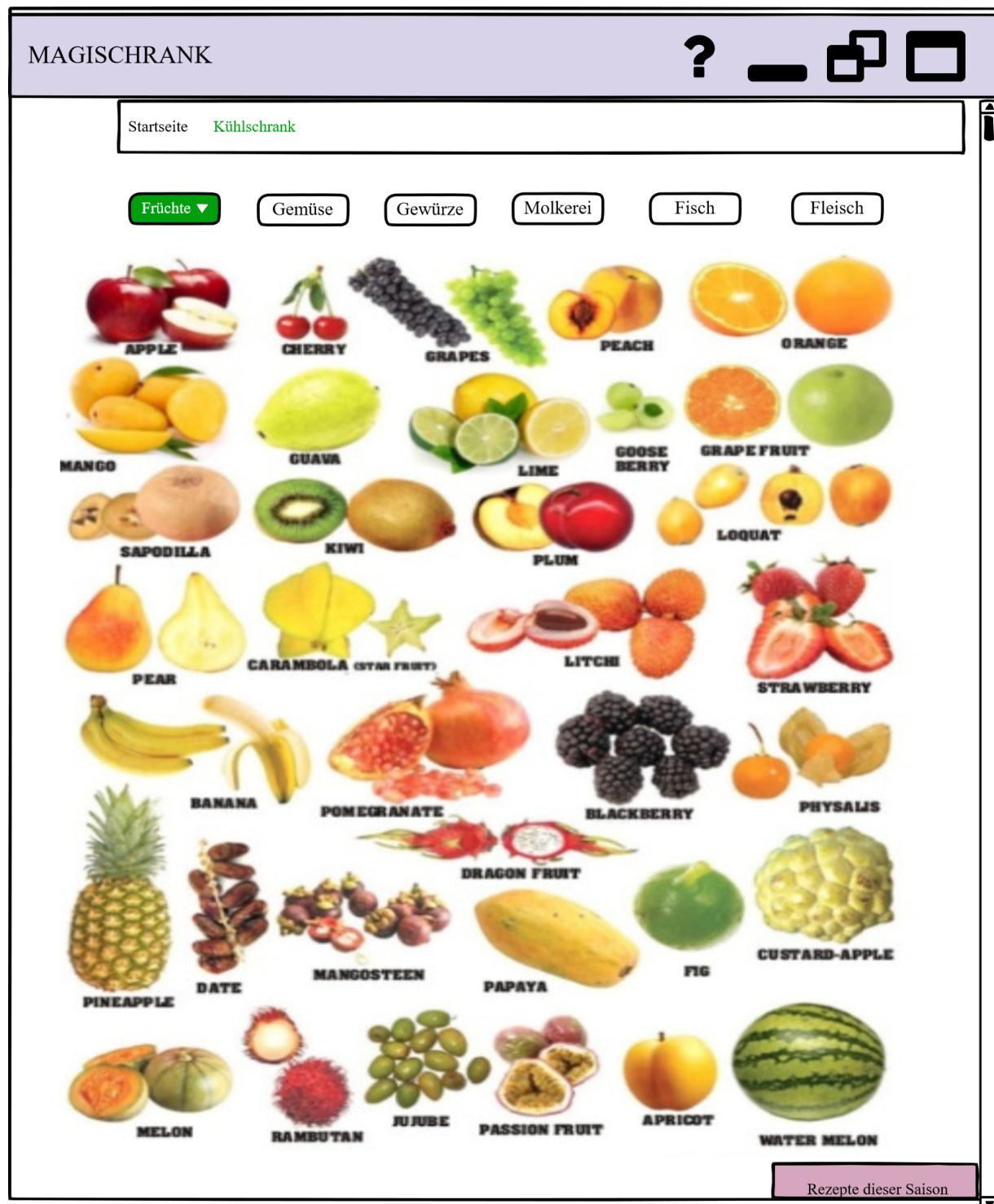


Abb. 6: Kühlschrank

Optional können die vorrätigen Lebensmittel auch über Bilder und Kategorien ausgewählt werden.

