جامعة أم القرى
UMM AL-QURA UNIVERSITY

Umm Al-Qura University
Computer & Info. Systems College
Computer Science Department

Maqadeer

PROJECT TEAM:

Rehab Al-Idrisi

Afnan Farouqui

Ekram Jamous

Nouf Twakkul

Fadia Alshaarani

Project of Advanced Programming Course

INSTRUCTOR: I.Bushra Algotiml

FISRT SEMESTER 2021-2022

1. Project Title

Maqadeer

2. Project Idea and Aims

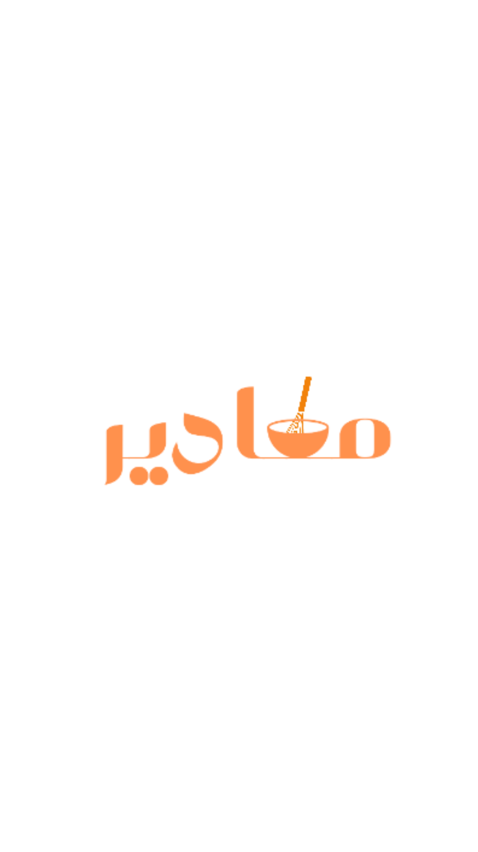
If you are one of those who are perplexed and overwhelmed by the constant question “What am I going to cook today?” then this app is for you.

It keeps all your foodstuffs to give you a suggested meal to cook. Also, it will keep all your needs in a menu list either manually or by relying on food staff that has been finished recently. Finally, it`s a good place to keep all recipes from social media that you plan to cook. in simple words, this app is your right hand in cooking.

4. Design and Implementation

A. Graphical User Interface

In our application, we used 14 interfaces to help the user benefit from our services:

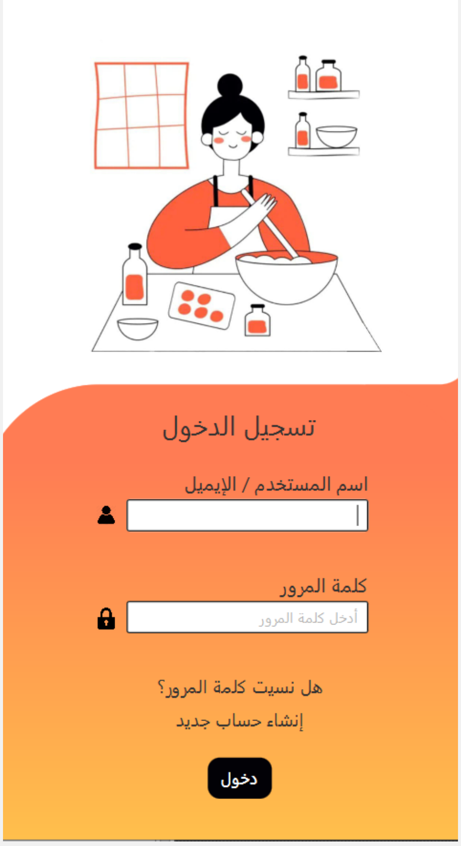


The logo interface

Contain the name of our project and one of the letters designed to be a bowl and a whisk that turn back and forth representing the main idea of this app ingredients and cooking .

4. Design and Implementation

A. Graphical User Interface



The Log-in interface

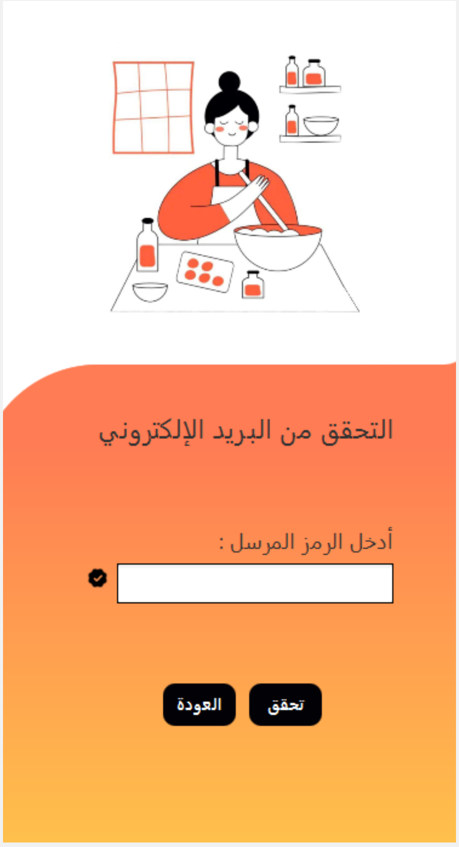
Contain a text field that will take the username or email and a password field to enter the password and check if it is matching any stored data in the database Under each field there is a label to present any error may occur like the username or email is not found or incorrect password

