



System Specification

Project Name	Fitness For Geeks		
Project Manager			
Document Owner	Requirements Engineer		
Created on			
Last modified	16/03/2005 19:54		
State	X	in Work	
		vorgelegt	
		Released	
Dokumentablage	SystemSpecificationTemplate.doc		

Document History

Revision			Chapter	Modification	Author
Nr.	Date	Version			
1		0.1	All	Initial Draft	

Content

1 Initial Situation and Goal

1.1 Initial Situation

A lot of people want to lose weight, gain weight or just maintain their weight while eating healthier in a simple way without having to do intense research on the subject. We want to help the user with choosing the right recipe for their meal. A lot of applications like [Balance](#) have a calories counter where the user must search for each product/food in their database and then add it to the list of eaten food. The application [MyFitnessPal](#) does it in a similar way but also provides a barcode scanner, a forum and the users have a private database for their own recipes which they can add to their daily calories later. Our application will be like MyFitnessPal, but will combine eating healthy, fitness and socializing a lot more. It will have a base database like the ones of the before mentioned applications, but the users will also be able to add recipes to our database with tags, which can be later specifically searched for. The forums of our application will be like the ones of [reddit](#), where users can post about whatever they want, but also add recipes to the forum which will be also added to our default database with the tag of the forum. Because our recipes will be community driven we also provide a rating system, so the user knows what the other people like and what not and therefore decide what to try and what not. The user will get their recommended nutrition, which can be influenced by adding a workout, and then see all recipes which are valid for their current recommended nutrition (after adding a recipe or workout the current amount changes).

1.1.1 Application Domain

Figure to illustrate the domain of application

Every person must eat enough to fill their total daily energy expenditure and decide on what to eat. That is why the user gets all recipes which are valid for their remaining nutrition needs. The user might want to change things up by trying something different that he has not tried before or a recipe from a friend, so the database must be able to be changed by the community. If a hobby chef or even a real chef commits a recipe to our site and wants to know what the users think, we must provide a rating system, so the owner of the recipe can see whether the recipe is very good or not. (Of course, the owner must consider that the person rating the recipe might have screwed up while cooking)

1.1.2 Glossary

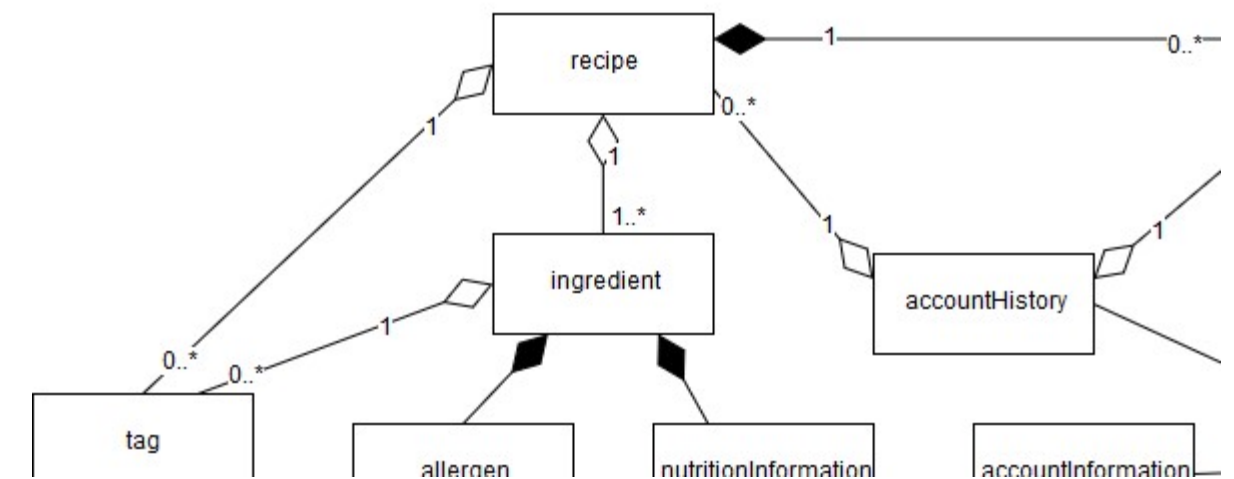
Forum

A forum is basically a sub view of all recipes, where users can post and talk about whatever they want.