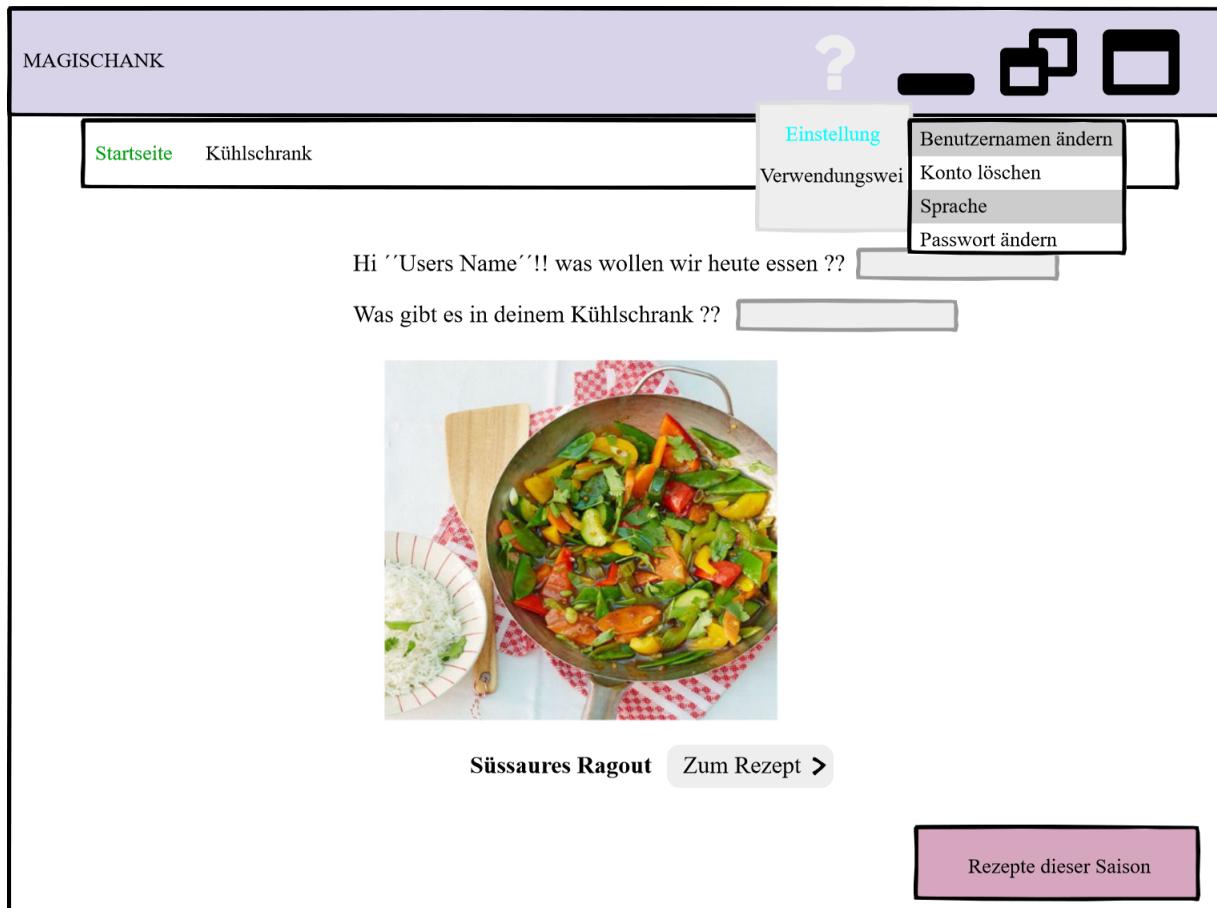


Abb. 7: Infos

Hier hat der/die Benutzer*in Zugriff (als optionales Feature) auf verschiedene Einstellungen wie Sprache, Kontoeinstellungen und auch Zugriff auf das Benutzerhandbuch im Falle, dass der Login im Projekt implementiert wird.

