FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

SYSTEM INTEGRATION AND ARCHITECTURE II
T2-TECHNICAL
Software Construction and Testing & System Integration

TABLEDIN: SMART RESERVATIONS AND SEAT COORDINATION SYSTEM

GRADE

Submitted by:

			1X1 PICTURE
Nares, Jerry Reivrick Conde Course Section	Reyes, John Edwin Crisostomo Course Section	Verzosa, Zrone Jinrx Jbryl Flores Course Section	LN, FN MI Course Section

Submitted to:

Ms. Geliza Alcober
Professor
September 2021

PROJECT PLAN

List at least 10 functional and 5 non-functional requirements for the chosen system.

Organize & Prioritize: Use a prioritization technique (e.g., MoSCoW, Kano Model).

A. Functional Requirements

Requirement ID	**Functional Requirement**	**Kano Category**
FR1	The system shall allow customers to create an account and log in securely.	Must-Have
FR2	The system shall allow customers to manage their profile (update details and change password.	Must-have
FR3	The system shall allow customers to book tables by selecting date, time, number of guests, and seat.	Must-have
FR4	The system shall allow customers to edit or cancel reservations.	Must-have
FR5	The system shall provide notifications (confirmation, reminders, updates, cancellations).	Must-have

FR6	The system shall allow admins/staff to log in with staff credentials.	Must-have
FR7	The system shall allow staff to manage reservations (approve, modify, cancel).	Must-have
FR8	The system shall allow staff to manage seating (view available, assign/reassign tables).	Must-have
FR9	The system shall provide queue management features for walk-in customers.	Must-have
FR10	The system shall allow staff to generate reports (daily, weekly, monthly)	Must-have

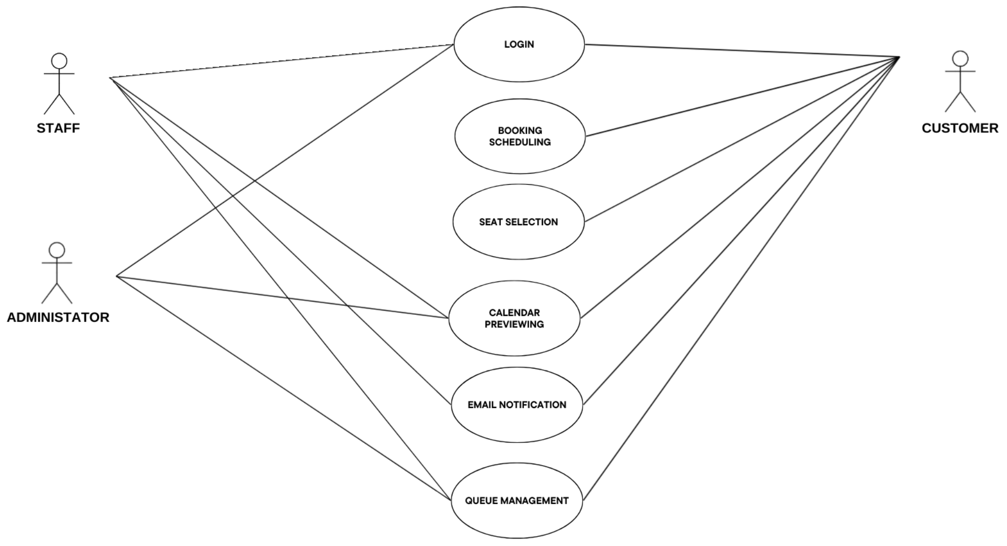
B. Non-functional Requirements

Requirement ID	**Non-Functional Requirement**	**Kano Category**
NFR1	The system shall be available 100% of the time during business hours.	Must-have
NFR2	The system shall process bookings within 10 seconds.	Performance
NFR3	The system shall ensure data security and privacy	Must-have