Rating System

The rating system will be a basic point system from one to ten, where ten is the best and one is the worst.

1.1.3 Model of the Application Domain



1.2 Goal Definition

Our target group are the average persons, for people that are serious about nutrition, our application will be too simple.

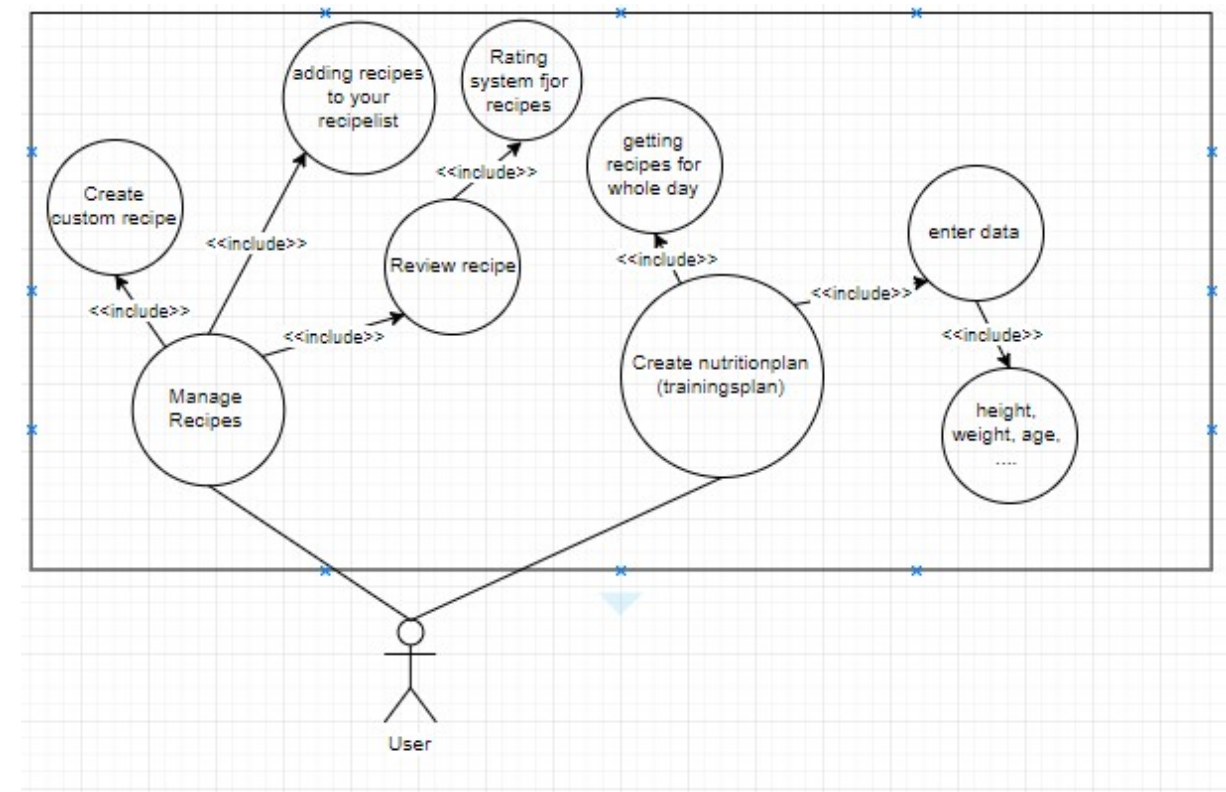
The goal of us is to make the user feel like eating healthy is not a chore, but something they can be proud and accomplished about.

We do not think that the users must have any special knowledge about nutrition while using our application.

