CookieApp
Software Architecture Document

Version 1.1

Revision History

Date	Version	Description	Author
19.11.2014	1.0	Document created Text Written	CVerdion
19.11.2014	1.1	Created and inserted Views and Diagrams	CVerdion

Table of Contents

Software Architecture Document

1. Introduction

This documents provides a basic architectural overview of the system CookieApp.

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions, which have been made on the system.

1.2 Scope

This document is supposed to help the software development process. It provides the basic architecture of the project and serves as construction plan for the classes and their structure.

2. Architectural Representation

The project will be constructed using the Model-View-Controller (MVC) architectural pattern for applications with user interfaces. This pattern separates the application in 3 sections, one for the view, which is the user interface, which is a graphical user interface in this case. Then there is the model, which contains all the entities and entries in the database. The controller will take the data from the model and provides in to the view, so the view can display the content, the user requested.
Through this separation the application will be flexible and easy to maintain and add new features to it. Also packages of this application can be reused for other software projects.

3. Architectural Goals and Constraints

This system takes use of the Spring MVC Framework.
This features a vast number of java extensions, which can be used in Java-Applications to build all kind of applications after the MVC Architectural Pattern

- Controller: Spring
- Model / DataAccess: Eclipse EMF / CDO
- View: Eclipse RAP
-

4. Use-Case View

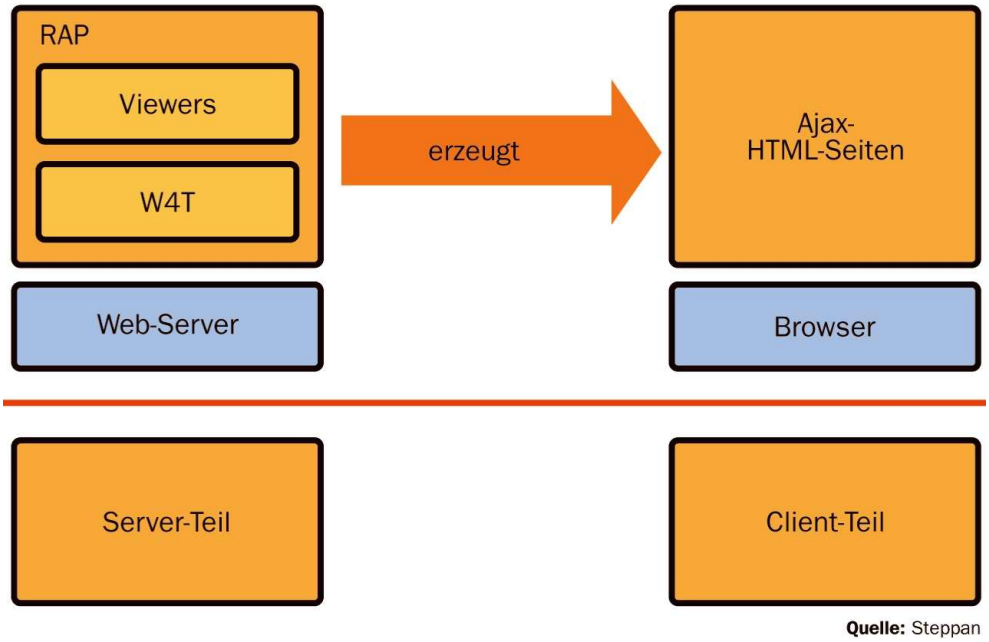
(n/a)

4.1 Use-Case Realizations

(n/a)

