**Restaurant Reservation System**

**Final Report**

Sharmila Ray

ANALYSIS-AND-DESIGN – SUMMER 2021

**INTRODUCTION**

This document is the culmination of the semester-long project that will be displayed in the three fronts: planning, analysis and design.

The planning feature includes a description of the problem domain, stating that we are creating a high-end restaurant reservation system that eases the process for customers and restaurant employees alike. This system allows for accurate forecasting of inventory and staffing that ultimately saves money for the restaurant.

The analysis feature includes the activity, use case, initial class, and sequence diagrams that depict the connection of the systems within the database. Furthermore, this feature includes the use case descriptions that detail the specifics of each use case and how they are to interact with each other.

The design feature shows the final class diagram, package diagram, database design and data access and manipulation design. Together they show the design modeling of our system containing each method and variable that must be used in order for the system to be functional.

**INDEX**

Category

**Part 1: Planning3**

1.1 Introduction about the problem domain3

1.2 System Request4

1.3 Feasibility analysis5 & 6

1.3 Requirements Definition7 & 8

**Part 2: Analysis9**

2.1 Activity Diagram9

2.2 Use Case Diagram10

2.3 Use Case Description11,12,13,14,15 & 16

2.4 Initial Class Diagram17

2.5 Sequence Diagram18,19,21,22 & 23

**Part 3: Design24**

3.1 Final Class Diagram24 & 25

3.2 Package Diagram26

3.3 Database Design27 & 28

3.4 Data Access and Manipulation Design29

## Part 1: PLANNING

* 1. **Introduction about the problem domain**

As a team at Saffroon, we have analyzed the current restaurant reservation system for this high-end chain restaurant and determined the need for an updated system. It will be used at various locations across the US, with a base system that can be adjusted per each specific restaurants’ needs.

This system will include three fronts/options: mobile app, traditional website booking, and calling the restaurants directly. Calling the restaurants will require an administrative model that allows for creating a reservation for the customer as well as managing previous reservations for any inquiries from customers via phone.

The numbers used to predict how long customers occupy tables and turnover between tables will be based on observations from the current system. However, there will be a predicted period of 90 days in order to account for adjustments to the new system to provide optimal accuracy. With reservations being made exclusively through our system, the restaurant will be able to use forecasted numbers in order to adjust staffing, inventory, and seating arrangements nightly.

Having accurate predictions of expected business, including customers and revenue, will allow for the restaurant to schedule staff as needed. Previous to the system we are creating, the restaurant did not have accurate prediction software in order to create precise scheduling. Additionally, being a reservation-only restaurant with an explicit number of reservations and customers every night allows for inventory to be ordered in unambiguous quantities. There will be no under or overestimation of required stock. Seeing as the reservations will change nightly, so will the seating arrangement. This is to allow for the optimal number of reservations, as well as comfort for the guests upon arrival.

Supplementary features of the system include a cyclical menu with 7 options corresponding to 7 different days of the week and prepay on the system to avoid cancellations or error in the number of reservations.

A rotating menu allows for exclusivity of the restaurant, as well as creating an incentive for customers to come back on various days of the week to taste all 7 menu items.

To avoid cancellations, leading to an error in staffing and inventory, prepay will be required. If a customer would like to cancel within a designated time period, there will be a cancellation fee. This allows for the system to prevent frequent cancellations affecting the daily operations of the system. These features will facilitate an even greater value added to the business based on our restaurant reservation system.

* 1. **System Request**

**Project sponsor:** Vice President of Operations for Restaurant Holding Group, Inc.

**Business need**: This project has been initiated in order to create an application that allows customers to make their own reservations via phone app or web which optimizes internal planning, staffing and the inventory management.

**Business requirements**:

Provide customers incentive to download & make reservations using phone app to view open reservation date/times, menu, make reservations and even prepay using phone app or website. Additionally, restaurant personnel will have access to a backend interface that will allow them to fulfill traditional call-in reservations, optimize seating capacity in concert with staff scheduling, operations planning and adjustments to product stock mins and maxes based on the systems business intelligence and forecasting ability. The functionality that the system should have is as follows:

* Restaurant location selection.
* Customer view of available reservation dates and times.
* Testing Menu.
* Customer online pre-payment for testing menu.
* Customer reservation cancellation feature.
* The above will be available on the phone app and website.

**Business value**:

Potential to increase revenue through customer retention, increased frequency of customer dinning, monetary savings through better staff shift forecasting, better product purchase forecasting leading to timely fresher products, accurate quantities and overall, better internal planning and operations management.

**Special issues**:

From a marketing perspective there is a lot of opportunity to stay in touch with the customers on a regular and even daily basis pushing directly to customer phone apps marketing lunch specials and other promotions that are directed at increasing frequent visitation in addition to evening dining and holiday festivities.

* 1. **Feasibility Analysis**

John Smith and Karry Washington presided over the following feasibility analysis for the Saffroon Online Reservation System Project. A thorough feasibility study was conducted, and the culmination of the feasibility analysis are as follows:

* **Technical feasibility**
* Restaurant reservation systems are very technically feasible and impose low risk.
  + Restaurant reservation systems are one of the most common technologies implemented in restaurants today.
  + There are hundreds of thousands of restaurants across the U.S. who have already implemented a reservation system.
  + There are countless restaurant reservation applications that exist in the marketplace.
* The technology used to create a reservation system is a familiar concept, so it imposes a low risk factor to implement.
  + Although this technology is new to our restaurant and staff, the tools to develop an easy-to-use system are available in the marketplace and have been studied by our staff.
  + Our staff has had personal experience with reservation systems from the customer side since most of our staff use reservation systems in their everyday lives. Operating the reservation system should be uncomplicated since the understanding from our staff and from the customer side allows them to have a better understanding for how to operate and design it from the operating side with the customer’s best interest in mind.
  + Developers who have designed and created reservation systems in their previous projects are readily available and there to assist should there be need to consult those with expertise while developing this system.
* The project size for a restaurant reservation system for few restaurants is small and easy to implement therefore presenting a low risk.
  + The project team will need less than 10 people.
  + Hostesses and managers will be the only staff that need to be trained on how to use and operate this system which should have a short learning curve.
  + Project timeframe will be less than 6 months since this is a very low complex project.
* Integration of reservation systems is not a foreign concept; it is straightforward and uncomplicated for both the user (customer) and the operator (restaurant staff) to operate.
  + Integration will not be conflicted at all since there is no current reservation system in place for this restaurant that could interfere with implementation.
* **Economic Feasibility**

*Intangible Costs and Benefits*

* + This database system will allow our restaurant to compile data to track the restaurant capacity to employees ensuring a profitable ratio of employment of staff to sales.
  + No shows are detrimental to restaurants. No one wants an empty restaurant because customers reserve tables and don’t show up. Restaurants have reported that reservation systems decrease the amount of no shows significantly since app and online features are easy to access to cancel reservations.
  + Reservation systems are important to customers and will help increase the customer retention rate. In a world that is becoming more technology centered, 54% of customers stated that online reservation is the most important technology feature. (Toast, 2019) *2019 Restaurant Success Report*.
* **Organizational feasibility**
  + From an organizational stance, implementing this reservation system poses little risk. This reservation system has a significant value to our establishment as well as to the customers. Our customers expect us to keep up with the demands of being internet accessible as the world moves forward in technological advances. This system will allow us to optimize financially as well as keep up with customer needs and wants which align with the goals of our company.
  + It has been proven that customers want easily accessible online reservation systems, so there is no doubt that users will accept this system. Our management is also on board with implementing this system since it’ll help ease the complications of managing seating as well as staff.
  + The project has a project champion, John Smith, Vice President of Operations for Restaurant Holding Group, Inc., who is more than qualified to sponsor this project and will assist our team as needed. All upper management approve of this project and are eager to have it up and running.
* **Additional Comments** 
  + Our company views implementation of this system as leverage and financial gain for the company. This system will add significant worth and utility to our company by increasing customer satisfaction and practical for everyday use for our staff.
  + Hiring additional staff will not be needed to operate this system. We will use our current staff (managers and hostesses) to operate the system.
  1. **Requirements Definition**

**Nonfunctional Requirements**

1. **Operational Requirements**
   1. The web application should be able to run on Google Chrome, Microsoft Edge,

Mozilla Firefox, or Apple's Safari.

* 1. The phone application should support both Android and iOS.
  2. System backup should be taken every night during maintenance window.

1. **Performance Requirements**
   1. System should be highly reliable and should not crash.
   2. System should be retrievable.
2. **Security Requirements**
   1. User should be able to select unique user ID and Password.
   2. For each location there should be unique login account for the restaurant manager.
   3. The restaurant manager’s login account should be highest priority.

1. **Cultural Requirements**

4.1. No special cultural and political requirement are assumed.

**Functional Requirements**

1. **Manage Login**
   1. Customer should be able to select location from the landing page.
   2. Customer can login using existing ID / Password, create new ID password or can

use guest login (with valid email and phone number).

* 1. Manager should be able to login using admin account and select / check the total

number of appointments on a given date.

1. **Reservation Handling**

2.1. After successful login user (Customer / Reservation assistant) should be able to

create new reservation, modify, and cancel existing reservation.