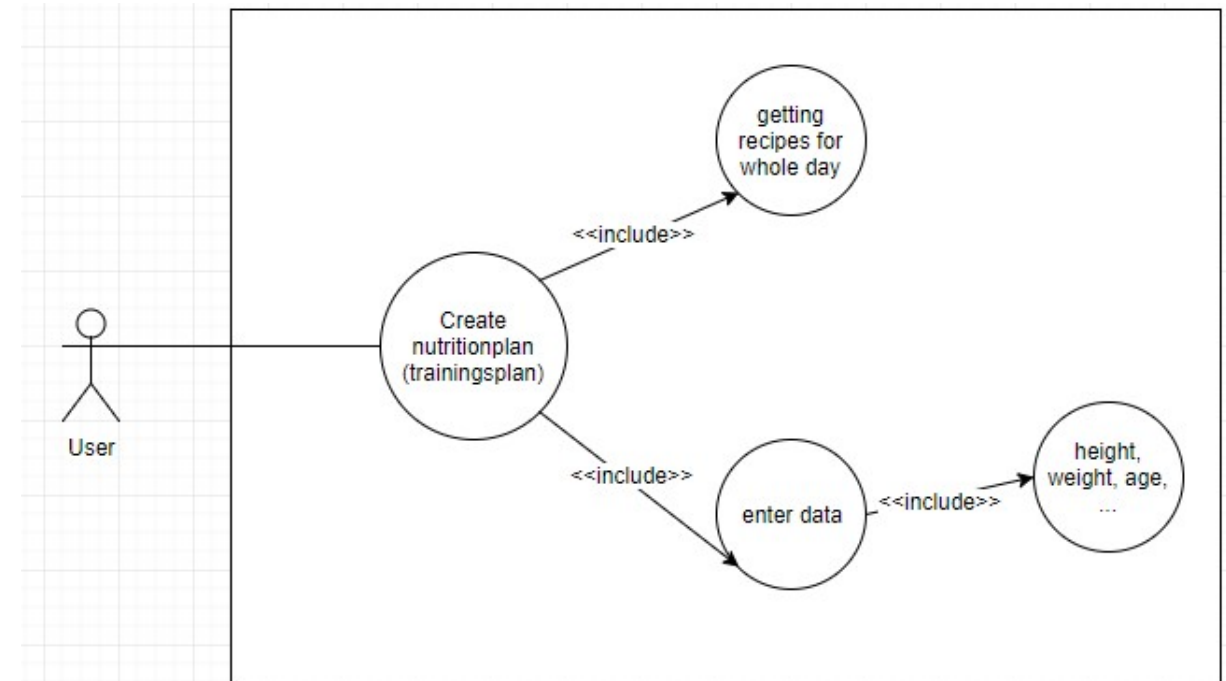
Our application will be web based, since web applications can be accessed by every operating system while still maintaining only one codebase.

