Software Requirements Specification (SRS) Document

Fork and Share

branhenr/Group-8-Project

05/08/2025

Final Version

Brandon Vazquez, Elizabeth Spratt

1. Project General Description (Elizabeth Spratt)

The goal of the “Fork and Share” web application is to connect users to a wide variety of recipes by allowing them to like recipes , save recipes, and follow content creators. The app will also allow content creators to publish their recipes to the app, as well as interact with users who leave replies.

2. Product Features (Brandon Vazquez)

The Fork and Share application is designed for users to be able to interact with other users to explore and share recipes to try out. Below are the key features and functions of the applicaiton:

Content Preference Customization: Users can filter the recipes that they see by adding preferences such as vegan, kosher, lactose intolerant, food allergies, gluten free, halal, etc.

Personalized Profile Creation: Users can create their own personalized profiles showing what recipes they like and which chefs they follow.

Interactive Content Interaction: Users can like recipes that they enjoy and be able to save any recipes to their own page.

Recipe Creation and Sharing: Provides a platform for chefs to post recipes for other users to try out and be able to share with others.

3. Functional Requirements

User (Elizabeth Spratt):

-FR 0: The app will allow the user to create a profile

-FR 1: The app will allow the user to edit their profile

-FR 2: The app will store the user’s information

-FR 3: The app will allow the user to search recipes based on preferences they set

-FR 4: The app will allow the user to view different recipes

-FR 5: The app will allow the user to like different recipes

-FR 6: The app will allow the user to view liked recipes

-FR 7: The app will allow the user to leave reviews on recipes

-FR 8: The app will allow the user to add a recipe to their own saved list.

Chef (Brandon Vazquez):

-FR 0: The app will allow chefs to create a profile.

-FR 1: The app will allow chefs to edit their profile.

-FR 2: The app will store the chef's information.

-FR 3: The app will allow the chef to reply to reviews.

-FR 4: The app will allow the chef to pin their best recipes.

-FR 5: The app will allow chefs to post recipes.

-FR 6: The app will allow chefs to modify their recipes.

4. Non-Functional Requirements

User (Elizabeth Spratt):

NFR 1: On the main page it will generate 6 recipes in a grid like style. From there the user can set their preferences or click on the next page to see the next 6 recipes. This will promote a quick system by not displaying all posts at once.

NFR 2: The User will be able to view their saved recipes in no more than 5 seconds. It will display a max of 10 at a time.

NFR 3: Removing recipes from the users “list” should take no longer than 3 seconds.

Chef (Brandon Vazquez):

NFR 0: Posting recipes should take less than 5 seconds.

NFR 1:Deleting recipes should take less than 3 seconds.

NFR 2: Loading up the chef’s profile should take no more than 10 seconds.

5. Scenarios

a. Users - *Forky Mcshare* (Elizabeth Spratt)

I.

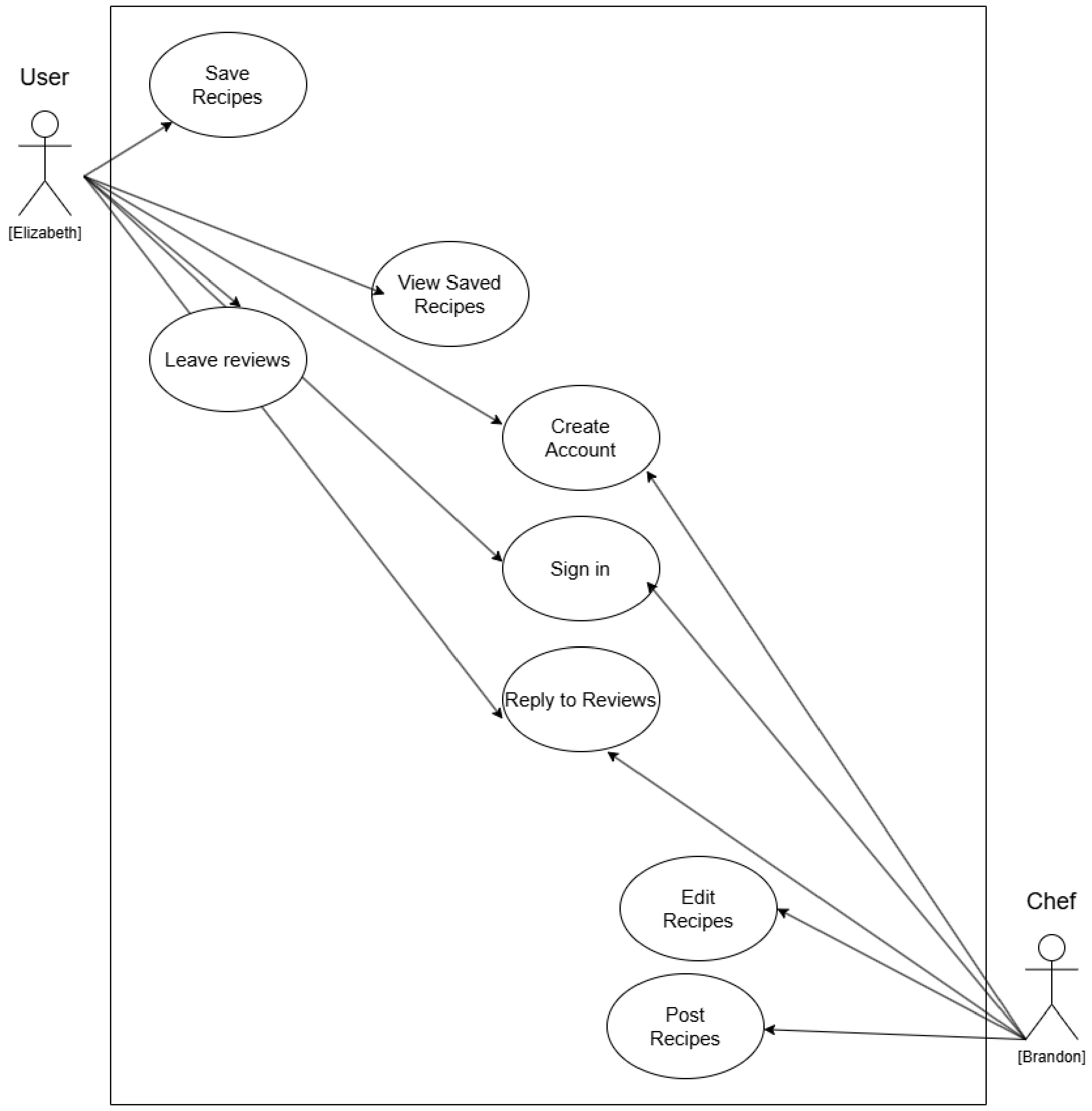
- **Initial Assumption**: The user has access to the web application and is logged in and on their profile page of ‘Fork and Share’.
- **Normal**: The user will be able to navigate to the main page where the web application will generate multiple recipes. From there the users can like recipes they are interested in.
    - The users will be able to set preferences to filter recipes.
- **What Can Go Wrong**: The user can accidentally add a preference that they did not mean to add. They will be able to remove preferences individually.
- **Other Activities**: The user will be able click on a “view more details button” to see more information on particular recipes.
- **System State on Completion**: The user will be prompted to view saved recipes.