There are two labels forget password if the user has an account, but he forgot the password and a label for creating an account if it is the user first visit. If the user forgot his password, he must enter his email in the username/email field to check if this email exists otherwise a massage in the label under the text field will be shown that the user must enter his email first, and the user will not be able to go to the page. If the email exists in the database, the user will go to the verify account page

if the user successfully entered all the data correct, he can press the login button to take him to the home page

4. Design and Implementation

A. Graphical User Interface



التحقق من البريد الإلكتروني

أدخل الرمز المرسل :

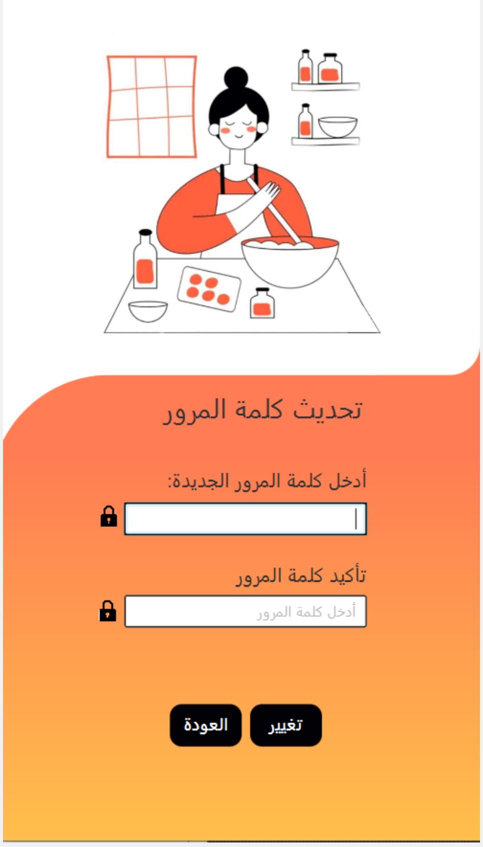
The Verify Account interface

the page can be reach only after the user entered an exists email where automatically a massage will be sent to the email contain a code

the Verify Account page contain a text field to enter the code that been sent and two buttons back to take you back to the login page and verify to check if the code is correct to take the user to the Forget Password page.

4. Design and Implementation

A. Graphical User Interface



The Forget Password interface

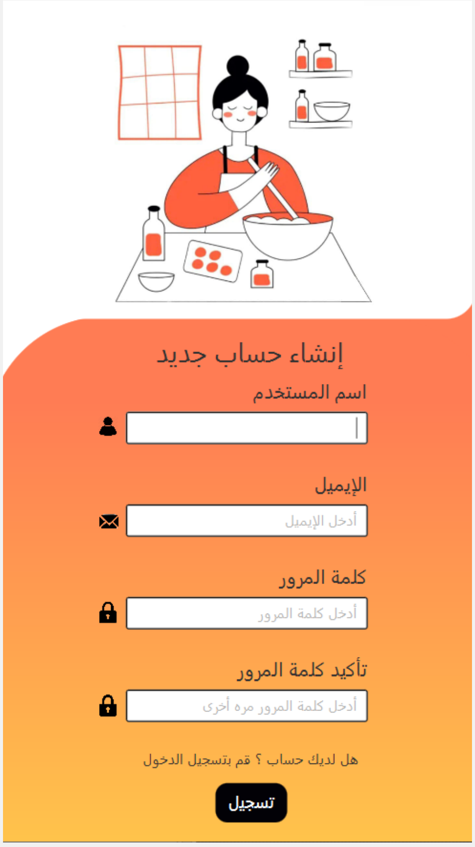
this page contain two password field the first one to enter the new password under the password field a label shows a massage if the password is not as required to be an acceptable password should contain (small letters, capital letters, digits, special character)

the second password field for extra checking that the two entered passwords are identical also a label under the second password field if the password doesn't match

at the bottom there are two buttons back button to return to the login page and change button to change the password and go back to the login page so the user login with the new password