2 Functional Requirements

2.1 Use Case Diagrams



2.2 Use Case ID 1 – Create nutritionplan

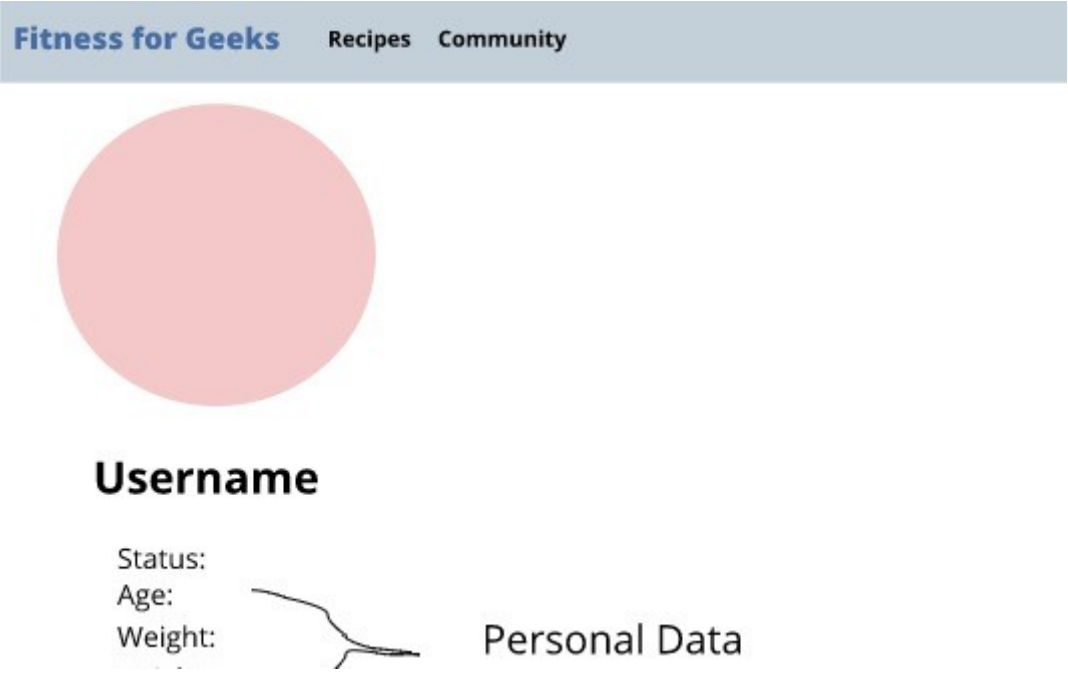


The main function of this use case is that the user gets a nutritionplan and optionally also a trainingsplan. Included is getting the recipes for the whole day so that the user knows what healthy foods to eat. And for the recipes to be precise the user needs to enter data such as height, weight, age, optional bf% and so on.

2.2.1 Characteristic Information

Superior business process:	Process-ID: 1 – Create nutritionplan (and trainingsplan)
Goal:	Create nutritionplan so that the user knows what to eat
Precondition:	The user needs to enter the personal data such as height, weight, age, (optional) bf% and so on
Postcondition:	The user receives a nutrionplan (and trainingsplan) based on the previously entered data
Involved User:	The person using the program
Triggering Event:	The user wants to get a nutritionplan (and trainingsplan)

2.2.2 GUI to call the use case



<i>Input field</i>	<i>Valid inputs</i>
Status	String
Age	Integer

<i>Input field</i>	<i>Valid inputs</i>
Weight	Supports both metric and imperial system
Height	Supports both metric and imperial system

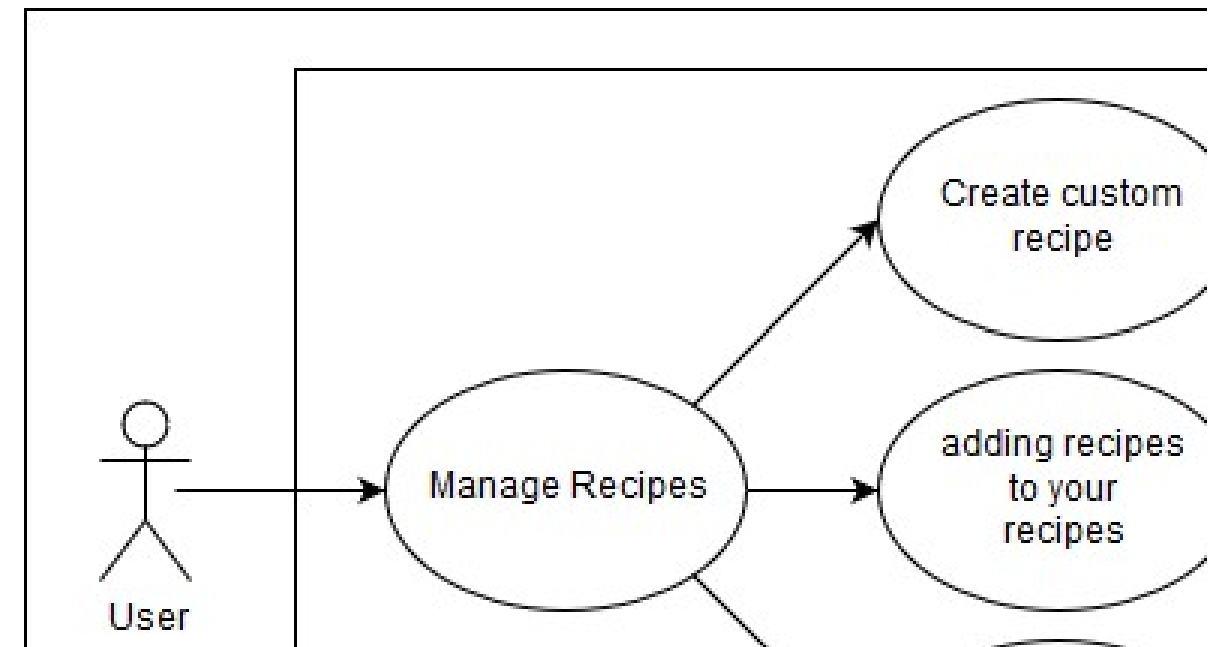
Scenario for the standard use (good case)