2.2. Customer should be able to enlist in a waitlist on a given date / time.

2.3. Manager should be able to view / modify reservations on a given date.

2.4. Manager should be able to dynamically handle cancellation and clear waitlist.

2.5. The software should be able to handle cancellation and waitlist confirmation till

the previous day of the reservation date with automated email confirmation and

waitlist updates.

2.6. Chef should be able to view reservations on a given date and plan to refresh

inventory.

1. **Payment Handling**

3.1. User (Customer / Reservation assistant) should be able to complete the payment

for the reservation successfully if all credentials are provided correctly.

3.2. Customer should be able to get refund if reservation is cancelled within the

cancellation window.

3.3. Manager should be notified of each financial transaction for proper accounting

and bookkeeping purposes.

1. **Display Tasting Menu**

4.1. User (Customer / Reservation assistant) should be able to view the tasting menu

on a given date and location if it is available and complete a reservation or get into a waitlist.

4.2. Manager should be able to upload and view the tasting menu on a given date and

location.

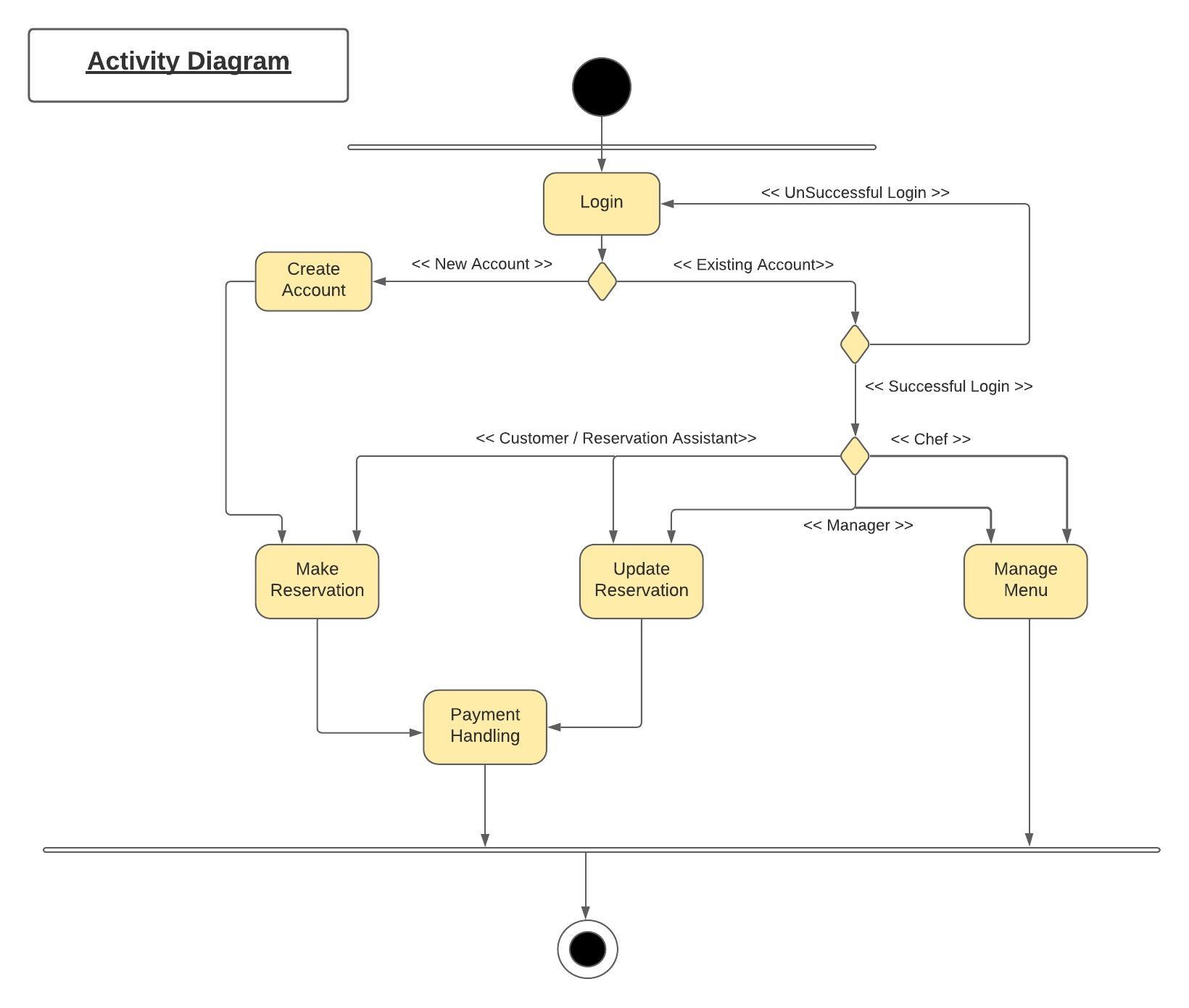
4.3. Chef should be able to upload and view the tasting menu on a given date.

**Part 2: ANALYSIS**

**2.1 Activity diagram**

The first part of our analysis is illustrated in the Activity Diagram shown below.

Four types of users can access the system:Customer, Reservation Assistant, Manager, and Chef. Based on the login permission, upon successful Login, possible options include Make Reservation, Payment Handling, Update Reservation and Manage Menu. For new accounts, the user can Create Account proceed to Make Reservation and subsequently Payment Handling activity.

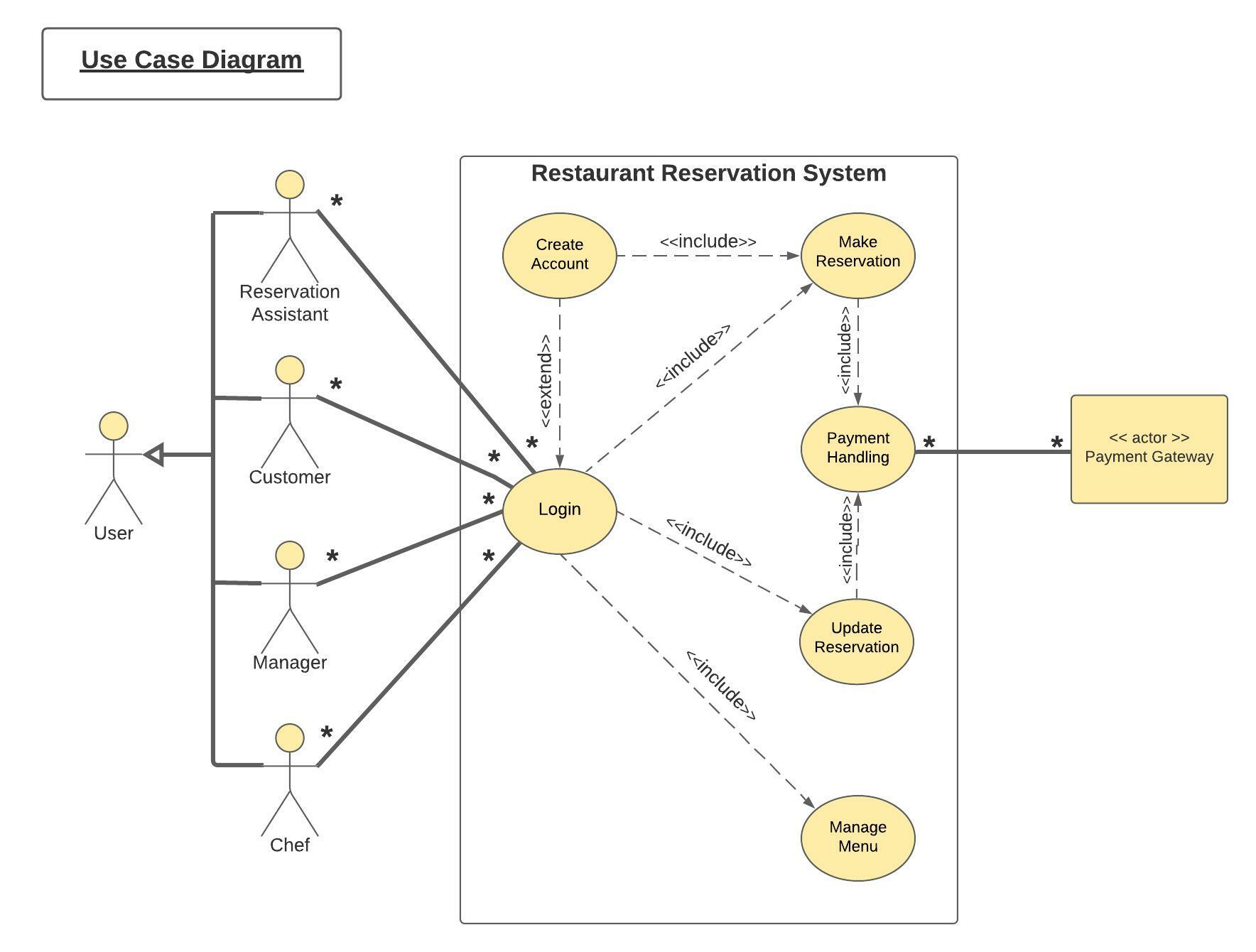


## 2.2 Use Case Diagram

We designed six Use Cases for the Restaurant Reservation System:

* Login
* Create Account
* Make Reservation
* Payment Handling
* Update Reservation
* Manage Menu

The Use Cases Relationships are illustrated in the Use Case Diagram below.



