Bits to Bites Mobile Application

Software Requirements Specification

Version 1.0

Nov 15th, 2020

Nikola Vojinovic
Corradina Dinatale
Noah Huboux
Stefan Grujic

Prepared for
COMP 3059—Capstone Project I
Instructor: Anjana Shah
Fall 2020

Revision History

Date	Description	Author	Comments
Nov 10th, 2020	Version 1.0	Nikola Vojinovic Corradina Dinatale Noah Huboux Stefan Grujic	First Revision

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
Nikola Vojinovic	Nikola Vojinovic	Developer	Nov 15th, 2020
Corradina Dinatale	Corradina Dinatale	Developer	Nov 15th, 2020
Noah Huboux	Noah Huboux	Developer	Nov 15th, 2020
Stefan Grujic	Stefan Grujic	Developer	Nov 15th, 2020

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to delineate the functional and non-functional specifications and requirements of the Bits to Bites Mobile Application. It is intended to serve as a guide to both the stakeholders and the developers of the system by providing a detailed description of the features of our application, its parameters, and goals.

The purpose of The Bits to Bites Mobile Application is to be an easily accessible and cost-effective source of quality recipes based on ingredients already available in the home. This application aims to streamline the process of making meals at home for both experienced and inexperienced home cooks with the goal of reducing household food waste. It aims to accomplish this by providing users with access to a database of user-uploaded recipes as well as tools that will make the process of cooking faster and easier.

1.2 Scope

1.2.1 In Scope

The Bits to Bites Mobile Application will be a recipe search application available for download on any android device and targeted to home cooks of all skill levels. It will be designed with the goal of helping to reduce household food waste and it will accomplish this by providing home cooks with the ability to input, save and keep track of a list of ingredients that they already have available at home. Using these ingredients as a parameter, the user will be able to search a database of recipes and sort them based on time, cuisine type, dietary restrictions, cost, and difficulty level. This sorting functionality aims to make the application as inclusive and as useful to as many people as possible.

The user will also be able to log an ingredient's expiry date when adding it to the pantry and choose whether they would also like to receive an expiry date reminder. Automating the process of tracking ingredient expiry dates will help cooks keep track of their ingredients more easily and avoid having to track dates manually which can often lead to food waste.

The user will also be able to upload recipes and share them with other users. Preformatted recipe templates will be used to provide a uniform review process for the admin as well as keep the content under control. Users will also be able to save their favourite recipes and add helpful notes to them.

The application will include a unit-converter to convert a recipe's ingredient measurements to the user's desired unit. This will allow users to work with the unit types they are most comfortable with. It will also include an ingredient multiplier feature which will multiply ingredient amounts based on user input. This feature will help the busy home cook save time by performing the calculation for them so they can avoid having to do it manually.

The application will also provide the ability to keep track of calories consumed/nutritional information based on recipes chosen by the user. This will help people make sure that they are meeting their daily nutritional requirements. The application will also include a kitchen timer feature as well as an automatic screen lock feature to facilitate the process of following a recipe and avoid having to use multiple tools outside of the application.

1.2.2 Out of Scope

The Bits to Bites Mobile Application will not be compatible across all platforms as it will be a strictly android application. There will be no website version so users will not be able to access the application through a browser. It will not include a built-in shopping list feature as this application aims to encourage users to use ingredients they already own. It also will not provide users with videos to guide them through the process of following a recipe. This will help avoid the hassle of managing video content and will help keep costs low. Users will not be able to navigate through the app with voice commands and will not be able to scan ingredients using the device's camera.

2. System Overview

2.1 Project Perspective

The Bits to Bites System will be a new self-contained system as the Bits to Bites company does not currently have any other systems to build off of. The system will be offering a recipe generating and uploading application for users. The design of the system will be catered to mobile devices, specifically any Android devices. The system origin will be constructed by the developer team and the team will use agile methodologies and practices to design the most versatile, robust system. The team of developers, consisting of Noah Huboux, Stefan Grujic, Corradina Dinatale and Nikola Vojinovic created the concept for the system proposition.

2.2 System Context

The business case for the Bits to Bites Application outlined that the needs fulfilled by the Bits to Bites Application were to address growing food waste concerns, to offer inexperienced cooking enthusiasts a helpful system which encourages more home cooking and give expert cooking enthusiasts ways to share creative and unique recipes with others.

The software that will be implemented to address these specific issues will be as follows. Firstly, the implementation of an expiration date tracker for ingredients entered by the user will address the concerns of food waste; giving users the ability to track their ingredients and make the most effective use out of all their ingredients, before they expire. Secondly, multiple features will address the concerns of the more inexperienced user base when it comes to cooking; one feature that will directly address their concern is the implementation of difficulty tracking level for all recipes created, as well as time to make. This feature will allow inexperienced users to choose the recipes that are easy and fast to create. The unit converter feature, that will convert recipe measurements to

preferred unit of measurement, will indirectly aid the inexperienced users, giving them measurements they are familiar with. The ingredient multiply feature will allow users to change the measurements for a given recipe based on the amount of people they are looking to make the meal for, this will also indirectly help inexperienced users as they will be able to dynamically change the recipe to best suit their current needs.
Finally, the feature that will allow users to upload their own recipes to a database for all other users to see; will allow experienced users to share their recipes. This will also address the strategic issue that is the lack of extensive data on recipes, making users upload their own recipes, will allow our database collection of recipes to grow at an exponential rate, while not incurring any additional up-front costs.

2.3 General Constraints

The general constraints for the Bits to Bites application are few and far in between. It is not anticipated that they will be difficult to resolve with sufficient planning and oversight. The constraints have not changed since the last sprint; the application design, testing and implementation, must be completed at no cost to any parties associated with the project. The final constraint is the application's deadline; the entire application must be in working condition according to the High Level Requirements by April 1st, 2021.

2.4 Assumptions and Dependencies

The assumptions and dependencies for the Bits to Bites application are:

- There are enough original recipes available to create an extensive base that will meet the needs of the customer base. It will also create the base by which users can add additional recipes and it is assumed that they will do so.

- The customer base is assumed to be similar to the project team; and thus no market research is necessary.

- There are no delays; each team member will adhere to the project and communication plans.

- Team members all have the required skill and ability to execute their portion of the plan.

- The funding for the project is limited and it is assumed that the team will work within those constraints.

- The team also assumes that the results of the sprint are successful and bug fixes are identified and fixed in a timely manner before the next sprint.

- Technically, the project assumes that mobile and android platform technology will remain stable and will not fundamentally change in the foreseeable future. The application will not be iOS compatible.

- The team assumes that the backend will be a database that is scalable. It will contain the both the individual ingredients and recipes that will use a variety of those ingredients.

• We assume that ingredients are valid and available in the North American marketplace.

• There is an assumption that there is no validation of ingredients (i.e spelling, actual food)

3. Functional Requirements

3.1.1 Add ingredients to Account

Use Case Name	Add ingredients to Account
Trigger	The Registered User selects to Add ingredient button
Precondition	The Registered User has accessed the Bits to bites main page
Basic Path	1. The system presents a choice of ingredients to Add. 2. The user chooses what ingredients to add 3. The system prompts the user to add an Expiry Date 4. The user confirms 5. The system prompts the number of ingredients to Add. 6. The user chooses the number of each ingredient to add 7. The system prompts confirmation to add ingredient 8. The user confirms and submits
Alternative Paths	
Postcondition	The ingredient/s are added to the account database
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.2 Edit stored ingredients

Use Case Name	Edit stored ingredients
Trigger	The Registered User choose the edit/delete ingredients button.
Precondition	The Registered User has accessed the View Ingredient Page
Basic Path	1. The Registered User selects an ingredient to edit. 2. The system presents a page with information about the selected ingredient 3. The Registered User edits the information and submits 4. The system verifies the information
Alternative Paths	1. The Registered User selects an ingredient to delete. 2. The system asks for confirmation. 3. The user approves
Postcondition	The ingredient/s information is updated or the ingredient/s is deleted from the database
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.3 View my stored ingredients

Use Case Name	View my stored ingredients

Trigger	The Registered User choose the View my stored ingredients button
Precondition	The Registered User has already Logged into an Account and has stored ingredients
Basic Path	• The system presents a list of all the stored ingredient on there account. • The system presents sorting options to the Registered User. • The user chooses a sorting options
Alternative Paths	If there are no stored ingredients the Add ingredients to Account use-case is started
Postcondition	The system returns the Registered User to the Bits to bites main page
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.4 Search for Recipes

Use Case Name	Search for Recipes
Trigger	The Registered User choose the Search for Recipe Button
Precondition	The Registered User has accessed the Bits to bites main page
Basic Path	1. The system displays default list of recipes based on the users stored ingredients 2. The system displays a set of categories and sorting fields for the user 3. The user chooses the a sorting method 4. The system displays a list of recipes based on the users choice
Alternative Paths	
Postcondition	The system returns the Registered User to the Bits to bites main page
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.5 Favorite Recipe

Use Case Name	Favorite Recipe
Trigger	The Registered User chooses the Favorite button
Precondition	The Registered User has selected a Recipe
Basic Path	1. The system displays confirmation of the Favorite. 2. The recipe is added to the users Favorite list.
Alternative Paths	1. The User choose an already Favorite Recipe 2. The system prompts the User if they wish to remove the Recipe from there Favorites 3. The User confirms
Postcondition	The Recipe is added or removed from the user favorite list

Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.6 Start Recipe

Use Case Name	Start Recipe
Trigger	The Registered User chooses a Recipe to start cooking
Precondition	The Registered User is on the *Recipe Search* page
Basic Path	1. The system prompts the user to multiply the ingredients used to increases the portions of the recipe 2. The User set the ingredient multiple and continues 3. The system prompts the user to set a kitchen timer 4. The User choose whether to set a kitchen timer and continues. 5. The system prompts the user to enable auto screen lock 6. The User chooses whether to enable auto screen lock 7. The system Page showing step by step instructions for the recipe 8. The User confirms when they are done cooking. 9. The system calls the Remove used ingredients from Account use-case
Alternative Paths	If the user chooses a recipe that was in progress the case starts from what step they where last at. The system returns the Registered User to the Recipe Main Page
Postcondition	The system returns the User to the Recipe Main Page
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.7 Remove used ingredients from Account

Use Case Name	Remove used ingredients from Account
Trigger	The Start Recipe use-case is finished
Precondition	The Start Recipe use-case was started
Basic Path	1. The system gets all the ingredients used in the Start Recipe use-case 2. The system deletes all the used ingredients from the Registered User Account
Alternative Paths	
Postcondition	The used ingredients are Removed from the users stored ingredients
Exception Paths	The attempt may be abandoned at any time.
Other	If an ingredient used in the recipe is not stored on the user account it will be ignored.

3.1.8 Sort Recipes

Use Case Name	Sort Recipes
Trigger	The Registered User chooses a sort option
Precondition	The Registered User has Search for Recipes
Basic Path	1. The User pick a category to sort by 2. The system lists all recipe matching that category
Alternative Paths	
Postcondition	The Recipes of chosen category are listed
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.9 Upload Custom Recipe

Use Case Name	Upload Custom Recipe
Trigger	The Registered User chooses the Upload Recipe Button
Precondition	The User must be logged into there account and not banned from Uploading Recipes
Basic Path	1. The system presents the user a multi-page form 2. The user fill out the form and submits 3. The form is sent to Custom Recipe database to be reviewed by and Admin for approve
Alternative Paths	
Postcondition	The Recipe is upload to the database
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.10 Edit Account info

Use Case Name	Edit Account info
Trigger	The Registered User chooses the Edit Account Button
Precondition	The user has accessed their Account Main Page
Basic Path	1. The system presents a list of all the information the User can edit. 2. The Registered User chooses a field to edit. 3. The User edit the filled/s and submits 4. The system checks that all fields are not blank and verifies the data in all the fields.
Alternative Paths	In step 5 if the field is blank the user is asked to fill them
Postcondition	The User Account info is updated
Exception Paths	• The attempt may be abandoned at any time. • If any of the data entered throws an exception
Other	None

3.1.11 Delete Account

Use Case Name	Delete Account
Trigger	The Registered User chooses the Delete Account Button
Precondition	The User must be logged into there account
Basic Path	1.0 The system prompts the user to enter their password 2.0 The user fills out the password and submits 3.0 The system prompts a final confirmation from the user 4.0 The user confirms and the system deletes their account from the database
Alternative Paths	If in step 3 the user cancels the attempt is abandoned
Postcondition	The users account is delete from the database
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.12 Add note to recipe

Use Case Name	Add note to recipe
Trigger	The Registered User chooses the Add note button
Precondition	The Registered User has already selected a recipe from the Favorite Recipe Page or Recipe Main Page
Basic Path	1. The system presents an empty text filed for the user to write in. 2. The User fills out the text filled and Submits.
Alternative Paths	1. The system presents an existing note 2. The user is given the option to edit the text filed 3. The user submits 4. The system asks for confirmation to overwrite the old note 5. The user accepts
Postcondition	The note is created or updated
Exception Paths	The attempt may be abandoned at any time.
Other	None

3.1.13 Create Account

Use Case Name	Create Account
Trigger	The visitor clicks on “Create new account” link
Precondition	The visitor is on the application login page
Basic Path	• The Visitor fills in registration form with personal details. • The Visitor clicks on “Create account” button. • The system validates form details and if successful, adds visitor to the database as a user. • The system redirects visitor to user dashboard.
Alternative Paths	In step 3, if validation is not successful, the system outputs an error message. Return to step 1.
Postcondition	A new account has been successfully created.
Exception Paths	The Visitor may abandon the account creation attempt at any time.
Other	The registration form fields include the username, email, password and password confirmation.

