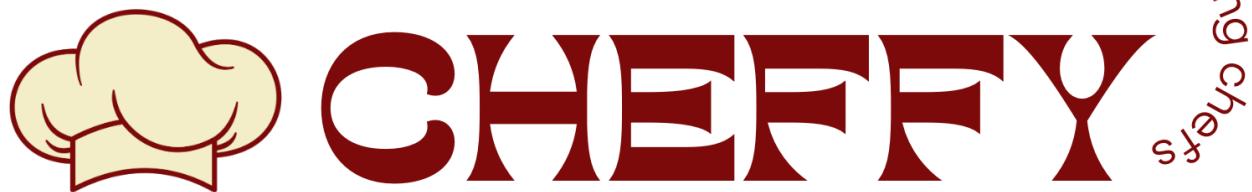




CS 526 Mobile Application Programming
Term 2 – 2023/2024



Supervised by:
L. Hanan K. Aljasim

Group no. 4

Members of the group:

#	Name	ID	Role
1	May M. Alotaibi	2200004606	Leader
2	Ghala M. Alkhaldi	2200003157	Member
3	Jumana Y. Aljassim	2200006269	Member
4	Bushra M. Alshehri	2200004242	Member

Table of Contents

Introduction.....	3
Project Objectives.....	3
Comprehensive List of Application Requirements and Core Functionalities	4
Mobile App User Flow Diagram	6
Mobile App Flowchart.....	7
App Interfaces:.....	9
Both Side (Chefs, Restaurants)	13
Chef Side	19
Restaurants Side	25
Conclusion	35

Table of Figures

Figure 1 User Flow Diagram	6
Figure 2 Splash Interface	9
Figure 3 Onboarding Interfaces	10
Figure 4 Log In Interface	11
Figure 5 Sign Up Interface.....	12
Figure 6 Home Interface	13
Figure 7 Search Interface	16

Introduction

CHEFFY is the culinary world's new hub, uniting talented chefs and visionary restaurants on a single platform. Chefs can showcase their culinary expertise, share signature recipes, and connect with potential employers seeking their unique skills. Meanwhile, restaurant owners and managers can discover top culinary talent, acquire innovative recipes to elevate their menus, and streamline their hiring process. CHEFFY fosters a dynamic community where culinary passions thrive and opportunities abound.

Project Objectives

- Establish a Thriving Culinary Community:**

Connect chefs and restaurants in Saudi Arabia to foster collaboration and growth within the local culinary scene.

- Empower Culinary Professionals:**

Provide a platform for chefs to showcase their skills and connect with exciting collaboration opportunities in Saudi Arabia.

- Enhance Restaurant Operations:**

Help restaurants discover and collaborate with top talent, streamline their recruitment process, and access innovative recipes to enhance their menus.

Comprehensive List of Application Requirements and Core Functionalities

To ensure a successful development process, a thorough understanding of the following requirements is essential:

1. Functional Requirements (What the app does):

- **User Management:**
 - **Chef Profiles:** Create, edit, and view detailed chef profiles with experience, culinary style and ratings.
 - **Restaurant Profiles:** Create, edit, and view restaurant profiles showcasing cuisine type, photos and location.
 - **Authentication/Authorization:** Secure login and registration for chefs and restaurants.
- **Core Features:**
 - **Recipe Marketplace:**
 - Chefs: Create, edit, list recipes with detailed descriptions, ingredients and photos.
 - Restaurants: Browse, search, and filter recipes, send purchase offers, and negotiate terms with chefs via direct messages.
 - **Messaging:**
 - Direct communication between chefs and restaurants regarding recipe offers, or general inquiries.
 - Real-time notifications for new messages and updates.
- **Additional Features:**
 - **Reviews and Ratings:** Allow users to rate chefs, restaurants, and recipes.
 - **Social Sharing:** Enable chefs to share recipes with the community of chefs and restaurants.
 - **In-App Payments:** Integrate payment gateway for recipe purchases.

2. Non-Functional Requirements (How the app performs):

- **Performance:**
 - **Fast Loading Times:** Ensure quick page loads and responsiveness.
- **Usability:**
 - **Intuitive Interface:** Design a user-friendly layout with easy navigation.
 - **Accessibility:** Consider accessibility guidelines.
- **Security:**
 - **Authentication:** Implement strong authentication measures (Goole firebase authentication).
- **Reliability:**
 - **Error Handling:** Implement proper error handling mechanisms to prevent crashes and data loss.

- **Scalability:**
 - **Architecture:** Design the system to accommodate future growth in user base and data.

3. Technical Requirements (Technologies and tools):

- **Programming Language:** Swift
- **Development Environment:** Xcode
- **Database:**
 - Choose a suitable database “Google firebase” for efficient data storage and retrieval.