b. Chef - *Billybob* (Brandon Vazquez)

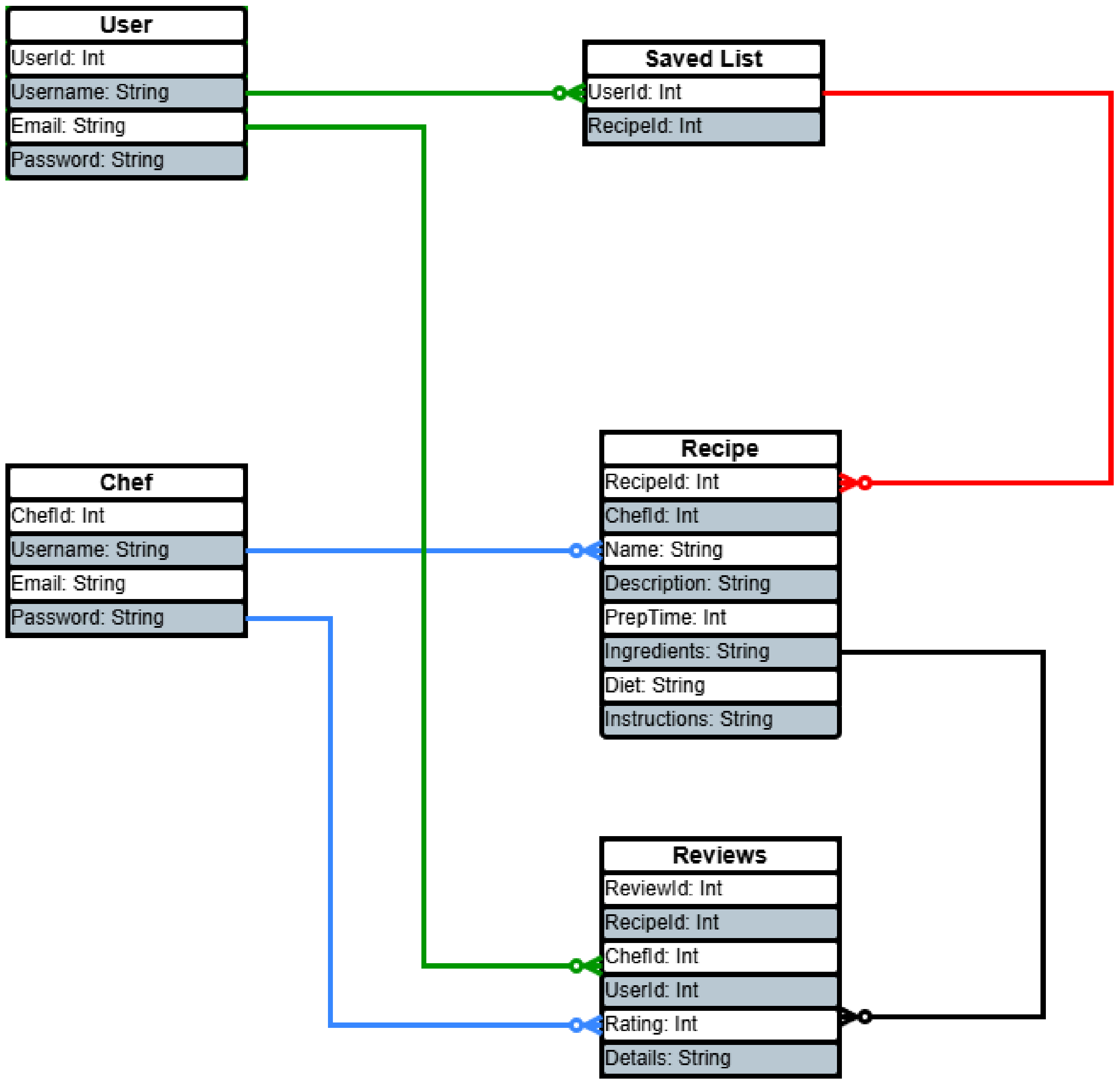
I. Posting recipes (Create services)