3.1.14 Log into Account

Use Case Name	Log into Account
Trigger	The visitor accesses the application
Precondition	The visitor is on the application login page
Basic Path	1. The Visitor fills in login form with personal credentials. 2. The Visitor clicks on “Login” button. 3. The system validates credentials and if successful, redirects visitor to user dashboard.
Alternative Paths	In step 3, if validation is not successful, the system outputs an error message. Return to step 1.
Postcondition	The Visitor has been logged into their account.
Exception Paths	The Visitor may abandon the login attempt at any time.
Other	The registration form fields include the username or email and password.

3.1.15 Add Recipe to Database

Use Case Name	Add Recipe to Database
Trigger	The admin clicks on “Pending Approval” button
Precondition	The admin is on the “View Recipes” page linked from the main admin dashboard.
Basic Path	1. The system presents the admin with a list of user uploaded recipes from the database, sorted by date. 2. The admin selects a recipe. 3. The admin reviews the recipe. If it satisfies the requirements, they will press the “Add Recipe” button.

	4. The system adds recipe to the database.
Alternative Paths	The "Edit Uploaded Recipe" use case may be invoked.
Postcondition	The Recipe has been added to the database.
Exception Paths	The Admin may abandon the operation at any time. In step 3, if the recipe does not satisfy the requirements, the "Delete Recipe" use case may be invoked.
Other	None

3.1.16 Delete Recipe from Database

Use Case Name	Delete Recipe from Database
Trigger	The admin clicks on "Delete Recipe" button
Precondition	The admin is on the "View Recipes" page linked from the main admin dashboard.
Basic Path	1. The system presents the admin with a list of user uploaded recipes from the database, sorted by date. 2. The admin selects a recipe. 3. The admin reviews the recipe. If it does not satisfy the requirements, the admin clicks on the "Delete Recipe" button. 4. The system deletes the recipe from the database. 5. The system sends an automatic recipe rejection email to the user.
Alternative Paths	None
Postcondition	The recipe has been deleted from the database.
Exception Paths	The Admin may abandon the operation at any time.
Other	The rejection email will include a general message notifying the user that the recipe has been rejected and list all possible reasons for the rejection.

3.1.17 Edit Uploaded Recipe

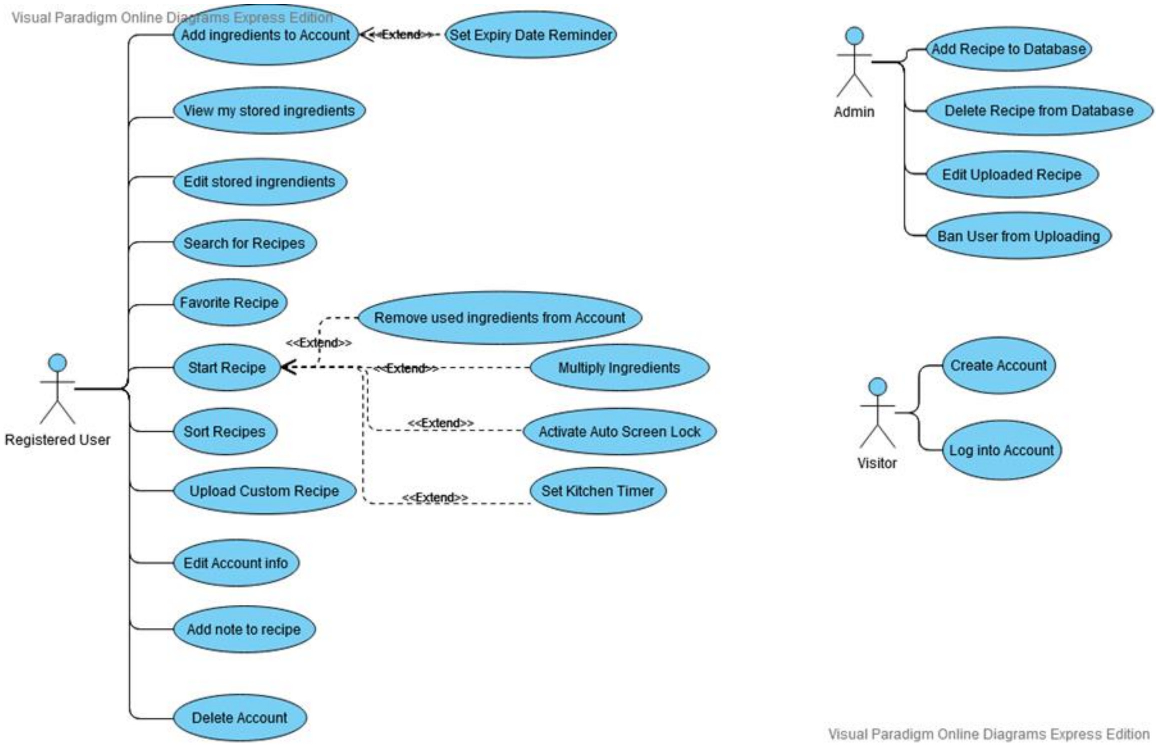
Use Case Name	Edit Uploaded Recipe
Trigger	The admin clicks on "Edit Recipe" button
Precondition	The admin is on the "View Recipes" page linked from the main admin dashboard.
Basic Path	1. The system presents the admin with a list of user uploaded recipes from the database, sorted by date. 2. The admin selects a recipe to update. 3. The admin clicks on "Edit Recipe" 4. The system presents the recipe fields in grid form for modification. 5. The admin makes the necessary changes to the recipe and

	clicks the “Save Recipe” button to submit the form. 6. The system updates the recipe in the database.
Alternative Paths	None
Postcondition	The recipe has been updated in the database.
Exception Paths	The Admin may abandon the operation at any time.
Other	This use case can be used to correct typographical errors, or to enter an updated version of a recipe.

3.1.18 Ban User from Uploading

Use Case Name	Ban User from Uploading
Trigger	The admin clicks on “Ban User” button
Precondition	The admin is on the “Manage Users” page linked from the main admin dashboard.
Basic Path	1. The admin chooses how to search the database for a user. The choices are by username or email. 2. If the search is by username, the admin enters the user’s username in the search box and clicks on the button “Search”. 3. The system returns user information that matches that username from the database. 4. The Admin selects the user and clicks on “Ban User”. 5. The system prompts the admin to confirm banning of user.
Alternative Paths	In step, 2 if the admin chooses to search for a user by email, the admin will enter the user’s email into the search box. 3. The system will return user information that matches that email from the database. Return to step 4.
Postcondition	The user has been banned from uploading recipes.
Exception Paths	The Admin may abandon the operation at any time.
Other	This use case can be used if a user has repeatedly uploaded inappropriate content or recipes that do not meet the requirements.

3.2 Use Cases



3.1.3 Use Case: Add Ingredients to Account

Brief Description
The Registered User enters a new ingredient to their account.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already Logged into an Account.

1. The Registered User selects to Add ingredient.
2. The system presents a choice ingredient to Add.
3. The system presents the number of ingredients to Add.
4. The Registered User confirms their choice
5. The system returns the Registered User to the Bits to bites main page.

3.2.2 Use case: View my stored ingredients

Brief Description
The Registered User views all the ingredient stored on their account.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already Logged into an Account.

5. The Registered User selects to View stored ingredient.
6. The system presents a list of all the stored ingredient on there account.
7. The system presents sorting options to the Registered User.
8. The system returns the Registered User to the Bits to bits main page.

3.2.3 Use case: Edit stored ingredients

Brief Description
The Registered User choose the edit/delete ingredients button.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already Logged into an Account and Viewing the stored ingredient page.

1. The Registered User selects an ingredient to edit/delete.
2. If the Registered User chooses to edit an ingredient the system presents a page with information about the selected ingredient
3. The Registered User edits the information and submits
4. The system verifies the information and returns the Registered User to the View Ingredient Page.

3.2.4 Use case: Search for Recipes

Brief Description
The Registered User searches for Recipes based on their stored ingredients.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already Logged into an Account.

3. The Registered User chooses to search by category, or keyword.
4. The system displays the choices to the Registered User
5. The Registered User selects the desired Recipe.

3.2.5 Use case: Favorite Recipe

Brief Description
The Registered User adds a recipe to their favorite list.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has selected a Recipe.

1. The Registered User chooses to Favorite the Recipe.
2. The system displays confirmation of the Favorite.
3. The system returns the Registered User to the Recipe Main Page.

3.2.6 Use case: Start Recipe

Brief Description
The Registered User chooses a recipe to start cooking.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User is on the Recipe Search page.

1. The Registered User chooses a Recipe to start cooking.
2. The Registered User is given the option to change the units of measurements the recipe uses
3. The system displays step by step instructions for the Recipe.
4. The Registered User is given the option to start a kitchen timer
5. The system returns the Registered User to the Recipe Main Page

3.2.7 Use case: Sort Recipes

Brief Description
The Registered User chooses a sorting method for their search.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User is on the Recipe Search page.

1. The Registered User chooses a sorting category.
2. The system presents a list of Recipes from that category.

3.2.8 Use case: Upload Custom Recipe

Brief Description
The Registered User fills out a forum to upload a new recipe to the database.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already Logged into an Account.

1. The Registered User chooses the Upload Recipe Button
2. The system presents the user a form to fill out
3. The Registered User fills in the information and submits the form.
4. The system verifies the information and Returns the user to Recipe Main Page.

3.2.9 Use case: Edit Account info

Brief Description
The Registered User edit the information on their account.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already Logged into an Account.

1. The Registered User chooses the Edit Account Button.
2. The system presents a list of all the information the User can edit.
3. The Registered User chooses a filled to edit.
4. The system verifies the information and Returns the user to Edit Account Page

3.2.10 Use case: Delete Account

Brief Description
The Registered User Deletes there account.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already Logged into an Account.

1. The Registered User chooses the Delete Account Button
2. The system prompts the user to enter there password
3. The user fill out the password and submits
4. The prompts a final confirmation from the user
5. The user can confirm or cancel. If the user confirms the system deletes the users account from the database and sends the user to the Visitor Main Page

3.2.11 Use case: Add note to recipe

Brief Description
The Registered User adds a note to selected recipe.

Initial Step-By-Step Description
Before this use case can be initiated, the Registered User has already selected a recipe from the Favorite Recipe Page or Recipe Main Page.

1. The Registered User chooses the Add note button.
2. The system presents a text filed for the user to write in.
3. The Registered User fills out the text filled.
4. The Registered User submits the changes.
5. The system Returns the user to Favorite Recipe Page or Recipe Main Page witch ever they started at.

3.2.12 Use case: Create Account

Brief Description
The Visitor User wants to create an account.

Initial Step-By-Step Description
Before this use case can be initiated, the Visiting User has the application open.

1. The user clicks the Login button.
2. They press the Register Account section.
3. The Visiting User fill out the Register Account form
4. They will press create the account button

3.2.13 Use case: Log into Account

Brief Description
The Visiting User wants to enter already made Account.