## Use Case Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case Name:** Login | | **ID:** 1 | **Importance Level:** high |
| **Primary Actor:** User (Customer, Manager, Chef and Reservation Assistant (RA)) | **Use Case Type:** Detail, essential | | |
| **Stakeholders and Interests:**  Customer – wants to access Create Account, Make Reservation and Update Reservation modules.  Manager - wants to access Update Reservation and Manage Menu modules.  Chef – wants to access Manage Menu module.  Reservation Assistant – wants to access Create Account, Make Reservation and Update Reservation modules. | | | |
| **Brief Description:** This use case describes how the Primary Actors log into the system and access desired modules. For phone-in reservations, the customer must provide their email/phone number to the Reservation Assistant. | | | |
| **Trigger:** The Primary Actors click on the “Login” button.  **Type:** External | | | |
| **Relationships:**  **Association:** Customer, Manager, Chef and Reservation Assistant  **Include:** Make Reservation, Update Reservation, Manage Menu  **Extend:** Create Account  **Generalization:** | | | |
| **Normal Flow of Events:**   1. The Customer/Reservation Assistant/Manager/Chef clicks on the “Login” button in home page. 2. The Customer/Reservation Assistant/Manager/Chef enters their existing login username and password. 3. The user id and password are validated, and the correct user type is verified to be either Customer, Reservation Assistant/Manager or Chef. 4. The verified Customer/Reservation Assistant with Guest Login may choose sub flow S-1, the verified. Manager may choose sub flow S-2 and the verified Chef may choose sub flow S-3. 5. The Customer/Manager/Chef may choose to logout at any point which will return them to the Login page. | | | |
| **Sub flows:**  **S-1:** Login as verified Customer/Reservation Assistant:  The user sees a page showing “Make Reservation” and “Update Reservation” buttons.  **S-2:** Login as verified Manager:  The user sees a page showing “Update Reservation” and “Manage Menu” buttons.  **S-3:** Login as verified Chef:  The user sees a page showing “Manage Menu” button. | | | |
| **Alternate/Exceptional Flows:**   * 1. If the Customer doesn’t have an existing account, direct to “Create Account” button from Login page.   3.1: If the user ID or password is wrong, the system shows Login error message and is returned to home page.  4.1: If the phone-in customer doesn’t have an existing account, the Reservation Assistant should click the “Create Account” button and proceed. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case Name:** Create Account | | **ID:** 2 | **Importance Level:** high |
| **Primary Actor:** Customer and Reservation Assistant (RA) | **Use Case Type:** Detail, essential | | |
| **Stakeholders and Interests:**  Customer – wants to create a new account and access Make Reservation module.  Reservation Assistant – wants to create a new guest-login account for phone-in customer valid email/phone number and access “Make Reservation” module. | | | |
| **Brief Description:** This use case describes how the Primary Actors access desired modules. For phone-in reservations, the customer must have provided their valid email/phone number to the Reservation Assistant. | | | |
| **Trigger:** The Primary Actors clicks on the “Create Account” button  **Type:** External | | | |
| **Relationships:**  **Association:** Customer and Reservation Assistant  **Include:** Make Reservation  **Extend:**  **Generalization:** | | | |
| **Normal Flow of Events:**   1. The Customer/Reservation Assistant access the system as verified user. 2. The Customer/Reservation Assistant clicks on “Create Account” button. 3. This opens a blank record for the Customer/Reservation Assistant to fill in the new account related information (e.g., Name, DOB, Address, Email ID, Phone Number). 4. The Customer/Reservation Assistant types in the new username and password. 5. The Customer/Reservation Assistant clicks on “Submit” button to complete the account creation step. This will add the new account to the database. 6. The Customer/Reservation Assistant clicks “Make Reservation” button after successful creation of a new regular/phone-in customer account OR may choose to go back to home page / logout. | | | |
| **Sub flows:** | | | |
| **Alternate/Exceptional Flows:**   * 1. If the Customer does not want to create an account, allow them to access “Make Reservation” with a unique guest-login session ID created using their valid email/phone number. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case Name:** Make Reservation | | **ID:** 3 | **Importance Level:** high |
| **Primary Actor:** Customer and Reservation Assistant (RA) | **Use Case Type:** Detail, essential | | |
| **Stakeholders and Interests:**  Customer – wants to make a new reservation in the restaurant.  Reservation Assistant – wants to make a new reservation in the restaurant for a phone-in customer. | | | |
| **Brief Description:** This use case describes how the Primary Actors complete a restaurant reservation. For phone-in reservations, the customer must have provided their valid email/phone number to the Reservation Assistant | | | |
| **Trigger:** The Primary Actors click on the “Make Reservation” button  **Type:** External | | | |
| **Relationships:**  **Association:** Customer, Reservation Assistant  **Include:** Payment Handling  **Extend:**  **Generalization:** | | | |
| **Normal Flow of Events:**   1. The Customer/Reservation Assistant checks the available reservation slots on a given date/time, for the total number of guests to be added to the reservation. 2. The Customer/Reservation Assistant selects an available slot. 3. The Customer/Reservation Assistant completes the reservation by clicking the “Complete Reservation” button which issues a call to “Payment Handling” module. | | | |
| **Sub flows:** | | | |
| **Alternate/Exceptional Flows:**   * 1. The Customer chooses not to make any reservation (e.g., due to unavailability of suitable slots).   2. The Customer chooses to enlist in a wait list.   3. The phone-in customer chooses not to make any new reservation (e.g., due to unavailability of suitable slots) and the Reservation Assistant aborts the step.   4. The phone-in customer chooses to enlist in a wait list and the Reservation Assistant adds to the wait list. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case Name:** Update Reservation | | **ID:** 4 | **Importance Level:** high |
| **Primary Actor:** Customer, Manager and Reservation Assistant (RA) | **Use Case Type:** Detail, essential | | |
| **Stakeholders and Interests:**  Customer – wants to update an existing reservation in the restaurant.  Manager – wants to update an existing reservation in the restaurant.  Reservation Assistant – wants to update an existing reservation in the restaurant for a phone-in customer. | | | |
| **Brief Description:** This use case describes how the Primary Actors updates an existing reservation. For phone-in reservations, the customer must have provided their valid email/phone number to the Reservation Assistant. | | | |
| **Trigger:** The Primary Actors click on the “Update Reservation” button after successful login.  **Type:** External | | | |
| **Relationships:**  **Association:** Customer, Manager and Reservation Assistant  **Include:** Payment Handling  **Extend:**  **Generalization:** | | | |
| **Normal Flow of Events:**   1. The Customer/Manager/Reservation Assistant checks existing reservation(s) related to the regular/phone-in customer. 2. The Customer/Manager/Reservation Assistant modifies/cancel the reservation via the “Modify Reservation” or “Cancel Reservation” button resulting in sub flow S-1 or S-2. | | | |
| **Sub flows:**  **S-1:** Modify Reservation   1. Open the availability table to check the alternate slots available for accommodating required number of guests on a different date/time. 2. Select a suitable slot and confirm the modified reservation.   **S-2:** Cancel Reservation   1. Cancellation would trigger a refund via a call to “Payment Handling”. 2. Once refund is successful, clear up the cancelled slot and make that slot available. 3. The Manager checks the wait list, change the appropriate number of waitlisted reservations to confirmed and clear those positions in the wait list. | | | |
| **Alternate/Exceptional Flows:**   * 1. The Customer chooses not to modify/cancel any existing reservation and goes back to home page.   2. The phone-in customer chooses not to modify/cancel any existing reservation and the Reservation Assistant aborts the step and goes back to home page. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case Name:** Payment Handling | | **ID:** 5 | **Importance Level:** high |
| **Primary Actor:** Customer, Manager, Reservation Assistant (RA) and Payment Gateway | **Use Case Type:** Detail, essential | | |
| **Stakeholders and Interests:**  Customer – wants to confirm a new reservation by completing the payment OR cancel an existing reservation in the restaurant and get back the refund amount  Manager – wants to cancel an existing reservation in the restaurant and process refund for the customer  Reservation Assistant – wants to confirm a new reservation by completing the payment OR cancel an existing reservation in the restaurant and process refund for a phone-in customer | | | |
| **Brief Description:** This use case describes how the Primary Actors interacts with the system to complete a payment or issuing a refund. For phone-in reservations, the customer must have provided their valid email/phone number to the Reservation Assistant | | | |
| **Trigger:** The Primary Actors click on the “Complete Reservation” in Use Case #3 or “Cancel Reservation” in Use Case #4  **Type:** External | | | |
| **Relationships:**  **Association:** Customer, Manager, Reservation Assistant and Payment Gateway  **Include:**  **Extend:**  **Generalization:** | | | |
| **Normal Flow of Events:**   1. In case of making a new reservation, the Customer/Reservation Assistant fills up appropriate payment information and completes the payment via the Payment Gateway. 2. On successful payment, the call goes back to the “Make Reservation” module and confirms the reservation. 3. In case of cancelling an existing reservation, the Customer/Manager/Reservation Assistant triggers the refund via the Payment Gateway. 4. On successful refund issued, the call goes back to the “Update Reservation” module and cancels the reservation. | | | |
| **Sub flows:** | | | |
| **Alternate/Exceptional Flows:**   * 1. In case of error in payment process (e.g., due to credit card declined, wrong payment info provided), abort the payment process and go back to the calling process to start the reservation process again.   2. In case of error in refund process (e.g., due to timeout occurring with payment gateway etc.), abort the refund process and go back to the calling process to start the cancellation process again. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-Case Name:** Manage Menu | | **ID:** 6 | **Importance Level:** high |
| **Primary Actor:** Chef and Manager | **Use Case Type:** Detail, essential | | |
| **Stakeholders and Interests:**  Chef – wants to upload and view the tasting menu as well as check reservation status on a given date and plan  to refresh the pantry accordingly.  Manager – wants to upload and view the tasting menu as well as check reservation status on a given date. | | | |
| **Brief Description:** This use case describes how the Primary Actors can upload/view the menu as well as check the reservation status of the restaurant on a given date. | | | |
| **Trigger:** The Primary Actors click on the “Manage Menu” button.  **Type:** External | | | |
| **Relationships:**  **Association:** Chef and Manager  **Include:**  **Extend:**  **Generalization:** | | | |
| **Normal Flow of Events:**   1. The Manager/Chef clicks on the “View Reservation” button and selects a date for which they want to view the reservation status. Refer to sub flow S-1 for the next step. 2. The Manager/Chef clicks on the “View Menu” button and selects a date for which they want to view the menu. Refer to sub flow S-2 for the next step. 3. The Manager/Chef clicks on the “Update Menu” button and selects a date for which they want to update the menu. Refer to sub flow S-3 for the next step. | | | |
| **Sub flows:**  **S-1:** View Reservation   1. The reservation status on the selected date is displayed in a separate window. 2. The user clicks “Back” to go back to the “Manage Menu” page.   **S-2:** View Menu   1. The menu on the selected date is displayed in a separate window. 2. The user clicks “Back” to go back to the “Manage Menu” page.   **S-3:** Update Menu   1. The user sees a link to upload the menu from a storage location in the device/cloud. 2. The user clicks “Back” to go back to the “Manage Menu” page. | | | |
| **Alternate/Exceptional Flows:**   * 1. If the display of reservation status runs into some errors e.g., database access issues, the system would pop up an Error window and will automatically take the user back to the “Manage Menu” page.   2. If the display of menu runs into some errors e.g., database access issues, the system would pop up an Error window and will automatically take the user back to the “Manage Menu” page.   3. If uploading the menu runs into some errors e.g., database access issues, the system would pop up an Error window and will automatically take the user back to the “Manage Menu” page. | | | |