3. Realisierung

3.1 Allgemeines

Für unser Projekt verwenden wir in Visual Studio die Programmiersprache C# und nutzen Windows Forms zur Darstellung unseres GUI. Dazu verwenden wir frei zugängliche externe Schnittstellen, um den Arbeitsaufwand zu reduzieren. Mehr Details zu den externen Schnittstellen im Abschnitt 3.4.

Veranschaulichung der Schnittstellen zwischen den Komponenten:

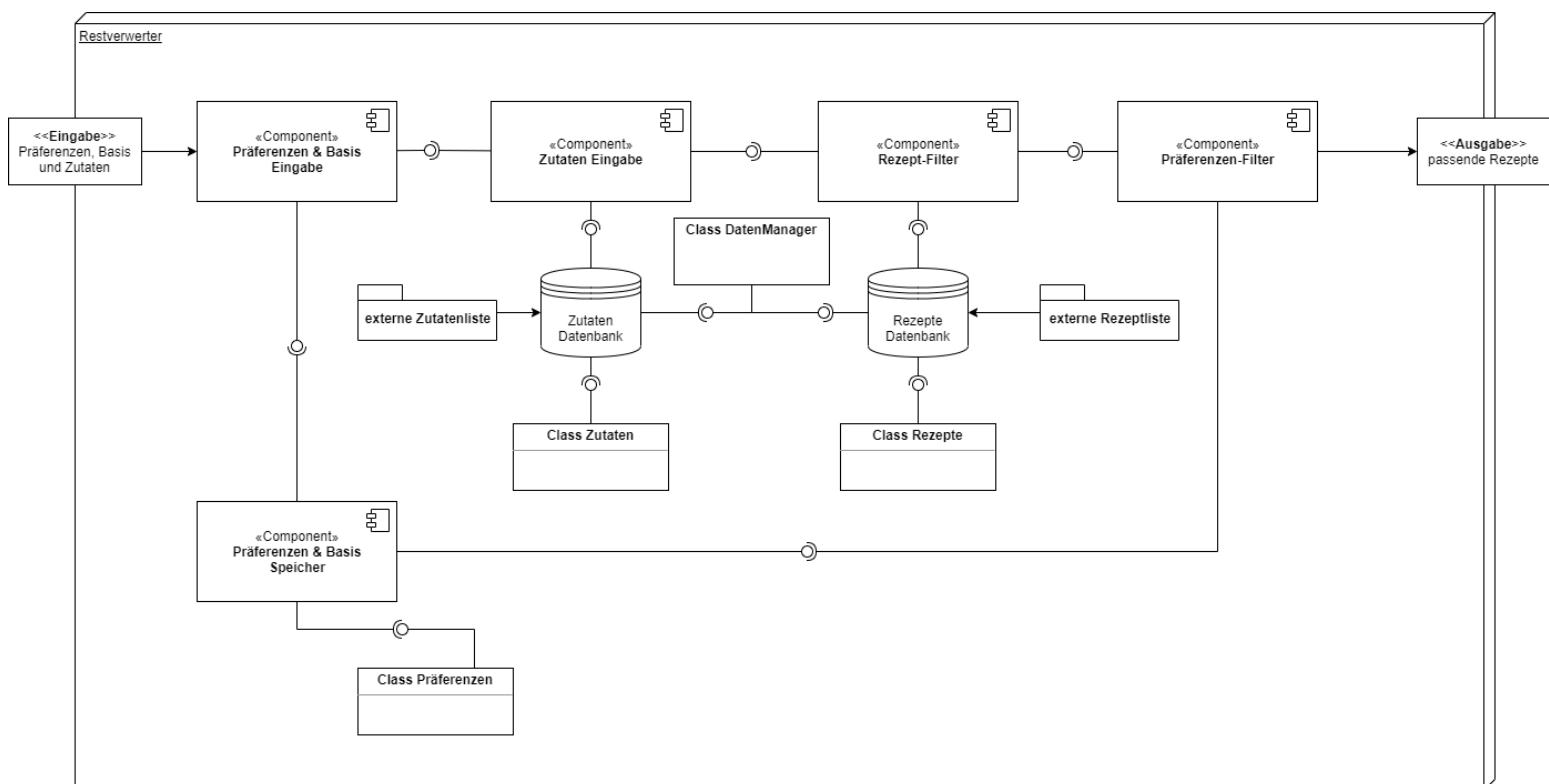


Abb. 8: Komponentendiagramm

3.1 Interne Schnittstellen

Veranschaulichung der Klassen und Interfaces:

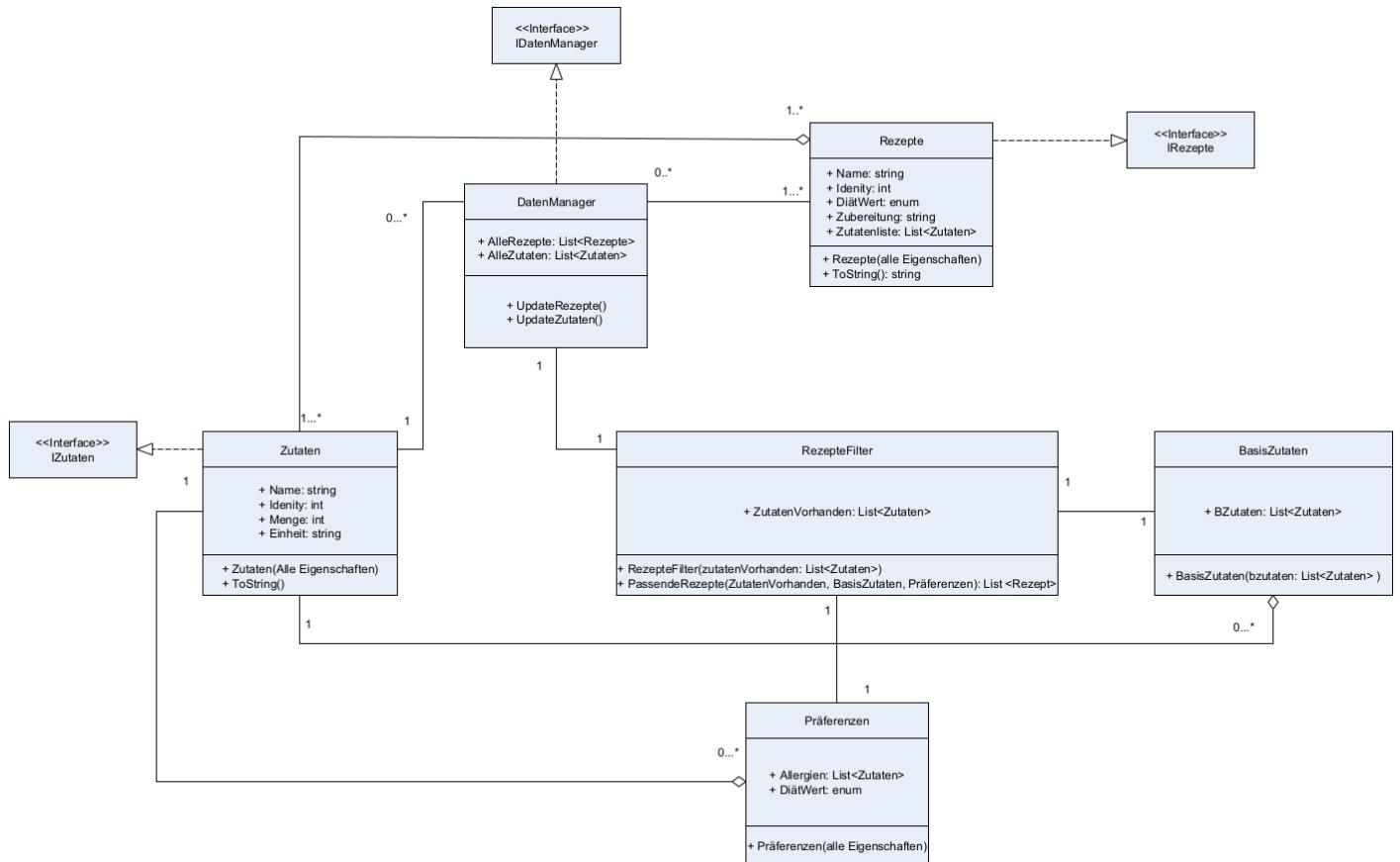


Abb. 9: Klassendiagramm

Zuständigkeiten der Programmierer für die jeweiligen Komponenten, Schnittstellen & Klassen:

<u>Schnittstelle/ Klasse</u>	<u>Programmierer</u>
IDatenManager + DatenManager	Leon
IRezepte + Rezepte	Ariol
IZutaten + Zutaten	Reynaldo
Präferenzen	Reynaldo
BasisZutaten	Ariol
RezepteFilter	Ariol, Reynaldo, Leon
GUI	Ariol
Herunterladen & Einbinden externer Schnittstellen (Zutaten & Rezepte)	Leon
Matching von den Rezepten, Zutaten & dem Rezeptefilter	Reynaldo

3.2 Visual-Studio-Projektsetup

Im Gitlab im Ordner Pflichtenheft unserer Gruppe liegt eine Klassenbibliothek namens "CommonInterfaces" und ein Ordner "GUI". Im Ordner "Programm" im GitLab befindet sich unser Visual Studio Projekt von dem aus wir in verschiedenen Branches (mehr Details im Bereich 4 "Test und Implementierungsphase") beginnen werden das Programm zu implementieren. Dort befinden sich alle im Klassendiagramm enthaltenen Klassen & Interfaces und GUI Visualisierungen wie bereits im Meeting besprochen. Desweiteren sei darauf hingewiesen, dass im GUI Ordner vorerst Bilder sind, die als Platzhalter dienen. Im CommonInterface sind die Eigenschaften & Methoden bereits deklariert. Implementierungen finden in Phase 2 & 3 statt.

```
1 Verweis
interface IDatenManager
{
    1 Verweis
    public List<Rezepte> AlleRezepte { get; set; }
    1 Verweis
    public List<Zutaten> AlleZutaten { get; set; }
    1 Verweis
    public List<Rezepte> UpdateRezepte(List<Rezepte> alleRezepte)
    {
        return null;
    }
    1 Verweis
    public List<Rezepte> UpdateZutaten(List<Zutaten> alleZutaten)
    {
        return null;
    }
}
```

Abb. 10: Beispiel vom Interface IDatenManager

3.3 Externe Schnittstellen

Das sind die externen Schnittstellen, die wir favorisiert haben für unser Projekt. Es befinden sich noch einige andere im Internet. Wir werden wahrscheinlich nicht alle Schnittstellen verwenden, sondern die implementieren, die sich für unser Projekt am geeignetsten ergeben. Die hier aufgelisteten Schnittstellen und Datensätze können Sie im Gitlab unter dem Ordner "Externe Schnittstellen" auffinden und alle sind frei zugänglich.