4. Design and Implementation

A. Graphical User Interface



إنشاء حساب جديد

اسم المستخدم

الإيميل

أدخل الإيميل

كلمة المرور

أدخل كلمة المرور

تأكيد كلمة المرور

أدخل كلمة المرور مره أخرى

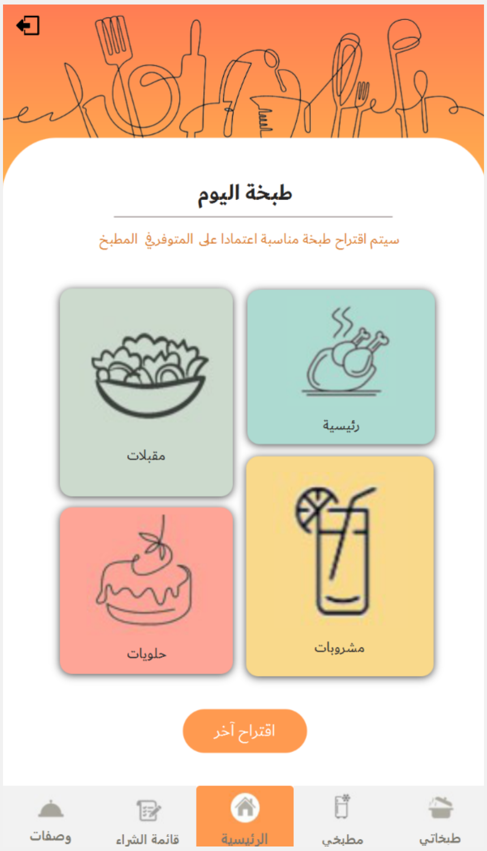
هل لديك حساب ؟ قم بتسجيل الدخول

The Sign-up interface

the sign-up page is for the user who does not have an account and want to create one the sign-up page contains a text field to enter the username under the username text field there is a label if the name already exists in the database also, a text field for email and label if the entered is not the format of an email or the email do exist in the database finally, a password field the first one to enter the new password under the password field a label shows a massage if the password is not as required to be an acceptable password should contain (small letters, capital letters, digits, special character). the second password field for extra checking that the two entered passwords are identical also a label under the second password field if The page contains a label if the user does have an account and want to login the label will take him to the login page if the user successfully entered all the data correct, he can press the sign button to take him to the Verify Account page

4. Design and Implementation

A. Graphical User Interface



طبخة اليوم

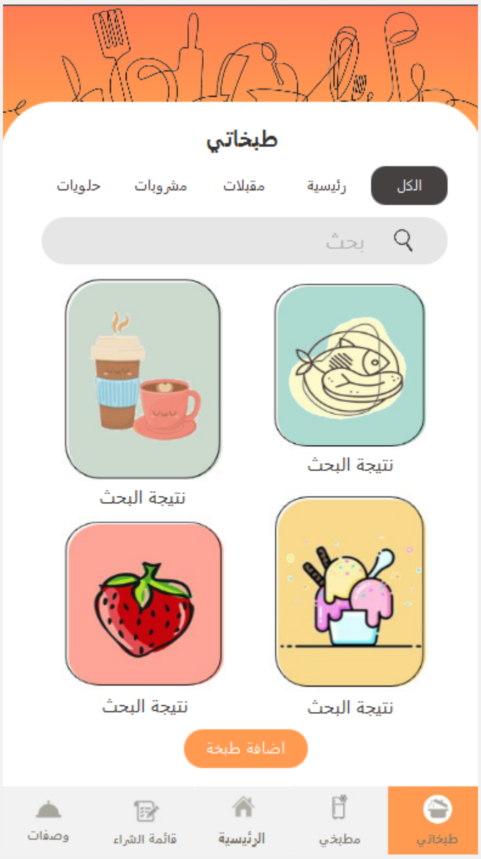
سيتم اقتراح طبخة مناسبة اعتمادا على المتوفرفي المطبخ

The Home Page interface

The main page: shows you a suggested recipe , based on the category you selected (main course, dessert, appetizer, drink), by search in your recipes based on the available and unexpired ingredients that you registered in the page of food purchases. If you do not like the suggestion presented and want to suggest another button (Another suggestion) will allow you to suggest another recipe. At the top of the main page there is a logout button in case the user wants to log out of his account. Also, the main page contains a navigation pane, which is a component of Radio Button that enables you to move between pages according to the selected button.

4. Design and Implementation

A. Graphical User Interface



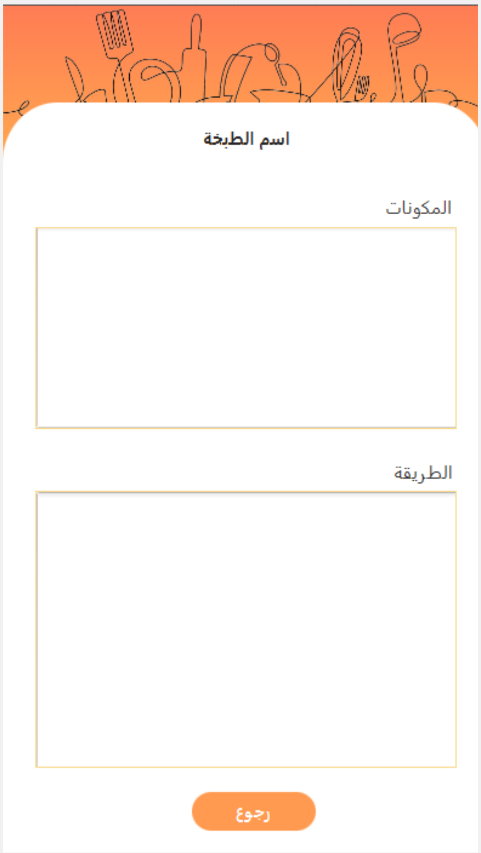
My Recipes interface

if the user chooses the (My Recipes) button from the navigation pane on the main page, it will transfer him to the (My Recipes) page:

On the My Recipes page, all the recipes that the user has added in the application are displayed according to the category chosen by the user (all, main dish, appetizers, drinks, desserts) or he can access a specific recipe by typing the name of the cook in the search text field.