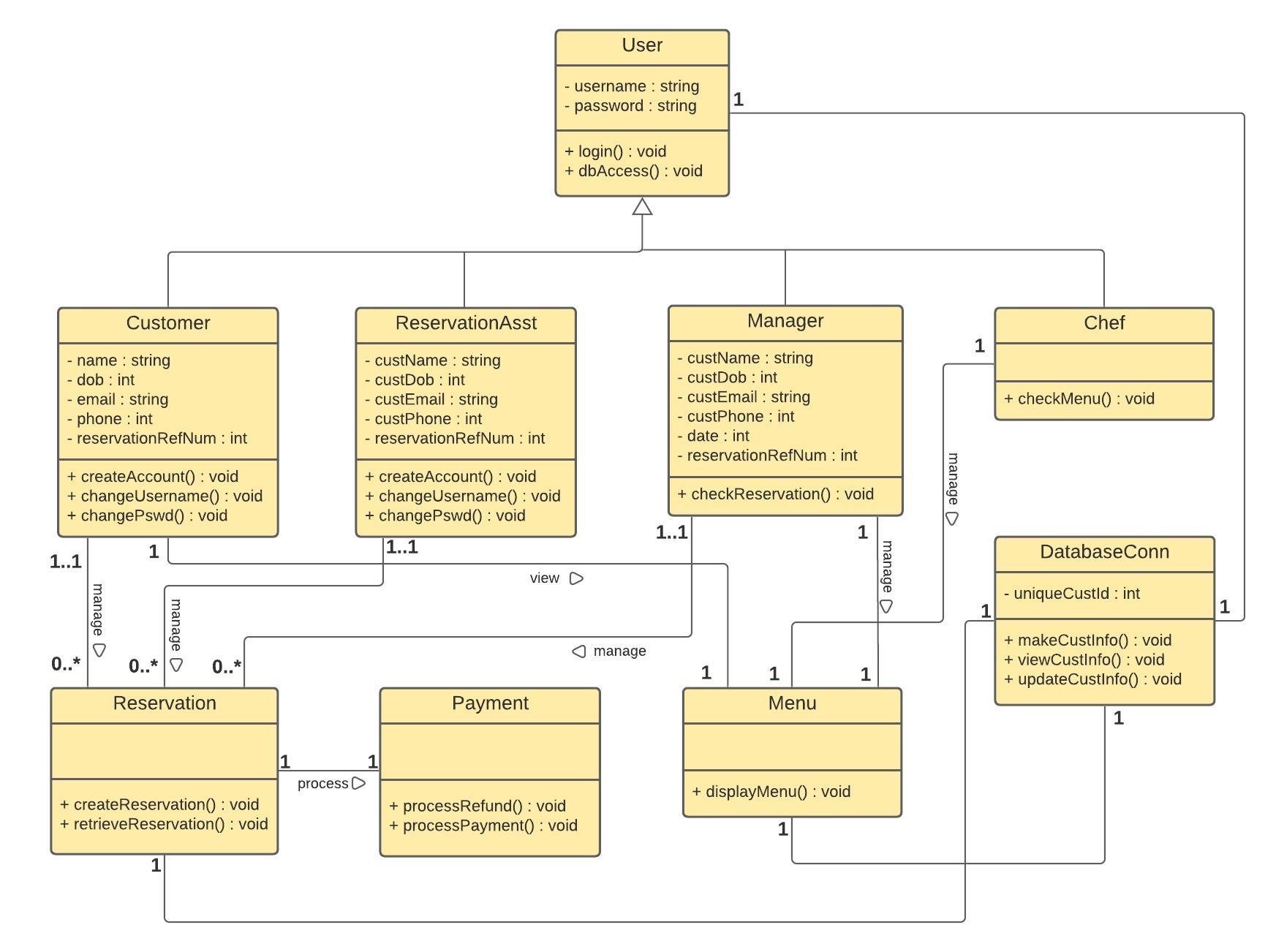
## Initial Class Diagram

This is our initial Class Diagram. We have produced nine classes for our project: User Class, Customer Class, Reservation Assistant Class, Manager Class, Chef Class, Reservation Class, Payment Class, Menu Class, and DBConnection Class.

There is a generalization relationship between User Class, Customer Class, Reservation Assistant Class, Manager Class, and Chef Class. Reservation Class and Payment Class are special cases of Customer and Reservation Assistant Classes. The Menu is a is a special case of Customer and Chef Class.

The Customer Class and Reservation Assistant Class can create user accounts, make reservations, change reservations, and take payments, Manager Class can check reservations and update the menu, and the Chef Class can update the menu.