- **Initial Assumption**: The chef has access to the web app and is logged in and on their profile page.
- **Normal**: There will be a button on the profile page that enables the chef to post/create a recipe. The chef will be able to put in their recipe with all the information that goes with it.
- **What Can go Wrong:** The chef makes a typo or accidentally posts the recipe incomplete. The chef will be able to modify the recipe with an edit button on the recipe.
- **Other Activities**: The chef can also choose to delete the recipe instead of editing it.
- **System State on Completion**: The user will be prompted with a message that their recipe has been successfully modified.

6. Use-Case Model

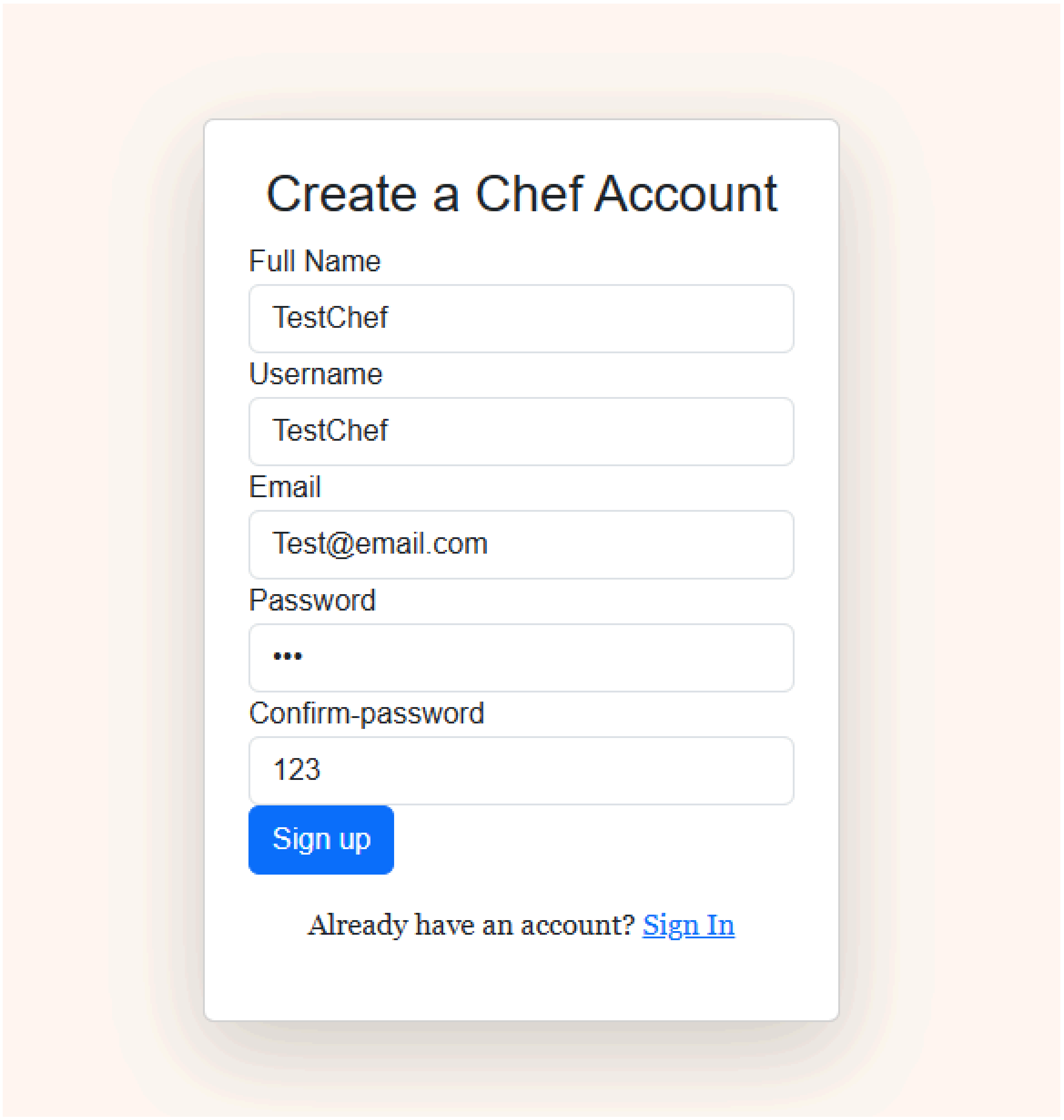


7. Database Schema



8.Scenarios with Screenshots

Chef Scenario(Brandon Vazquez):