4. Design and Implementation

A. Graphical User Interface



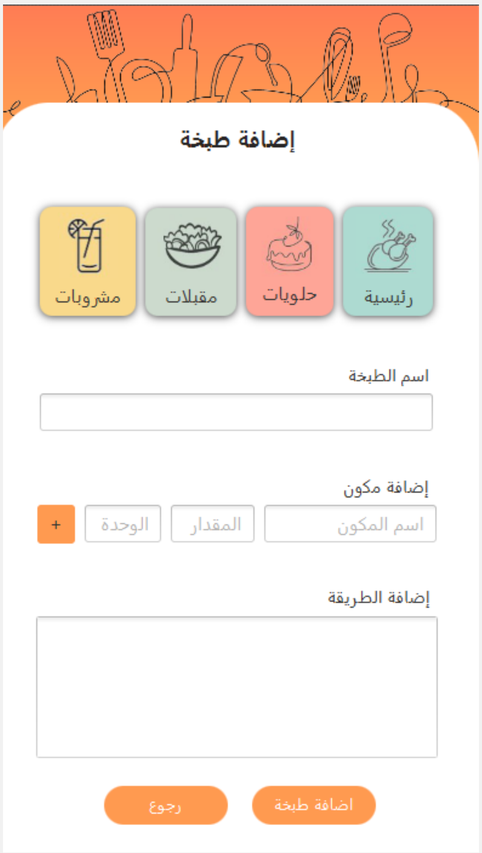
View Recipes interface

if the user chooses a specific recipe, he will be transferred to the recipes display interface, which will display the ingredients and method of the recipe.

In the event that the user wants to return to the My Cooks page, he can press the (back) button.

4. Design and Implementation

A. Graphical User Interface

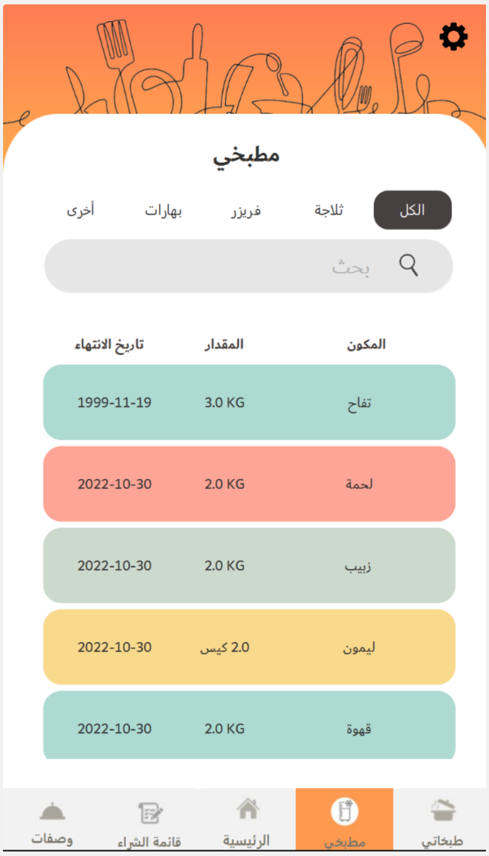


Adding Recipes interface

On the (Adding Recipe) page, the user must choose the appropriate classification, then fill in the form by entering the name of the recipe and its ingredients, adjusting the amount of each ingredient according to the appropriate unit and writing the method in the text area.

4. Design and Implementation

A. Graphical User Interface



المكون	المقدار	تاريخ الانتهاء
تفاح	3.0 KG	1999-11-19
لحمة	2.0 KG	2022-10-30
زبيب	2.0 KG	2022-10-30
ليمون	2.0 كيس	2022-10-30
قهوة	2.0 KG	2022-10-30

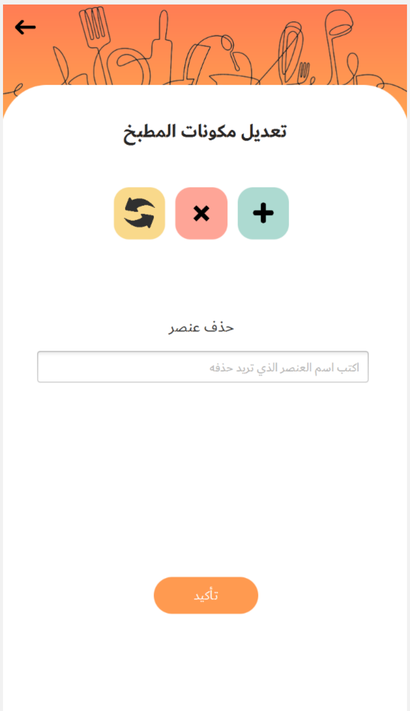
My Kitchen interface

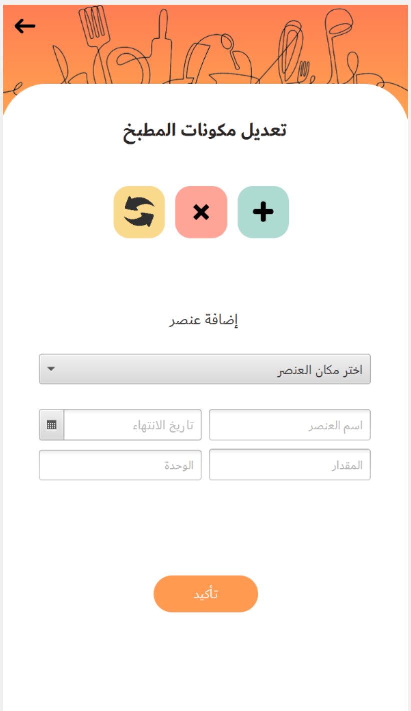
If the user chooses the (My Kitchen) button from the navigation pane on the main page, it will transfer him to the (My Kitchen) page:

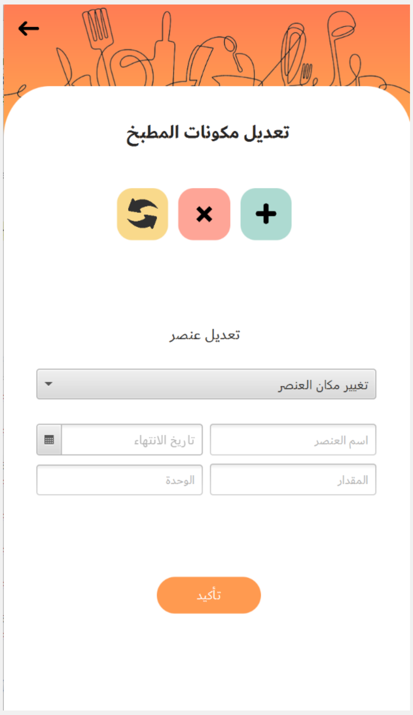
On the (My Kitchen) page, all kitchen components are displayed according to the classification chosen by the user (all, refrigerator, freezer, spices, other), and the user can reach a specific item by typing the component in the search field. If the user wants to add a new component, delete a component, or make an adjustment, he can click on the icon at the top of the page on the right.

4. Design and Implementation

A. Graphical User Interface





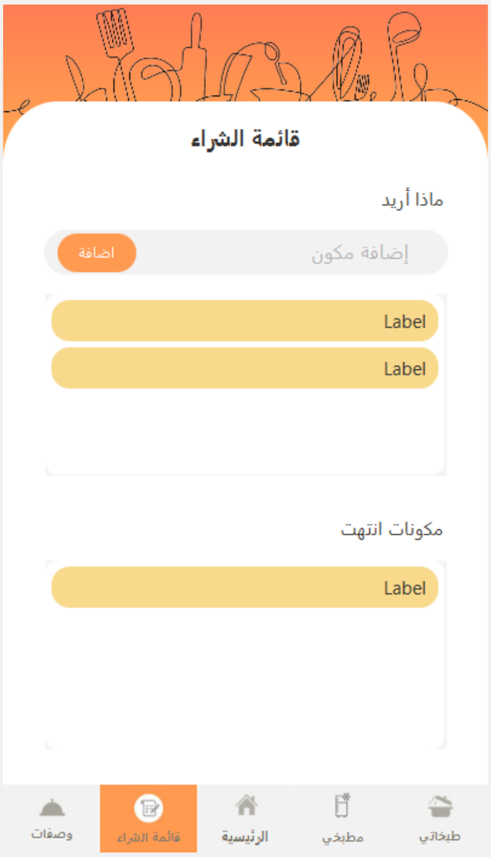


Kitchen Setting interface

On the () page, the user will see three buttons through which he can choose what suits him according to his need (add, delete, modify), then choose from the combo box to classify the item he wants to add or modify and fill in the name of the component, its expiration date and its amount in the appropriate unit for the amount, but if Wanted to delete an item, just type the item name in the text field. Finally, the user presses the button (Confirm) to confirm the operation. And if the user wants to go back to page(My Kitchen), he will press the icon at the top of the page on the right.

4. Design and Implementation

A. Graphical User Interface

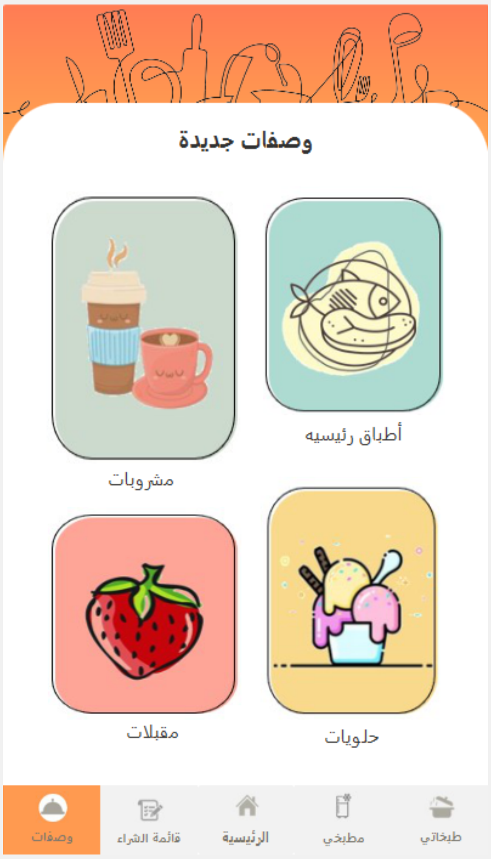


Shopping Listinterface

If the user chooses the (Shopping List) button from the navigation pane on the main page, it will transfer him to the (Shopping List) page: On the (Shopping List) page, the user can add components to be purchased later. There is also a pane to display the components that are about to expire.