<i>Step</i>	<i>User</i>	<i>Activity</i>
1	User	Log in
2	User	Open the menu and click the “Create nutritionplan” button
3	User	Edit the personal data if the user wants to
4	User	Choose between custom or generated nutrionplan
5	User	The user can adjust the nutrionplan if needed
6	User	Log out

2.2.3 Scenarios for non-standard uses (bad cases or work around cases)

<i>Step</i>	<i>User</i>	<i>Activity</i>
1	User-Connection	The user loses connection – the website can only be used with internet connection (100% of the time due to caching)

2.3 Use Case ID 2- Manage Recepes



The main function of this use case is that the user can manage his recipes, here the user can create custom recipes which he can add to their recipelist and the user can also rate and leave comments on recipes, here he can inform other users about the experience he had with this particular recipe.

2.3.1 Characteristic Information

Superior business process:	The user can change his saved recipes, create custom ones and review them.
Goal:	The user maintains his recipelist.
Precondition:	The user wants to change his recipelist.
Postcondition:	The user now has a updated recipelist.
Involved User:	The person using the program
Triggering Event:	The user wants to change his recipelist.

2.3.2 GUI to call the use case



<i>Input field</i>	<i>Valid inputs</i>
Button	Click

Scenario for the standard use (good case)

<i>Step</i>	<i>User</i>	<i>Activity</i>
1	user	The user clicks on the button to change his recipelist.
2	user	The user adds ore deletes recipes from his list, creates new ones or comments on them.
3	user	The user clicks a button to save the changes he made or to submit the comment he wrote.

2.3.3 Open Points

- Internet connection all the time
- The data can be (unintentionally) entered incorrectly (age 80 but the person is 20)

3 Non-functional Requirements

4 ID:	1
Name:	Usability requirement
Type:	USE
Description:	The website should be useable in under 150ms. We are going to try to cache as much as possible.
Assigned use cases:	1, 2

ID:	2
Name:	Legal requirement
Type:	LEGAL
Description:	We will have to ask the user for their permission, so we can use their data. We will also need to inform them about our use of cookies.

5 Quantity Structure

We need a server for the profile information so that people can edit their data (height, weight, bf%, ...) and are also able to write comments on recipes and save those recipes to their personal profile. Also crucial is the use of a relational database.

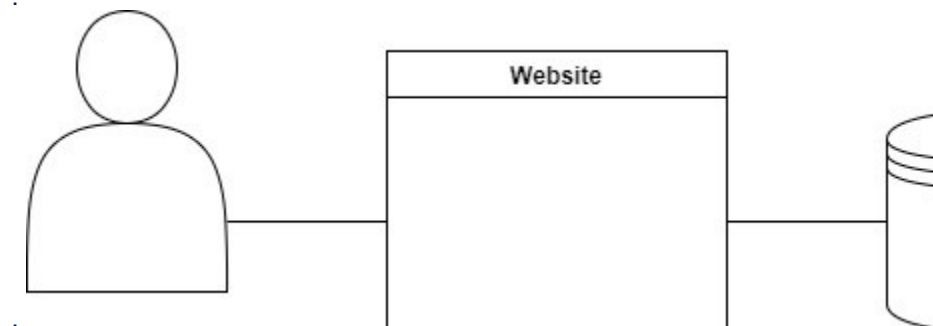
Profile

Each user will have a profile so we need about 150B for each profile as well as 1MB for the recipe storage. Totally we need about 100GB for the first release. However, 100GB will not be enough for: say 100 000 users.

Conclusion

All in all we need about 1TB disk space – to be on the safe side.

2 System Architecture and Interfaces



The user can for example manage their recipes and information through our web application, which then notifies the server and the server sends queries to the database accordingly.

Acceptance Criteria

2.1 1 – Create nutritionplan

Test Step	Expected Behaviour
Create nutritionplan	The nutritionplan can be now seen an is editable
Customize nutritionplan	The changes get saved
Save nutritionplan	The nutritionplan gets saved and is viewable

2 – Manage Recipes

Test Step	Expected Behaviour
Create custom recipe	The recipe can be used and optionally turned public so other users can see it
Review recipe	The recipe can now be seen under reviews/comments
Adding a recipe to the recipe list	The recipe now has a higher priority than recipes that are not in the recipe list

3 **List of Abbreviations**

Abbreviation	Description
Bf%	Bodyfat percentage

4 **References**

5 **List of Figures**