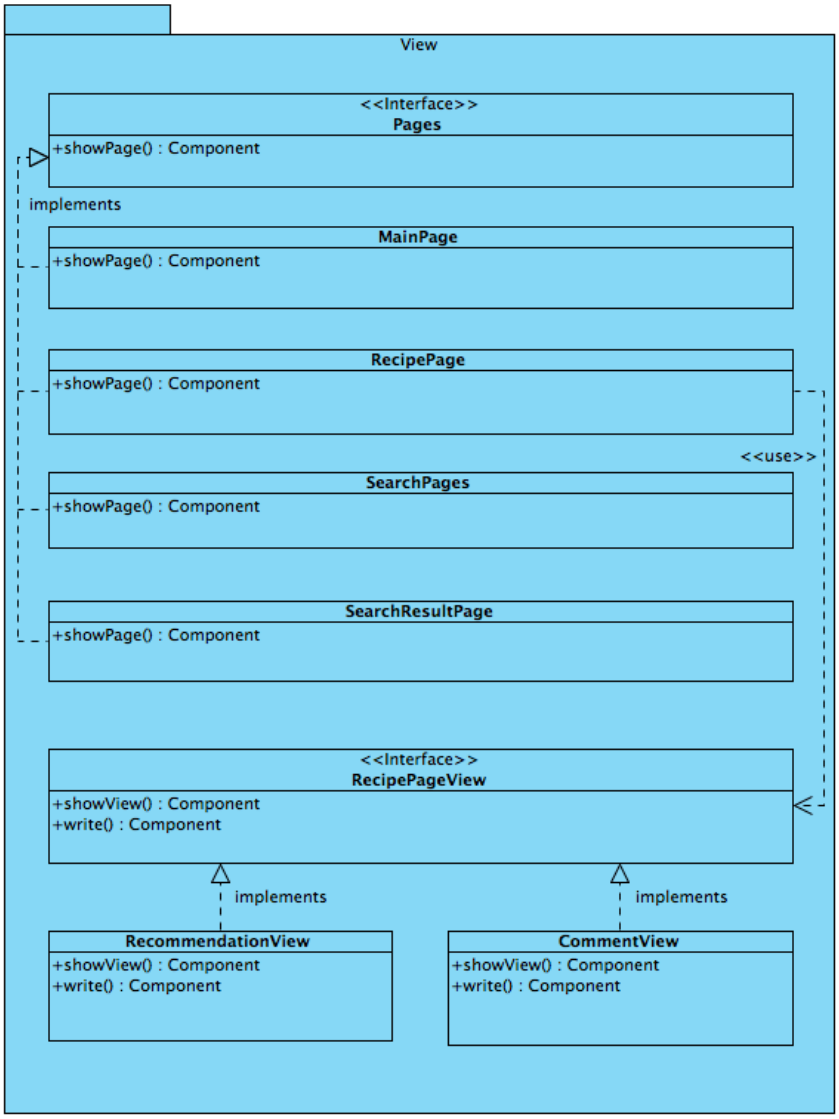
5. Logical View

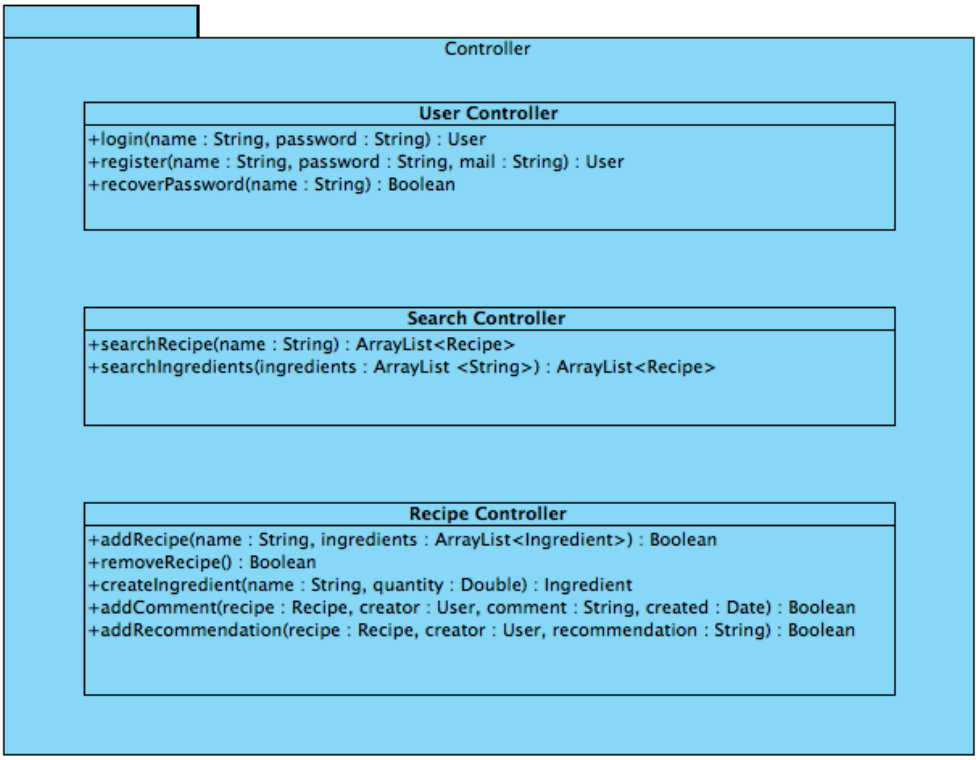
5.1 Overview

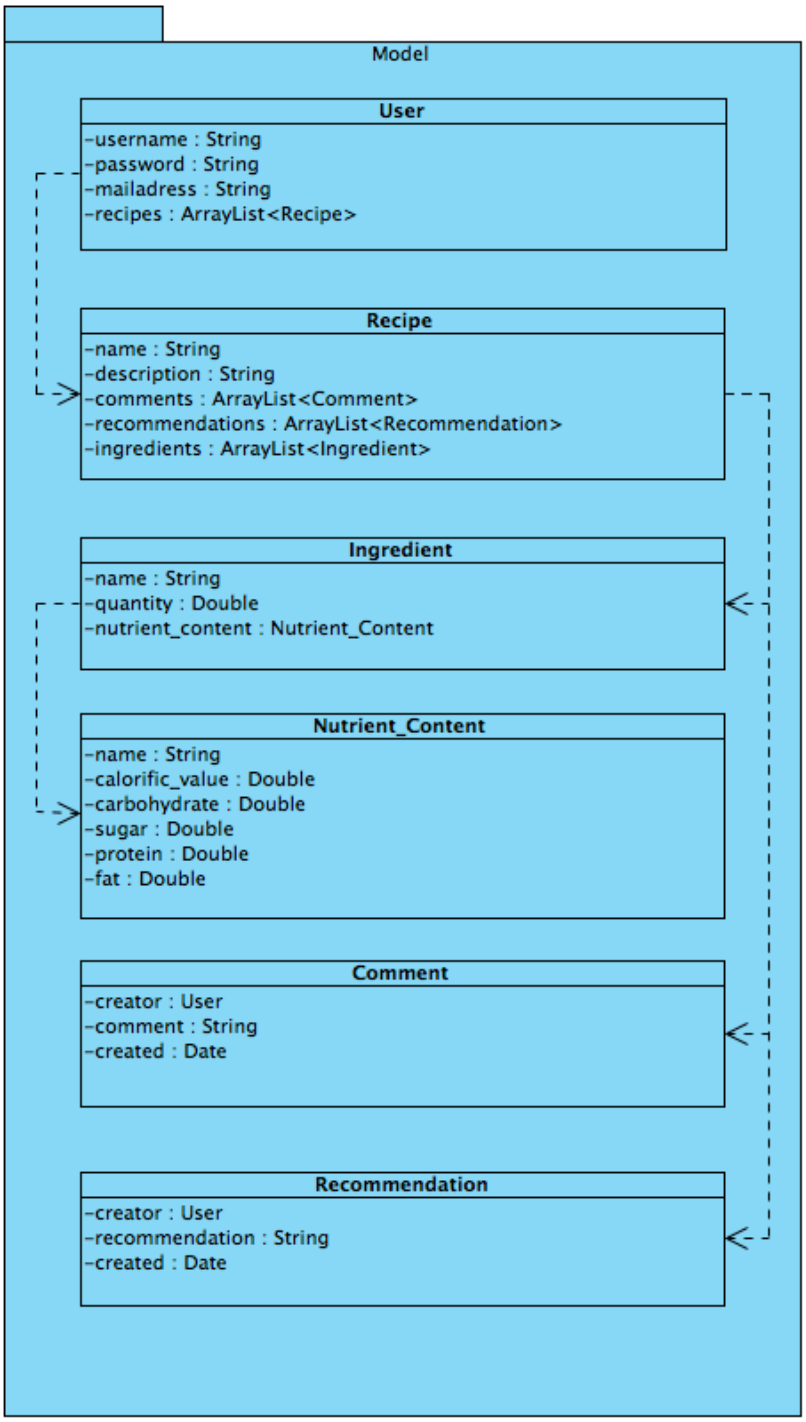


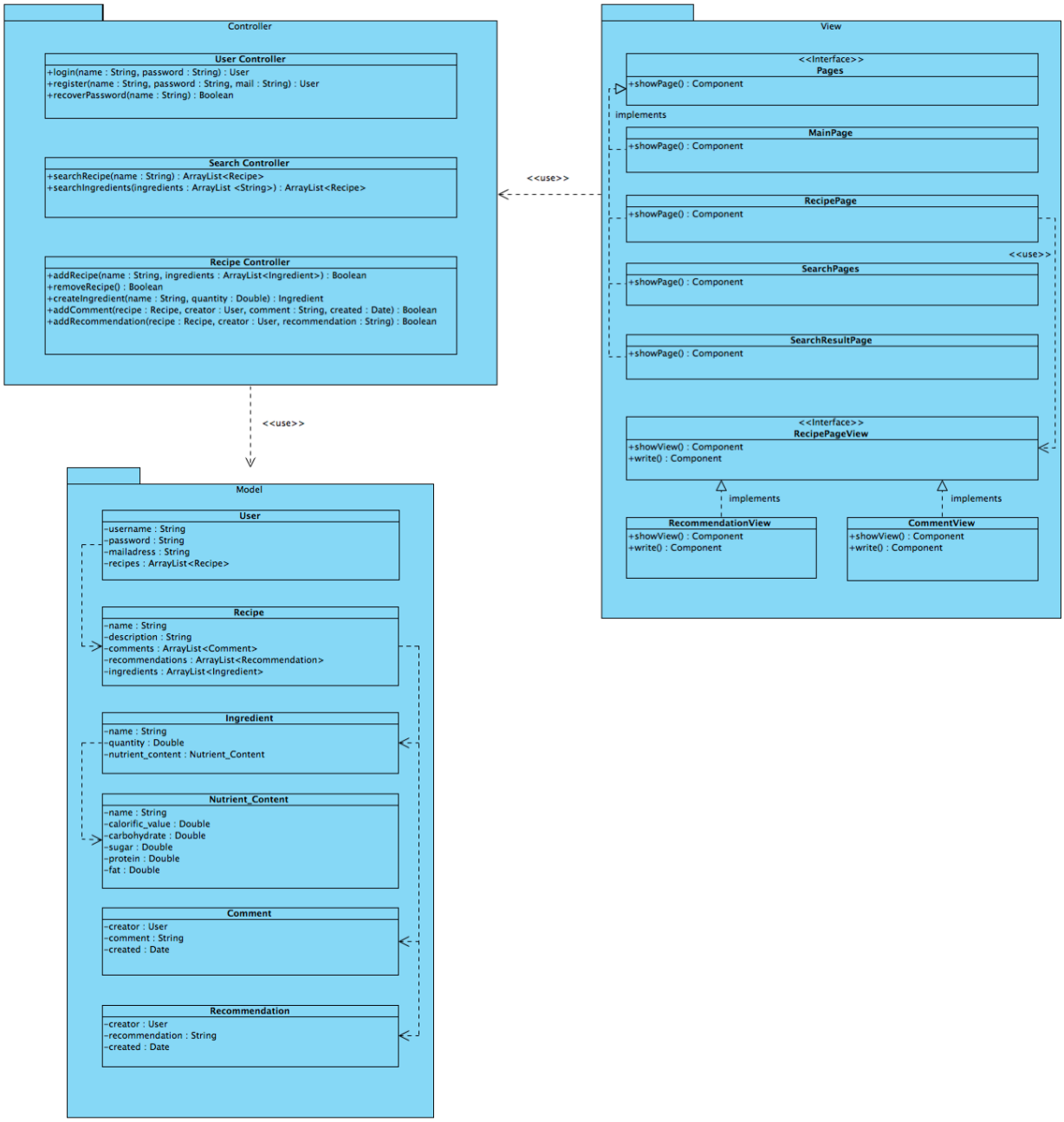
Quelle: Steppan

5.2 Architecturally Significant Design Packages




Controller
User Controller
+login(name : String, password : String) : User
+register(name : String, password : String, mail : String) : User
+recoverPassword(name : String) : Boolean
Search Controller
+searchRecipe(name : String) : ArrayList<Recipe>
+searchIngredients(ingredients : ArrayList <String>) : ArrayList<Recipe>
Recipe Controller
+addRecipe(name : String, ingredients : ArrayList<Ingredient>) : Boolean
+removeRecipe() : Boolean
+createIngredient(name : String, quantity : Double) : Ingredient
+addComment(recipe : Recipe, creator : User, comment : String, created : Date) : Boolean
