

Istanbul Technical University
Faculty of Computer and Informatics
Computer Engineering Department



BLG 411E
Requirements Specification
Group 20

Project Name: BastırChef

Muhammed Necati Polat - 070200734

Atacan Utkusavaş - 070190148

Merve İnal - 150200003

Hızır Emirhan Keskin - 150190015

Fadime Buse Yediparmak - 150190053

Nov 21,
FALL 2023

Contents

1	Introduction	2
2	System Requirements	2
2.1	Functional Requirements	2
2.2	Non-functional Requirements	2
3	Use Cases	3
3.1	User Types	3
3.2	User Scenarios	3
3.3	Use Case Diagram	5
3.4	Use Cases	5
3.4.1	Register for User	5
3.4.2	Register for Admin	6
3.4.3	Login for User	7
3.4.4	Login for Admin	7
3.4.5	Profile Page	7
3.4.6	Creating Recipe	8
3.4.7	Editing Recipe	8
3.4.8	Recipe Display	9
3.4.9	Recipe Reaction	10
3.4.10	Shopping List Viewing and Editing	10
3.4.11	Storage Update	11
3.4.12	Recipe Searching	11
3.4.13	Recipe Reporting for User	12
3.4.14	Recipe Reporting for Admin	13
3.4.15	Recipe Delete for User	13
3.4.16	Recipe Delete for Admin	14
3.4.17	Ingredient Add	14
3.4.18	Ingredient Delete	14
3.4.19	Sending Ingredient Adding Request	15
3.4.20	Checking Ingredient Adding Requests	15
3.4.21	Favorite Recipe Addition	16
3.4.22	Recipe Suggestion	16
3.4.23	Creating Recipe List	17
3.4.24	Recipe Addition to Recipe List	17
3.4.25	Recipe Removal from Recipe List	18
3.4.26	Adding Ingredient to User Fridge	18
3.4.27	Updating Ingredient from User Fridge	19

1 Introduction

This document includes system functional requirements and its use cases, use case scenarios, use case diagram of our app, BastırChef.

This app is a vibrant platform connecting chefs and home cooks passionate about food. Our app enables chefs to share their favorite recipes, while users can effortlessly search for dishes based on their available ingredients. Users can create custom food lists, easily share them with friends, and discover new recipes tailored to their tastes. The app also offers intelligent recommendations, suggesting fresh recipes based on users' previous likes and preferences.

2 System Requirements

2.1 Functional Requirements

Requirement ID	Task ID (project plan)	Description
1	FR-2	Users should be able to register for an account and log in to access for recipes.
2	FR-3	Users must be able to browse through a list of recipes.
3	FR-3.1	There should be a search feature allowing users to find recipes based on ingredients.
4	FR-8.1	Clicking on a recipe should display its details, including ingredients, preparation steps, and reactions.
5	FR-7	Users must be able to like, comment on recipes or add them to their favorites.
6	FR-4	The app should recommend recipes based on the user's past interactions and preferences.
7		User can add or delete ingredient to his/her storage (storage means like fridge,my ingredients etc.) This storage is used for search and shopping list.
8	FR-6	App should be able to add ingredients from a recipe to a digital shopping list of a user.
9	FR-6	User can delete an ingredient from shopping list and add it to his/her storage if he/she bought it.
10	FR-8	Users can submit their own recipes to the app for others to view.
11	FR-8.1	User can edit their recipes.
12	FR-5	User can make recipe playlists.
13		User can report a recipe to admin.
14	FR-1	Admin can see reports and can delete recipes from Admin Panel.
15		User can create a request to admin to add an ingredient.
16	FR-1	Admin can check ingredient adding requests and can add ingredients to the system.
17	FR-1	Admin can delete ingredients from the system.

2.2 Non-functional Requirements

- **Usability:** The user interface ought to be straightforward and the user's interaction with the system should be pleasant.
- **Security:** User credential security will be protected.
- **Scalability:** The system should scale for several hundreds of users using it during busy course registration times.
- **Performance:** Application pages should open in a short acceptable time.

- **Modularity:** The program structure will be modular with different tasks bound to different services and modules.
- **Reliability:** Modules and framework will be integrated correctly so that both applications will perform as intended in any case.

3 Use Cases

3.1 User Types

- User : Searches for recipes, adds them, reviews them
- Admin: Controls recipes and ingredients.