Mobile App User Flow Diagram

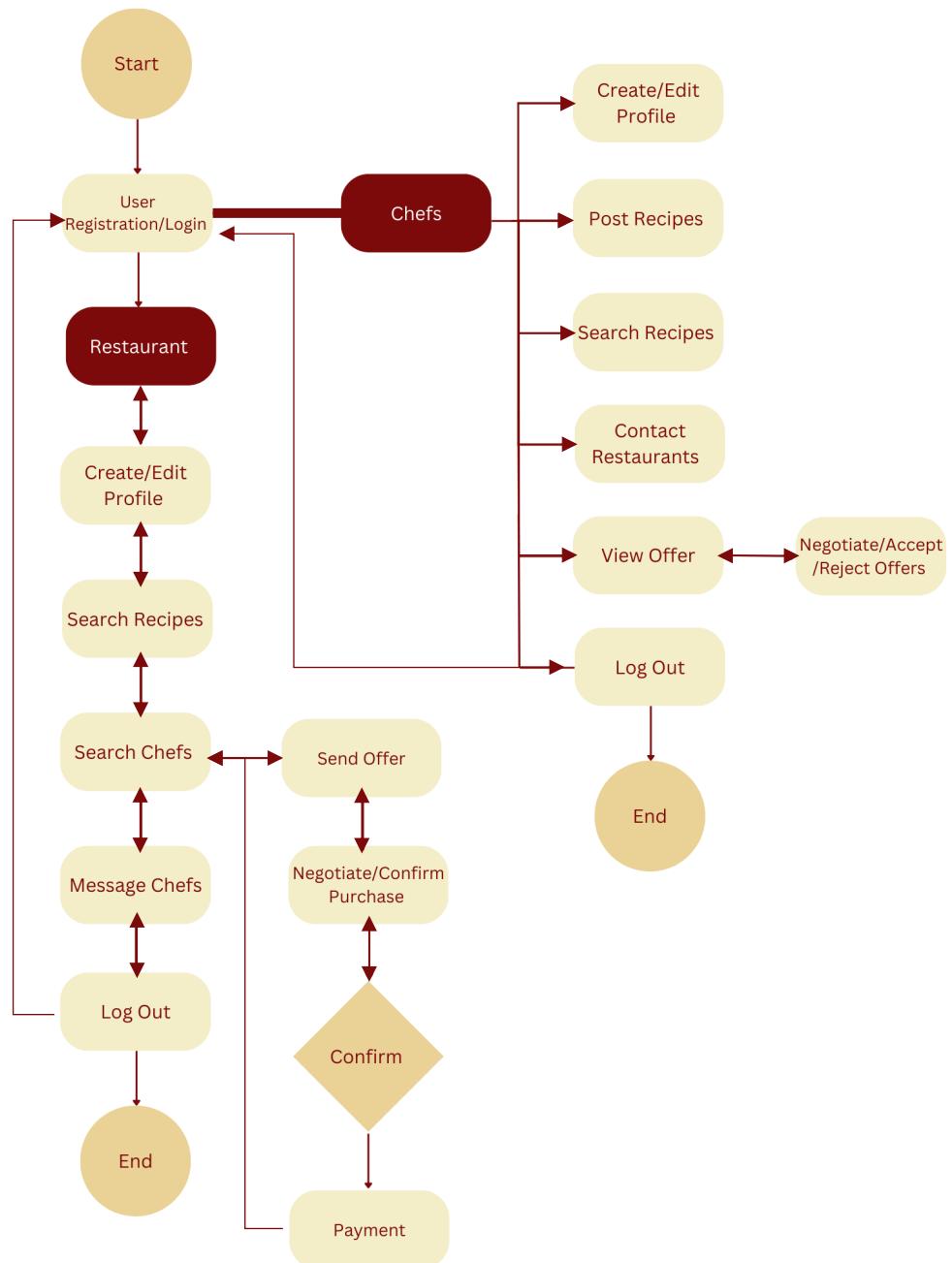
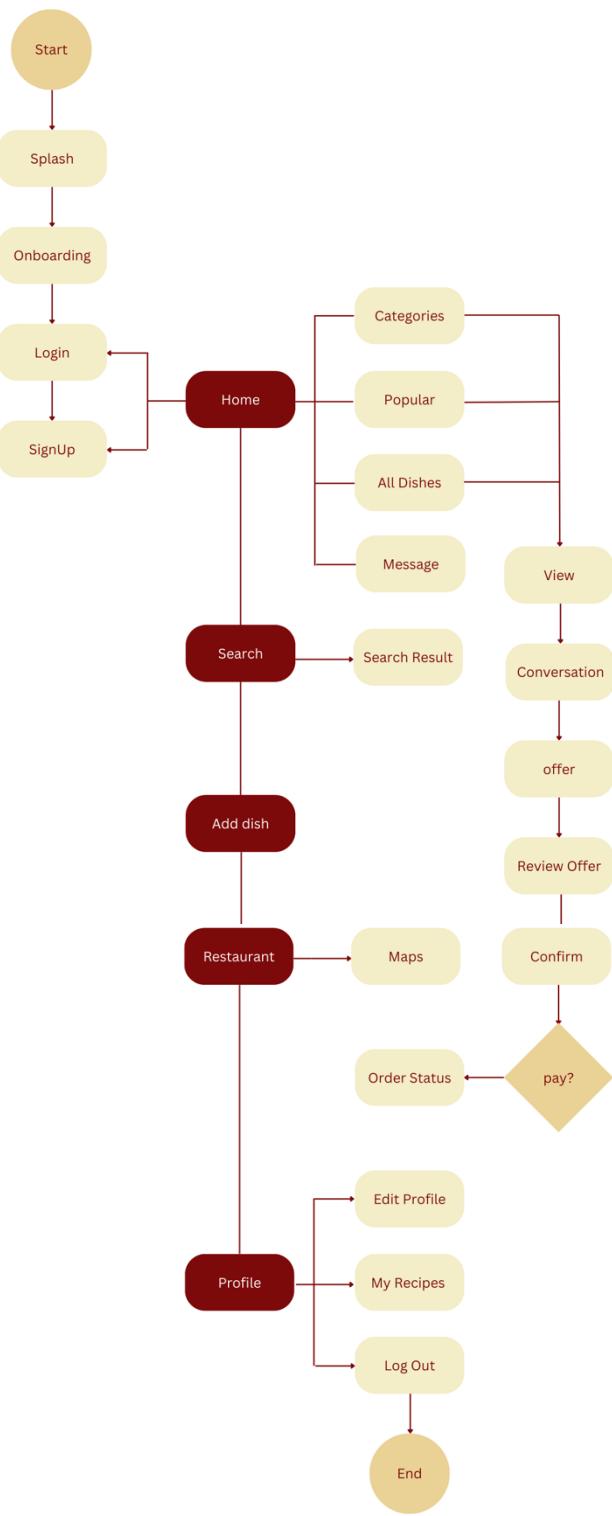