Initial Step-By-Step Description
Before this use case can be initiated, the Visiting User has the application open.

1. The Visiting User clicks the Login button.
2. They will enter the Login form fields.
3. The Visiting User will press the Login button.

3.2.14 Use case: Add Recipe to Database

Brief Description
An admin wants to add a Recipe to the Database.

Initial Step-By-Step Description
Before this use case can be initiated, the admin must be logged into their account.

1. The admin goes to the recipes page.
2. They will press the Pending Recipe button.
3. The Admin reviews the recipes awaiting acceptance.
4. If it satisfies the requirements, then they will press the Add Recipe button.
5. The system will add the recipe to the database.

3.2.15 Use case: Delete Recipe from Database

Brief Description
An admin wants to delete a recipe from the Database.

Initial Step-By-Step Description
Before this use case can be initiated, the admin must be logged into their account and there is a recipe to delete.

1. The admin goes to the recipes page.
2. The admin will choose a recipe.
3. They will press the Delete Recipe button.
4. The admin will confirm that they wish to delete the recipe.
5. The system will delete the recipe from the database.

3.2.16 Use case: Edit Uploaded Recipe

Brief Description
An admin wants to Edit an uploaded Recipe.

Initial Step-By-Step Description
Before this use case can be initiated, the admin must be logged into their account and there is a recipe to edit.

1. The admin goes to the recipes page.
2. They will choose a recipe.
3. The admin will press the Edit Recipe button.
4. The admin will edit the recipe.
5. They will press the Save Recipe button.
6. The system will edit the recipe in the database.

3.2.17 Use case: Ban User from Uploading

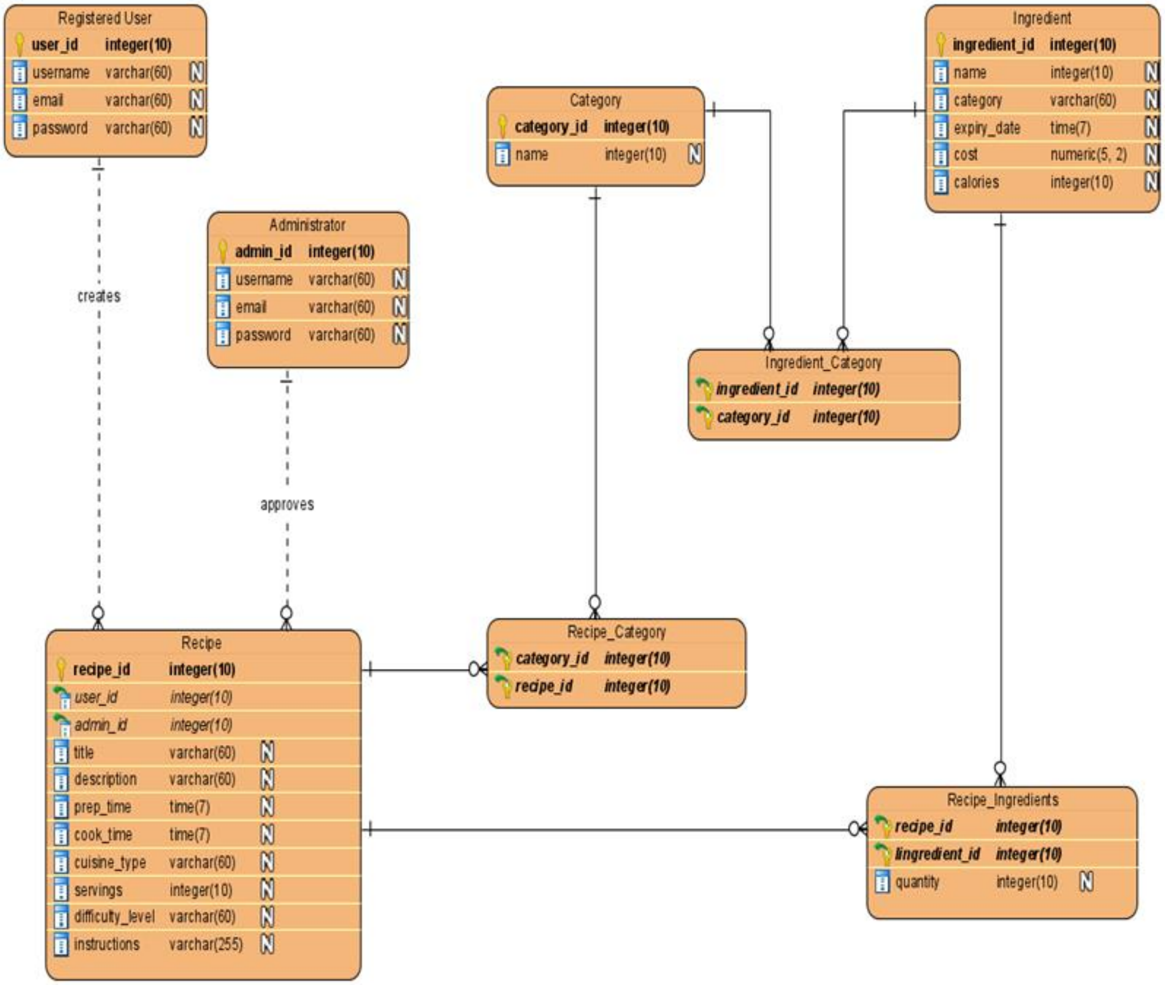
Brief Description
An admin wants to Ban a user from uploading recipes.

Initial Step-By-Step Description
Before this use case can be initiated, the admin must be logged into their account and there is a recipe user to ban.

1. The admin selects the user to ban.
2. They click Ban User from Uploading Recipes.
3. The admin confirms that they want to ban the user from uploading recipes.