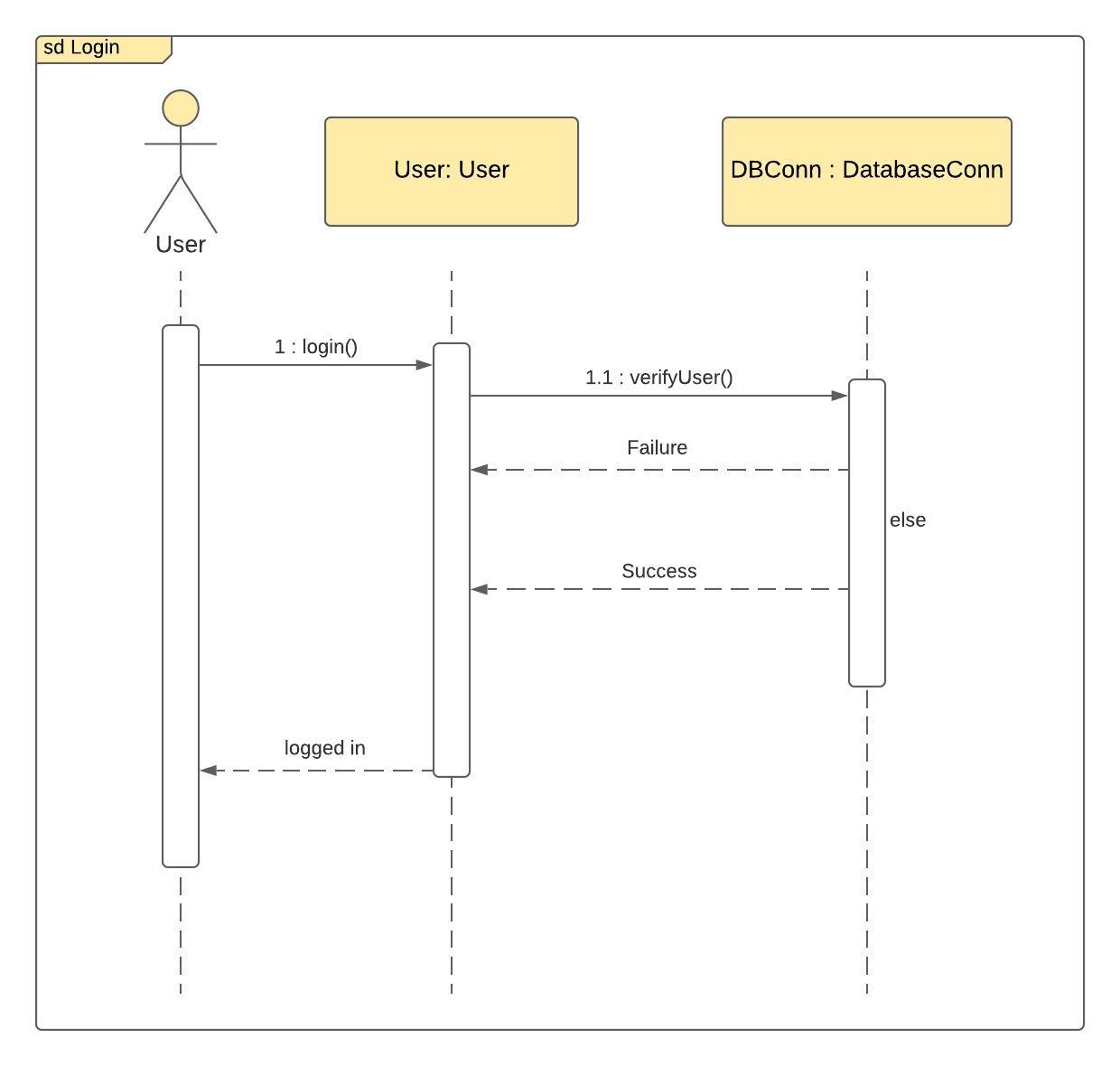
DBConnection class is used to connect the system with the database



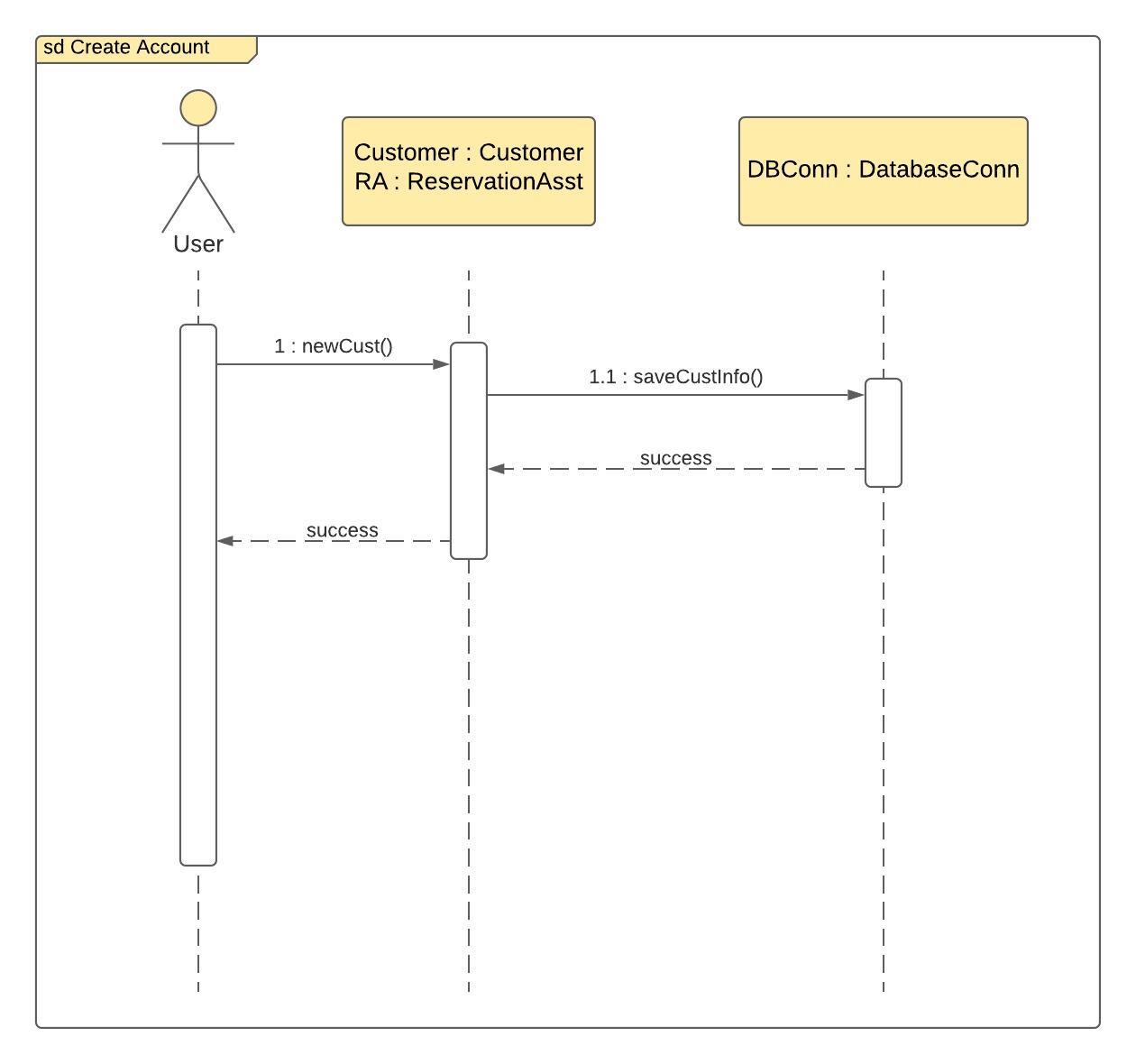
## 2.5 Sequence Diagrams

The sequence diagrams corresponding to each use case are shown below.

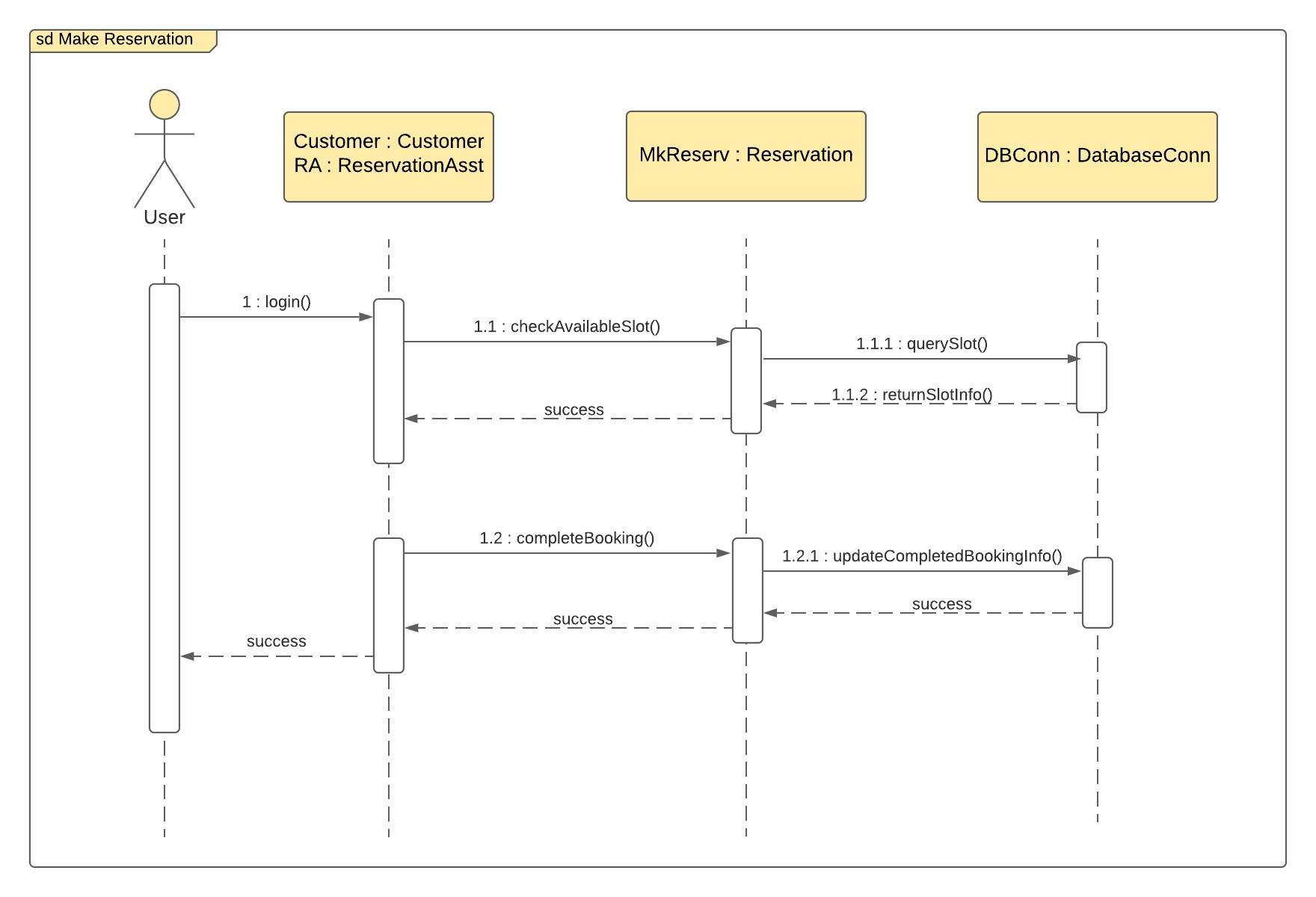
**Login**:



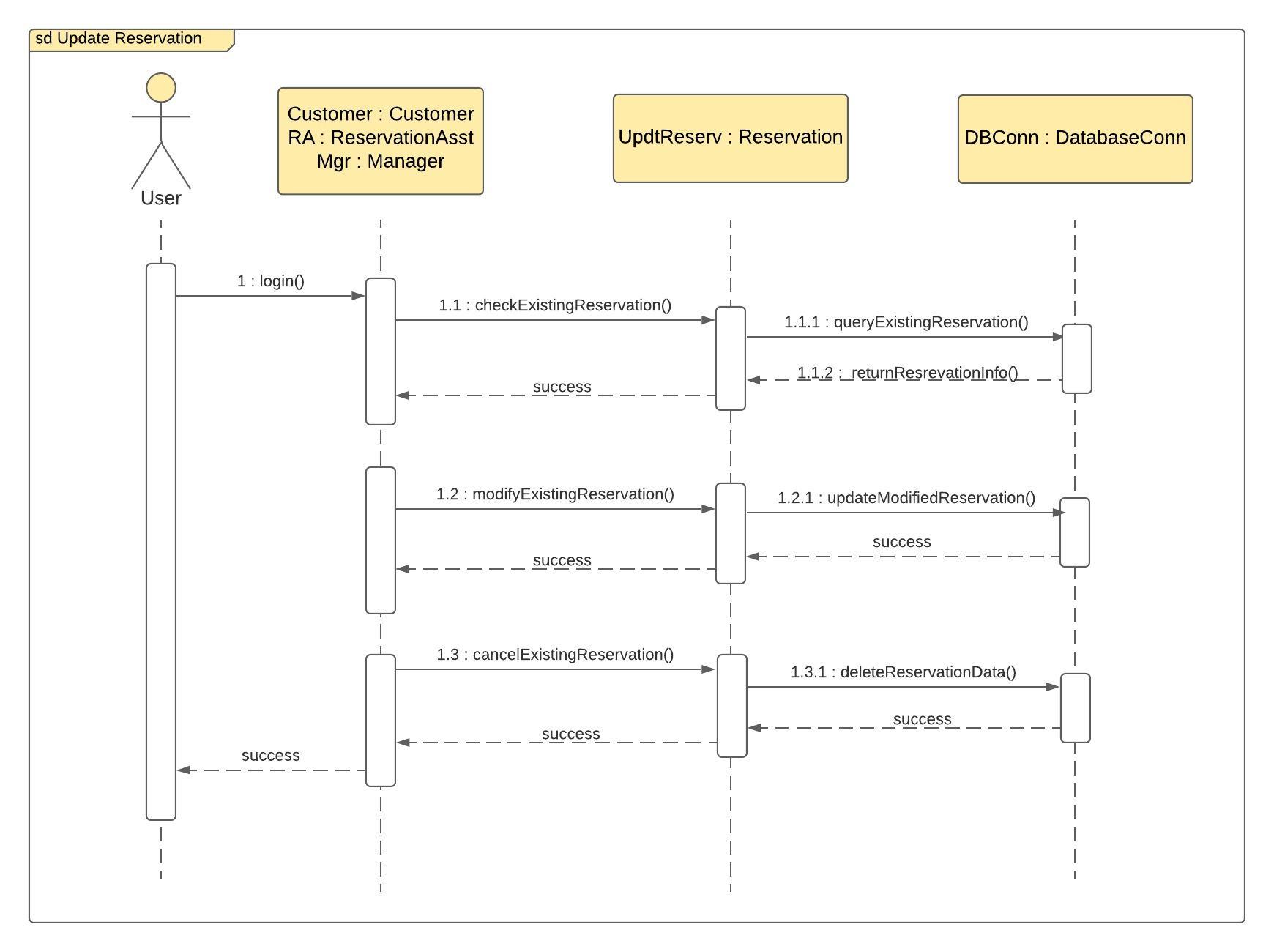
**Create Account**:



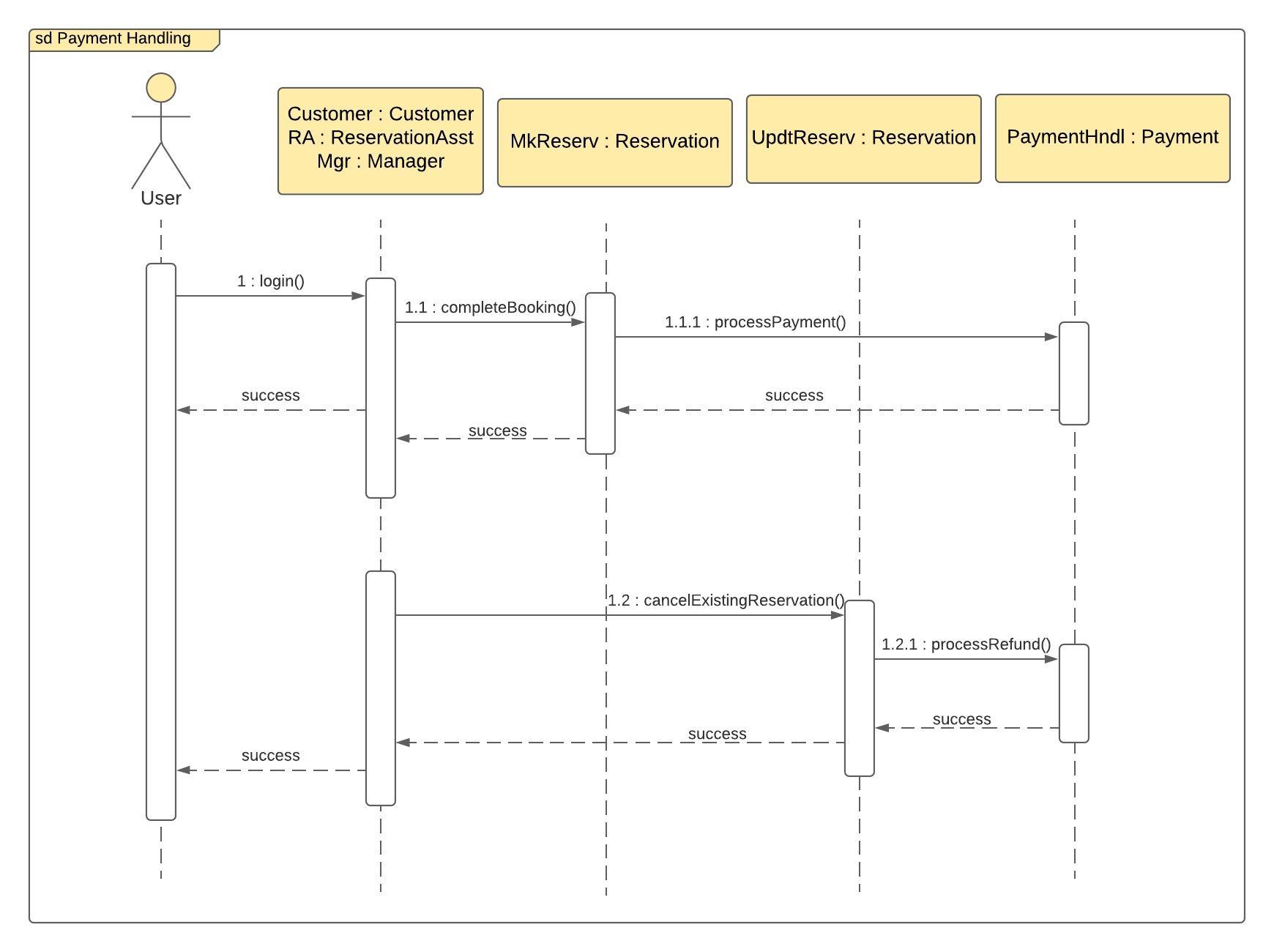
**Make Reservation**:



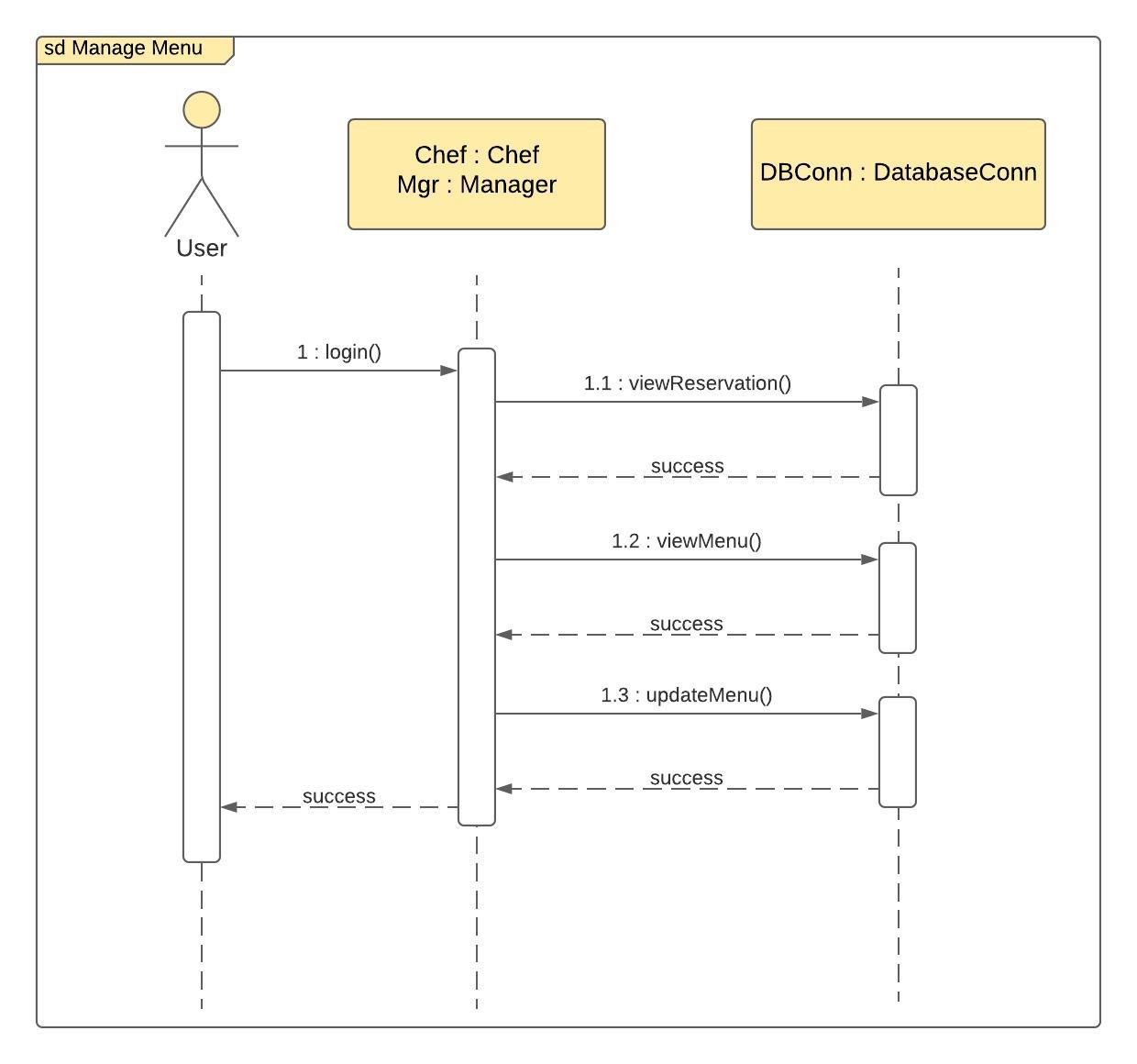
**Update Reservation**:



**Payment Handling**:



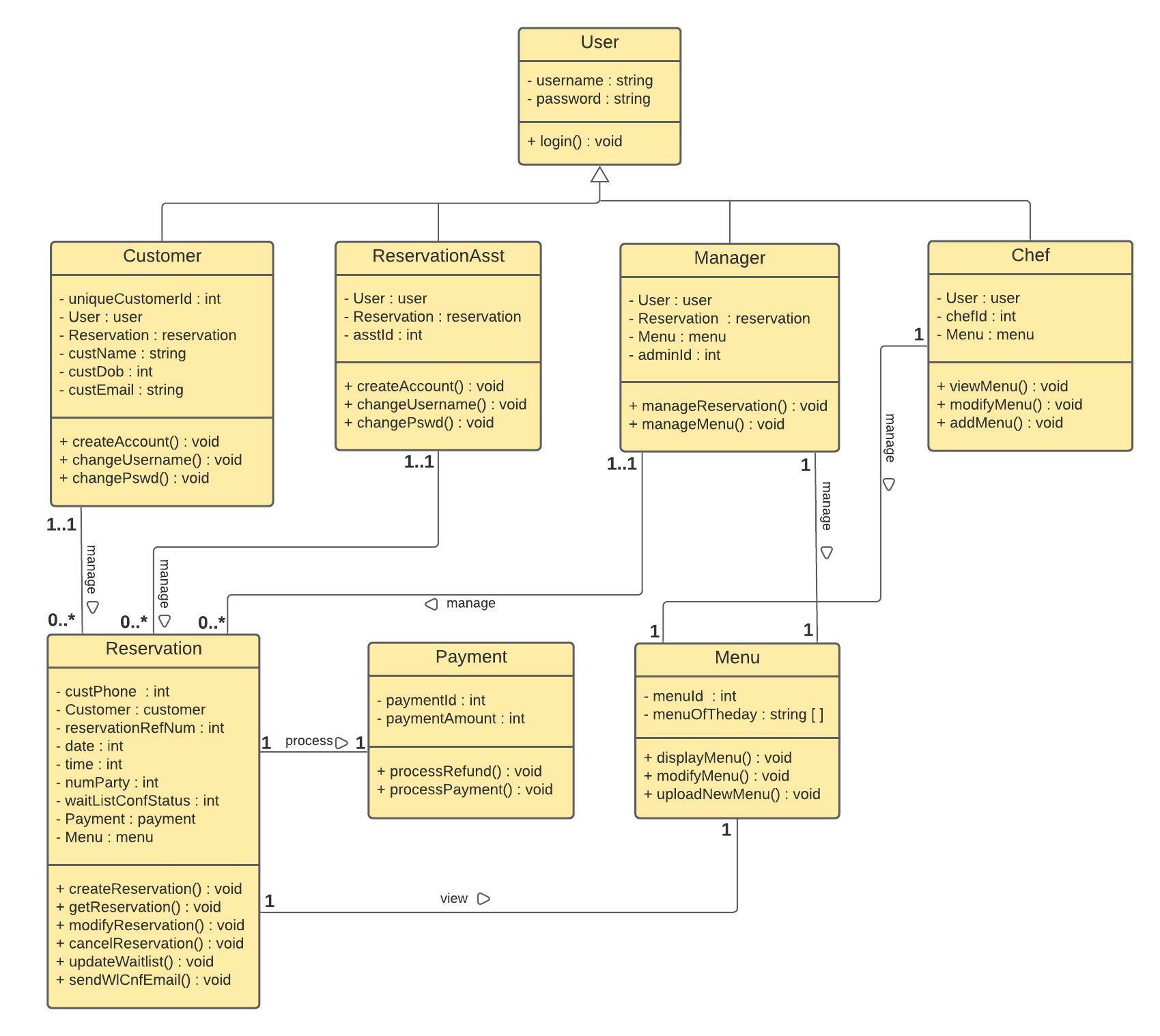
**Manage Menu**:



## Part 3: DESIGN

## Final Class Diagram

The final class diagram of the Restaurant Reservation System design is shown below. It is similar to the initial class diagram in most parts with the only change being the class DatabaseConn now removed, as the functionality of this class will be covered by the DAM classes in the Data Access design. Each of the classes are now populated with the detailed attributes and methods. The relationships amongst the objects are also defined in the class diagram below. Appropriate references between various class objects have also been added as per the Foreign Key attributes defined in the Final Database Diagram of Section 3.3.



The restaurant Reservation System is built on the following assumptions:

1. It's a boutique fine dining restaurant with a fixed-tasting menu, that is unique for a day of the week.
2. There will not be any bill/cheque handling. All payments will be completed during completing reservation via internet/phone.