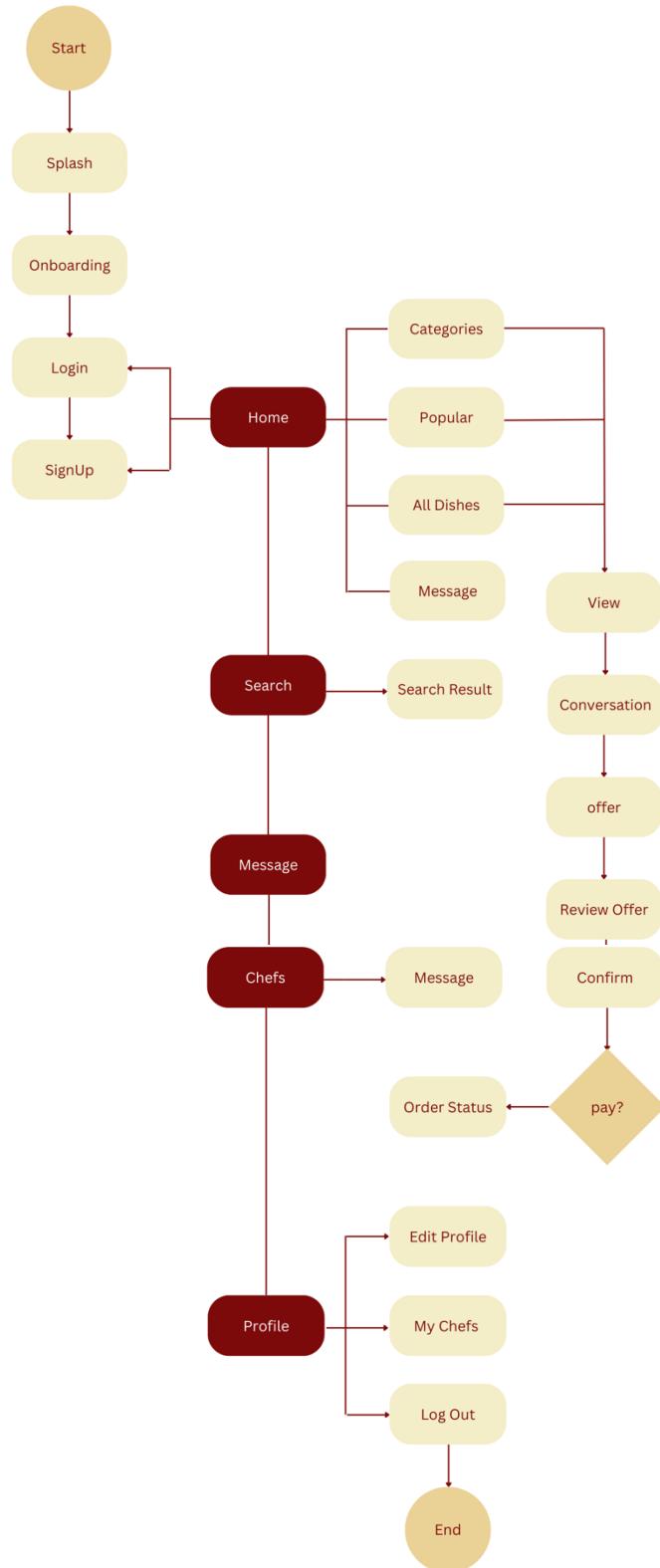
Figure 1 User Flow Diagram

Mobile App Flowchart

Chef:



Restaurant:



App Interfaces:

The following interfaces illustrate the complete user flow of the Cheffy app, showcasing various functionalities that work together to achieve the app's objectives.

Splash interface:

The following image **Figure 2** showcases the splash screen interface that greets users upon app launch for a moment.



Figure 2 Splash Interface

Onboarding interface:

The following image **Figure 3** showcases the onboarding screen interfaces that guide users through the initial steps of using the app.

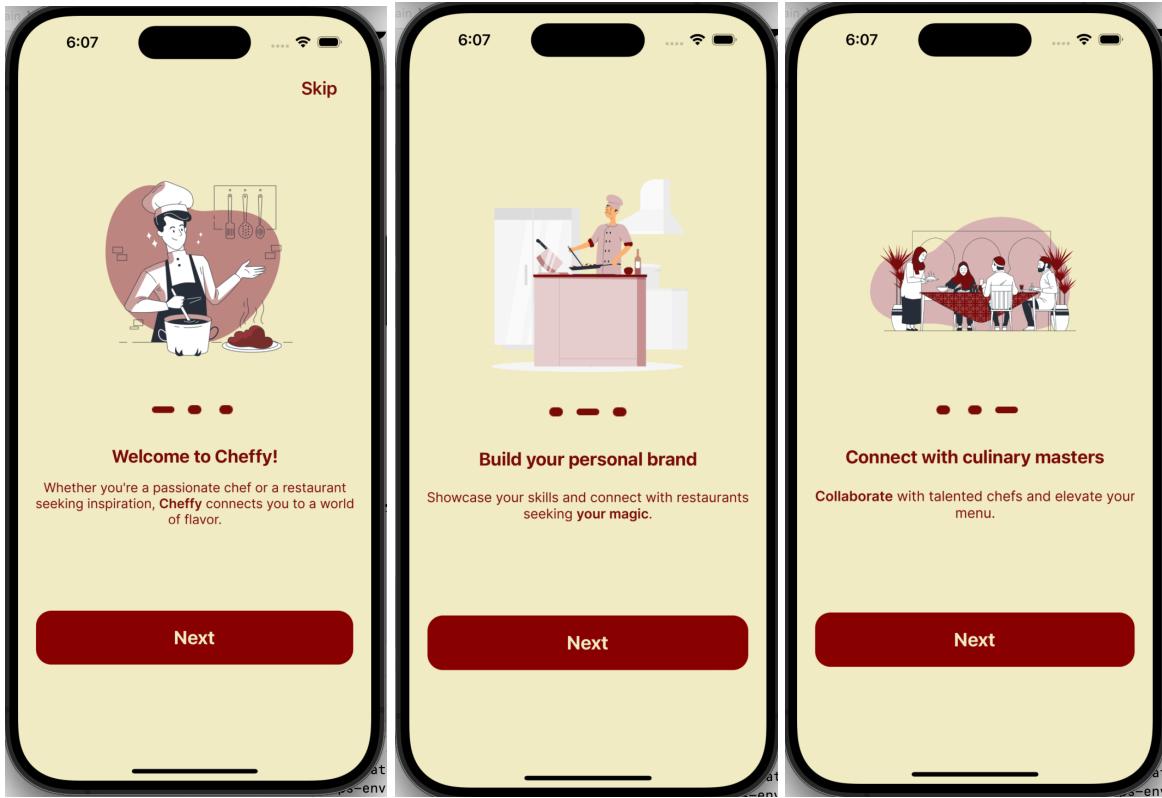


Figure 3 Onboarding Interfaces

Login interface:

The login interface, as shown in Figure 4, serves as a secure gateway for both chefs and restaurants to access their respective accounts within the Cheffy app. By requiring valid credentials, this interface ensures that only authorized individuals can utilize the app's features and functionalities, thus maintaining the platform's integrity and user privacy.

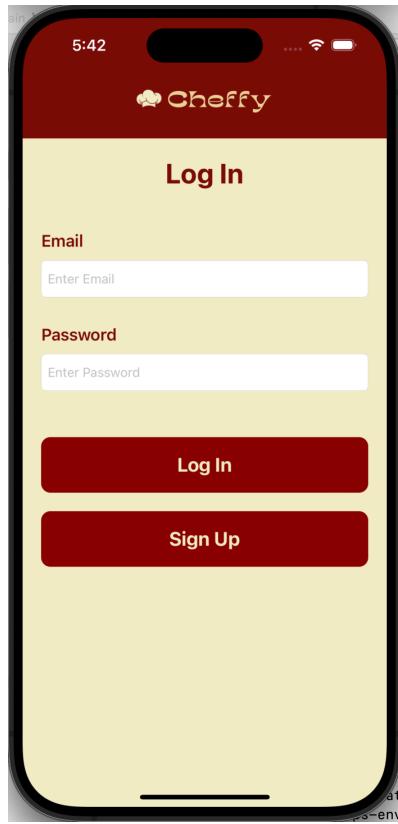


Figure 4 Log In Interface

Sign-Up Interface:

The sign-up interface, displayed in Figure 5, offers an optional pathway for both chefs and restaurants to create new accounts within the Cheffy app. This interface facilitates the registration process, allowing users to input their necessary information and establish their profiles on the platform.