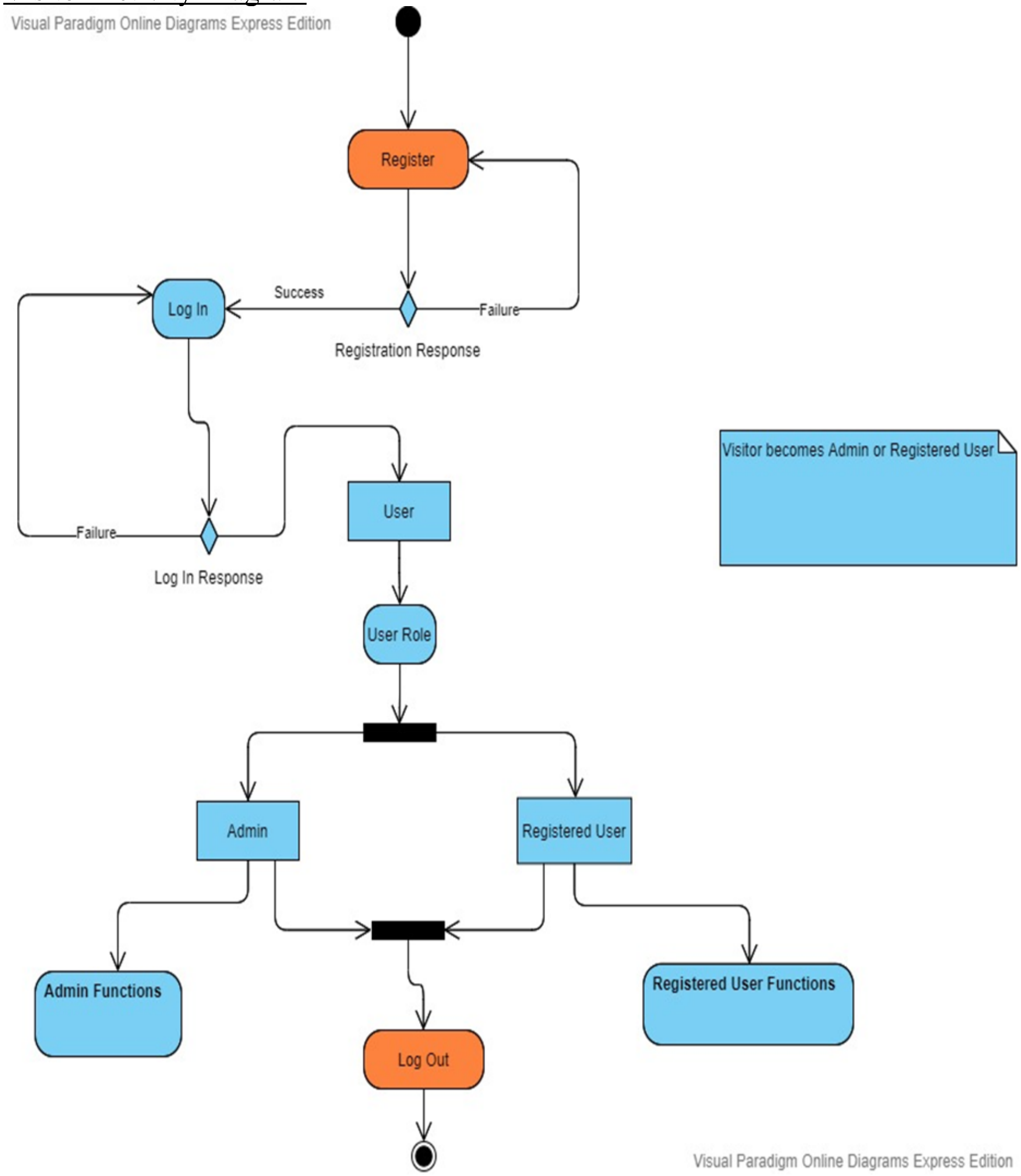
3.3 Data Modelling and Analysis

3.3.1 Normalized Data Model Diagram

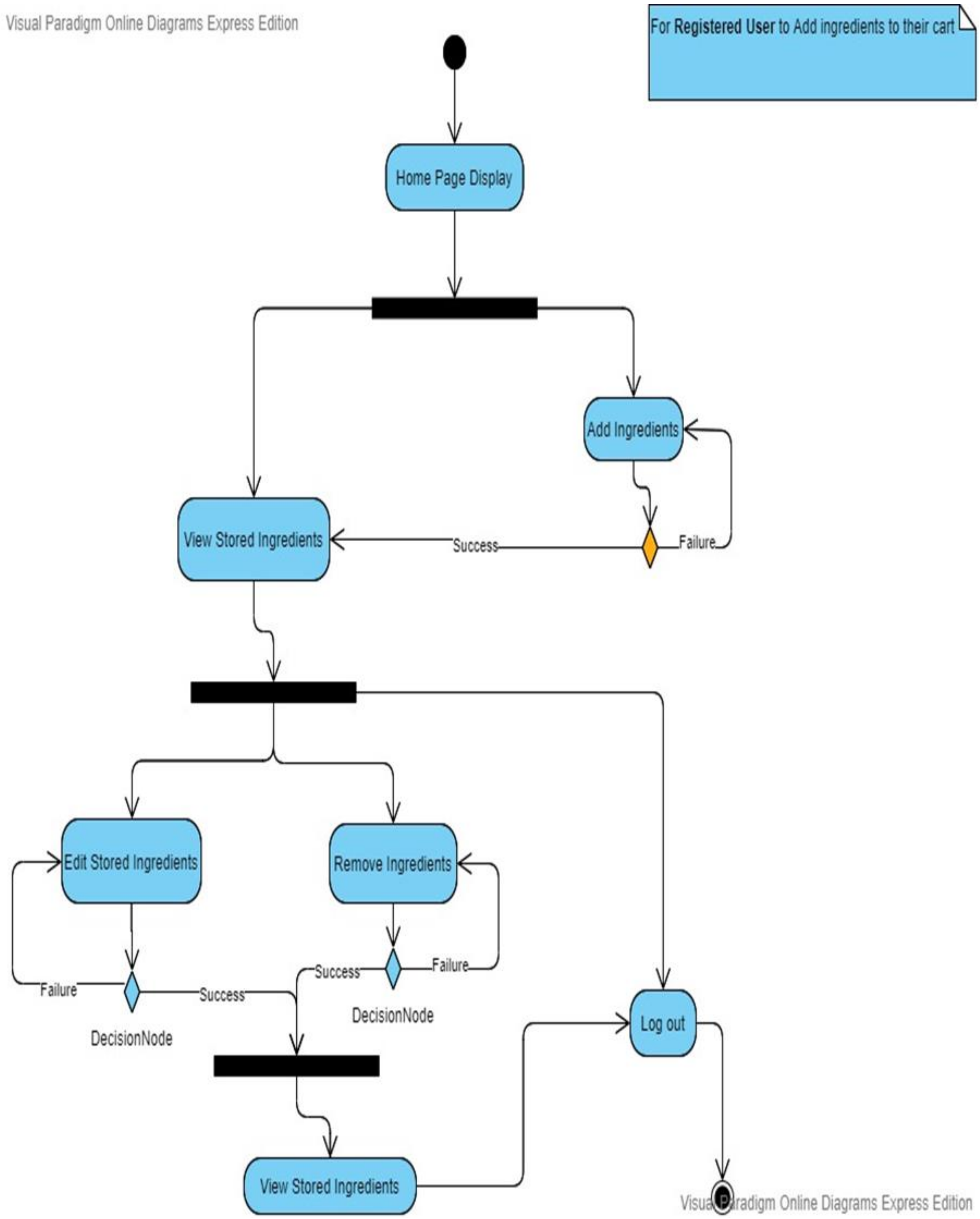


3.3.2 Activity Diagrams

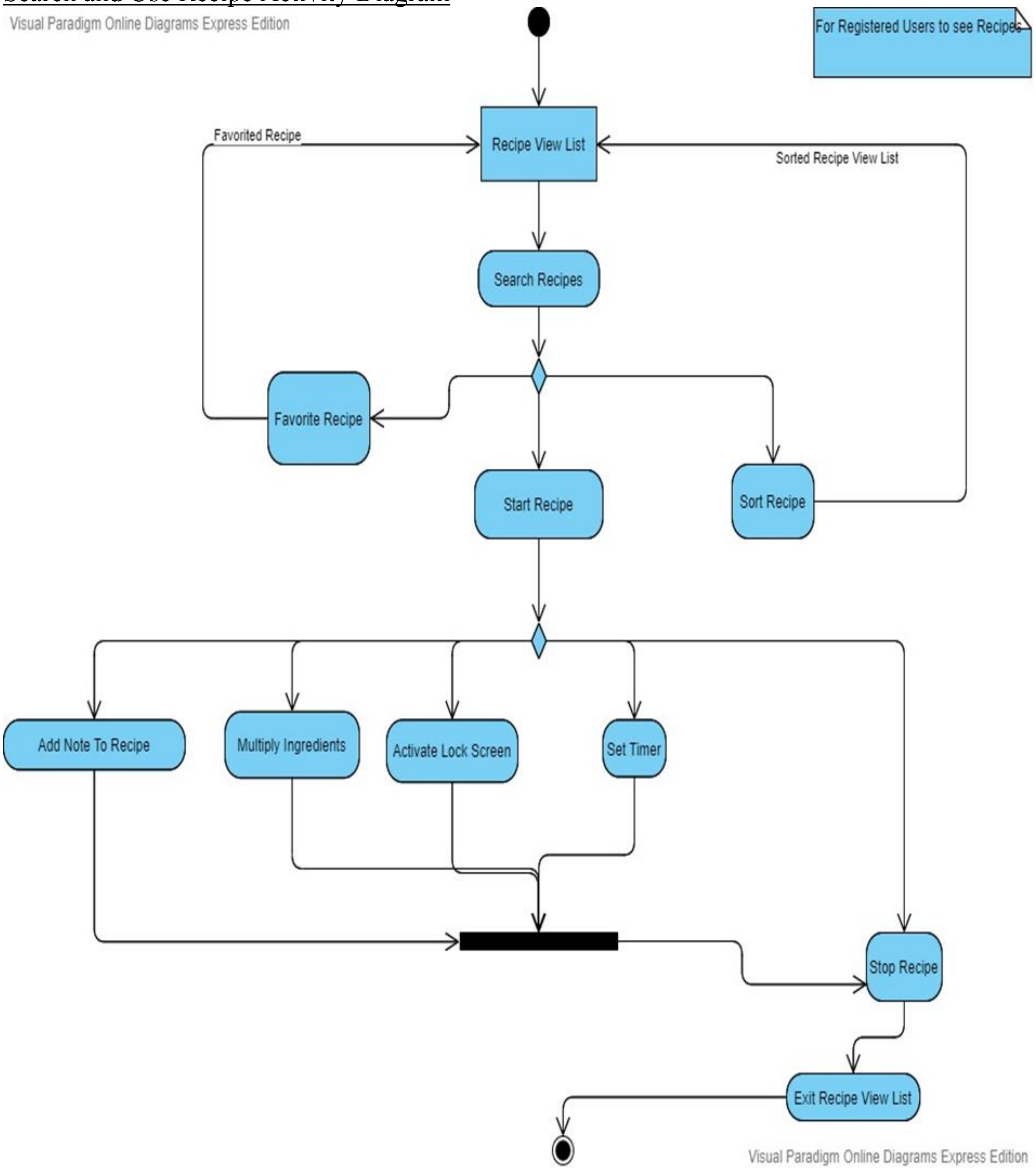
Visitor Activity Diagram


Visual Paradigm Online Diagrams Express Edition
Register
Success
Failure
Log In
Registration Response
Visitor becomes Admin or Registered User
Failure
User
Log In Response
User Role
Admin
Registered User
Admin Functions
Registered User Functions
Log Out
Visual Paradigm Online Diagrams Express Edition

