WhatsAppToEat

<Iteration/ Master> Test Plan

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 13/06/2013 | 1.0 | Whats App To Eat – Rezepte Verwaltung | Sebastian Bantz,  Thomas Franken, Thomas Lang |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Intended Audience 6

2. Target Test Items 6

3. Outline of Planned Tests 6

3.1 Outline of Test Inclusions 6

3.2 Outline of Other Candidates for Potential Inclusion 6

3.3 Outline of Test Exclusions 6

4. Test Approach (1) 7

4.1 Modify Grades 7

4.2 Test Procedure: 7

4.3 Activity-Diagram: 8

4.4 Cocumber-Code: 8

5. Test Approach (2) 9

5.1 Modify Grades 9

5.2 Test Procedure: 10

5.3 Activity-Diagram: 10

5.4 Cocumber-Code: 11

6. Test Approach (3) 11

6.1 Modify Grades 11

6.2 Test Procedure: 11

6.3 Activity-Diagram: 12

6.4 Cocumber-Code: 13

Test Plan

# Introduction

## Purpose

Dieses Dokument soll eine Übersicht über die in dem Projekt “Whats App To Eat” durchgeführten Test. Für die durchzuführenden Tests wurden fünf UseCases ausgewählt. Je nach Erfordernis wird bei jedem UseCase ein Functiontest bzw. ein GUI-Test durchgeführt (siehe Kapitel 2).

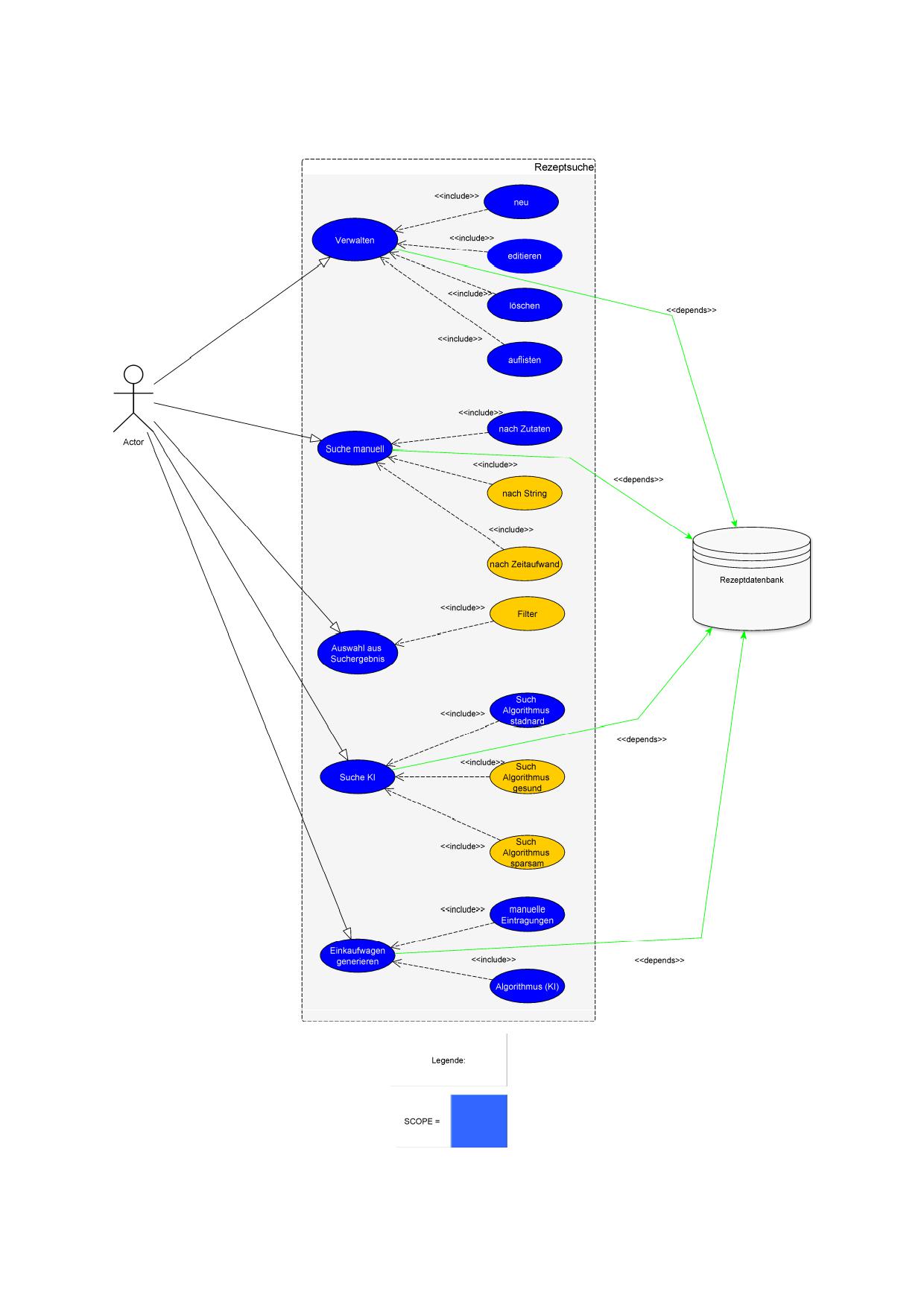
getestete UseCases:

* zu Verwalten navigieren
* zu Suche navigieren
* Rezept hinzufügen
* Rezeptliste anzeigen
* Rezept löschen

• Die obengenannten UseCases wurde in unserem Projekt komplett implementiert.

## Scope

Folgende Punkte wurden als Scope unseres Projektes definiert:



## Intended Audience

Dieses Dokument soll aufzeigen nach welchen Methoden die GUI- und Functionaltest durchgeführt wurden. Es beschreibt welche UseCases getestet wurden und zeigt welche Ergebnisse die einzelnen Test geliefert haben.

# Target Test Items

Der folgende Punkt gibt eine Übersicht über die zu testenden Elemente der einzelnen UseCases.

|  |  |  |
| --- | --- | --- |
| Scope´s | GUI Testing | Functional-Testing |
| Zu Verwalten navigieren | X |  |
| Zu Suche navigieren | X |  |
| Rezepte hinzufügen | X | X |

# Outline of Planned Tests

Folgende Tests wurden geplant.

* Zu Verwalten navigieren
* Zu Suche navigieren
* Rezept hinzufügen

## Outline of Test Inclusions

(siehe Punkt 3)

## Outline of Other Candidates for Potential Inclusion

Es wäre empfehlenswert die folgenden Tests zu einem späteren Zietpunkt noch durchzuführen:

- Änderung der Performance testen, wenn die Menge der Datenbankelemente steigt.

- Messung des Stromverbrauchs der App bei steigenden Anzahl Datenbankeinträgen

## Outline of Test Exclusions

* Zu Verwalten navigieren
* Zu Suche navigieren
* Zu Einkaufsliste navigieren
* Rezept hinzufügen
* Rezeptliste anzeigen
* Rezept anzeigen
* Rezept löschen
* Rezept bearbeiten
* Such mit Standardalgorithmus
* Suche Rezepte nach Zutaten
* Rezepte zu Einkaufwagen hinzufügen
* Einkaufswagen löschen
* Einzelne Rezepte aus Einkaufswagen löschen
* Menge der Zutaten im Einkaufswagen ändern

# Test Approach (1)

Testduchführung für den UseCase „ Zu Verwalten Navigieren“

## Modify Grades

1. **Location**Navigate zu Verwalten
2. **Feature Pass/Fail Criteria**

Pass:  
Navigation vom Startscreen zu Verwalten erfolgt  
Fail:  
Es kann vom Startscreen nicht zu Verwalten navigiert werden.

1. **Means of Control**

Der User kann auf den Screen „Verwalten“ wechseln indem er den Button „Verwalten“ anklickt. Anschließend wir der Screen „Verwalten“ aufgerufen und angezeigt.

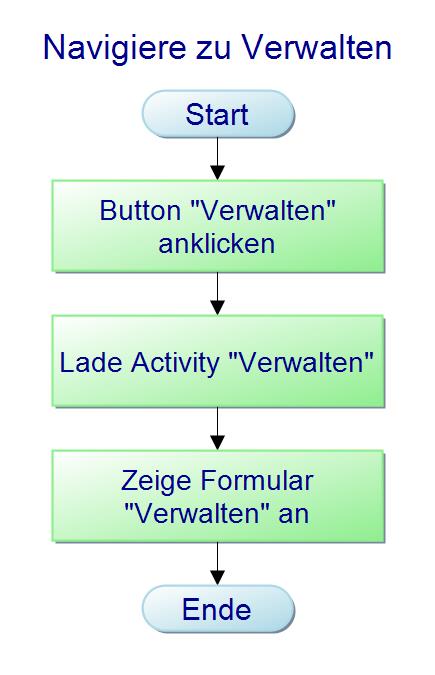
## Test Procedure:

1. User befindet sich auf dem HomeScreen.
2. User klickt auf Verwalten.
3. Neuer Screen wird geöffnet.
4. User befindet sich auf dem Screen Verwalten.