A typical flow for making a reservation is as follows:

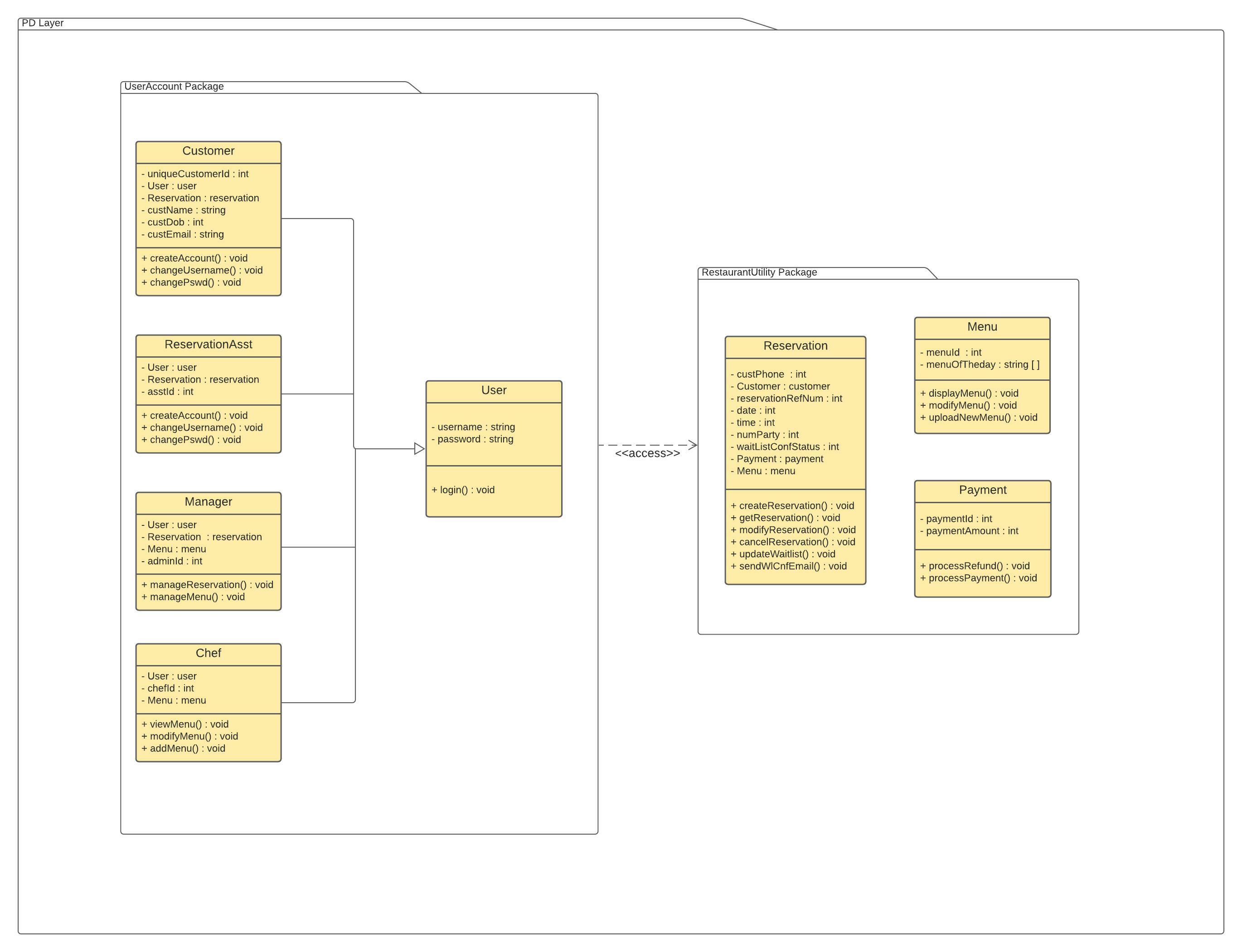
1. The *reservation* object will invoke the *createReservation*() method, which will retrieve the *menuOfTheDay* for the given date via the *menu* object. The attribute *menuOfTheDay* is a list of string type to store the menu on a given date.
2. The *menu* object will just display what's on offer on the Tasting Menu for the selected date. If the customer likes the menu, they can go ahead with booking a table. The menu price is fixed per person.
3. The method *createReservation*() will get the *numParty* value and calculate the total cost of the reservation = *numParty* \* Price Per Person. Then it will reference the *payment* object to complete the payment with the Payment Gateway.

## Package Diagram

We have divided the classes into 2 packages: User Account Package and RestaurantUtility Package.

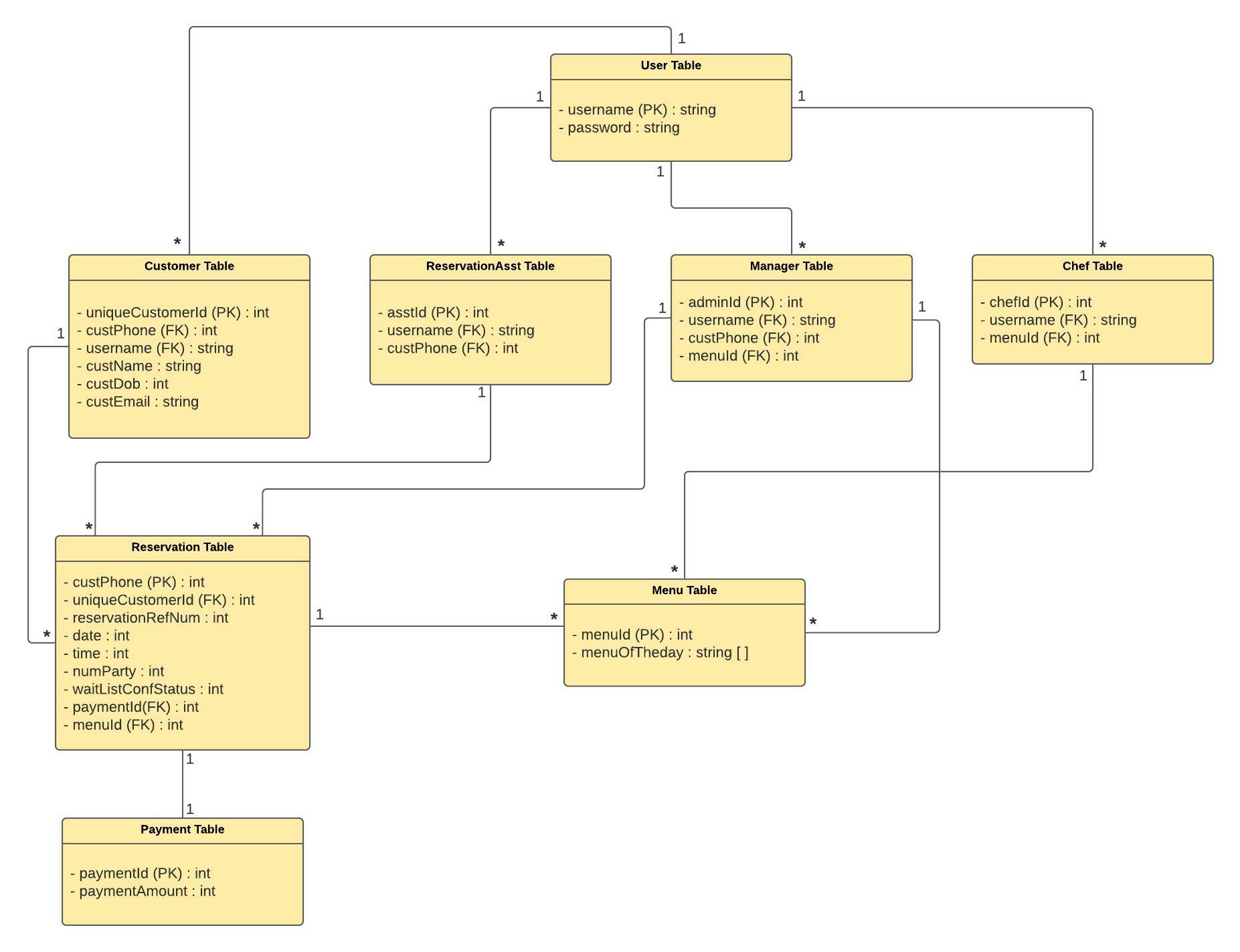
The User class is packaged together with Customer, ReservationAsst, Manager and Chef classes as these are different types of users of the Restaurant Reservation System.

Each of the Reservation, Payment and Menu classes handle some Restaurant Reservation related attributes and operations and hence are packaged together into RestaurantUtility Package.



## Database Design

The Database Design diagram is shown below.



The User Table has a 1-to-many relationship with the different types of user tables – Customer Table, ReservationAsst Table, Manager Table and Chef Table.

The Customer Table uses the *uniqueCustomerId* attribute as PK and needs to be unique for a customer. By design, each *uniqueCustomerId* is attached to a unique value of *custPhone* which is a FK in Customer Table. The Customer Table can create, access and modify the reservation data for each customer in the Reservation Table via *uniqueCustomerId*. It’s a 1-to-many association. The ReservationAsst Table and Manager Table can also create, access and modify the reservation data of each customer in the Reservation Table by accessing it via the *custPhone* attribute.

In the Reservation Table, the Primary key *custPhone*, uniquely defines each customer from the Customer Table. Each customer can have any number of reservations attached to their *custPhone* which in turn refers to their *uniqueCustomerId*. Each reservation will be referenced using the *reservationRefNum* attribute once created. Each *reservationRefNum* in turn has a unique *paymentId* which is the PK for the Payment Table. The *paymentAmount* of each reservation is tracked via *paymentId* in the Payment Table and is sent to the Payment Gateway for payment processing or refund processing in case of a cancellation.

The Chef Table and Manager Table can access and modify the Menu table via *menuId* attribute.

The Customer Table and ReservationAsst Table can access the Menu Table indirectly via the Reservation Table using the *menuId* attribute which is the PK of the Menu Table and an FK in Reservation Table.

## Data Access and Manipulation Design

The Data Access and Manipulation (DAM) classes are designed to translate between the objects of the problem domain classes and the corresponding database objects i.e. problem domain classes use the DAM class to connect to the corresponding database table. We followed the rule stated in the textbook that there should be one DAM class for each problem domain class and based on that, our Data Access and Manipulation Design is shown in the diagram below.