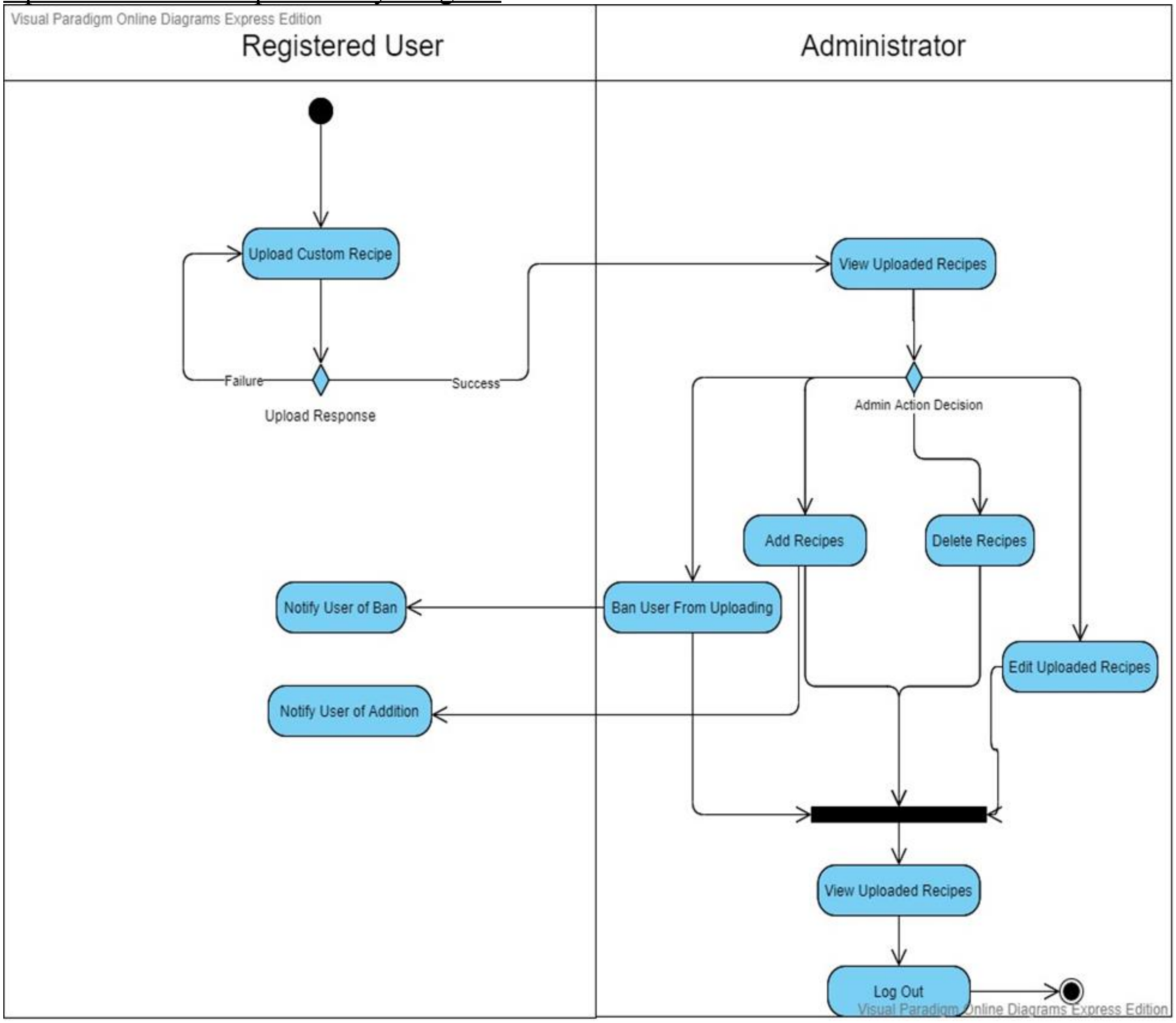
Ingredient Storage Activity Diagram



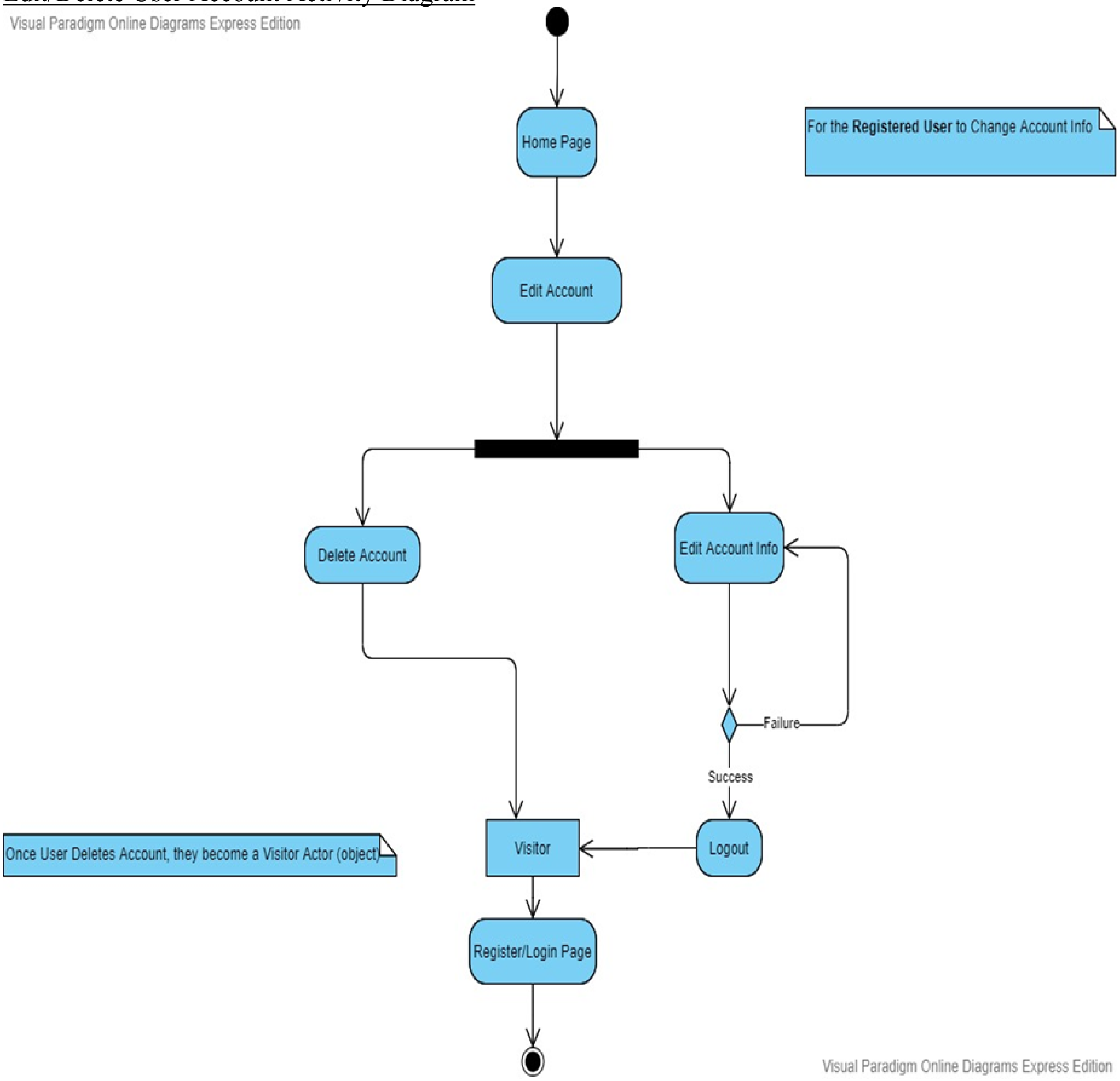
Search and Use Recipe Activity Diagram



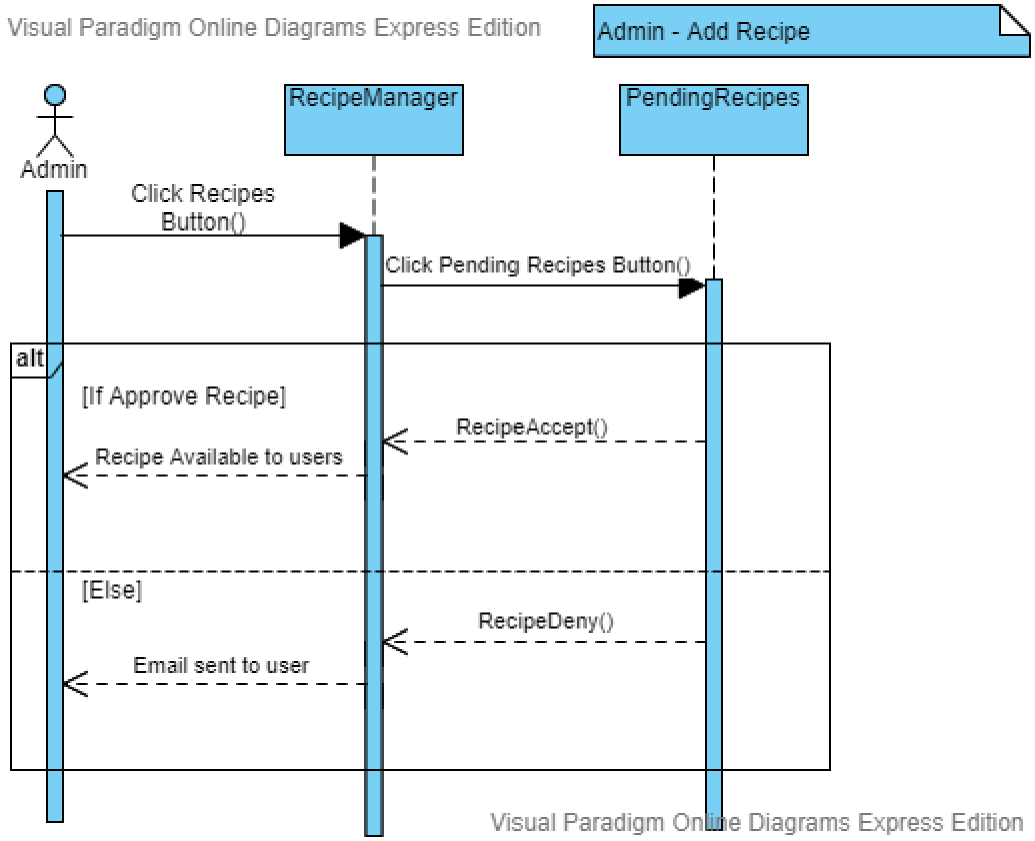
Upload Custom Recipe Activity Diagram

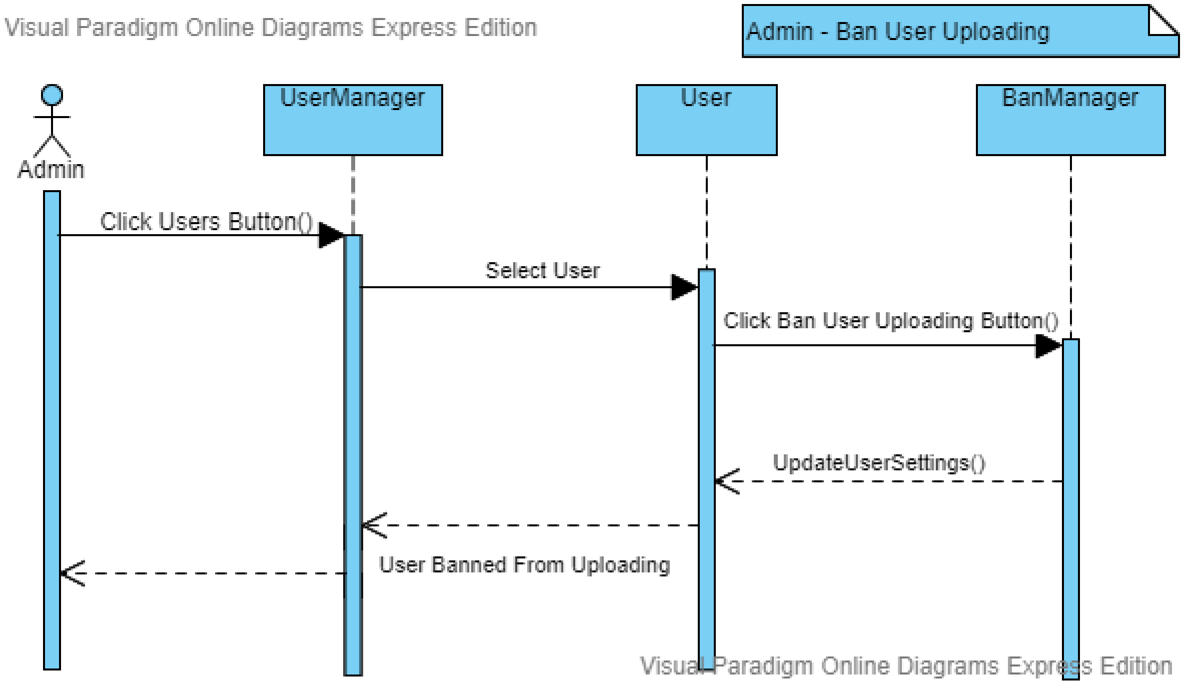


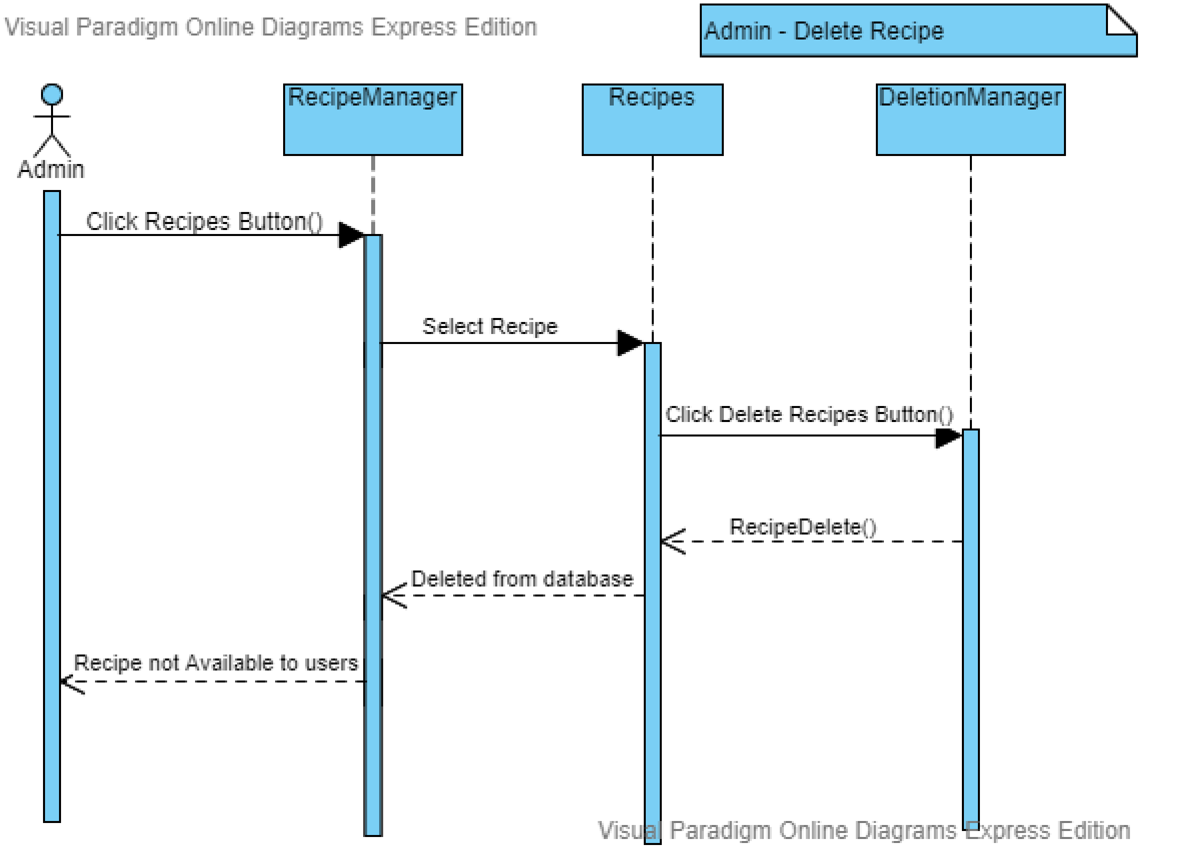
Edit/Delete User Account Activity Diagram

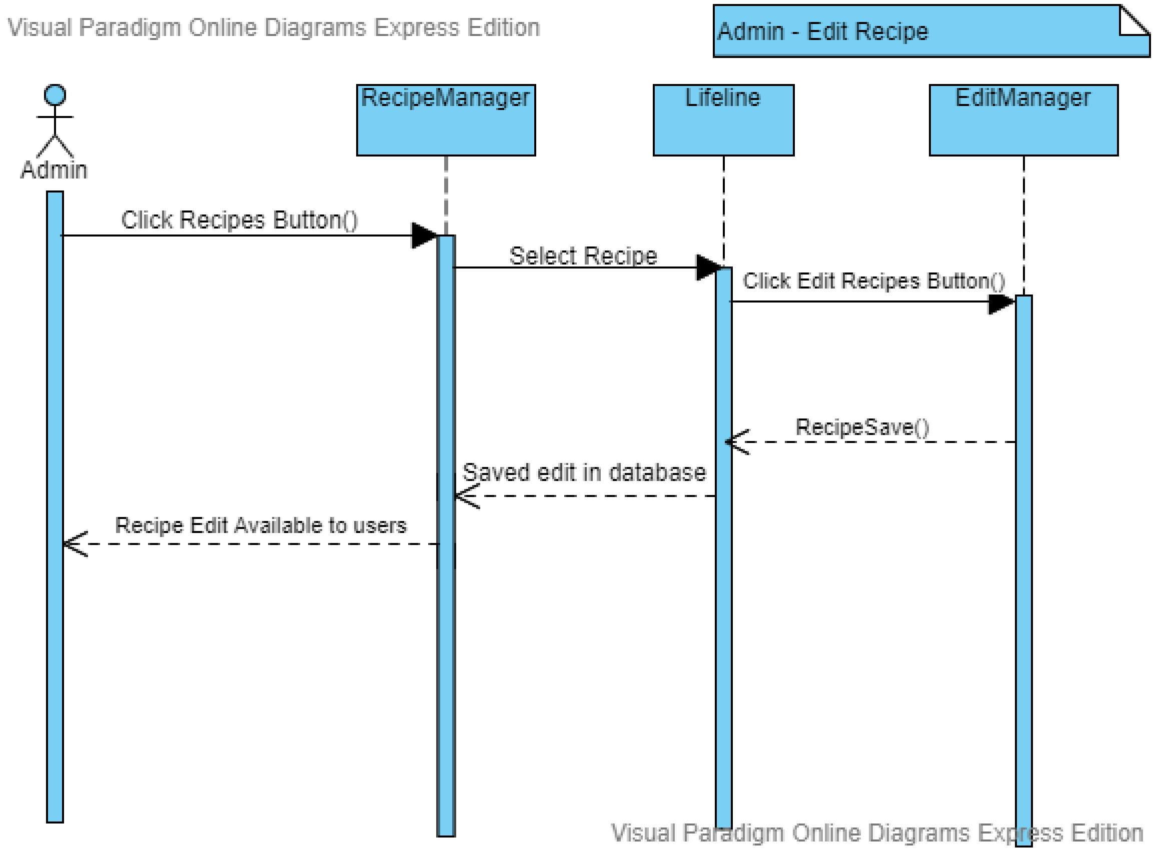

Visual Paradigm Online Diagrams Express Edition
Home Page
For the Registered User to Change Account Info
Edit Account
Delete Account
Edit Account Info
Failure
Success
Once User Deletes Account, they become a Visitor Actor (object)
Visitor
Logout
Register/Login Page
Visual Paradigm Online Diagrams Express Edition