Folgende Ergebnisse können erzielt werden:

* Screen wechselt zu Verwalten
* Screen wechselt nicht zu Verwalten

## Activity-Diagram:



## Cocumber-Code:

In order to manage my recipes

As a user

I want to navigate to Verwalten

Given I am in Rezepte

When I click on Verwalten

Then I should see the screen Verwalten

And there should be a Liste anzeigen button

And there should be a Neuer Eintrag button

# Test Approach (2)

Testduchführung für den UseCase „ Zu Suche Navigieren“

## Modify Grades

1. **Location**Navigate zu Suche
2. **Feature Pass/Fail Criteria**

Pass:  
Navigation vom Startscreen zu Suche erfolgt  
Fail:  
Es kann vom Startscreen nicht zu Suche navigiert werden.

1. **Means of Control**

Der User kann auf den Screen „Suche“ wechseln indem er den Button „Suche“ anklickt. Anschließend wir der Screen „Suche“ aufgerufen und angezeigt.

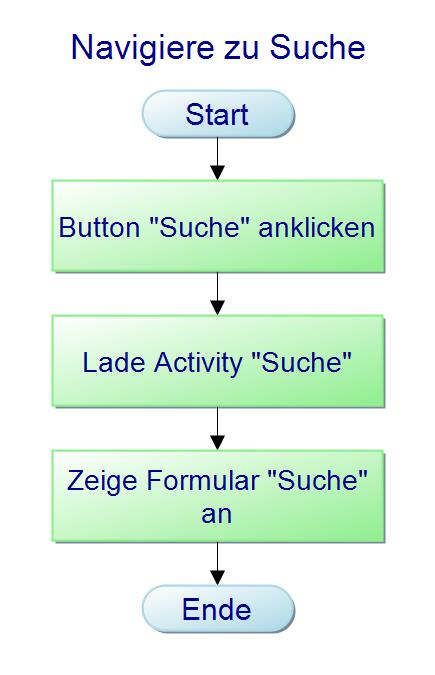
## Test Procedure:

1. User befindet sich auf dem HomeScreen.
2. User klickt auf Suche.
3. Neuer Screen wird geöffnet.
4. User befindet sich auf dem Screen Suche.

Folgende Ergebnisse können erzielt werden:

* Screen wechselt zu Suche
* Screen wechselt nicht zu Suche

## Activity-Diagram:



## Cocumber-Code:

In order to search for a recipe

As a User

I want to navigate to Suche

Given I am in Rezepte

When I click on Suche

Then I should see the screen Suche

And there should be a Algemeine Suche button

And there should be a Zutatensuche button

# Test Approach (3)

Testduchführung für den UseCase „ Zu Suche Navigieren“

## Modify Grades

1. **Location**Verwalten
2. **Feature to tested:**  
   Rezept hinzufügen
3. **Feature Pass/Fail Criteria**

Pass:  
Rezept kann hinzugefügt werden.  
Fail:  
Hinzufügen nicht möglich

1. **Means of Control**

User kann ein Rezept hinzufügen, indem er im Verwalten-Screen auf Rezept hinzufügen klickt und anschließend das Formular ausfüllt.

Sobald der Benutzer auf hinzufügen klickt und das Rezept erfolgreich in die Datenbank eingetragen wurde, gelangt der User zum Erfolgsmeldung-Screen. Von Dort gelangt er mit einem Klick auf „Liste anzeigen“ zur Rezeptübersicht.