3.2 User Scenarios

- **Creating a Recipe:** Walter, the Chef, wants to share his recipe with his friends. Walter launches the app and navigates to main menu. Then, he clicks the add button and fills the data fields. After that, he clicks the Share button and the recipe is shared.
- **Editing a Recipe** Walter added the recipe before. However, Walter forgets a small but important detail. So, he clicks on the recipe and clicks on Edit button to edit the recipe he shared. Then, recipe creation page is displayed again and Walter fills the data fields again. After that, he clicks the Submit button and the recipe is edited.
- **Search Recipe with Ingredients:** Jesse, the Cook, wants to cook a recipe with the ingredients he has. He launches the app and navigates to search page. Then, he selects the ingredients he has and app lists the possible recipes to cook with selected ingredients. Jesse selects a recipe and clicks on it to see details. If he likes the recipe, he can add it to his favorites and send reactions under the recipe.
- **Admin Deletes Recipe:** Gus, the Admin, wants to delete some recipes which misguide users. He opens the Admin Panel and navigates to recipes page. He selects the false recipe and clicks the delete button. System displays a pop up to confirm the operation. Gus clicks the delete button in the pop up and recipe is deleted.
- **Recipe Suggestion:** Mike, the grandpa, wants to cook a special dish for his grandchildren. He opens the app and navigates to main menu. Main menu displays the recipe suggestions according to Mike's favorite recipes. Mike selects any recipe that he wants to cook and clicks on it. Then, recipe information is displayed.
- **Adding to Shopping List:** Mike, the grandpa, selected a recipe and clicks to start recipe button. However, app warns Mike since Mike lacks some ingredients for this recipe. So, Mike clicks on Add to List button and missing ingredients are added to Shopping List.
- **Favorites List:** Skyler, the wife, wants to cook her favorite dish to her husband. So, she launches the app and navigates to Profile page. Then, she clicks on Favorites button and the app lists her favorite dishes. She selects the dish she likes the most and clicks on it to see recipe description.
- **Adding recipes to recipe lists:** Saul, the food enthusiast, wants to discover different recipes. So, he launches the app and scrolls the recipe flow in the main page. He finds a very easy and delicious recipe in the flow, so he adds the recipe to his food list named "Quick Dishes". After scrolling a bit more, he encounters a special dish for a romantic date. So he adds that dish to another food list which he named "Very Special Dishes".
- **Viewing a recipe list:** Saul, the lover, wants to cook a very special dish for a romantic night with his girlfriend. So, he launches the app and navigates to Profile page. Then, he clicks on the Food Lists button to see his food lists. After that, he selects the food list named "Very Special Dishes" which he stores the dishes that he would like to cook for his girlfriend. He selects a dish from that list and recipe description page is opened.

- **Recipe Reporting:** Gale, inexperienced chef, puts a recipe on the system where he forgets the cooking instructions for raw chicken. Walter, the Chef, notices this recipe and clicks on the recipe to see the details. Because of the absence of the cooking instructions for raw chicken, Walter has a concern that this recipe can give food poisoning to other cooks if it is made. Thus, Walter clicks the report button to report the recipe. Gus, the Admin, logs into his admin account from the admin panel. He clicks the reports tab and sees the report of the Gale's recipe. Gus looks at the details of the recipe by clicking on the report. He sees what Walter saw about the recipe. He backs to the reports page and clicks the delete button next to the report and recipe is deleted.
- **Ingredient Request:** Lewis, exotic lover, knows different types of recipes and wants to share his experience with the community. Lewis writes the instructions for the recipe and adds the necessary ingredients and amounts for ingredients however Lewis can not find Kimchi in the ingredients. Lewis creates a request to add Kimchi to ingredients of the app. Admins views the request and see the need for adding ingredient to the database. Admins approves request and Lewis can happily add Kimchi to ingredients of the recipe.
- **Write Comment and Give Like to a Recipe:** Lewis, exotic lover, tries a different recipe that he saw today and enjoys it very much. Lewis wants to share his delightful experience with the community. Lewis clicks on recipe and comments how he felt about the recipe. Also he gives a like to recipe. Other users sees how Lewis felt after enjoying the dinner and they also tried the recipe.
- **User Register:** Mary, who downloaded the app, wants to create an account for this app. She enters username,password and email. She clicks to register and if username and email is not taken by another user, she registered to app.
- **User Login:** Mary, who registered to the app, wants to enter her account. She clicks to login button and enters her username and password. If she enters true, she launches the main menu.