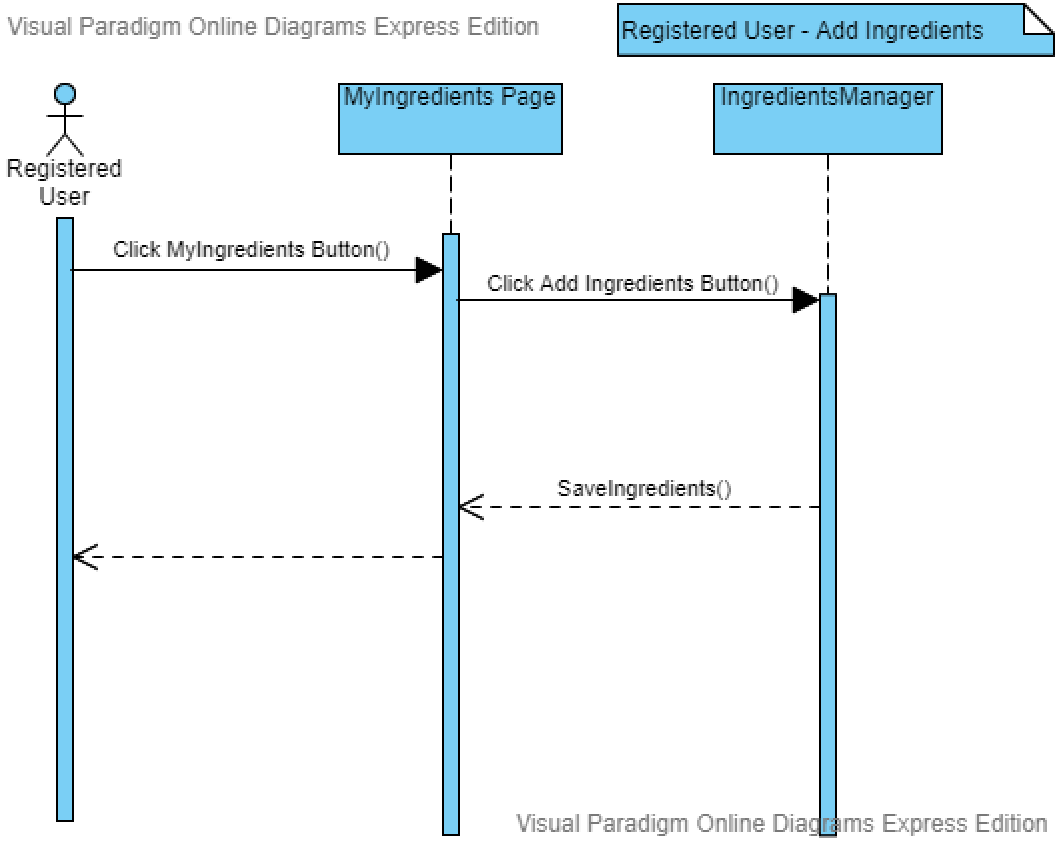
3.3.3 Sequence Diagrams

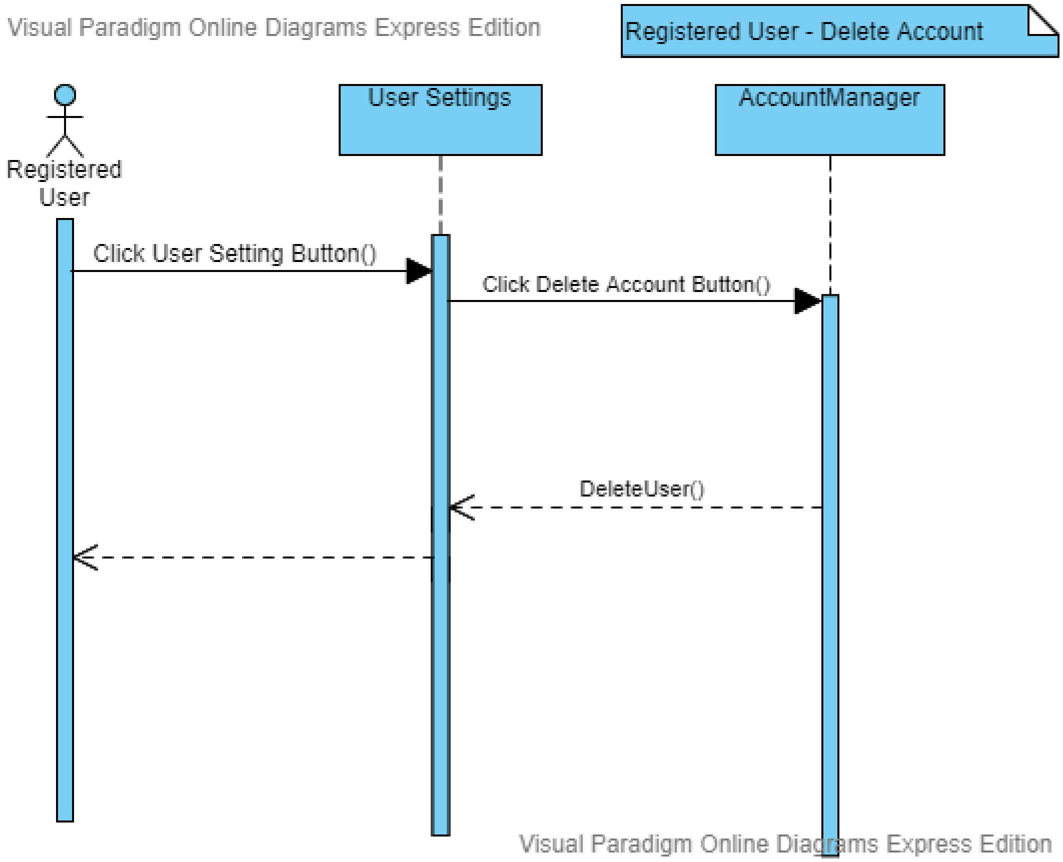

Visual Paradigm Online Diagrams Express Edition
Admin - Add Recipe
Admin
RecipeManager
PendingRecipes
Click Recipes Button()
Click Pending Recipes Button()
alt
[If Approve Recipe]
RecipeAccept()
Recipe Available to users
[Else]
RecipeDeny()
Email sent to user
Visual Paradigm Online Diagrams Express Edition

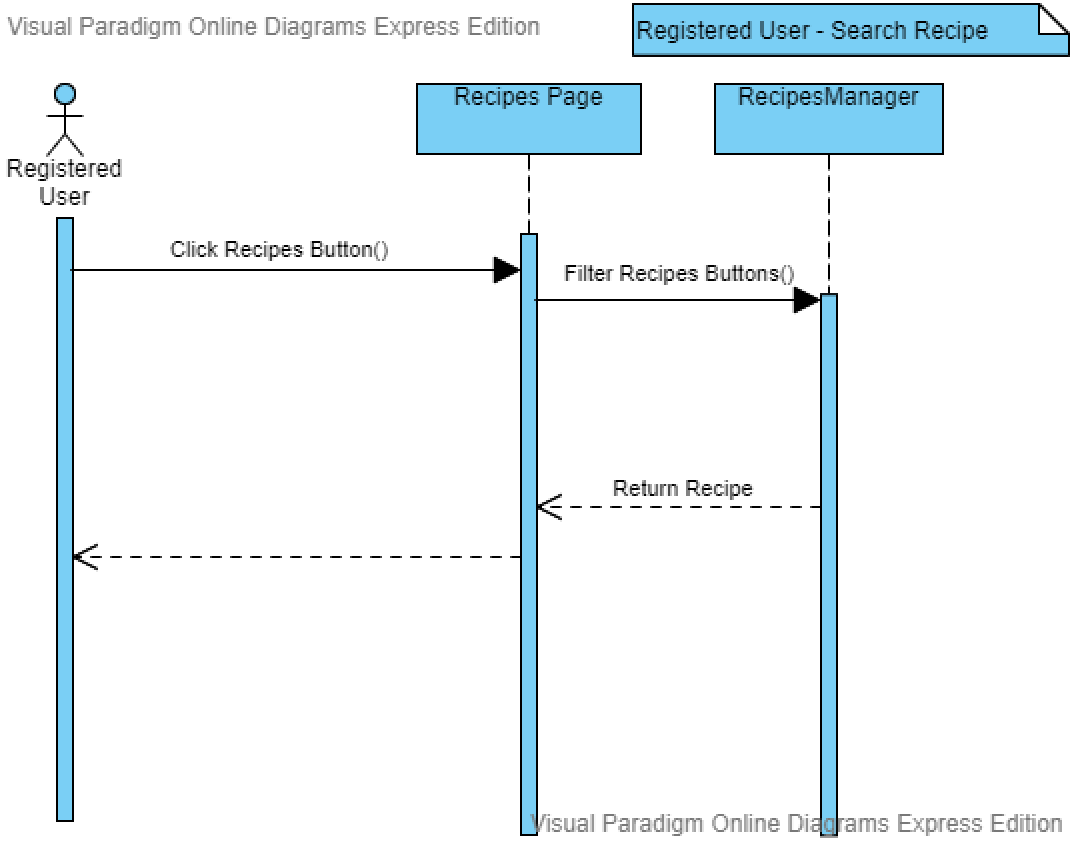

Visual Paradigm Online Diagrams Express Edition
Admin - Ban User Uploading
Admin
UserManager
User
BanManager
Click Users Button()
Select User
Click Ban User Uploading Button()
UpdateUserSettings()
User Banned From Uploading
Visual Paradigm Online Diagrams Express Edition

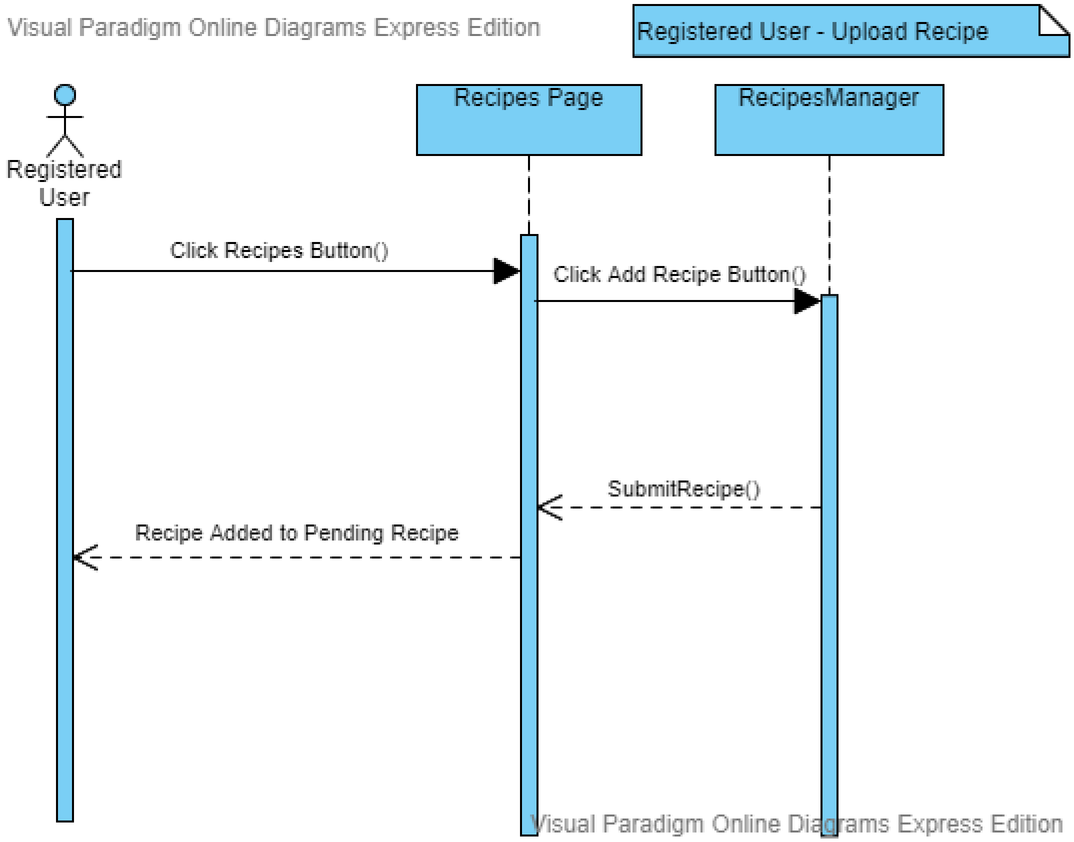

Visual Paradigm Online Diagrams Express Edition
Admin - Delete Recipe
Admin
RecipeManager
Recipes
DeletionManager
Click Recipes Button()
Select Recipe
Click Delete Recipes Button()
RecipeDelete()
Deleted from database
Recipe not Available to users
Visual Paradigm Online Diagrams Express Edition