4. Design and Implementation

A. Graphical User Interface

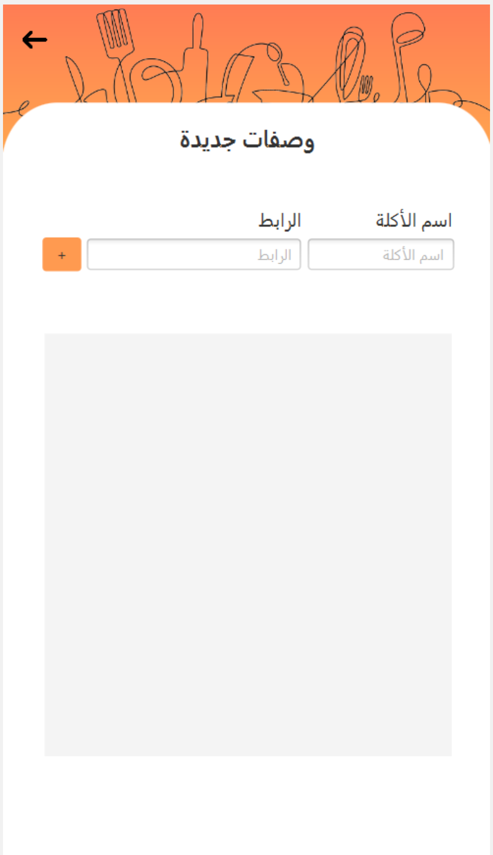


The New Recipes interface

this page contains 4 buttons under each button a label specifies the category of the recipes (main course, appetizer, drinks, desert) if the user press any of those buttons the user will be transfer to the Display Recipes page of the chosen category. at the bottom four buttons at the navigation bar the second button from the left will transfer the user to the Shopping List page. pressing the third button from the left will transfer the user to the home page and pressing the fourth button from the left will transfer the user to My Kitchen page, finally pressing the first button form the right will transfer the user to My Recipes page.

4. Design and Implementation

A. Graphical User Interface



Display Recipes interface

this page contains 4 buttons under each button a label specifies the category of the recipes (main course, appetizer, drinks, desert) if the user press any of those buttons the user will be transfer to the Display Recipes page of the chosen category. at the bottom four buttons at the navigation bar the second button from the left will transfer the user to the Shopping List page. pressing the third button from the left will transfer the user to the home page and pressing the fourth button from the left will transfer the user to My Kitchen page, finally pressing the first button form the right will transfer the user to My Recipes page.

4. Design and Implementation

B. Event-Driven Programming

On the Sign up page, We have two types of Event, Action Event and Mouse Event. The source for the action event is the "register" button at the bottom of this page. This event has been programmed in the "signUp" method, which in turn checks that all fields are filled in and also to verify that all the data entered in the field are correct and meet the required restrictions . If all the data entered are correct, this event will send an email to the user and it will be taken to the verify Account page.
The source for Mouse Event is the label "Do you have an account? login".This event has been programmed in the "goLogPage" method
Which in turn will take us to the login page.

On the verify Account page, there is only one type of event, which is Action Event . The source for this event is two buttons.
The first button is a "verification" button in which its special event has been programmed in a "verify" method, which in turn checks if the code entered by the user matches the code that was sent to the user or not. If it matches, therefore according to the previous page you came from, this event will do one of two tasks:
If I am coming from the Sign Up page, this event will take me to the homepage.
If I am coming from a login page, this event will take me to the forgot password page

4. Design and Implementation

B. Event-Driven Programming

The second button is the "back" button. in which its special event has been programmed in the "backAction" method. According to the previous page you came from, this event will do one of two tasks:

If I am coming from the Sign Up page, this event will bring me back to the SignUp page.

If I am coming from a login page, this event will bring me back to the Log in page .

On my recipes page, there are two types of events, action event and mouse clicked event.
The first source for Action Event is the text field for the search process. This event is programmed in the "searchResult" method , which in turn takes the name of the recipe entered in the text field and searches for it in the data base based on the category it falls under. If the recipe is found, its name will be displayed under one of the images on this page. If it is not found, a label will appear as "Search results are not available".

The second source of Action Event is the "Add Recipe"button. This event has been programmed in the "goToAddingPage" method. This event will take me to the Adding Recipe page.

4. Design and Implementation

B. Event-Driven Programming

The source for mouse clicked is the four images on the page. This event has been programmed in the 4 method for each image (viewAction1,
viewAction2, viewAction3, viewAction4) which in turn takes the name of the recipe under the image and retrieves the method and ingredients of this recipe from the data base.
This event will take me to view recipe page where the method and ingredients of this recipe are displayed there.

On the view recipe page, there is only one type of event , which is Action Avent. The source for this event is a "back" button. This event has been programmed in "btBack" method where this event will take me back to my recipes page.