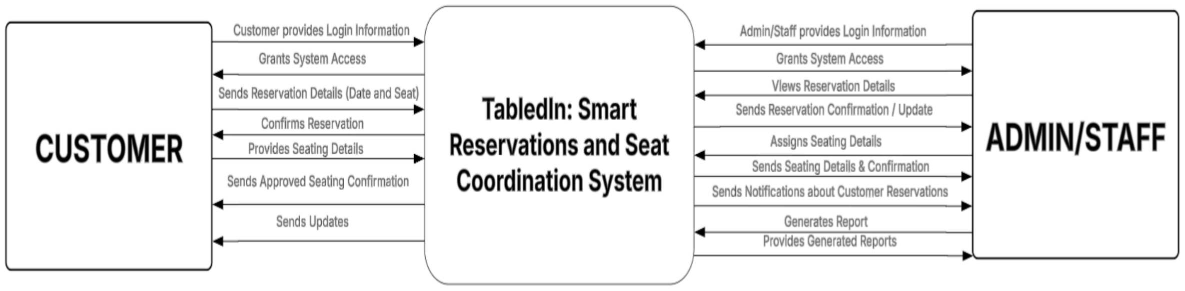
	(encrypted passwords and secure login).	
NFR4	The system shall be user-friendly and intuitive, requiring minimal training.	Attractive
NFR5	The system shall be scalable to handle multiple branches and increased users in the future.	Performance

SYSTEM FUNCTIONALITY

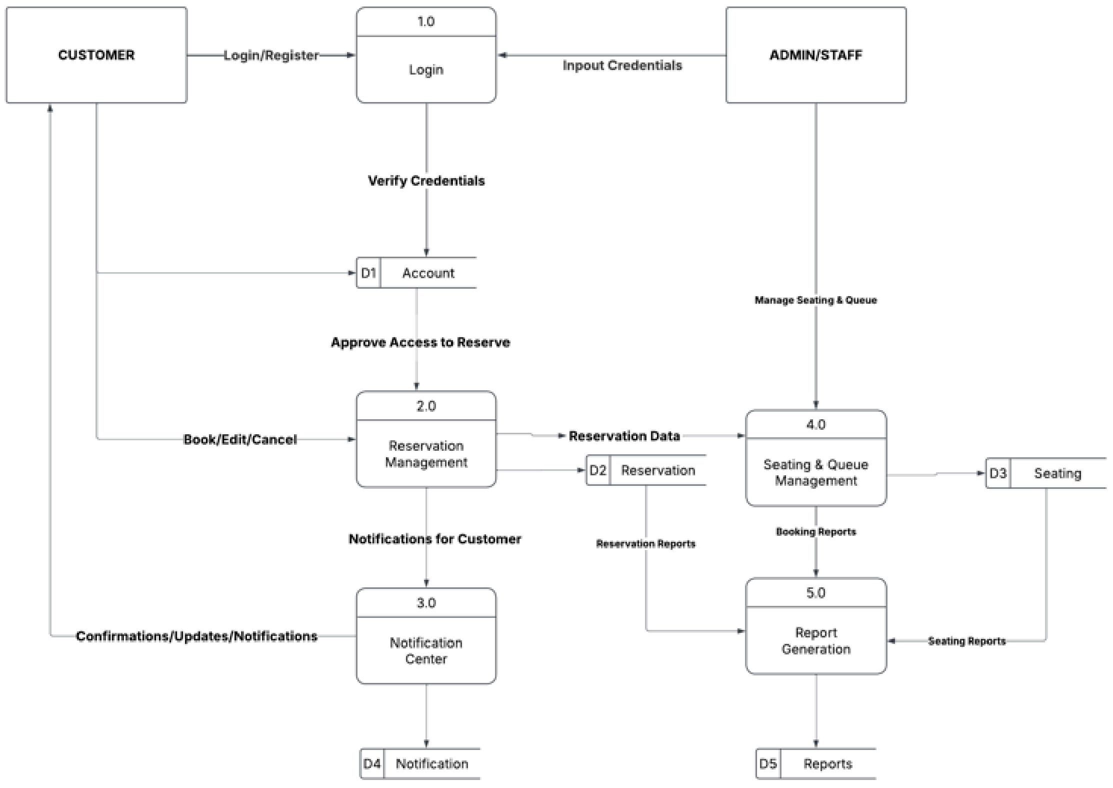
Embed the system’s Use Case Diagram, CFD, DFD, ERD, and Activity Diagram


STAFF
ADMINISTATOR
LOGIN
BOOKING SCHEDULING
SEAT SELECTION
CALENDAR PREVIEWING
EMAIL NOTIFICATION
QUEUE MANAGEMENT
CUSTOMER