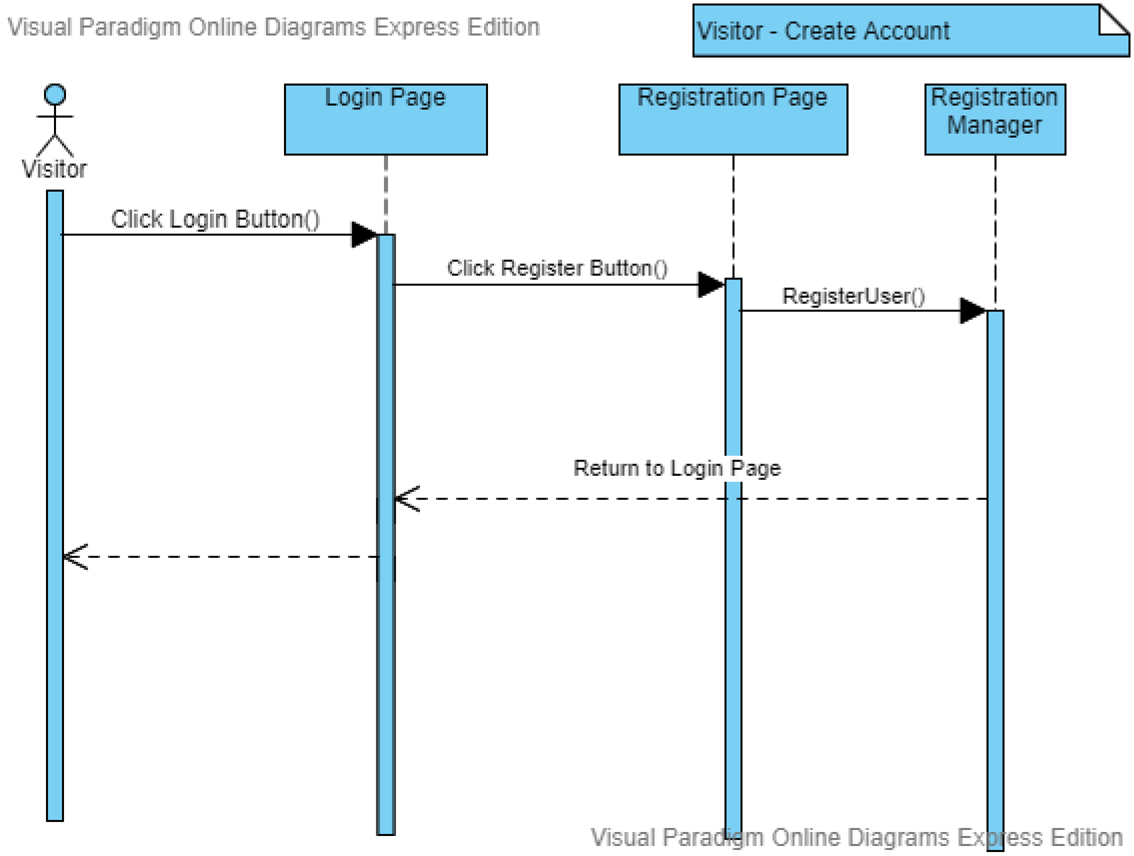

Admin - Edit Recipe
Admin
RecipeManager
Lifeline
EditManager
Click Recipes Button()
Select Recipe
Click Edit Recipes Button()
RecipeSave()
Saved edit in database
Recipe Edit Available to users

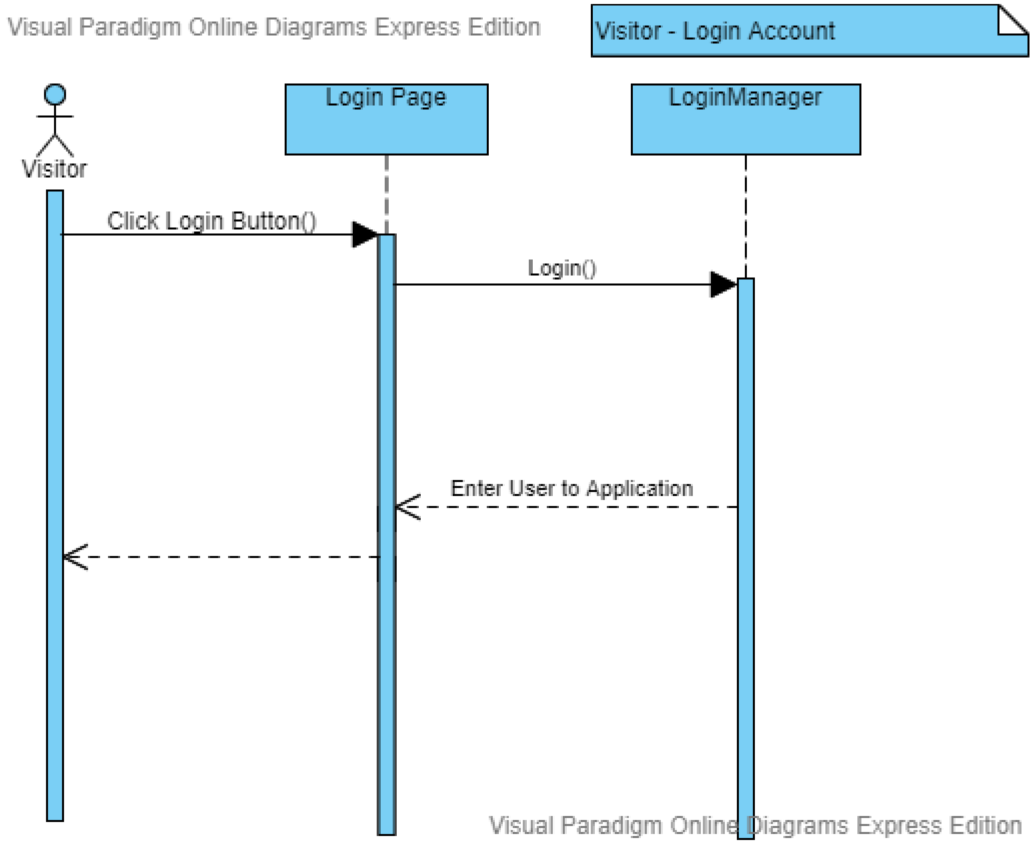

Registered User - Add Ingredients
Registered User
MyIngredients Page
IngredientsManager
Click MyIngredients Button()
Click Add Ingredients Button()
SaveIngredients()


Visual Paradigm Online Diagrams Express Edition
Registered User - Delete Account
Registered User
User Settings
AccountManager
Click User Setting Button()
Click Delete Account Button()
DeleteUser()
Visual Paradigm Online Diagrams Express Edition

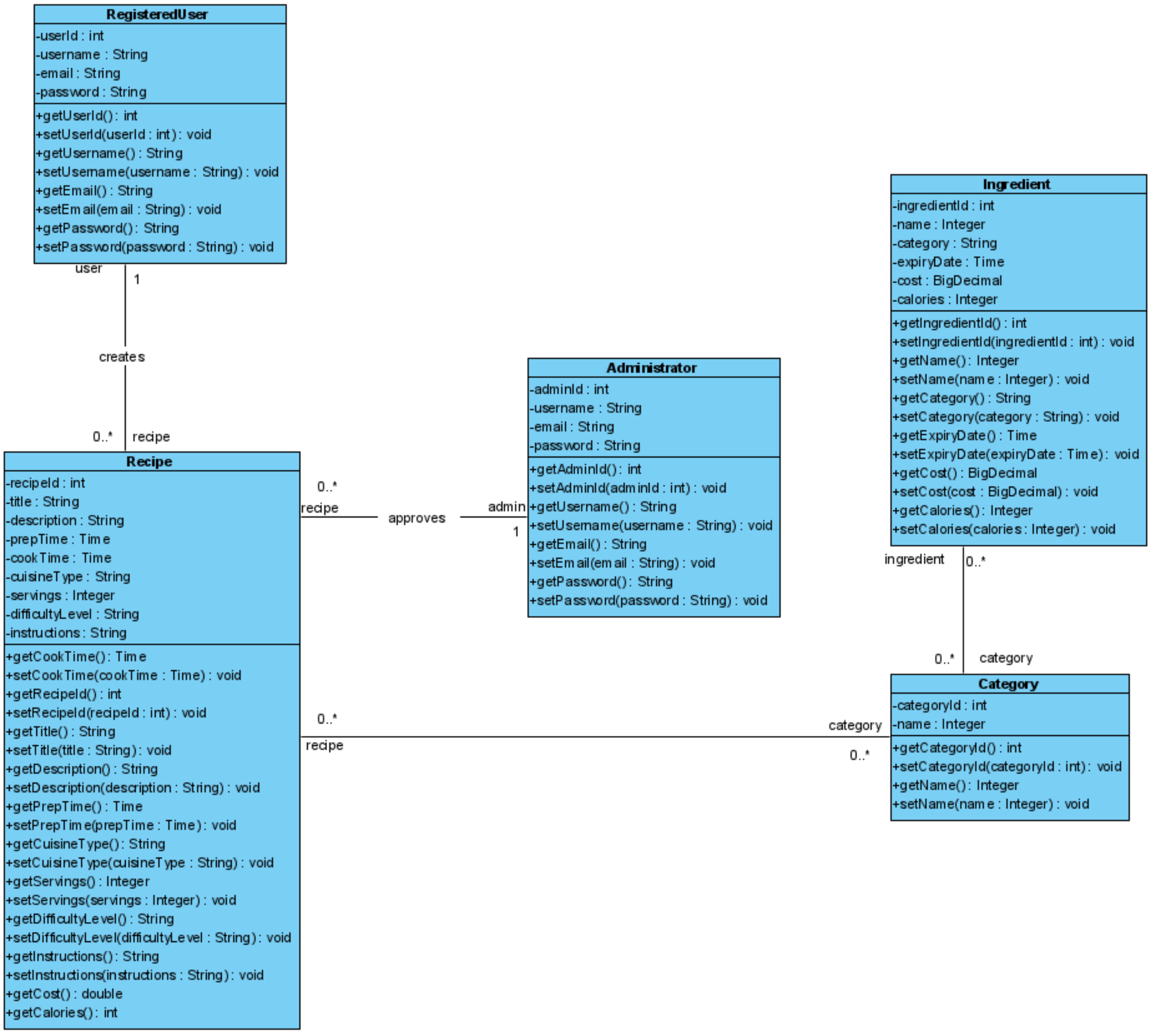

Visual Paradigm Online Diagrams Express Edition
Registered User - Search Recipe
Registered User
Recipes Page
RecipesManager
Click Recipes Button()
Filter Recipes Buttons()
Return Recipe
Visual Paradigm Online Diagrams Express Edition


Visual Paradigm Online Diagrams Express Edition
Registered User - Upload Recipe
Registered User
Recipes Page
RecipesManager
Click Recipes Button()
Click Add Recipe Button()
SubmitRecipe()
Recipe Added to Pending Recipe
Visual Paradigm Online Diagrams Express Edition