Create a Chef Account

Full Name: TestChef

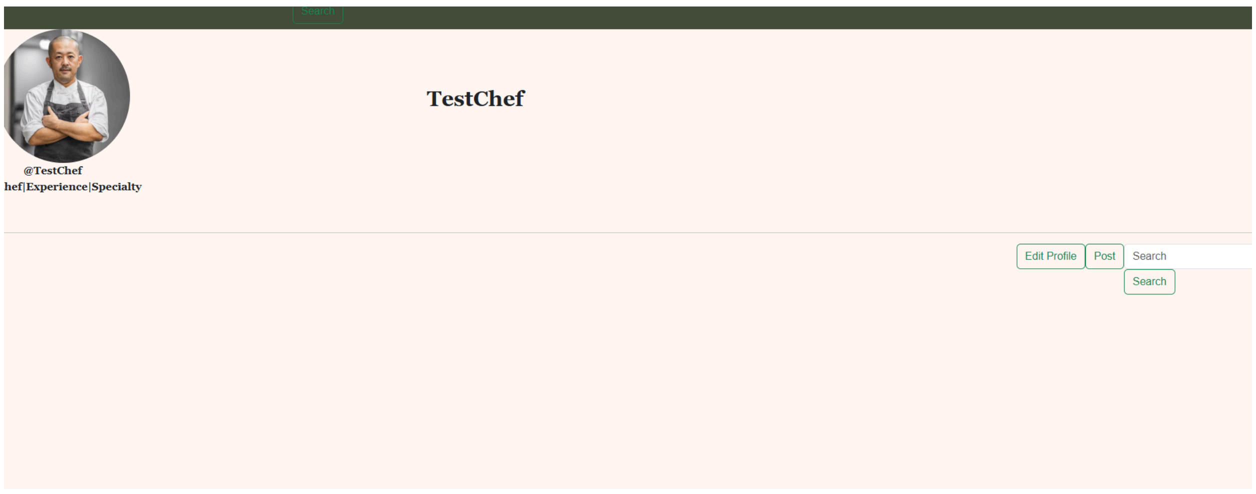
Username: TestChef

Email: Test@email.com

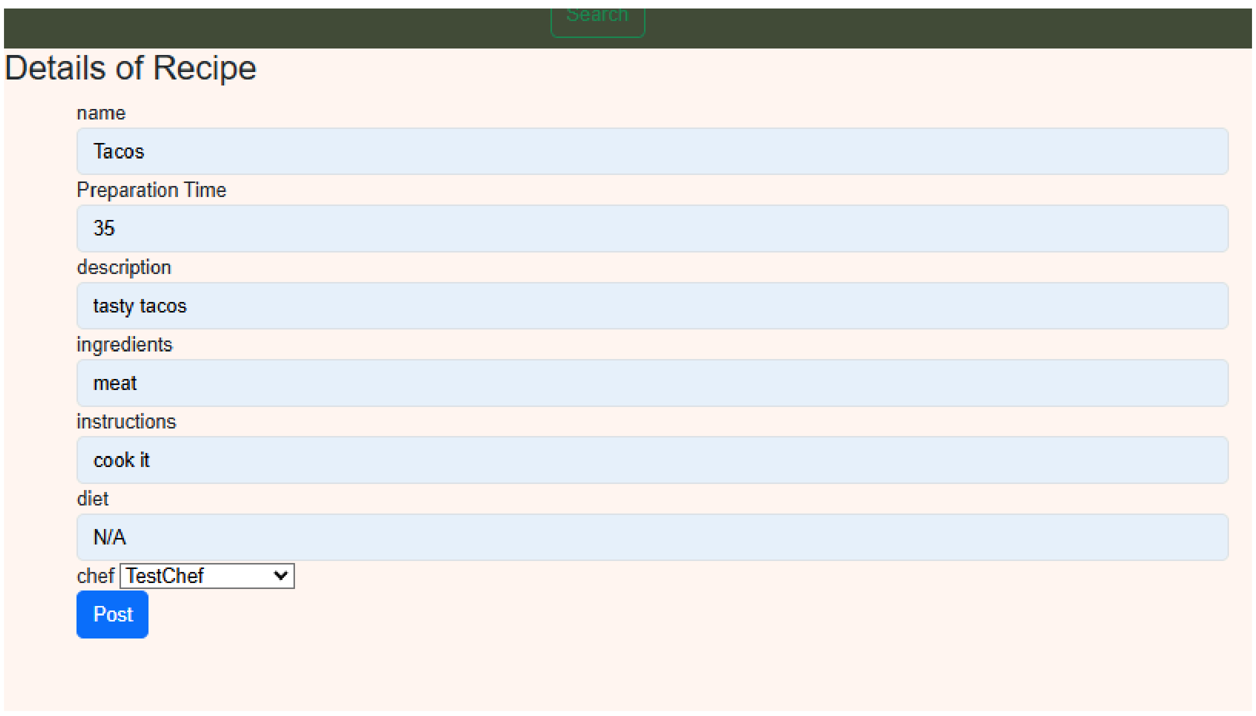
Password: •••

Confirm-password: 123

Creating the chef account.