+addRecommendation(recipe : Recipe, creator : User, recommendation : String) : Boolean


Model
User
-username : String
-password : String
-mailadress : String
-recipes : ArrayList<Recipe>
Recipe
-name : String
-description : String
-comments : ArrayList<Comment>
-recommendations : ArrayList<Recommendation>
-ingredients : ArrayList<Ingredient>
Ingredient
-name : String
-quantity : Double
-nutrient_content : Nutrient_Content
Nutrient_Content
-name : String
-calorific_value : Double
-carbohydrate : Double
-sugar : Double
-protein : Double
-fat : Double
Comment
-creator : User
-comment : String
-created : Date
Recommendation
-creator : User
-recommendation : String
-created : Date



6. Process View

[This section describes the system's decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.]

7. Deployment View

[This section describes one or more physical network (hardware) configurations on which the software is

*deployed and run. It is a view of the Deployment Model. At a minimum for each configuration it should indicate the physical nodes (computers, CPUs) that execute the software and their interconnections (bus, LAN, point-to-point, and so on.) Also include a mapping of the processes of the **Process View** onto the physical nodes.]*

8. Implementation View

[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]

8.1 Overview

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers.]

8.2 Layers

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

9. Data View (optional)

(n/a)

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]

10. Size and Performance

(n/a)

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

11. Quality

(n/a)

[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated.]