Visual Paradigm Online Diagrams Express Edition
Visitor - Create Account
Visitor
Login Page
Registration Page
Registration Manager
Click Login Button()
Click Register Button()
RegisterUser()
Return to Login Page
Visual Paradigm Online Diagrams Express Edition

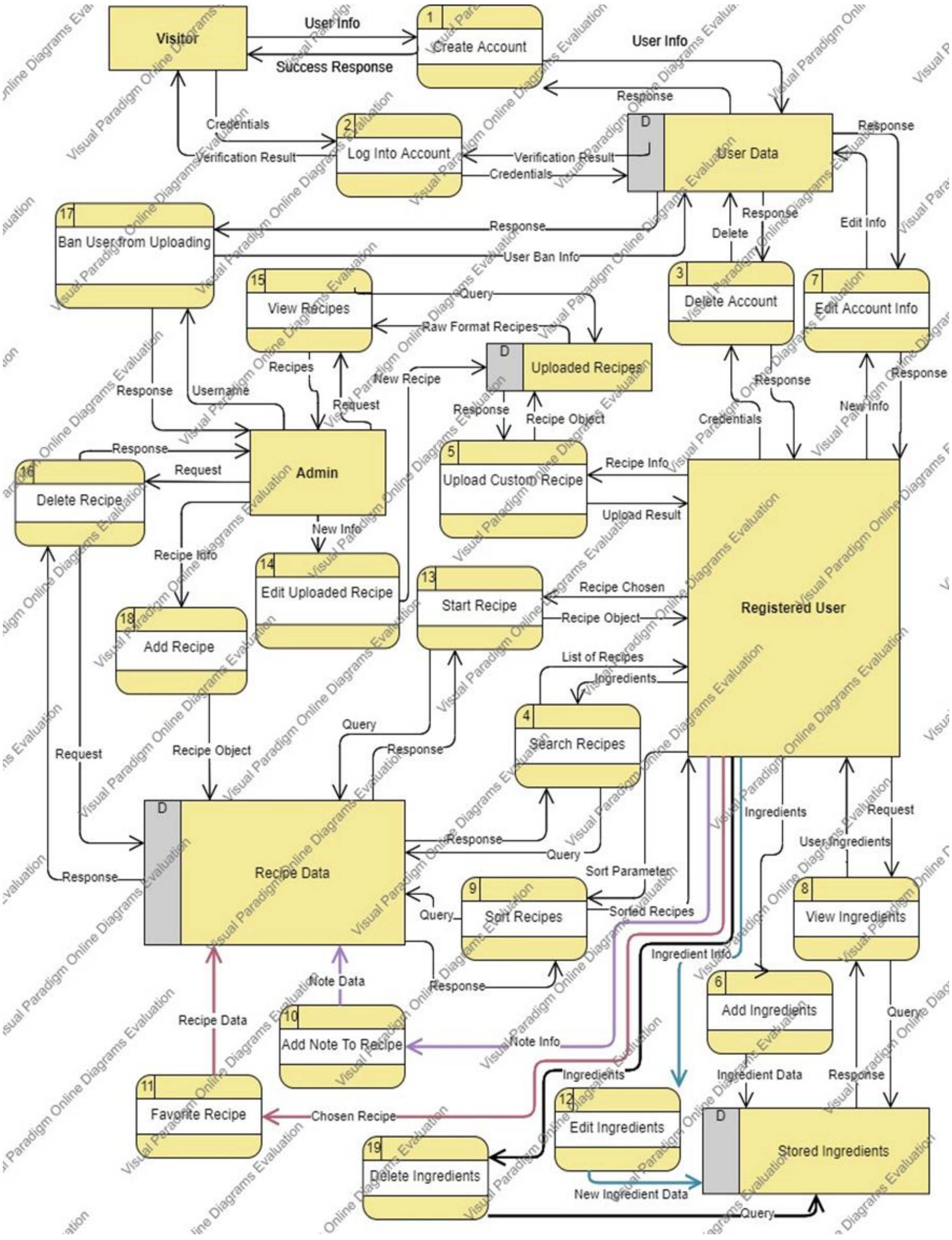

Visual Paradigm Online Diagrams Express Edition
Visitor - Login Account
Login Page
LoginManager
Visitor
Click Login Button()
Login()
Enter User to Application
Visual Paradigm Online Diagrams Express Edition

3.3.4 UML Class Diagram


RegisteredUser
-userId : int
-username : String
-email : String
-password : String
+getUserId() : int
+setUserId(userId : int) : void
+getUsername() : String
+setUsername(username : String) : void
+getEmail() : String
+setEmail(email : String) : void
+getPassword() : String
+setPassword(password : String) : void
user
1
creates
0..*
recipe
Recipe
-recipeId : int
-title : String
-description : String
-prepTime : Time
-cookTime : Time
-cuisineType : String
-servings : Integer
-difficultyLevel : String
-instructions : String
+getCookTime() : Time
+setCookTime(cookTime : Time) : void
+getRecipeId() : int
+setRecipeId(recipeId : int) : void
+getTitle() : String
+setTitle(title : String) : void
+getDescription() : String
+setDescription(description : String) : void
+getPrepTime() : Time
+setPrepTime(prepTime : Time) : void
+getCuisineType() : String
+setCuisineType(cuisineType : String) : void
+getServings() : Integer
+setServings(servings : Integer) : void
+getDifficultyLevel() : String
+setDifficultyLevel(difficultyLevel : String) : void
+getInstructions() : String
+setInstructions(instructions : String) : void
+getCost() : double
+getCalories() : int
0..*
recipe
approves
admin
1
Administrator
-adminId : int
-username : String
-email : String
-password : String
+getAdminId() : int
+setAdminId(adminId : int) : void
+getUsername() : String
+setUsername(username : String) : void
+getEmail() : String
+setEmail(email : String) : void
+getPassword() : String
+setPassword(password : String) : void
Ingredient
-ingredientId : int
-name : Integer
-category : String
-expiryDate : Time
-cost : BigDecimal
-calories : Integer
+getIngredientId() : int
+setIngredientId(ingredientId : int) : void
+getName() : Integer
+setName(name : Integer) : void
+getCategory() : String
+setCategory(category : String) : void
+getExpiryDate() : Time
+setExpiryDate(expiryDate : Time) : void
+getCost() : BigDecimal
+setCost(cost : BigDecimal) : void
+getCalories() : Integer
+setCalories(calories : Integer) : void
ingredient
0..*
0..*
category
0..*
recipe
category
0..*
Category
-categoryId : int
-name : Integer
+getCategoryId() : int
+setCategoryId(categoryId : int) : void
+getName() : Integer
+setName(name : Integer) : void

3.4 Process Modelling

3.4.1 Data Flow Diagram



4. Non-Functional Requirements

Performance

- Application should be capable enough to handle 10,000 users without affecting its performance.
- Each request should be processed within 3 seconds or less.
- The application should load in 3 seconds or less, when the number of simultaneous users are less than 10000.
- System downtime may not exceed 2 minutes per day.

Reliability

- The system should aim to have a 85 percent reliability during the first year of running the application, meaning that there is an 85 percent chance that the system will not experience critical failure.
- In case of critical failure system should aim to be restored within 72 hours.
- The system should aim to have a mean time between critical failures of at least 7 months.
- The system should aim to have at the most 1 critical failure for the first year.

Availability

- The application should be available 98 percent of the time during the first year, unless any necessary system changes need to be done.
- Emails should be sent with a latency of no greater than 24 hours.

Security

- Users should not be able to access Administrator pages and functions.
- Visitors should not be able to access User privileges or Admin privileges without creating and logging into an account.
- User private information such as Passwords should be unavailable to view for any User or Admin.

- User information in Database must be 99 percent secured and the necessary privacy precautions must be put in place.
- Databases and Application must be as 90 percent secure from potential security threats, such as malicious programs or hackers.

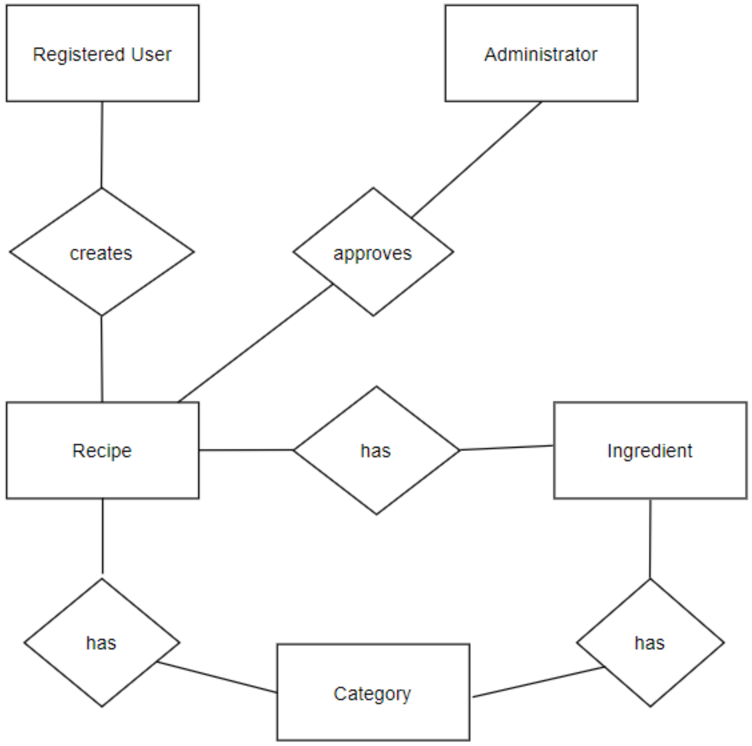
Maintainability

- Aim for 75 percent maintainability within 24 hours, which means there is a 75 percent chance that the component will be resolved within 24 hours.
- Aim for 100 percent maintainability within 72 hours, which means there is a 100 percent chance that the component will be resolved within 72 hours.
- Aim for 75 percent maintainability within 96 hours for critical system failures.
- Aim for 100 percent maintainability within 7 days for critical system failures.

Portability

- Application should be 100 percent available on Android mobile phones.
- Application should run on data networks as well as public and local wifi networks.
- The software code should aim to be at least 75 percent portable, so moving from one OS to another OS does not create any problem.
- The software should be 100 percent compatible with the firewall and antivirus protection programs on its corresponding OS.

5. Logical Database Requirements



Registered User Data Entity

Data Item	Type	Description	Comment
User Id	Integer	ID number of registered user	Primary Key
Username	String	Username of registered user	Unique
Email	String	Internet Address	Unique
Password	String	Password to access Account	Unique

Administrator Data Entity

Data Item	Type	Description	Comment
Admin ID	Integer	ID number of administrator	Primary Key
Username	String	Username of administrator	Unique
Email Address	String	Internet address	Unique
Password	String	Password to access account	Unique

Recipe Data Entity

Data Item	Type	Description	Comment
Recipe ID	Integer	ID number of recipe	Primary Key
User	Pointer	User Entity	User who created the recipe Foreign key
Admin	Pointer	Admin Entity	Admin who approved the recipe Foreign key
Title	String	Title of recipe	
Description	String	Short description of recipe	
Prep Time	Time	Total time to prep ingredients	
Cook Time	Time	Total time it takes to cook	
Cuisine Type	String	Type of cuisine	
Servings	Integer	Number of servings	
Difficulty Level	String	Difficulty of recipe	
Instructions	String	Body of recipe	

Ingredient Data Entity

Data Item	Type	Description	Comment
Ingredient ID	Integer	ID number of ingredient	Primary Key
Name	String	Name of ingredient	
Expiry Date	Date	Date ingredient will expire	
Cost	Double	Cost of Ingredient	
Calories	Integer	Calories of ingredient	

Category Data Entity

Data Item	Type	Description	Comment
Category ID	Integer	ID number of category	Primary Key
Name	String	Name of Category	

Recipe Ingredient Associative Entity

Data Item	Type	Description	Comment
Recipe	Pointer	Recipe entity	
Ingredient	Pointer	Ingredient entity	

Recipe Category Associative Entity

Data Item	Type	Description	Comment
Category	Pointer	Category entity	
Recipe	Pointer	Recipe entity	

Ingredient Category Associative Entity

Data Item	**Type**	**Description**	**Comment**
Category	Pointer	Category entity	
Ingredient	Pointer	Ingredient entity	

6. Other Requirements

There are no additional requirements.