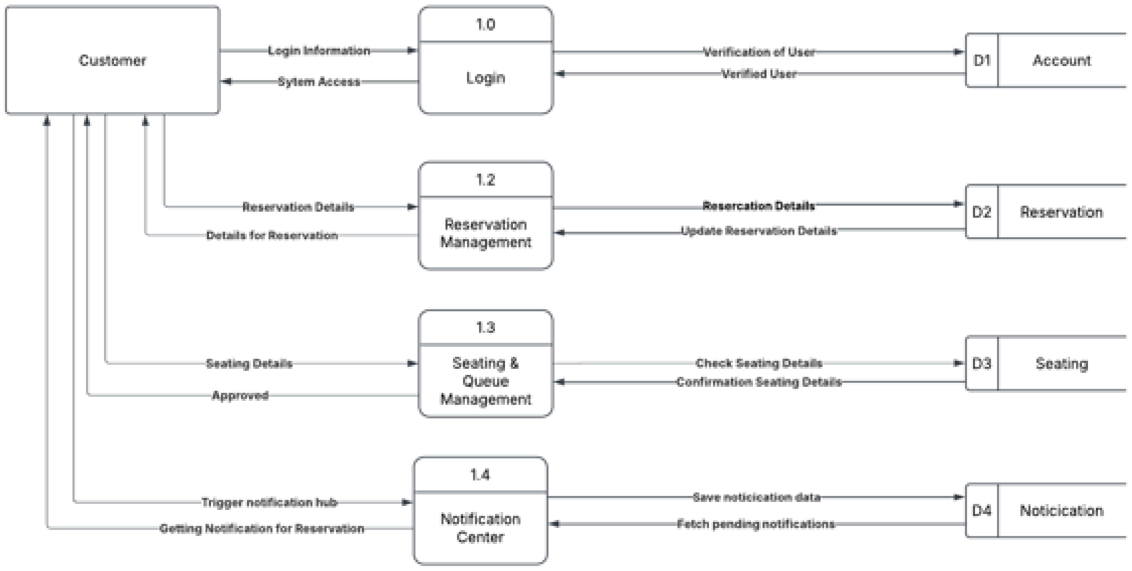
CFD


CUSTOMER
Customer provides Login Information
Grants System Access
Sends Reservation Details (Date and Seat)
Confirms Reservation
Provides Seating Details
Sends Approved Seating Confirmation
Sends Updates
TabledIn: Smart Reservations and Seat Coordination System
Admin/Staff provides Login Information
Grants System Access
Views Reservation Details
Sends Reservation Confirmation / Update
Assigns Seating Details
Sends Seating Details & Confirmation
Sends Notifications about Customer Reservations
Generates Report
Provides Generated Reports
ADMIN/STAFF