The chef is able to see their profile.



Details of Recipe

name
Tacos

Preparation Time
35

description
tasty tacos

ingredients
meat

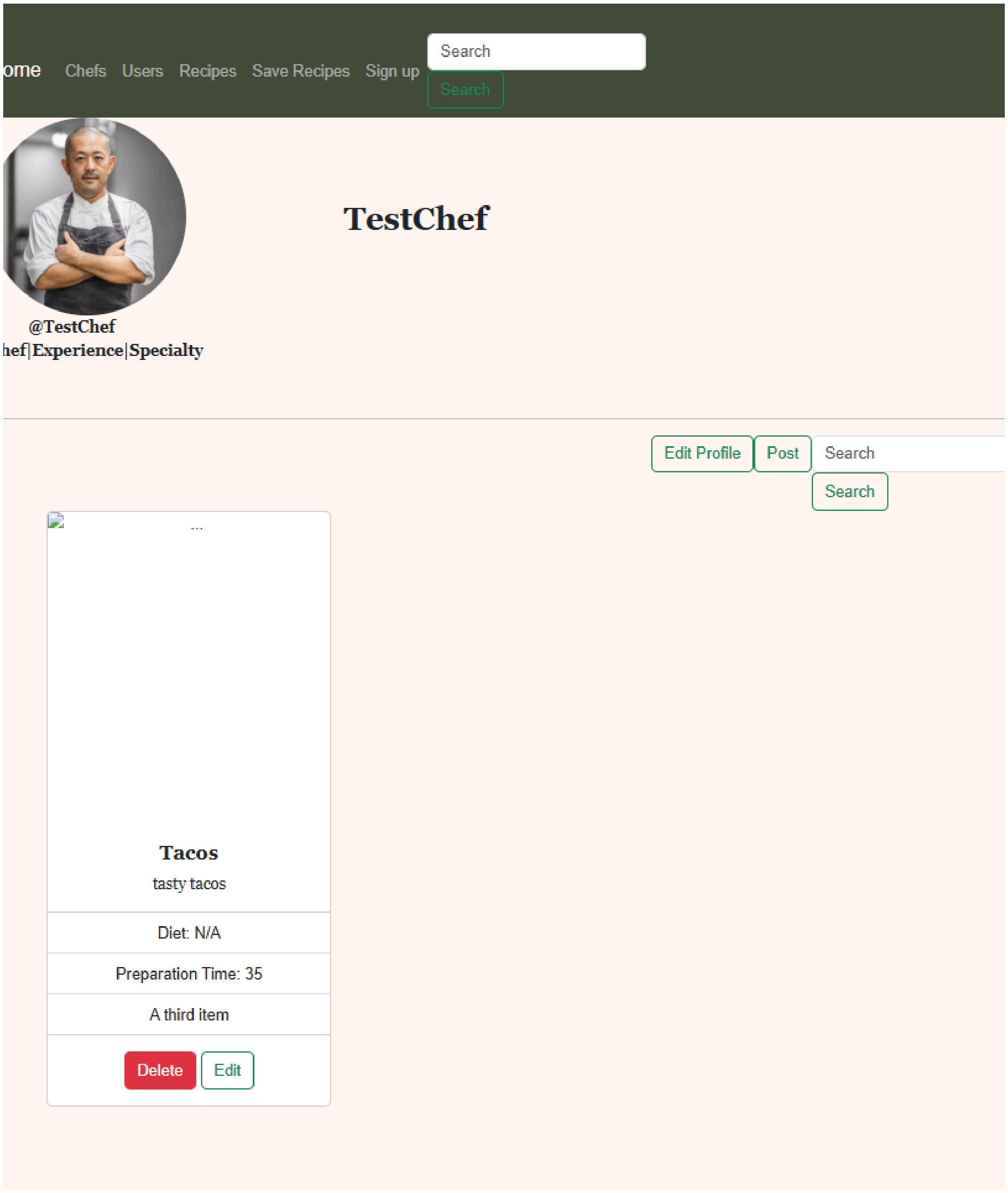
instructions
cook it

diet
N/A

chef TestChef

Post

The chef created a recipe to post.