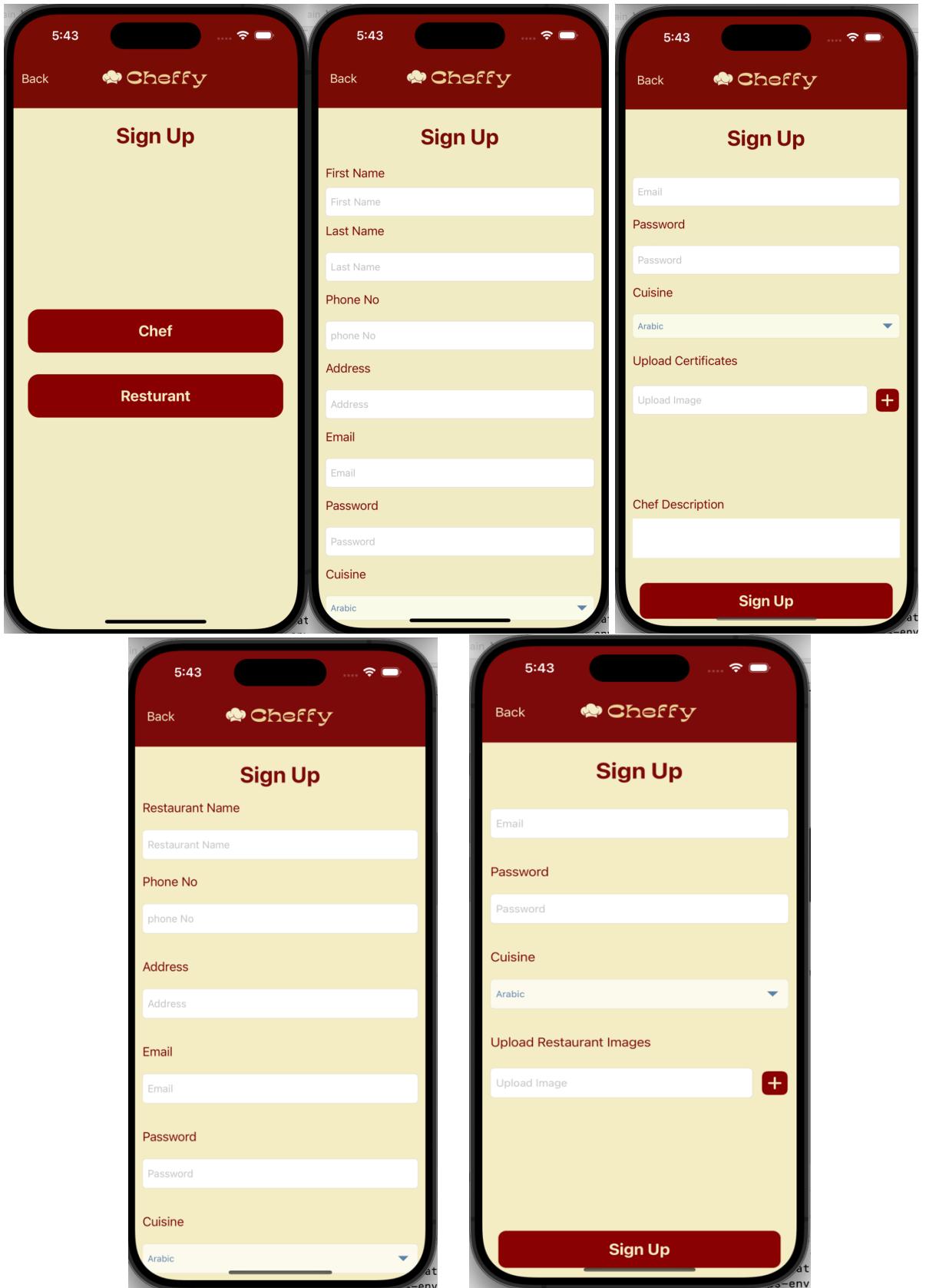


Figure 5 Sign Up Interface

Both Side (Chefs, Restaurants)

Home Interface:

The home interface, as shown in **Figure 6**, serves as the central hub of the Cheffy app. It showcases a curated selection of dishes and restaurants, inviting users to explore diverse culinary options. Popular dishes are displayed alongside their respective ratings, reflecting user preferences and aiding in decision-making. Icons conveniently located at the top of the interface facilitate direct communication with chefs or restaurants, enhancing the user experience and fostering interaction within the app's community.

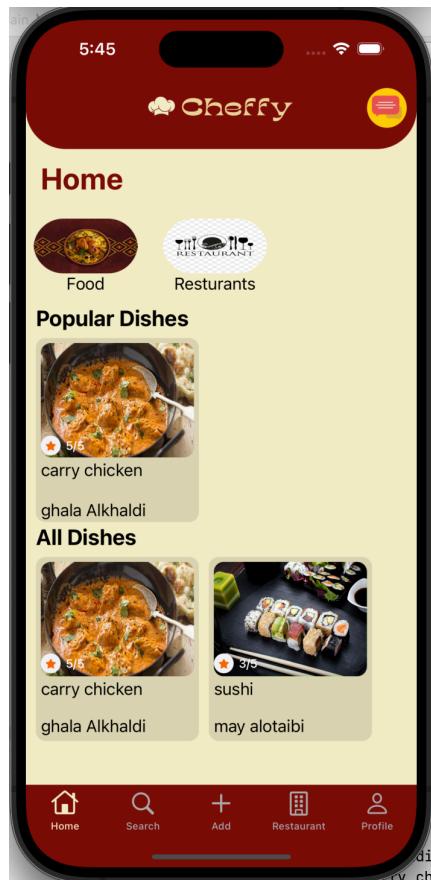


Figure 6 Home Interface

Search Interface:

The search interface, shown in **Figure 7**, is a key feature of the Cheffy app that allows users to easily discover dishes that pique their interest. It prominently features a search bar at the top of the interface, where users can input keywords or specific dish names. Below the search bar, the interface dynamically displays relevant search results, including enticing images of dishes along with the names of the chefs or restaurants that offer them. Users can then conveniently select a dish from the search results to access a more detailed view, which may include additional information about the dish, ingredients, preparation methods, and pricing. Furthermore, the interface provides options for users to directly contact the chef or restaurant responsible for the dish or share their feedback by rating it.