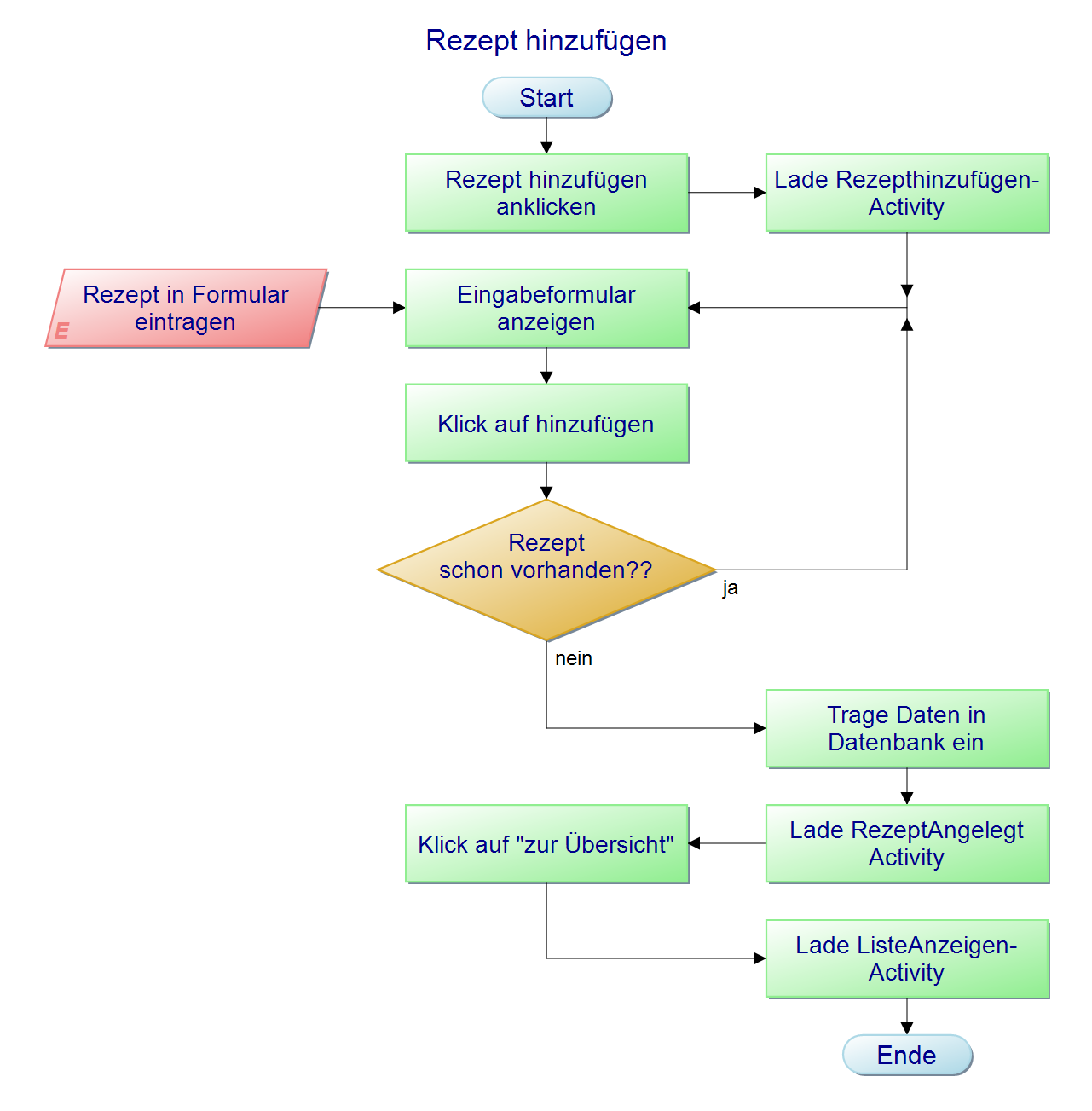
## Test Procedure:

1. User befindet sich auf dem Verwalten Screen
2. Der User klickt auf Rezept hinzufügen
3. Der Rezept hinzufügen-Screen wird geöffnet.
4. Der User füllt das Formular aus.
5. Der User klickt auf hinzufügen.
6. Der Benutzer bekommt eine Erfolgsmeldung.
7. Der Benutzer klickt auf „Zur Übersicht“
8. Der Benutzer kommt zum „Liste Anzeigen“ Screen.

Folgende Szenarios können entstehen:

* Die Daten sind nicht vorhanden: Das Hinzufügen hat nicht funktioniert.
* Rezept wurde erfolgreich hinzugefügt.

## Activity-Diagram:



## Cocumber-Code:

In order to search for a recipe

As a User

I want to navigate to Suche

Given I am in Rezepte

When I click on Suche

Then I should see the screen Suche

And there should be a Algemeine Suche button

And there should be a Zutatensuche button

Es war uns leider nicht möglich den von uns geschriebenen Testcode auszuführen, da wir auf nicht sprechende Fehlermeldungen gestoßen sind. Auch ausgedehnte Recherchen im Internet blieben mangels Information erfolglos.

Fehler:

Test run failed: Instrumentation run failed due to 'java.lang.ClassNotFoundException'  
Leider wird vom Android JUnit Test nicht genauer darauf eingeganben was die Ursache des Problems ist. Somit haben wir keinen Anhaltspunkt um diesen Fehler zu beheben, da für diese Art von Fehler sehr viele mögliche Ursachen verantwortlich sein können.

Wir haben zwar Hinweise auf mögliche Lösungen gefunden wie z.B.:

„Was the Buildpath probably configured.“   
- Es wurde von der Person die den Tipp gegeben hat nicht näher spezifiziert wie die korrekten Einstellungen für den BuildPath wären. Insofern kann es sich tatsächlich um die Fehlerquelle handeln, allerdings können wird das nicht feststellen.

“Target Package declared correctly”   
- Tatsächlich war unser „Target Package“ falsch deklariert, doch löste auch eine korrekte Deklaration das Problem nicht.

Darauf haben wir versucht auf die folgenden Testingtools umzusteigen:

- Androidtesting,  
- Cucumbar Android  
- UI testing  
- Monkey Runner

Allerdings haben wir bei den getesteten Alternativen ebenfalls feststellen müssen das teilweise nur sehr mangelhafte Dokumentationen verfügbar sind. Wir haben auch feststellen müssen, dass die Einarbeitungszeit in die alternativen Tools unsere verfügbare Zeit weit übersteigen würde.