3.3 Use Case Diagram

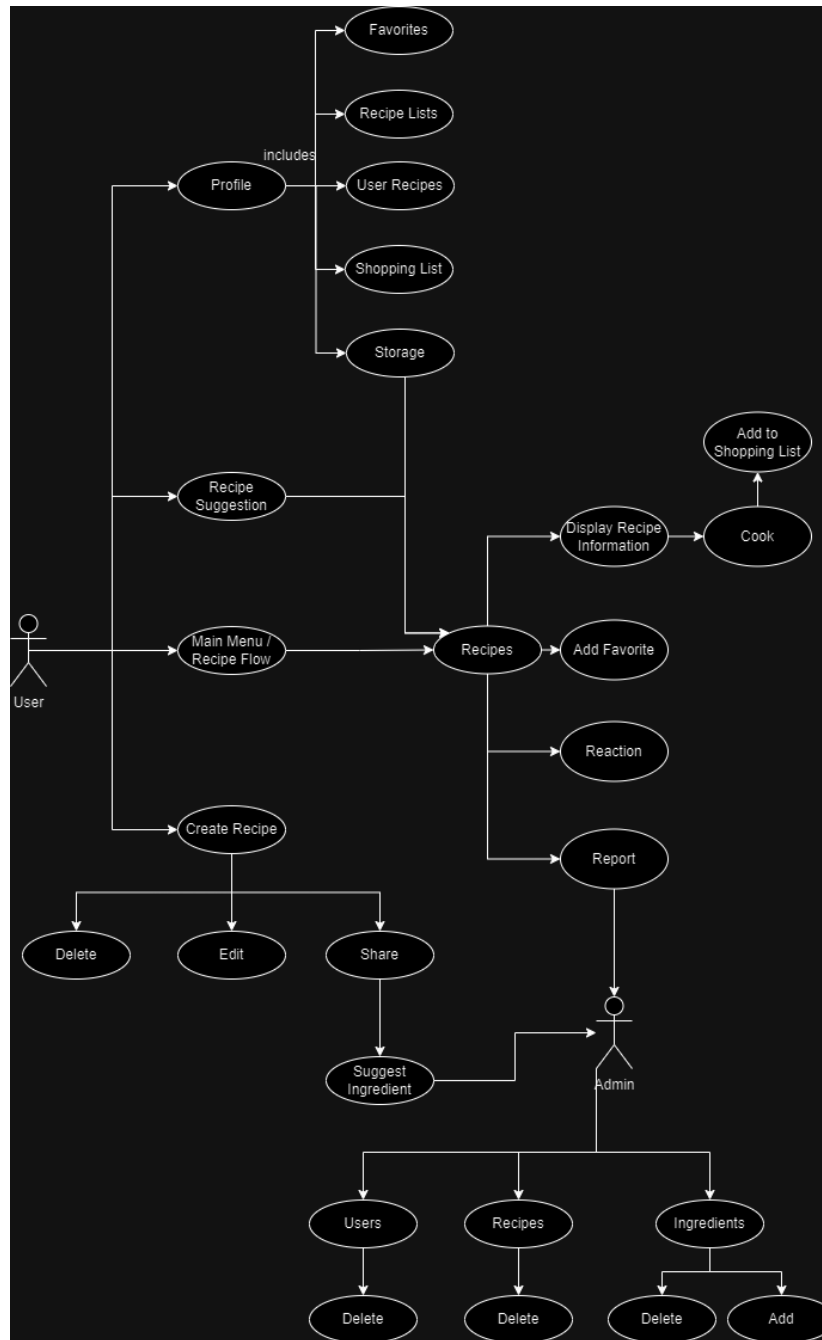


Figure 1: Use Case diagram for the whole system

3.4 Use Cases

3.4.1 Register for User

- **Purpose:** To create an account for BastırChef application (in mobile)
- **Actors:** User
- **Pre-conditions:** User shouldn't have any existing account.

- **Post-conditions:** User fills all the input fields. System saves the account with user's informations.
- **Main Flow:**

Actor Actions	System Actions
1) User opens the app	2) System displays login page
3) User selects register button	4) System displays register page
	5) System displays input fields for username, password and email.
6) User enters his/her username, password and email	
7) User clicks the register button.	8) System stores user's data and registers the user account.

- **Alternative Flows:**
 - **Step 6:** Username already taken by another user. Do not accept this username and give "username already taken" warning. Go to step 5.
 - **Step 6:** Invalid password. Do not accept this password until it is valid. Invalidity type can be short password or weak password. Give warning according to invalidity type. Go to step 5.
 - **Step 6:** User enters previously used email address. Do not accept this email and do not allow to continue. Go to step 5..
 - **Step 7:** User tries to register with empty information. Give "Please enter username/password/email." warning and do not allow to register.