This page code is :

```
extension CHSearchViewController: UISearchBarDelegate {
    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
        if searchText == "" {
            // If search text is empty, show all dishes
            arrfilterDishes = arrObjDishes
            isSearch = false
        } else {
            // Filter dishes based on search text
            arrfilterDishes = arrTempAllDishes.filter {
                ($0.title?.localizedCaseInsensitiveContains(searchText))!
            }
            isSearch = true
        }
        self.tblSearch.reloadData() // Reload table view with
        filtered data
    }
}

// Extension for UITableViewDelegate and UITableViewDataSource
extension CHSearchViewController: UITableViewDelegate,
UITableViewDataSource {
    func tableView(_ tableView: UITableView,
    numberOfRowsInSection section: Int) -> Int {
        return self.arrfilterDishes.count // Return number of
        rows based on filtered data
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

```

```

        let cell =
self.tblSearch.dequeueReusableCell(withIdentifier:
"CHSearchTableViewCell", for: indexPath) as!
CHSearchTableViewCell
        cell.ImgDish.layer.cornerRadius = 10
        let dish = self.arrfilterDishes[indexPath.row]
        let dishImage = dish.selectedImages as!
[CHDishImageModel]
        if dishImage.count > 0
        {
            cell.ImgDish.sd_setImage(with: URL(string:
dishImage[0].imageUrl ?? ""))
        }
        else
        {
            cell.ImgDish.image = UIImage(named: "Dishes")
        }

        cell.lblDishName.text = dish.title
        cell.lblDishCreatedBy.text = dish.postedByName
        return cell
    }

    func tableView(_ tableView: UITableView, heightForRowAt
indexPath: IndexPath) -> CGFloat {
        return 100.0 // Set height for each row
    }

    func tableView(_ tableView: UITableView, didSelectRowAt
indexPath: IndexPath) {
        // Handle row selection
        let vc =
(storyboard?.instantiateViewController(withIdentifier:
"CHDishDetailViewController") as? CHDishDetailViewController) !
        vc.objDish = self.arrfilterDishes[indexPath.item]
        vc.hidesBottomBarWhenPushed = true
        self.navigationController?.pushViewController(vc,
animated: true) // Push to detail view controller
    }
}

```

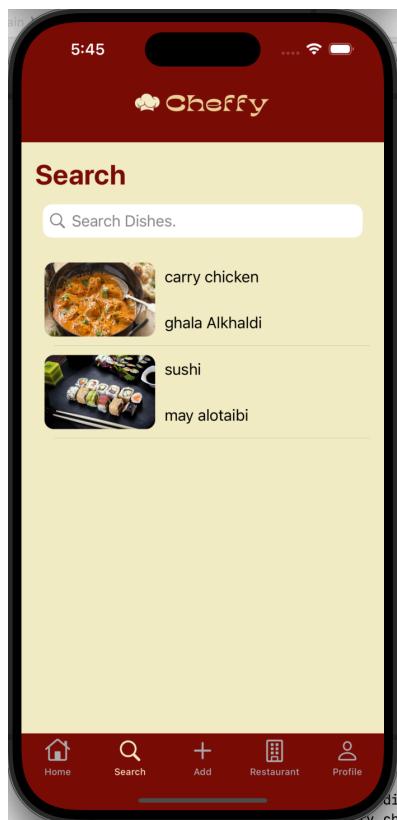


Figure 7 Search Interface

Massages Interface:

The messaging interface, as depicted in **Figure 8**, is a vital component of the Cheffy app, fostering a thriving community by facilitating seamless communication and collaboration among chefs and restaurants. This interface empowers chefs to connect with their peers, enabling them to exchange creative ideas, share valuable culinary insights, and explore potential partnerships that can elevate their culinary journeys. Additionally, chefs can leverage this platform to initiate conversations with restaurants, expressing their interest in the establishments' menus, showcasing their unique skills and expertise, and potentially opening doors to new career opportunities. The messaging interface prioritizes direct and efficient communication, ensuring that users can easily connect and engage with one another, fostering a vibrant and supportive community within the app.

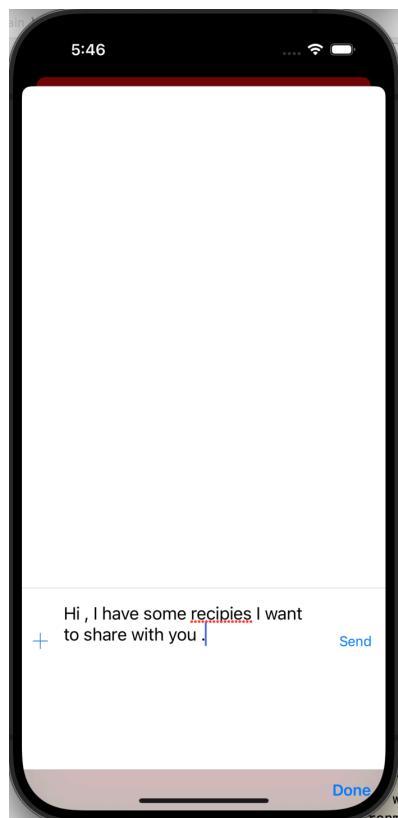


Figure 8 Contact with Restaurants

Log Out:

The **Figure 9** shows the log out interface for the Cheffy app. It includes a confirmation pop-up window asking the user "Are You Sure?" with options to either cancel the logout by tapping "No" or confirm it by tapping "Yes, Log Out.



Figure 9 Log Out

Chef Side

Add Dishes Interface:

The Add Dishes interface, as depicted in **Figure 10**, empowers chefs to effortlessly showcase their culinary creations on the Cheffy platform. Through this interface, chefs can input essential details about their dishes, including the title, an enticing image, a comprehensive description and the cuisine type. Once the information is complete, chefs can add the dish to the app, making it instantly available for users to discover on the Dishes interface or the Home interface. This seamless process ensures that the app's content remains dynamic and up-to-date, reflecting the latest offerings from the culinary community.