In the new recipes page, there is only one type of event, which is Action Event.
The source for this event is the four buttons on this page. Each button is dedicated to a specific section(main dishes,appetizers, drinks, desserts). For each section button, its event is programmed in 4 method (show_drinks_page, show_main_page,show_dessert_page,
show_appetizers_page). The function of this event is to retrieve all the names and links of the recipes listed under this section from the data base and then take me to the display recipe page where all these recipes will be displayed.

4. Design and Implementation

B. Event-Driven Programming

On the display recipe page, there is only one event type, which is Action Event. The source for this event is the "Add" button and a hyperlink.

The event of the add button has been programmed in the " add " method which in turn takes the name and link of the recipe entered in the text field and adds it in the data base and displays the name and link of this recipe inside a box with the rest of the recipes previously stored.

The hyperlink event has been programmed in the "setHyper" method This event will display for you the content of the hyperlink in another webpage.

4. Design and Implementation

B. Event-Driven Programming

login contoller in this page we have two types of events, an action event type which is fired when 'logIn' button is clicked that will transfer you to 'Home page' if the user enterd their account information right. inside the action event a method that checks if all fields are empety or not. then we get the user by their email or username from database, using HQL, we open a session and close it when we use sql queries. then check if it exists or not. if it exists we will check if the enterd password matches the one stored in database.if it matches it will transfer the user to home page, if it doesn't it will display an error message saying the password enterd is wrong. the other event type is a mouse event that's fired when 'goForgetPasswordPage' label is clicked, t it will transfer the user to another page when they can reset their password as it's titled, simlier to 'goSignUpPage' label that trasnfers the user to 'singUp' page to create a new account.

4. Design and Implementation

B. Event-Driven Programming

ShoppingList controller

we have created three methods. the first one overrided method

public void initialize(URL url, ResourceBundle rb) to call other two methods created. the secned method displaysll() that retrive data from database and store in a list type <ShoppingList> so we can add it as label in the Vbox.

the secend method is displayed() that retrive data from database and store in a list type <FoodStuff> so we can add it as label in the Vbox

we have an action type for the button 'tbadd' inside it, checks if the field is empty if it isn't it will check if the inserted item exists in database . if it
doesn't it will add it. if it does it will display a message. lastly an event that displays expired items after comparing the date in databse with current date.

.

4. Design and Implementation

B. Event-Driven Programming

ForgotPasswordPagecontroller

before transfering to this page the user must enter thier email in the 'loginPage' so we can make sure that the user is already registered so we can also send a verification code. we have two events of the type action for two buttons. the first button's 'btchange' method firstly checks if all fields are empty or not. if they are it will display an error messege conveying that. then it will check if the password matches the rules. then we will check if the confirm password field empty or not so can check if it matches the field that the password was first enterd in. if it does we will use HQL to open sessions to update the new password in the database. then we will display a message that it was updated. then the user will be transfered to the 'logInPage'.

the other button 'back' transfers the user to the previous page 'logIn'.

4. Design and Implementation

B. Event-Driven Programming

HomePageController simply has four buttons (Source Object) with ActionEvent on each representing the categories. After clicking on one of them, the program search for recipe which his ingredients available in food stuff then displays on label. If user need other suggestion, he will click on (other Suggest Button). Also, there are five Buttons to navigate between pages. finally, there is a logOut button in top left corner. to log from the user account.

MyKitchenController this page for searchinging and displaying content from the database. if you write in search field will display the results information in the labels. there are 5 button to specific searching. in the top right, there is button to navigating to Kitchen setting page. finally, same as Home Page there are 5 Buttons to navigate between pages

KitchenSettingController There is 3 buttons to determine which operation the user needs to apply in foodstuff. each will display all UI that needs to be done. Update Button, when write food name, if exist , then all field will display its information to help user remember it. Delete button has only one field to write food name. if exist will delete. otherwise, nonexist error will be displayed using label. Addinging will add if all conditions met. finally, the operation will be done if the user click on confirm button then sucessfully message will appears

4. Design and Implementation

B. Event-Driven Programming

AddingRecipeController this page has buttons to choose category, fields to write receipt information. button to add in database after checking all fields. if conditions not met there are a lot of labels will help user.

Navigaiotion class this since there are alot of pages has same button with same action for navigate, we make this class.

4. Design and Implementation

C. Java Database Programming

Data is the most important component of our project and All functionalities depend on it. For example, for suggesting a recipe, the program checks all known recipe ingredients for a user and the foodstuff that is in his kitchen. To keep data persistently we will create a database. This database is a relational database. It consists of six tables to cover all data in our project. First Table is for User, has usual information: name, email and password for signup and login, and has an id to be a foreign key for the rest tables that belong to him. The second table is Recipe, which has recipe information: id, name, method and category. For ingredients, which are multivalve attribute for a recipe, we create individual table for them. The forth table is Foodstuff which will keep all items in the kitchen in sections. It stores an id, name, quantity, unit and expiration date. The fifth table is NewRecipe is for keeping names and links for new recipes that the user plans to learn them. Finally, shoppinglist table will keep all items that users write to buy in the future. There is a UML provided to easy understanding our database.