Typ	genutzt für:	Author	Begründung
Datensatz (Excel)	Rezepte & Zutaten	Food.com	Sehr große Datenbank an Rezepten (über 180.000) mit jeweils einer Menge an enthaltenen Zutaten. Übersichtliche Auflistung.
Datensatz (json)	Rezepte & Zutaten	Yummly	kleinere Datenbank, dafür verständlicher & wahrscheinlich leichter ins Programm zu integrieren

4. Test und Implementierungsphase

Unsere Test & Implementierungsstrategie:

Phase 2:

*Alle Teammitglieder verfahren nach dem Vorgehensmodell **Scrum**. D.h. implementierte Komponenten werden, bevor weiter verfahren wird, sofort getestet. Das hilft dabei, Fehler gleich zu erkennen und reduziert die benötigte Zeit zur Fehlersuche. Alle Teammitglieder sprechen sich in Phase 2 ab, um Widersprüche, offene Fragen, etc. zu klären. Dennoch fertigt jedes Teammitglied selbstständig seinen Prototypen an. Jedes Teammitglied verwendet Branches zur Versionskontrolle und Datensicherung.*

Phase 3:

*In Phase 3 verfahren alle Teammitglieder ebenfalls nach **Scrum**. D.h. alle Teammitglieder versuchen eine Komponente/ Klasse/ Schnittstelle in ihrem eigenen Branch, von dem „Developer“ Branch abgeleitet, zu implementieren und zu testen.*

Das schließt mit ein, dass wir mehrmals in der Woche uns miteinander über Discord austauschen, um Widersprüche, Probleme zu lösen und einander zu helfen.

Hat ein Teammitglied eine Klasse/ Komponente fertiggestellt (Code fehlerfrei), so wird dieser funktionierende Teil des Codes mit den Komponenten und Klassen der anderen Teammitglieder gemerged. Das findet in dem Zwischen Branch „Developer“ statt, dort wird geprüft, ob alle Klassen & Komponenten miteinander „harmonieren“ und sich nicht einander stören.

Nach ausreichendem Testen auch mit eigenen Test-Klassen wird der fertige Code aus dem Developer Branch mit dem Master gemerged und dessen Funktionalität wird noch einmal geprüft.

5. Lizenz

Restverwerter Lizenz

Copyright (c) <2021> <Ariol Jospin Tsadjeu (567222), Reynaldo Domenico (574417), Leon Wagner (575232)>

Jedem, der eine Kopie dieser Software und der zugehörigen Dokumentationsdateien (die "Software") erhält, wird hiermit kostenlos die Erlaubnis erteilt, ohne Einschränkung mit der Software zu handeln, einschließlich und ohne Einschränkung der Rechte zur Nutzung, zum Kopieren, Ändern, Zusammenführen, Veröffentlichen, Verteilen, Unterlizenzieren und/oder Verkaufen von Kopien der Software, und Personen, denen die Software zur Verfügung gestellt wird, dies unter den folgenden Bedingungen zu gestatten:

Der obige Urheberrechtshinweis und dieser Genehmigungsnachweis müssen in allen Kopien oder wesentlichen Teilen der Software enthalten sein.

DIE SOFTWARE WIRD OHNE MÄNGELGEWÄHR UND OHNE JEGLICHE AUSDRÜCKLICHE ODER STILLSCHWEIGENDE GEWÄHRLEISTUNG, EINSCHLIESSLICH, ABER NICHT BESCHRÄNKKT AUF DIE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT, DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK UND DER NICHTVERLETZUNG VON RECHTEN DRITTER, ZUR VERFÜGUNG GESTELLT. DIE AUTOREN ODER URHEBERRECHTSINHABER SIND IN KEINEM FALL HAFTBAR FÜR ANSPRÜCHE, SCHÄDEN ODER ANDERE VERPFLICHTUNGEN, OB IN EINER VERTRAGS- ODER HAFTUNGSKLAGE, EINER UNERLAUBTEN HANDLUNG ODER ANDERWEITIG, DIE SICH AUS, AUS ODER IN VERBINDUNG MIT DER SOFTWARE ODER DER NUTZUNG ODER ANDEREN GESCHÄFTEN MIT DER SOFTWARE ERGEBEN.