Recipes Website

Analysis and Design Document

Student: Kando Edina

**Group:30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

The Recipes Website is going to be a web application meant for users that enjoy cooking. The app will support three types of users: regular user, chef and admin.

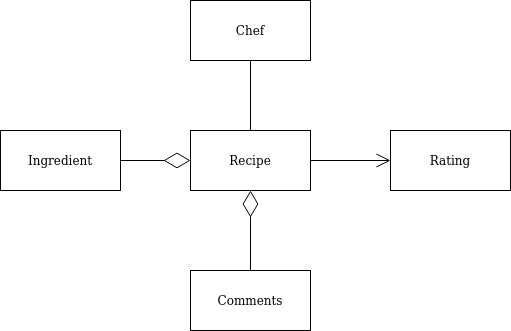
The regular users will have the most limited set of actions they can do on the website. These actions are: creating an account, browsing recipes, adding to favorites, rating, commenting and privately chatting with other users.

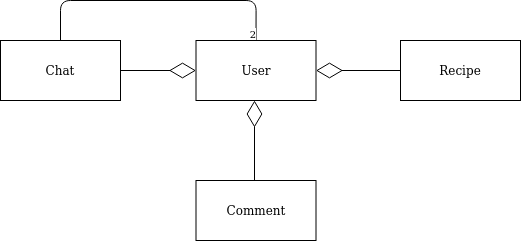
The chefs are a type of user that need to be validated. Besides the usual actions presented before, these users can also add/remove recipes (remove only recipes of their own).

The admin is responsible with approving chef role requests.

# Elaboration – Iteration 1.1

# Domain Model

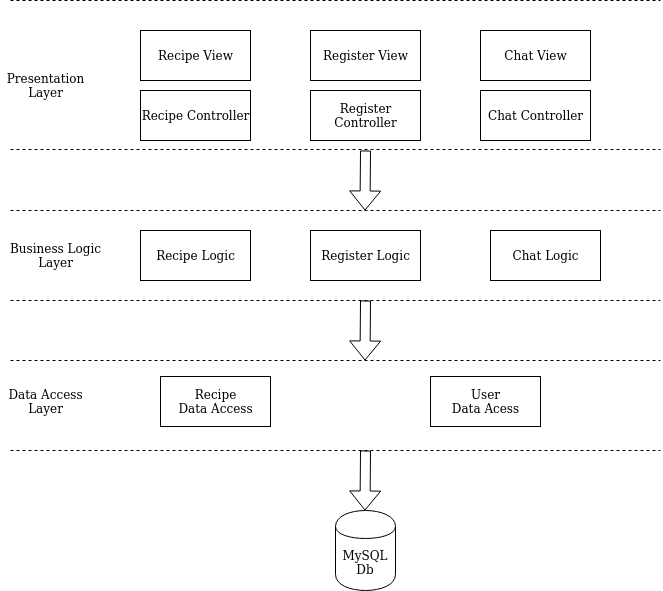




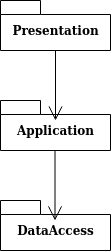
# Architectural Design

## Conceptual Architecture

Architectural Pattern: Layered Architecture

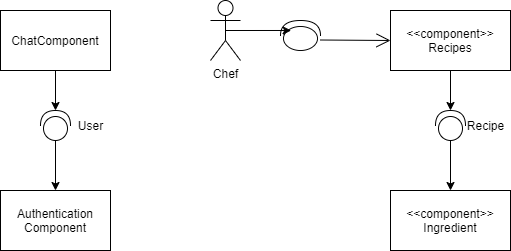


## Package Design



## Component and Deployment Diagrams

Component diagram:



Deployment diagram:

# 

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

1.1 .Test case for registration pop-up

1.Verify that registration popup appears

2.Verify that all the specified fields are present on the registration popup

3.Verify that the required/mandatory fields are marked with \* against the field

4. Verify that the popup has submit and cancel buttons

5. Verify that hitting submit sends data to the server

6. Verify that clicking cancel button closes the popup and stops the request

7. Verify that validation is present for the fields

1.2. Test case for adding a recipe

1. Verify that authenticated users with chef role get access to recipe creation panel

2. Verify that recipe creation panel is working fine for single recipe creation

3. Verify panel validation for checking mandatory field

4. Verify that recipe created by user get visible on the website after certain period of time

# Future improvements

* Editing account information;
* Editing recipes added;
* Sending email validations.

# Bibliography