4. Design and Implementation

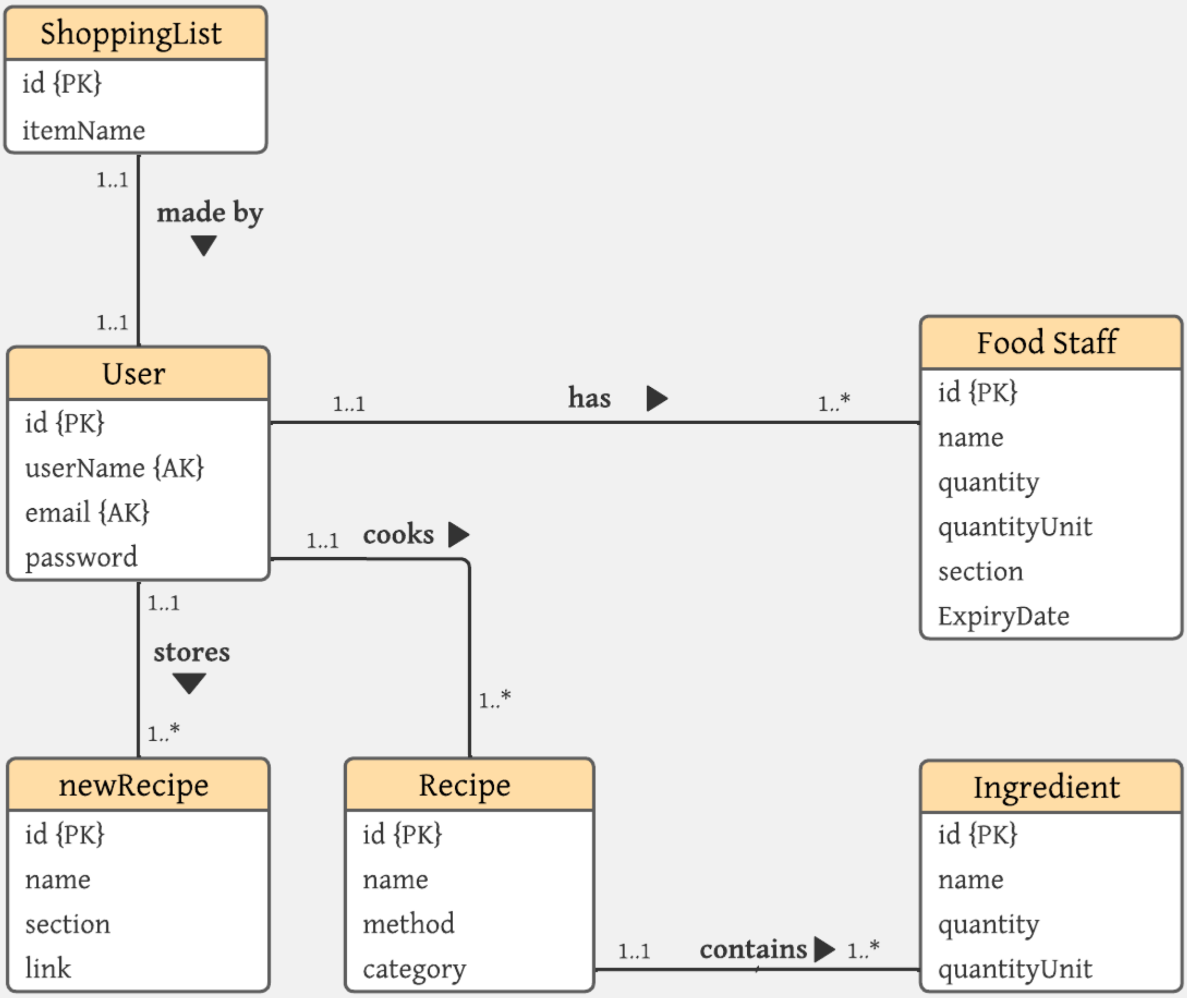
C. Java Database Programming

In our program, we have two types of databases: object model and relational mode. To map between them we use hibernate framework that has great methods for mapping between them. We use a lot of them for different operaions. Firstly, on Sign-up Page, the program check exists of a user and then keeps his information. On Log-in Page, check if the user is the correct person by validating his existing and his password and gets his email if he forgets his password. on Home Page, it checks recipes, ingredients and foodstuff tables to suggest recipes by clicking on the section button and then displaying the recipe name on a label. In My Recipe Page, it uses the database to search, retrieve and add recipes. On My Kitchen Page, which is Foodstuff, uses the database to search, display, update, add and delete the foodstuff using text field UI to get data from the user if needed and labels to display them. In New Recipe Page, will display recipe name under sections on labels after retrieving them from the database and can add through text fields. The last page is Shopping List, the same as New Recipe in retrieving and adding using database, checks for all foodstuff will end using Expire Data Attribute, then display them in labels.

Shortly, the app is nothing without database. Most components in all pages using one or more of methods that deals with database.

4. Design and Implementation

C. Java Database Programming



Maqadeer Databse UML

4. Design and Implementation

D. Model-View Controller

- We use javafx Scene Builder Gluon to write Maqadeer project, it helps us to implement our code in MVC pattern.
- Each scene has FXML file and Java controller, most of UI components are added via FXML file and then they are controlled from java. controller by some ids and functions.
- styling.css is used for some JavaFX-CSS effects.