3.4.2 Register for Admin

- **Purpose:** To create an Admin account for BastirChef application (in web)
- **Actors:** Admin
- **Pre-conditions:** Admin shouldn't have any existing account.
- **Post-conditions:** Admin fills all the input fields. System saves the account with admin's informations.
- **Main Flow:**

Actor Actions	System Actions
1) Admin opens the Admin Panel	2) System displays login page
3) Admin selects register button	4) System displays register page
	5) System displays input fields for username, password and email.
6) Admin enters his/her username, password and email	
7) Admin clicks the register button.	8) System stores admin's data and registers the admin account.

- **Alternative Flows:**
 - **Step 6:** Username already taken by another user. Do not accept this username and give "username already taken" warning. Go to step 5.
 - **Step 6:** Invalid password. Do not accept this password until it is valid. Invalidity type can be short password or weak password. Give warning according to invalidity type. Go to step 5.
 - **Step 6:** Admin enters previously used email address. Do not accept this email and do not allow to continue. Go to step 5.
 - **Step 7:** Admin tries to register with empty information. Give "Please enter username/password/email." warning and do not allow to register.

3.4.3 Login for User

- **Purpose:** To enter an account
- **Actors:** User
- **Pre-conditions:** User should have an existing account registered to the system.
- **Post-conditions:**
 - User successfully logged in if entered values are correct.
 - User could not log in and the system displayed an error message to inform user if entered values are false.
- **Main Flow:**

Actor Actions	System Actions
1) User opens the app	2) System displays login page
3) User enters his/her username and password	4) System processes user request.

- **Alternative Flows:**
 - **Step 2 & 3:** User clicks to register. Go to the register page.
 - **Step 3:** User enters invalid username or password. Give an "incorrect username or password" warning. Clear the fields of username and password entered on the login page.

3.4.4 Login for Admin

- **Purpose:** To enter an account
- **Actors:** Admin
- **Pre-conditions:** Admin should have an existing account registered to the system.
- **Post-conditions:**
 - Admin successfully logged in if entered values are correct.
 - Admin could not log in and the system displayed an error message to inform user if entered values are false.
- **Main Flow:**

Actor Actions	System Actions
1) Admin opens the website	2) System displays login page
3) Admin enters his/her username and password	4) System processes user request.

- **Alternative Flows:**
 - **Step 3:** Admin enters invalid username or password. Give an "incorrect username or password" warning. Clear the fields of username and password entered on the login page.

3.4.5 Profile Page

- **Purpose:** Editing and updating user information.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. User is logged in to their account.
- **Post-conditions:** The user successfully registers and gains access to apps features.

- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the Home Page
5) User selects the “edit profile” option	6) Edit page is displayed
7) User changes the information	8) System checks if the fields that are changed are valid.
9) User selects ”Save” option	10) Information is updated and a success message is shown.

- **Alternative Flows:**

- **Step 2 :** If the provided information is false, an error message that prompts the user to correct the information is shown.
- **Step 4:** An error message is shown if the new information is invalid or incomplete.
- **Step 5:** If the user didn’t change any information, It redirects to the Homepage.
- **Step 5:** If there is a system error during profile updating, the system informs the user and suggests trying again later.

3.4.6 Creating Recipe

- **Purpose:** Creating a new recipe
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user’s device. The user is logged in to their account
- **Post-conditions:** The user successfully registers and gains access to app features.
- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the Home Page
5) User selects the “create a recipe” option	6) System displays recipe creating page
7) User provides the information (name, portions, ingredients)	8) System checks if the information is valid
9) User selects ”Save” option	10) New recipe is created and user is directed to recipe information page

- **Alternative Flows:**

- **Step 2 :** If the provided information is false, an error message that prompts the user to correct the information is shown.
- **Step 4:** An error message is shown if the new information is invalid or incomplete.
- **Step 5:** If there is a system error during recipe creation, the system informs the user and suggests trying again later.

3.4.7 Editing Recipe

- **Purpose:** Editing a recipe
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user’s device. The user is logged in to their account

- **Post-conditions:** The user successfully registers and gains access to app features. Users can access the information of the recipe added by themselves.

- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the Home Page
5) User selects the “recipe list” option	6) System displays the list of recipes that are created by that user
7) User selects the “edit recipe” option	8) System displays the recipe editing page
9) User changes the previous information	10) System checks if the information is valid
11) User selects the ”Save” option	12) Recipe information is updated and a success message is shown.

- **Alternative Flows:**

- **Step 2 :** If the provided information is false, an error message that prompts the user to correct the information is shown.
- **Step 4:** An error message is shown if the new information is invalid or incomplete.
- **Step 5:** If the user didn’t change any information, It redirects to the Recipe list page.
- **Step 5:** If there is a system error during recipe creation, the system informs the user and suggests trying again later.

3.4.8 Recipe Display

- **Purpose:** To view the full recipe

- **Actors:** User

- **Pre-conditions:**

- User already has an account and is logged in.
- User should be in a recipe listing page (like recipe search, recipe suggestion or favorite recipes).

- **Post-conditions:**

- Recipe details (ingredients, instructions, etc.) are displayed correctly to the user.
- User interaction enabled. Users are able to interact with the recipe, such as liking, commenting, scrolling through instructions, adding to favorites.

- **Main Flow:**

Actor Actions	System Actions
1) User clicks to the recipe	2) System routes to display recipe page
	3) System displays recipe title, recipe description, recipe ingredients.
4) User view the recipe.	
5) User can make recipe reaction (like, comment, favorite).	6) System does the Recipe Reaction Use Case.

- **Alternative Flows:**

- **In every step:** User exits viewing the recipe. Return to previous page.

3.4.9 Recipe Reaction

- **Purpose:** To give like, write comment or add to favorites
- **Actors:** User
- **Pre-conditions:**
 - User already has an account and is logged in.
 - User should be in a recipe's display page.
- **Post-conditions:** The user's liking or commenting action was added to the recipe post and printed. If the user added it to favorites, the recipe was added to the user's favorite recipes list.
- **Main Flow:**

Actor Actions	System Actions
1) User opens the recipe	2) System displays recipe page
3) User clicks to "like" button	4) System increases the number of likes of the recipe post by 1.
5) User opens the comment section	6) System displays comments.
7) User writes a comment and sends it.	8) System adds this comment to recipe.

- **Alternative Flows:**
 - **Step 3:** User clicks to "already" clicked like button. Decrease the number of likes of the recipe post by 1.

3.4.10 Shopping List Viewing and Editing

- **Purpose:** To allow the users to see which ingredients they are lacking in order to complete/make their selected/favorite recipes and edit these ingredients.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** The user successfully generated shopping list based on their selected recipes.
- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
5) User clicks to the shopping list page button.	6) System directs the user to the shopping list page. Shopping list is displayed which contains the cumulative required ingredients that user does not have for the selected recipe(s).
7) User reviews the generated shopping list and clicks the edit button.	8) System changes the shopping list from viewing to editing mode where user can click delete buttons next to the existing ingredients to delete or user can add ingredients with the add button.

- **Alternative Flows:**
 - **Step 3:** If the provided information is false, an error message that prompts the user to correct the information is shown.

- **Step 5:** If there is no ingredient listed in the shopping list because user did not selected any recipe, display a message that there is no recipe selected. User is redirected to the recommendation page.
- **Step 7:** If there is a system error during shopping list editing, the system informs the user and suggests trying again later.

3.4.11 Storage Update

- **Purpose:** To keep the storage information updated
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** Shopping list is kept well.
- **Main Flow:**

Actor Actions	System Actions
1) User chooses the "Shopping List" option	2) System redirects the user to the Shopping List Page
3) User selects the delete option on an item	4) System Shows a pop-up to ask if the item is bought
5) User selects the "Bought" option	6) System asks the amount
7) User enters the amount	8) The specific ingredient on the storage is updated by adding the amount to the current stock and the item is deleted from the shopping list
	9) System redirects the user to the Shopping List Page

- **Alternative Flows:**
 - **Step 2:** If user can not get redirected to the page, a pop-up is shown.
 - **Step 5:** User may select the not bought option. Then the item is deleted without making any changes in the storage.
 - **Step 7:** User may enter an invalid amount. An error pop-up is shown.
 - **Step 8:** If the storage can not be updated, user is informed and the item is not deleted from the shopping list.