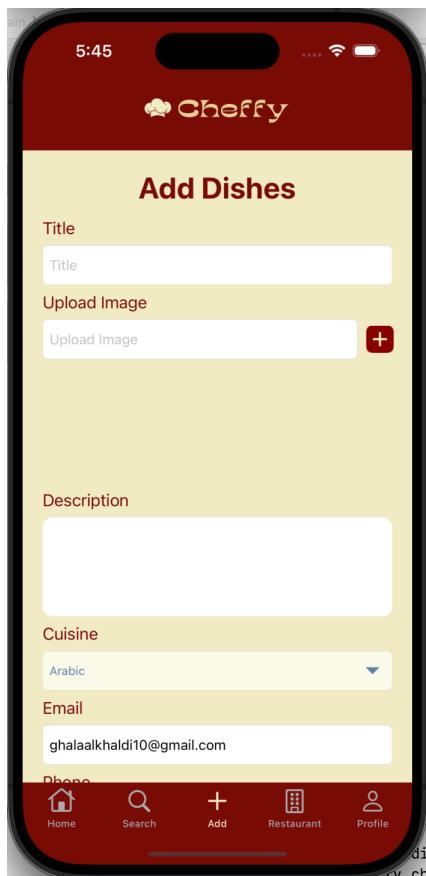


Figure 10 Add Dishes Interface

Profile Interface:

The profile interface, as illustrated in **Figure 11**, serves as a personalized hub for users within the Cheffy app. It offers four primary options:

1. **Edit Profile:** This feature empowers users to update and personalize their profiles by modifying their name, contact information, and profile picture, ensuring that their information remains accurate and up-to-date.

2. **Dishes:** For chefs, this section provides a comprehensive overview of their culinary creations. They can easily add new dishes, make edits to existing ones, or remove dishes from the platform, maintaining full control over their showcased offerings.
3. **My Orders:** This section offers a convenient way for users to track the status of their current orders and access their order history. It provides a consolidated view of all past and ongoing transactions, enhancing transparency and user experience.
4. **Log Out:** The log out option allows users to securely sign out of their Cheffy accounts, safeguarding their personal information and ensuring privacy.

Beyond these core features, the profile interface includes a navigation bar at the bottom, offering seamless access to other essential sections of the app. Users can effortlessly navigate to the Home interface, Search interface, Add interface, Restaurant interface, and return to the Profile interface itself. This intuitive navigation design enhances user experience and ensures smooth interaction with the app's diverse functionalities.

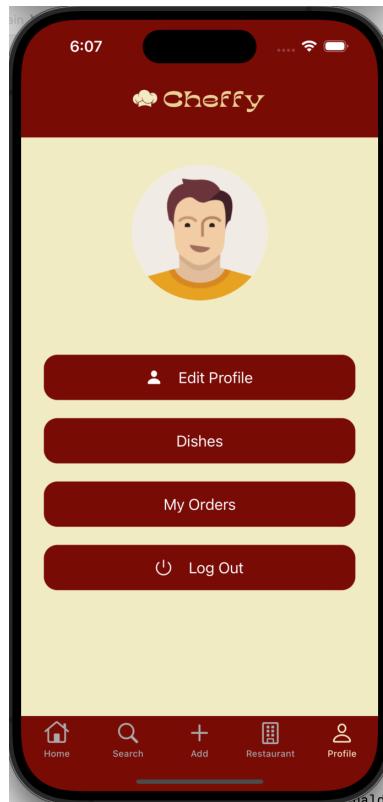


Figure 11 Profile Interface

View Restaurants Interface:

This interface showcases information about a registered restaurant, including its name (Macdonald), address (Al-Asyal Rd., Jalmudah, Al Jubai), and contact number (0530919668). Additionally, the interface provides options to "Show in Map" to view the restaurant's location, "Contact Restaurant" to get in touch with them, and "Cancel" to close the interface.

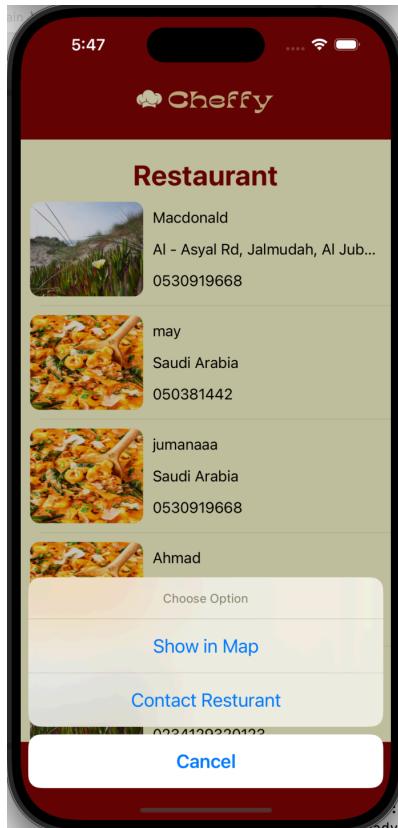


Figure 12 View Restaurants

The (Show in map) Implemented using this code :

```
func showInMap(address:String)
{
    // Format the address to be URL-friendly
    let formattedAddress =
address.addingPercentEncoding(withAllowedCharacters:
.urlQueryAllowed) ?? ""

    // Google Maps URL scheme
    if let googleMapsURL = URL(string:
"comgooglemaps://?q=(formattedAddress)",
UIApplication.shared.canOpenURL(googleMapsURL) {
```

```
// If Google Maps is installed, open the address in
Google Maps
    UIApplication.shared.open(googleMapsURL, options:
[:], completionHandler: nil)
} else {
    // If Google Maps is not installed, open the address
in Apple Maps
    if let appleMapsURL = URL(string:
"http://maps.apple.com/?q=(formattedAddress)") {
        UIApplication.shared.open(appleMapsURL, options:
[:], completionHandler: nil)
    }
}
```

View Chefs Interface:

The Dish Details Interface (**Figure 13**) in the Cheffy app provides a comprehensive view of a specific dish ("sushi"). It features an enticing image of the dish, along with essential details like its name, price ("150SR"), and a brief description ("Salmon sushi with special sauce"). Additionally, the interface showcases the chef's profile picture and name ("May alotaibi"), allowing users to connect the dish with its creator. Interactive buttons for messaging the chef and rating the dish enhance user engagement and feedback. This streamlined design effectively combines visual appeal, essential information, and interactive features to create a seamless user experience within the app.



Figure 13 View Chefs

My Order Interface:

The "My Orders" interface within the Cheffy app allows chefs to view and manage their incoming orders. It features three tabs:

1. **Open Orders:** This tab displays orders that are currently in progress and awaiting preparation by the chef. The interface showcases details of an open order, including a picture of the dish, the dish name ("carry chicken"), cuisine type (Indian), price (66SR), and order date (01/06/24). Chefs have the option to either "Prepare" or "Cancel" the order.
2. **Close Orders:** This tab would display orders that have been successfully completed and delivered to customers.
3. **Cancel Orders:** This tab would display orders that have been canceled by the chef.



Figure 14 My Order Interface

Restaurants Side

Chef information Interface:

The Chef Information Interface, as shown in the **Figure 15**, allows restaurants to view a chef's profile and gather essential information about them. This interface is accessible when a chef contacts a restaurant, providing the restaurant with a convenient way to learn more about the chef's background and culinary expertise.