TestChef

@TestChef
hef|Experience|Specialty

Tacos
tasty tacos

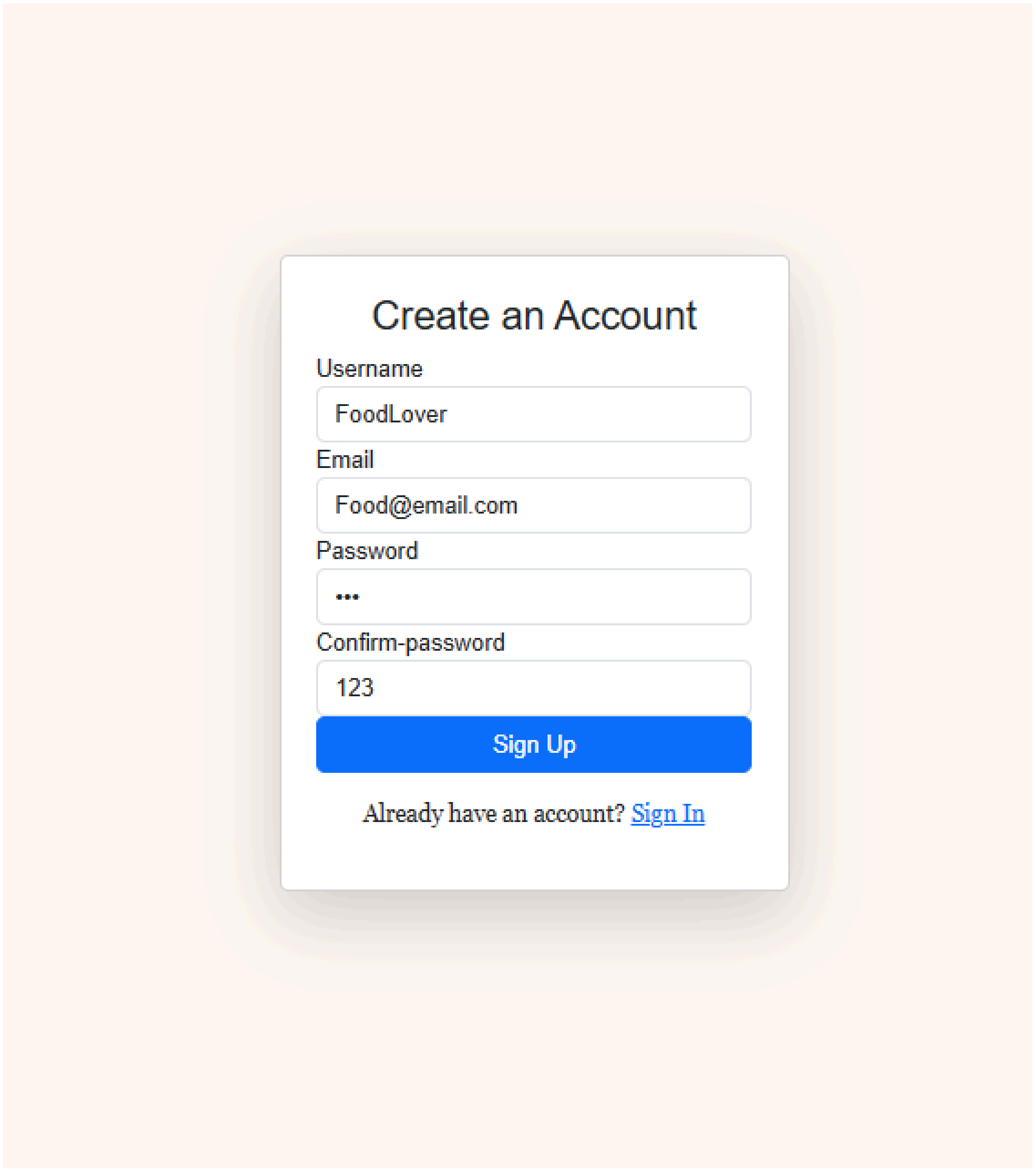
Diet: N/A

Preparation Time: 35

A third item

The recipe posted on the chef’s page and being able to be seen.

User Scenario(Elizabeth Spratt):