3.4.12 Recipe Searching

- **Purpose:** To allow the users to search suitable recipes according to the ingredients that he/she in his/her disposal.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** The user successfully searched and find a recipe according to the ingredients that he/she in his/her disposal.
- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
5) User clicks to search bar.	6) System directs the user to the ingredients page where user can input/select the ingredients that he/she has.
7) User selects ingredients from the pre-determined ingredient list and clicks the search button. User can also filter and sort the ingredients.	8) System evaluates the inputted ingredients by using the search algorithm. After the evaluation user is directed to the recipes page. Recipe page is displayed and shows the appropriate recipes that can be made with those ingredients.

- **Alternative Flows:**

- **Step 3:** If the provided information is false, an error message that prompts the user to correct the information is shown.
- **Step 5:** If there is a system error during searching, the system informs the user and suggests trying again later.
- **Step 7:** If the user does not select any ingredient and click the search button, display a message that there is no ingredient selected and give two options in form of buttons: continuing the ingredient selection process or going back to the home page.

3.4.13 Recipe Reporting for User

- **Purpose:** To allow the users to report unwanted, spam, abusive and inappropriate recipes in order to be deleted from the system.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** The user successfully reported the unwanted, spam, abusive and inappropriate recipes in order to be deleted from the system.
- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
5) User clicks on a recipe (This recipe can be obtained from "Search Recipe" or "Recipe Suggestion" use case flows).	6) System directs the user to the details page of the inappropriate recipe where user can also see a button for reporting.
7) User clicks the report button.	8) System send the admin a report request in order to remove the inappropriate recipe.

- **Alternative Flows:**

- **Step 3:** If the provided information is false, an error message that prompts the user to correct the information is shown.
- **Step 7:** If there is a system error during reporting, the system informs the user and suggests trying again later.

3.4.14 Recipe Reporting for Admin

- **Purpose:** To allow the admins to delete the unwanted, spam, abusive and inappropriate recipes that users are reported from the database.
- **Actors:** Admin
- **Pre-conditions:** BastırChef is opened from the web browser. The admin is logged in to their account from the admin panel. Database and server is in a ready state to have user processes.
- **Post-conditions:** The admin successfully deleted the unwanted, spam, abusive and inappropriate recipes from the database.
- **Main Flow:**

Actor Actions	System Actions
1) Admin signs in to the Admin Panel	2) Admin panel displays the main page
3) Admin clicks the reports tab/button.	4) Admin panel displays the reported recipes. Admin can see how many report requests from the users are sent for a particular recipe. Admin also can see the reported recipe's details by clicking on the report.
5) Admin clicks the remove button or reject button next to the recipe.	6) System deletes the reported recipe from the database if the admin clicks the remove button. If admin clicks the reject button recipe is not deleted and remains in the database.

- **Alternative Flows:**
 - **Step 1:** If the provided information is false, an error message that prompts the admin to correct the information is shown.
 - **Step 3:** If there are no reported recipes by the users when admin clicks the reports tab, instead of showing an empty page, system will tell the admin that there is no reported recipes currently by a message on the screen.
 - **Step 5:** If there is a system error during removing a reported recipe, the system informs the user and suggests trying again later.

3.4.15 Recipe Delete for User

- **Purpose:** To allow the users to delete their own recipes.
- **Actors:** User
- **Pre-conditions:** User should have been registered and signed in. User should have at least 1 recipe to be deleted.
- **Post-conditions:** User successfully deleted the selected recipe from the system.
- **Main Flow:**

Actor Actions	System Actions
1) User signs in to the app	2) System displays the main page.
3) User navigates to Profile	4) System displays the profile page
5) User clicks the My Recipes button	6) System displays the recipes of the user
7) User clicks Delete button of any recipe to be deleted	8) System displays a popup to confirm the action
9) User clicks Delete button in the popup	10) System deletes the recipe

- **Alternative Flows:**
 - **Step 9:** If the user closes the pop-up or clicks on Cancel button in the pop-up, system does not delete the selected recipe.

3.4.16 Recipe Delete for Admin

- **Purpose:** To allow the admin to delete misinformative or spammed recipes.
- **Actors:** Admin
- **Pre-conditions:** Admin should have been registered and signed in to the Admin Panel.
- **Post-conditions:** Admin successfully deleted the selected recipe from the system.
- **Main Flow:**

Actor Actions	System Actions
1) Admin signs in to the Admin Panel	2) Admin panel displays the main page.
3) Admin clicks the Recipes button.	4) Admin panel displays the Recipes page and lists the recipes
5) Admin clicks the Delete button of any recipe row	6) System displays a pop-up to confirm delete operation
7) Admin clicks Delete button in the pop-up	8) System deletes the selected recipe

- **Alternative Flows:**
 - **Step 7:** If the user closes the pop-up or clicks on Cancel button in the pop-up, system does not delete the selected recipe.