Figure 15 Chef Information Interface

Dish Details Interface:

The Dish Details Interface (**Figure 16**) in the Cheffy app provides a comprehensive view of a specific dish ("carry chicken"). It features an enticing image of the dish, along with essential details like its name, price ("66SR"), and a brief description ("Carry chicken can serve with rise"). Additionally, the interface showcases the chef's profile picture and name ("ghala Alkhaldi"). Interactive buttons for messaging the chef, rating the dish, and buying the dish enhance user engagement and facilitate transactions. This streamlined design effectively combines visual appeal, essential information, and interactive features to create a seamless user experience within the app.



Figure 16 Dish Details

The (Figure 17) within the Cheffy app is a pop-up window that appears when a user taps the "Rate Dish" button on the Dish Details Interface. It allows users to provide feedback on a specific dish ("carry chicken" in this case).

The interface includes:

- **Dish Image:** A small image of the dish being rated, serving as a visual reminder.
- **Rating Scale:** A 5-point rating scale represented by emoticons, ranging from unhappy to very happy. The user in the image has selected a rating of 5.
- **Submit Button:** A button that allows the user to submit their rating.
- **Close Button (X):** A button to dismiss the rating interface without submitting a rating.

This simple and intuitive interface enables users to quickly and easily rate dishes, contributing to the overall rating displayed on the Dish Details Interface. This feedback mechanism helps other users make informed decisions and provides valuable insights to chefs.

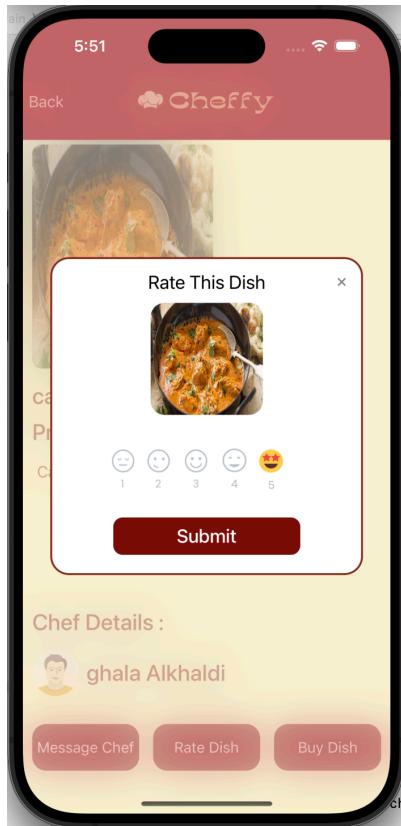


Figure 17 Rate Dish

Payment Details Interface:

The Payment Details Interface, as depicted in the **Figure 18**, is a pop-up window within the Cheffy app's order flow. It facilitates payment processing for users purchasing dishes from chefs. The interface includes the following components:

The rating is implemented using this code :

```
func averageRating(for ratings: [CHDishRatingModel]) -> Int {  
    let totalRating = ratings.reduce(0) { $0 + ($1.rating ?? 0) }  
    return ratings.isEmpty ? 0 : totalRating / ratings.count  
}
```

1. Header:

- **Title:** "Enter Card Details" indicates the purpose of the interface.
- **Close Button (X):** Allows users to exit the payment process if desired.

2. Card Information Fields:

- **Card Number:** A field for users to enter their credit or debit card number.
- **Date of Expiry:** Fields for the card's expiration month and year.
- **CVC:** A field for the card verification code (CVV or CVC).

3. Payment Action:

- **Pay Now Button:** A prominent button that, when tapped, initiates the payment transaction.

4. Additional Information:

- **Chef Details:** Below the payment form, the interface displays the chef's details (name and image) associated with the order.

This interface streamlines the payment process, allowing restaurants to securely enter their card information and complete the transaction within the app, ensuring a seamless purchasing experience.

The enter card conditions is :

```
func isValidInput() -> Bool {
    let txtDate = self.txtExpiryDate.text!
    let month = txtDate.prefix(2)
    let year = txtDate.suffix(2)
    let whitespace = CharacterSet.whitespacesAndNewlines

    // Check for whitespace and valid month/year values
    let rangeCVC: NSRange = (txtCVC.text! as NSString).rangeOfCharacter(from: whitespace)
    let rangeExpMonth: NSRange = (month as NSString).rangeOfCharacter(from: whitespace)
    let rangeExpYear: NSRange = (year as NSString).rangeOfCharacter(from: whitespace)

    if Int(month) ?? 3 > 12 {
        self.showAlertView(title: "", message: "Please enter
valid exp month")
        return false
    } else if Int(year) ?? 12 < 18 {
        self.showAlertView(title: "", message: "Please enter
valid exp year number")
        return false
    } else if rangeCVC.location != NSNotFound {
        self.showAlertView(title: "", message: "Please enter
valid CVC number number")
        return false
    }
}
```

```

        } else if rangeExpMonth.location != NSNotFound {
            self.showAlertView(title: "", message: "Please enter
valid exp month")
            return false
        } else if rangeExpYear.location != NSNotFound {
            self.showAlertView(title: "", message: "Please enter
valid exp year")
            return false
        } else {
            return true
        }
    }

    // Function to show alert messages
    func showAlertView(title: String, message: String) {
        let alertView = UIAlertController(title: title, message:
message, preferredStyle: .alert)
        alertView.addAction(UIAlertAction(title: "Ok", style:
.default, handler: nil))
        present(alertView, animated: true)
    }

    // Text field delegate function to handle character changes
    func textField(_ textField: UITextField,
shouldChangeCharactersIn range: NSRange, replacementString
string: String) -> Bool {
        if textField == self.txtCardNumber { // Check if the
modified textField is the card number field
            // Create a set of characters that are NOT numbers
or "." (the decimal point)
            let nonNumberSet = CharacterSet(charactersIn:
"0123456789.").inverted

            // Formatting card number with spaces every 4 digits
            if (textField.text?.count ?? 0) == 4 && string.count
!= 0 {
                // If there are already 4 digits and the user is
adding more, insert a space
                textField.text = "\(textField.text ?? "")\(" ")"
            } else if (textField.text?.count ?? 0) == 9 &&
string.count != 0 {
                textField.text = "\(textField.text ?? "")\(" ")"
            } else if (textField.text?.count ?? 0) == 14 &&
string.count != 0 {
                textField.text = "\(textField.text ?? "")\(" ")"
            }
        }
    }
}