Create an Account

Username

FoodLover

Email

Food@email.com

Password

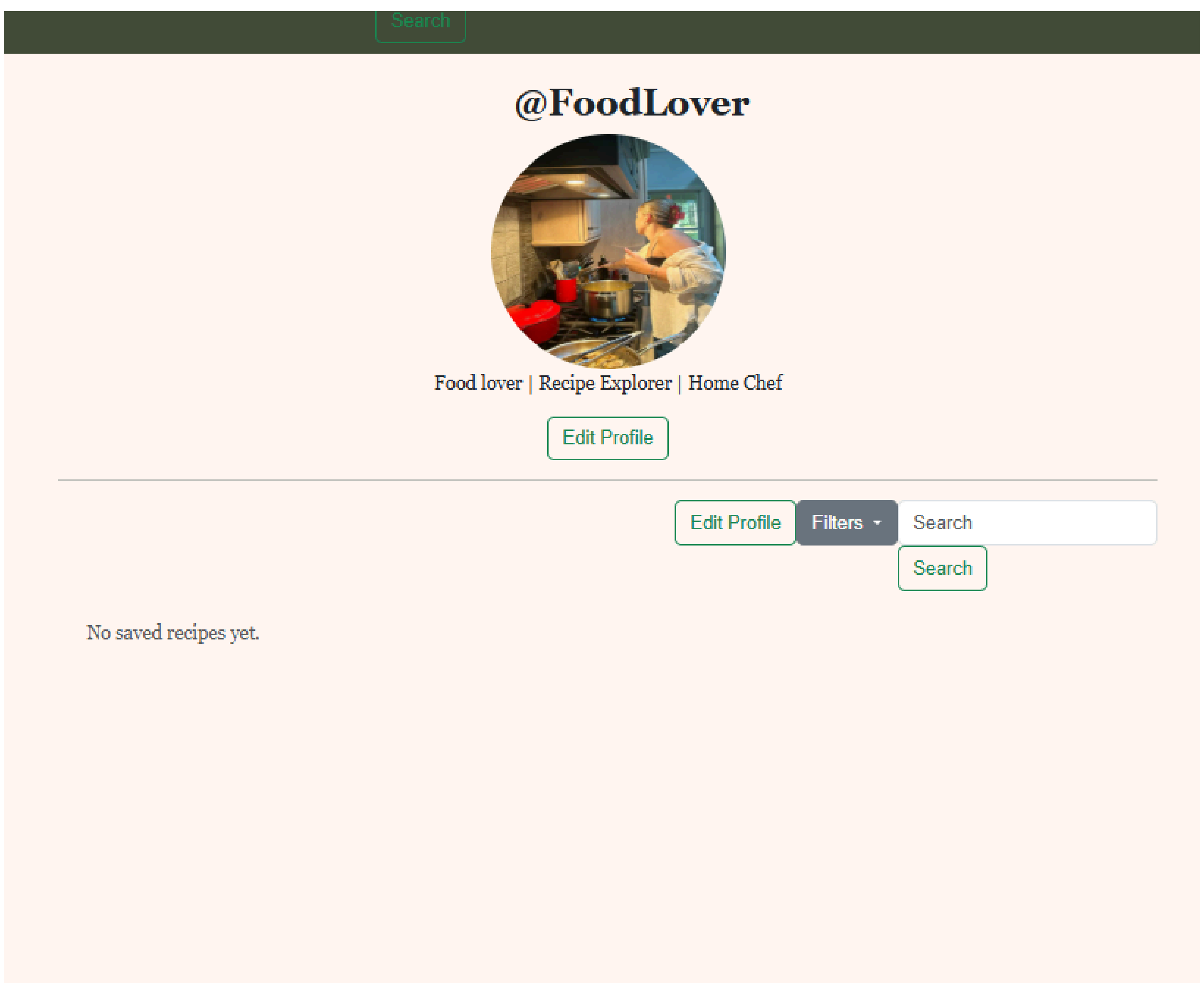
•••

Confirm-password

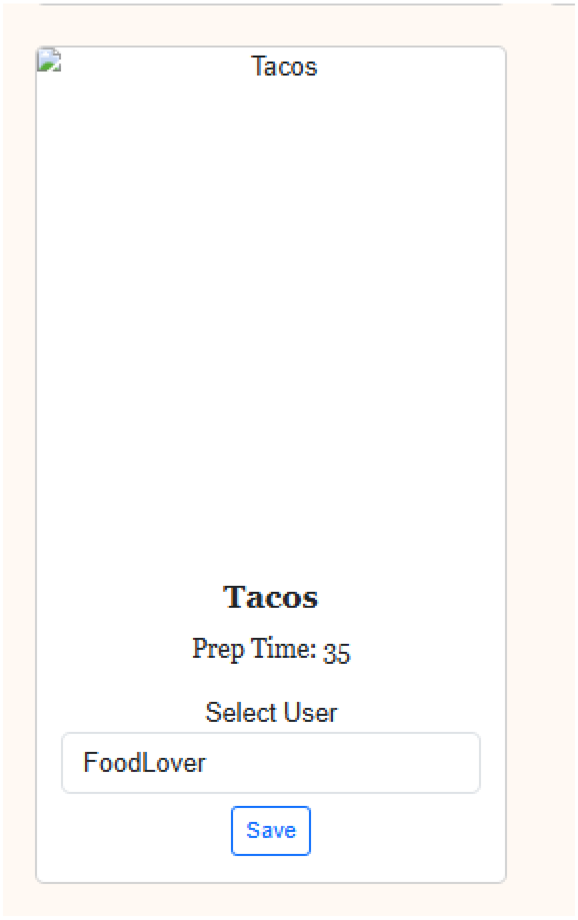
123

Already have an account? Sign In

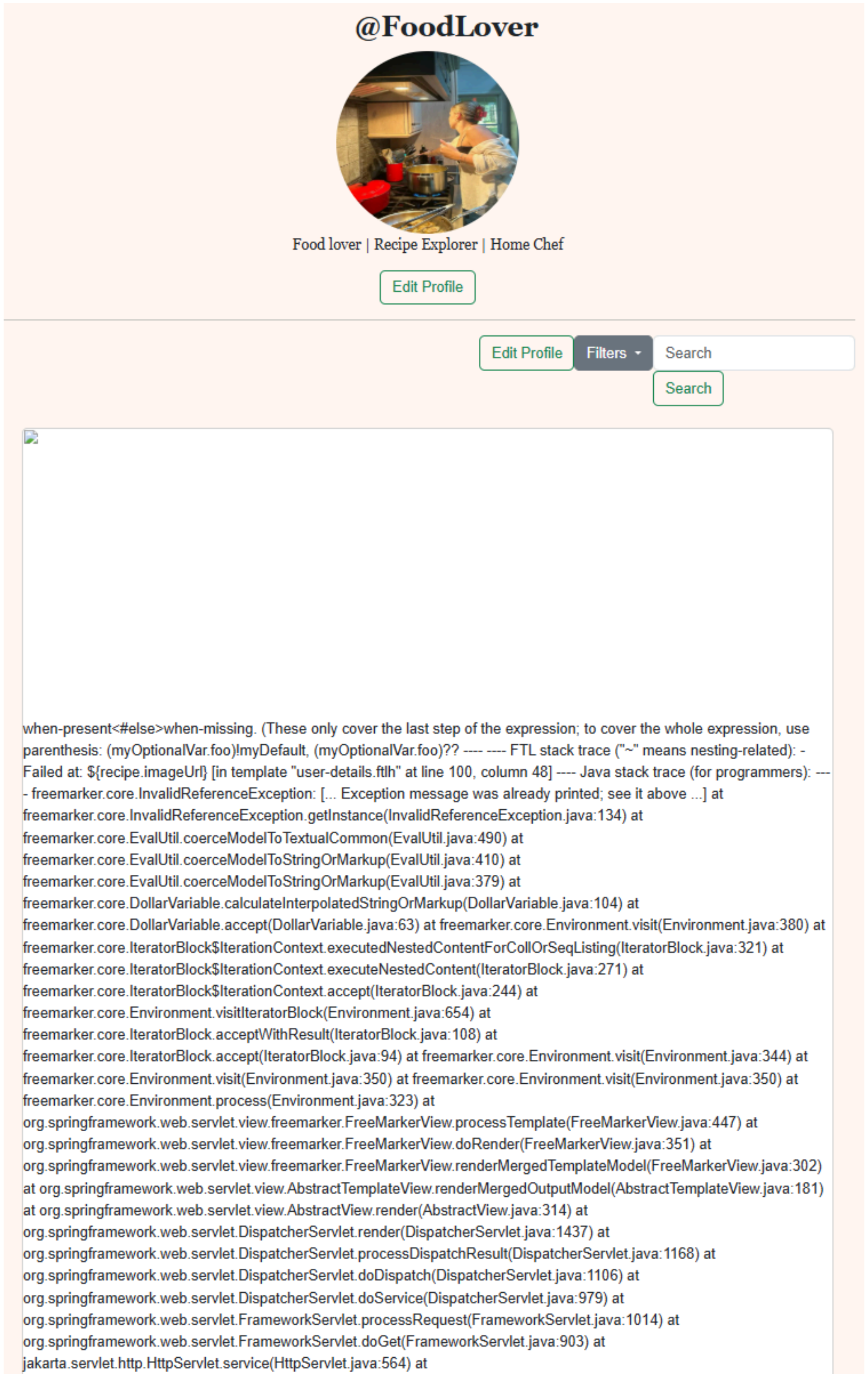
User signing up to create a profile.



User seeing their profile.



User finds a recipe they like and want to save.



```
when-present<#else>when-missing. (These only cover the last step of the expression; to cover the whole expression, use
parenthesis: (myOptionalVar.foo)!myDefault, (myOptionalVar.foo)?? ---- ---- FTL stack trace ("~" means nesting-related): -
Failed at: ${recipe.imageUrl} [in template "user-details.ftlh" at line 100, column 48] ---- Java stack trace (for programmers): ---
- freemarker.core.InvalidReferenceException: [... Exception message was already printed; see it above ...] at
freemarker.core.InvalidReferenceException.getInstance(InvalidReferenceException.java:134) at
freemarker.core.EvalUtil.coerceModelToTextualCommon(EvalUtil.java:490) at
freemarker.core.EvalUtil.coerceModelToStringOrMarkup(EvalUtil.java:410) at
freemarker.core.EvalUtil.coerceModelToStringOrMarkup(EvalUtil.java:379) at
freemarker.core.DollarVariable.calculateInterpolatedStringOrMarkup(DollarVariable.java:104) at
freemarker.core.DollarVariable.accept(DollarVariable.java:63) at freemarker.core.Environment.visit(Environment.java:380) at
freemarker.core.IteratorBlock$IterationContext.executedNestedContentForCollOrSeqListing(IteratorBlock.java:321) at
freemarker.core.IteratorBlock$IterationContext.executeNestedContent(IteratorBlock.java:271) at