3.4.17 Ingredient Add

- **Purpose:** To allow the admin to add new ingredients.
- **Actors:** Admin
- **Pre-conditions:** Admin should have been registered and signed in to the Admin Panel.
- **Post-conditions:** Admin successfully added new ingredients to the system.
- **Main Flow:**

Actor Actions	System Actions
1) Admin signs in to the Admin Panel	2) Admin panel displays the main page.
3) Admin clicks the Ingredients button.	4) Admin panel displays the Ingredients page and lists the ingredients
5) User clicks the Add button	6) System displays a pop-up with data fields
7) User fills the fields and clicks Add button	8) System adds the new ingredient with given data fields

- **Alternative Flows:**
 - **Step 7:** If the user does not fill all the fields, closes the pop-up or clicks on Cancel button in the pop-up, system does not add the new ingredient.

3.4.18 Ingredient Delete

- **Purpose:** To allow the admin to delete ingredients.
- **Actors:** Admin
- **Pre-conditions:** Admin should have been registered and signed in to the Admin Panel.
- **Post-conditions:** Admin successfully deleted the selected ingredients from the system.

- **Main Flow:**

Actor Actions	System Actions
1) Admin signs in to the Admin Panel	2) Admin panel displays the main page.
3) Admin clicks the Ingredients button.	4) Admin panel displays the Ingredients page and lists the ingredients
5) User clicks the Delete button of any ingredient	6) System displays a pop-up to confirm delete operation
7) User clicks Delete button in the pop-up	8) System deletes the selected ingredient

- **Alternative Flows:**

- **Step 7:** If the user closes the pop-up or clicks on Cancel button in the pop-up, system does not delete the selected ingredient.

3.4.19 Sending Ingredient Adding Request

- **Purpose:** To create requests by users to add ingredients that are not available as options in the application
- **Actors:** User
- **Pre-conditions:** The user is logged in to their account. User searched an ingredient to add to storage or recipe but there is no ingredient such that.
- **Post-conditions:** Request is sent to admin.
- **Main Flow:**

Actor Actions	System Actions
1) User enters to app. He searched for an ingredient (to search for a recipe or to add his storage) but there is no such ingredient.	3) If result is empty, system shows a request option.
3) "Create a request" option appears. User selects this option.	4) System shows fields to fill.
5) User fills information for missing ingredient.	
6) User sends the request by clicking "Send Request" button.	7) System gets the request and adds it to Admin Panel's requests part.

- **Alternative Flows:**

- **Step 5:** User leaves some information empty. Give "please fill empty fields." warning and wait for Step 6 to check this alternative flow again.

3.4.20 Checking Ingredient Adding Requests

- **Purpose:** To see request by users in admin panel for adding an ingredient as option to application.
- **Actors:** Admin
- **Pre-conditions:** Admin should have been registered and signed in to the Admin Panel.
- **Post-conditions:** The request is accepted and the material is added to the application, or the request is rejected by admin.

- **Main Flow:**

Actor Actions	System Actions
1) Admin enters to "Requests" page in Admin Panel.	2) System goes to the requests page and lists currently not answered requests.
3) Admin sees requests with two buttons: "apply" and "decline"	
4) Admin agree with a request and clicks to apply button.	5) System add this ingredient to ingredients datas and deletes this request.

- **Alternative Flows:**

- **Step 4:** Admin don't want to add this ingredient to application and clicks decline. Delete this request from requests list and do not add it to ingredients.

3.4.21 Favorite Recipe Addition

- **Purpose:** To allow users to store their favorite recipes to look after or share with other users.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** The user have successfully added recipe to its favorite list.
- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
5) User clicks to actions section of recipe.	6) System shows actions to user which includes add to favorites
7) User clicks the add to favorites.	8) System closes the actions section of recipe adds recipe to favorites and shows a pop up that says recipe is added successfully.

- **Alternative Flows:**

- **Step 7:** If an error occurs during the addition such as db crash a pop up shown that tells recipe is not added successfully and user have to try again.
- **Step 7:** If item is already on the favorites action tab will say remove from favorites. When this button is clicked system will remove recipe from favorites and inform the user

3.4.22 Recipe Suggestion

- **Purpose:** To show users recipes that are more likely to be liked and made.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** Probability of user liking the recipe is went up

- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
	5) System requests recipes to be displayed
	6) Algorithm chooses recipes probably will get more like to be displayed

- **Alternative Flows:**

- **Step 6:** If algorithm does not find any recipes according to user's favorites, an initial recipe list will be displayed.