```

```

        // Allow only numbers and space formatting
        if range.length == 1 { // If the user is deleting a
character, allow it
            return true
        } else if (textField.text?.count ?? 0) < 19 {
            // If the total length is less than 19 (max card
number length with spaces),
            // Check if the entered string ONLY contains
numbers (after removing any non-numeric characters)
            return string.trimmingCharacters(in:
nonNumberSet).count > 0
        } else {
            return false //Don't allow further input if max
length is reached
        }
    } else if textField == txtCVC { // Check if the modified
textField is the CVC field
        let nonNumberSet = CharacterSet(charactersIn:
"0123456789.").inverted

        // Allow only numbers and limit to 3 characters
        if range.length == 1 {
            return true
        } else if (textField.text?.count ?? 0) < 3 {
            return string.trimmingCharacters(in:
nonNumberSet).count > 0
        } else {
            return false
        }
    } else if textField == txtExpiryDate { // Check if the
modified textField is the expiry date field
        let nonNumberSet = CharacterSet(charactersIn:
"0123456789.").inverted

        // Formatting expiry date with a slash after 2
digits
        if (textField.text?.count ?? 0) == 2 && string.count
!= 0 {
            textField.text = "\\(textField.text ?? "")\(/\)"
        }

        // Allow only numbers and slash formatting
        if range.length == 1 {
            return true

```

```

        } else if (textField.text?.count ?? 0) < 5 { // Max
length of MM/YY is 5
            return string.trimmingCharacters(in:
nonNumberSet).count > 0
        } else {
            return false
        }
    } else { // For any other textField, allow input without
restrictions
        return true
    }
}

```

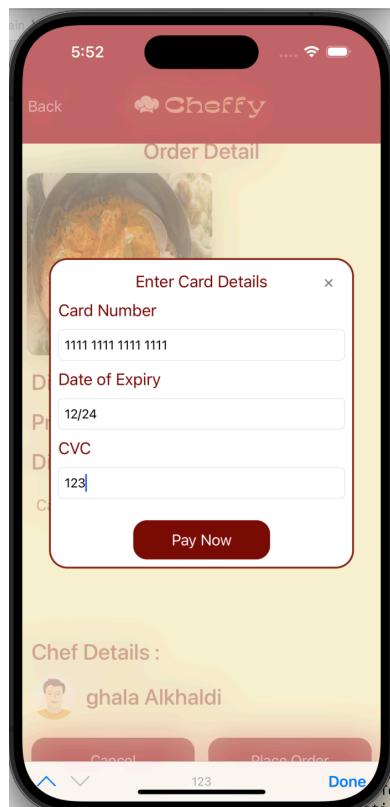


Figure 18 Payment Details Interface

The **Figure 19** shows a notification on the Cheffy app's "Order Detail" screen, confirming that an order has been successfully placed. This notification indicates that the order request has been sent to the chef and will appear in their "Open Orders" tab until they accept it.

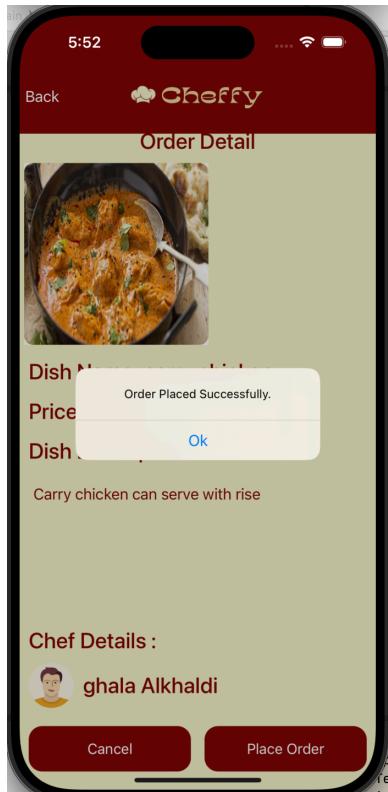


Figure 19 Order Placed Successfully

The **Figure 20** displays the "My Orders" interface within the Cheffy app, specifically the "Open Orders" tab. This interface shows orders that are currently pending, awaiting acceptance by the chef.

Key features of the Pending Orders Interface include:

- **Order Details:** Provides essential information about the pending order, including:
 - **Dish Name:** The name of the dish ordered (e.g., "carry chicken").
 - **Cuisine:** The type of cuisine the dish belongs to (e.g., "Indian").
 - **Price:** The cost of the dish (e.g., "66SR").
 - **Date:** The date the order was placed (e.g., "01/06/24").
- **Status:** Indicates that the order is pending and waiting for the chef's acceptance ("Wait until Chef accepts your order").
- **Image:** A visual representation of the dish.

This interface allows users to easily track the status of their pending orders and provides essential details about the dishes they have ordered. It keeps users informed about the progress of their orders and sets expectations for when the chef will likely accept and start preparing the dish.



Figure 20 My Order Update Interface

Features applied out of Mobile Application Programming content

The Cheffy app was meticulously crafted using the Swift programming language within the iOS ecosystem. This choice ensures a seamless and native experience for iPhone and iPad users. Although mobile app programming courses often focus on Java/Kotlin within Android Studio, Cheffy stands apart by leveraging Swift's modern syntax and performance advantages. Every feature is designed with iOS-specific considerations in mind, optimizing Apple's interface guidelines and user expectations.

Cheffy is a mobile app that extends beyond the traditional curriculum by including features that are not seen in a typical course. Using the potent ChatSDKFirebase library, real-time communication was one of the main goals of this study. Chefs and restaurant owners can communicate instantly owing to this interface, which makes order management and collaboration more effective. Cheffy helps chefs and owners improve their workflow and communication by emphasizing on real-time conversations.

Cheffy app's main feature is its real-time chat feature, which is powered by ChatSDKFirebase. Chefs can confirm orders, schedule deliveries, and respond quickly to client

inquiries. Restaurant owners can quickly get in touch with gifted chefs and extend employment offers to them. This open channel of communication encourages a more flexible and quick thinking method for managing restaurants. Ultimately, both cooks and restaurant owners benefit from the smooth integration of real-time chat, as it makes operations more profitable and efficient.

Conclusion

The Cheffy app interfaces demonstrate a comprehensive and user-friendly design that caters to both chefs and food enthusiasts. From the intuitive onboarding process and secure login to the diverse range of features like searching for dishes, managing orders, and fostering communication, the app aims to create a seamless and engaging experience for all users. The interfaces prioritize showcasing culinary creations, facilitating interaction between chefs and potential customers, and streamlining the ordering process. By incorporating user feedback mechanisms like ratings and reviews, the app encourages a continuous improvement cycle for both chefs and the platform itself.

For a more in-depth look at the app's functionality and codebase, please refer to the additional materials available at the provided link: https://drive.google.com/drive/folders/1An471-Yxkgn5_otkZ9BhayBfL5U0WmIy?usp=drive_link