freemarker.core.IteratorBlock$IterationContext.accept(IteratorBlock.java:244) at
freemarker.core.Environment.visitIteratorBlock(Environment.java:654) at
freemarker.core.IteratorBlock.acceptWithResult(IteratorBlock.java:108) at
freemarker.core.IteratorBlock.accept(IteratorBlock.java:94) at freemarker.core.Environment.visit(Environment.java:344) at
freemarker.core.Environment.visit(Environment.java:350) at freemarker.core.Environment.visit(Environment.java:350) at
freemarker.core.Environment.process(Environment.java:323) at
org.springframework.web.servlet.view.freemarker.FreeMarkerView.processTemplate(FreeMarkerView.java:447) at
org.springframework.web.servlet.view.freemarker.FreeMarkerView.doRender(FreeMarkerView.java:351) at
org.springframework.web.servlet.view.freemarker.FreeMarkerView.renderMergedTemplateModel(FreeMarkerView.java:302)
at org.springframework.web.servlet.view.AbstractTemplateView.renderMergedOutputModel(AbstractTemplateView.java:181)
at org.springframework.web.servlet.view.AbstractView.render(AbstractView.java:314) at
org.springframework.web.servlet.DispatcherServlet.render(DispatcherServlet.java:1437) at
org.springframework.web.servlet.DispatcherServlet.processDispatchResult(DispatcherServlet.java:1168) at
org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:1106) at
org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:979) at
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1014) at
org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:903) at
jakarta.servlet.http.HttpServlet.service(HttpServlet.java:564) at
```

The saved recipe is on their page.