3.4.23 Creating Recipe List

- **Purpose:** To allow users to create lists in order to group recipes that forms a concept and share their lists with other users.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** A Recipe List is created which in recipes can be added or deleted that can be shared among users.
- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
5) User clicks on Profile	6) System shows tabs that include Recipe Lists tab
6) User clicks on Recipe Lists	7) System shows current recipe lists and a button to add a recipe list
8) User clicks on Add Recipe List button	9) A form will open and take name of the list
	10) System will create a recipe list belonging to the user
	11) System will show a pop up that says recipe list created successfully

- **Alternative Flows:**

- **Step 8:** If an error occurs during the creation of recipe list system will pop up an error message that informs the user
- **Step 8:** If there is already a list with the same name system will warn the user and will not take further actions.

3.4.24 Recipe Addition to Recipe List

- **Purpose:** To allow users to create recipes to share and interact with community.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.
- **Post-conditions:** Recipe successfully added to the chosen recipe list and will be displayed inside of recipe list

- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
5) User clicks to actions section of recipe.	6) System shows actions to user which includes add to recipe list
7) User clicks the add to recipe list.	8) System shows a list of recipe lists
9) User chooses a recipe list	10) System adds recipe to recipe list
	11) System shows a pop up that tells it is added successfully

- **Alternative Flows:**

- **Step 10:** If an error occurred pop up says that try again an error occurred
- **Step 10:** If recipe is already in recipe list system notifies the user and does no further action

3.4.25 Recipe Removal from Recipe List

- **Purpose:** To allow users to manage their recipe lists by providing remove feature
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes. A recipe is added to the Recipe List before.
- **Post-conditions:** Recipe is successfully removed from recipe list and no longer will be displayed
- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the home page
5) User clicks to recipe list	6) System shows recipe lists
7) User clicks remove on a recipe list	8) System shows pop up that says are you sure?
9) User clicks to yes button	10) System deletes the recipe from recipe list
	11) System shows a pop up that tells it is deleted successfully

- **Alternative Flows:**

- **Step 10:** If recipe is not deleted successfully because of db error system notifies the user.

3.4.26 Adding Ingredient to User Fridge

- **Purpose:** To allow users to add their ingredients to their fridge in the application. To increase suggestion algorithm and provide a better shopping list feature.
- **Actors:** User
- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes. Ingredient is not added before to their fridge
- **Post-conditions:** Users fridge successfully updated and given amount is inserted into fridge

- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the Home Page
5) User selects the "Profile" tab	6) System displays the tabs in profile tab which includes "Fridge"
7) User selects the "Fridge" tab	8) System displays the ingredients in the users "Fridge"
9) User clicks on "Add Ingredient"	10) System shows Ingredient form to user
11) User fills the form and clicks "Add"	12) "Fridge" is updated and a success message is shown.

- **Alternative Flows:**

- **Step 11:** If addition is not successful due to db error system notifies the user.
- **Step 11:** If User tries to add an Ingredient that is not in the system yet a button will appear below the search bar that creates a Ingredient addition request. Admins approves the request if it is appropriate and User can add ingredient after.

3.4.27 Updating Ingredient from User Fridge

- **Purpose:** To allow users to update their ingredients amount in their fridge. To increase suggestion algorithm and provide a better shopping list feature.

- **Actors:** User

- **Pre-conditions:** BastırChef app is installed on the user's device. The user is logged in to their account. Database and server is in a ready state to have user processes.

- **Post-conditions:** The User's fridge is successfully updated and given amount of ingredient is removed from the fridge

- **Main Flow:**

Actor Actions	System Actions
1) User launches the app	2) Login screen is displayed
3) User enters login information	4) System validates the information. User is redirected to the Home Page
5) User selects the "Profile" tab	6) System displays the tabs in profile tab which includes "Fridge"
7) User selects the "Fridge" tab	8) System displays the ingredients in the users "Fridge"
9) User clicks on Ingredient	10) System shows Ingredient details and amount
11) User changes the amount and clicks "Save"	12) "Fridge" is updated and a success message is shown.

- **Alternative Flows:**

- **Step 12:** If deletion is not successful due to db error system notifies the user.
- **Step 12:** If User tries to remove from more than he/she have an error message will be displayed and no further action will be taken.