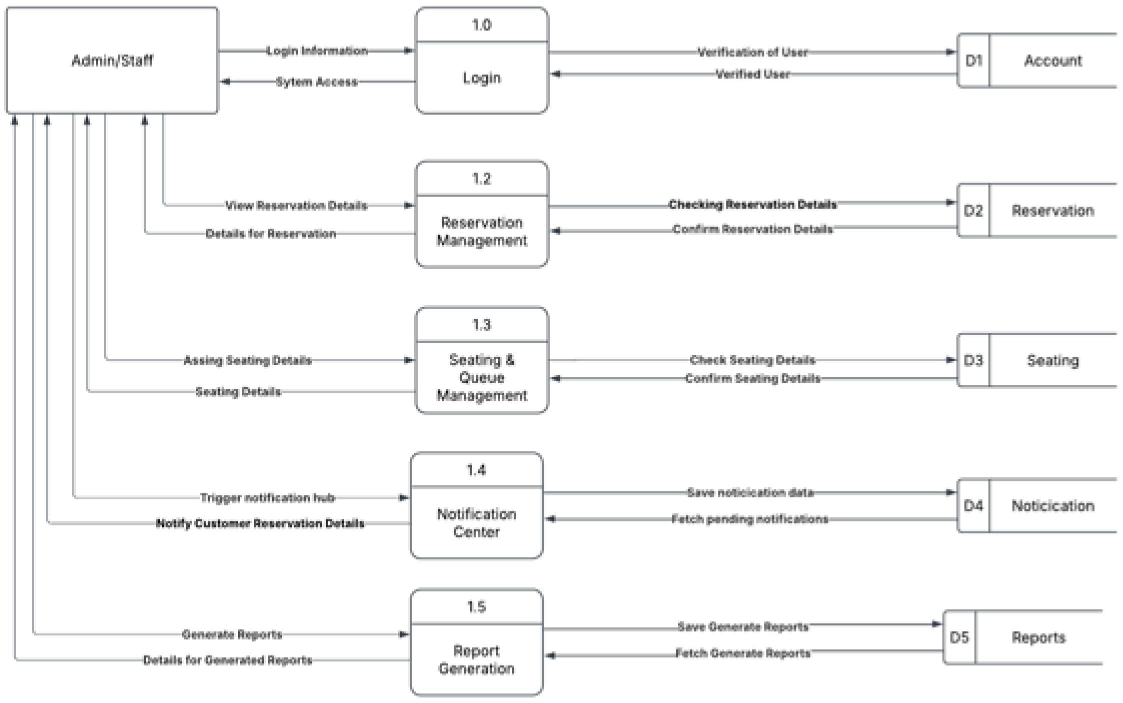
DFD



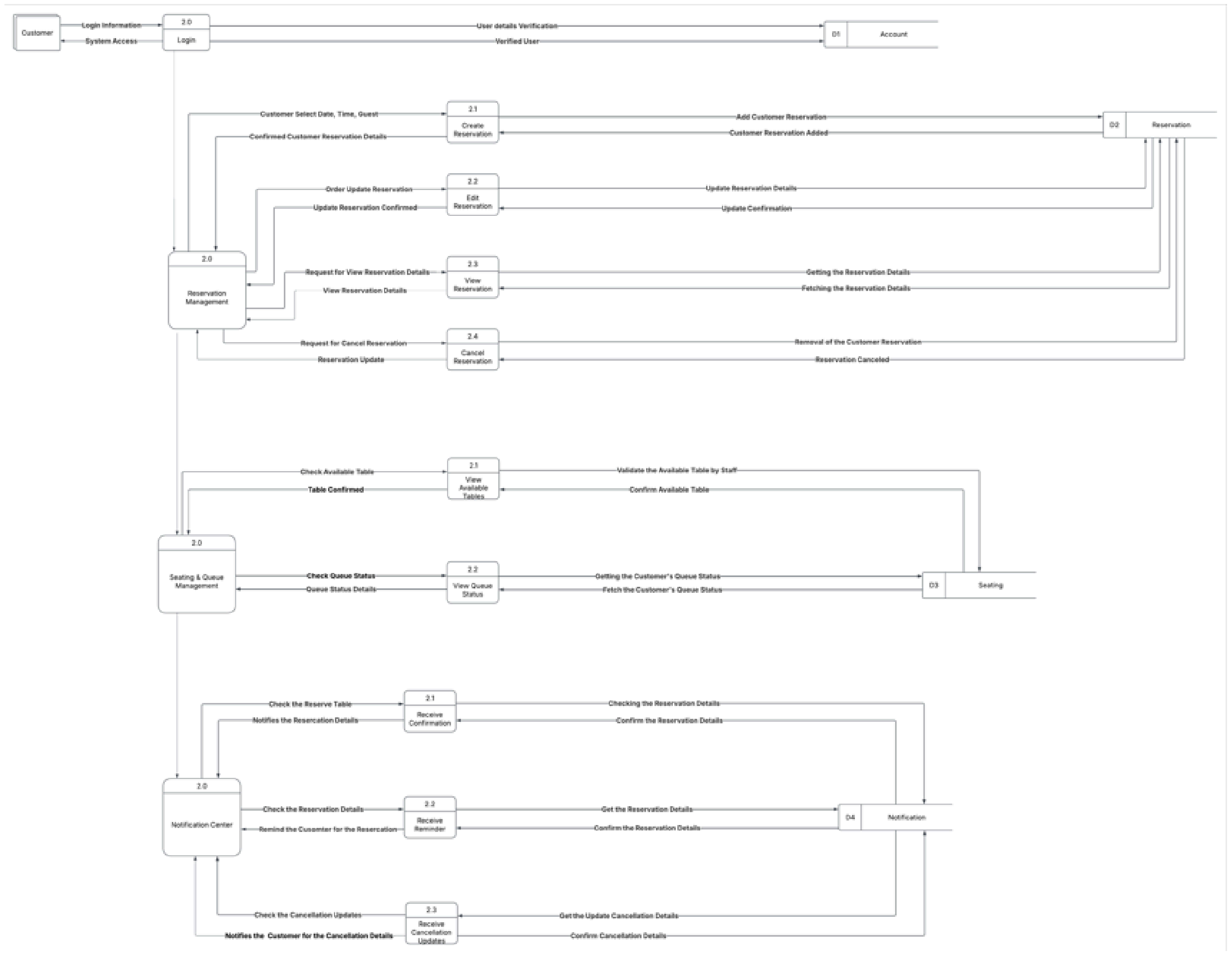
Level 1 DFD - Customer



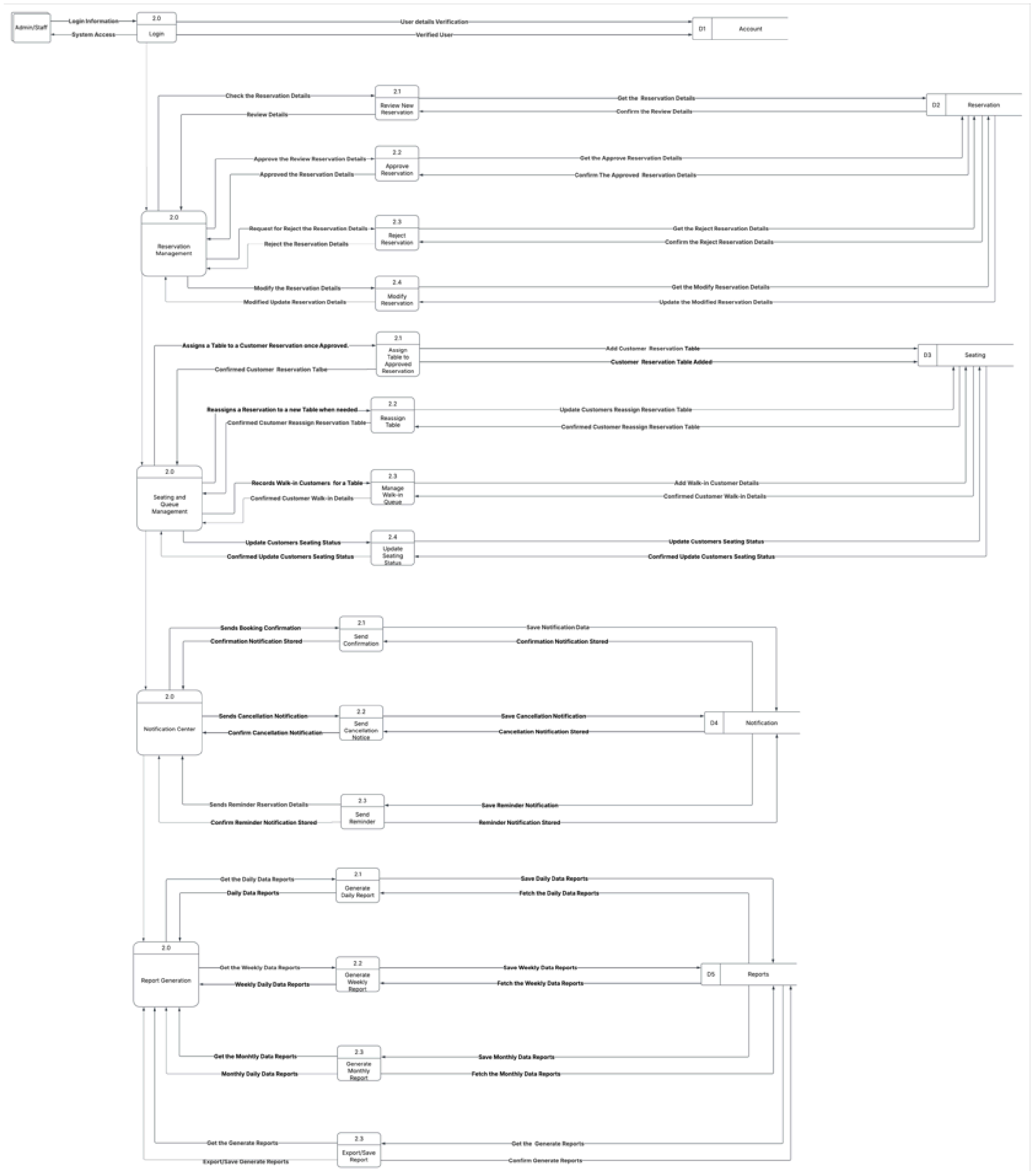
Level 1 DFD - Admin/Staff



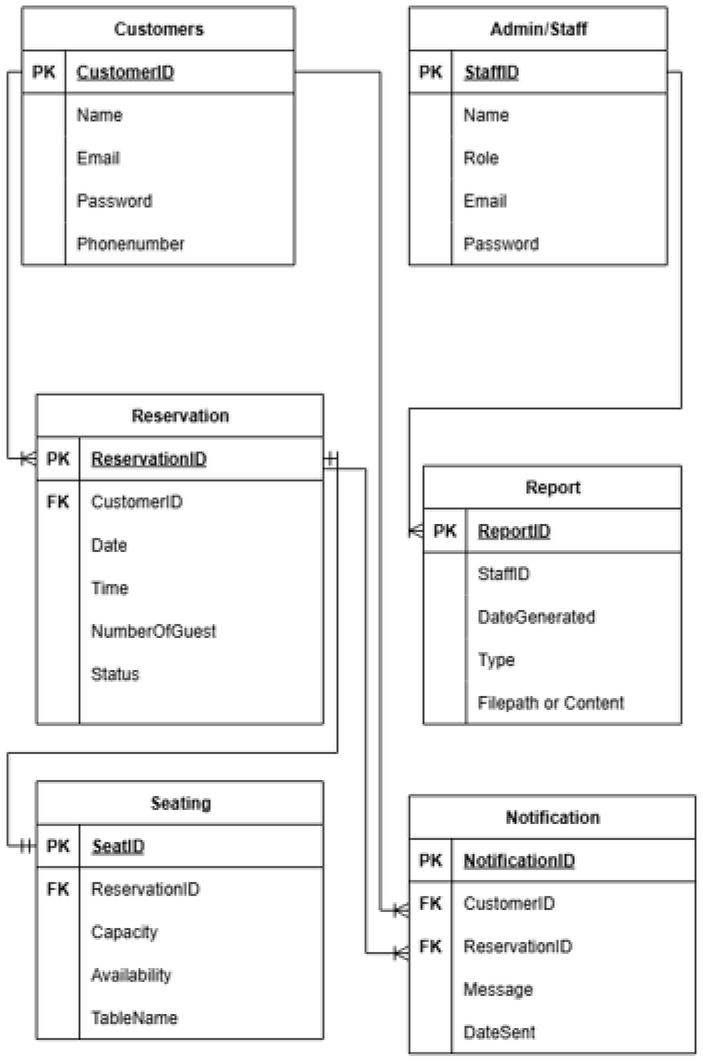
Level 2 DFD - Customer


Customer
Login Information
System Access
2.0
Login
User details Verification
Verified User
D1
Account
Customer Select Date, Time, Guest
Confirmed Customer Reservation Details
2.1
Create Reservation
Add Customer Reservation
Customer Reservation Added
D2
Reservation
Order Update Reservation
Update Reservation Confirmed
2.2
Edit Reservation
Update Reservation Details
Update Confirmation
2.0
Reservation Management
Request for View Reservation Details
View Reservation Details
2.3
View Reservation
Getting the Reservation Details
Fetching the Reservation Details
Request for Cancel Reservation
Reservation Update
2.4
Cancel Reservation
Removal of the Customer Reservation
Reservation Canceled
Check Available Table
Table Confirmed
2.1
View Available Tables
Validate the Available Table by Staff
Confirm Available Table
2.0
Seating & Queue Management
Check Queue Status
Queue Status Details
2.2
View Queue Status
Getting the Customer's Queue Status
Fetch the Customer's Queue Status
D3
Seating
Check the Reserve Table
Notifies the Resercation Details
2.1
Receive Confirmation
Checking the Reservation Details
Confirm the Reservation Details
2.0
Notification Center
Check the Reservation Details
Remind the Cusomter for the Resercation
2.2
Receive Reminder
Get the Reservation Details
Confirm the Reservation Details
D4
Notification
Check the Cancellation Updates
Notifies the Customer for the Cancellation Details
2.3
Receive Cancellation Updates
Get the Update Cancellation Details
Confirm Cancellation Details

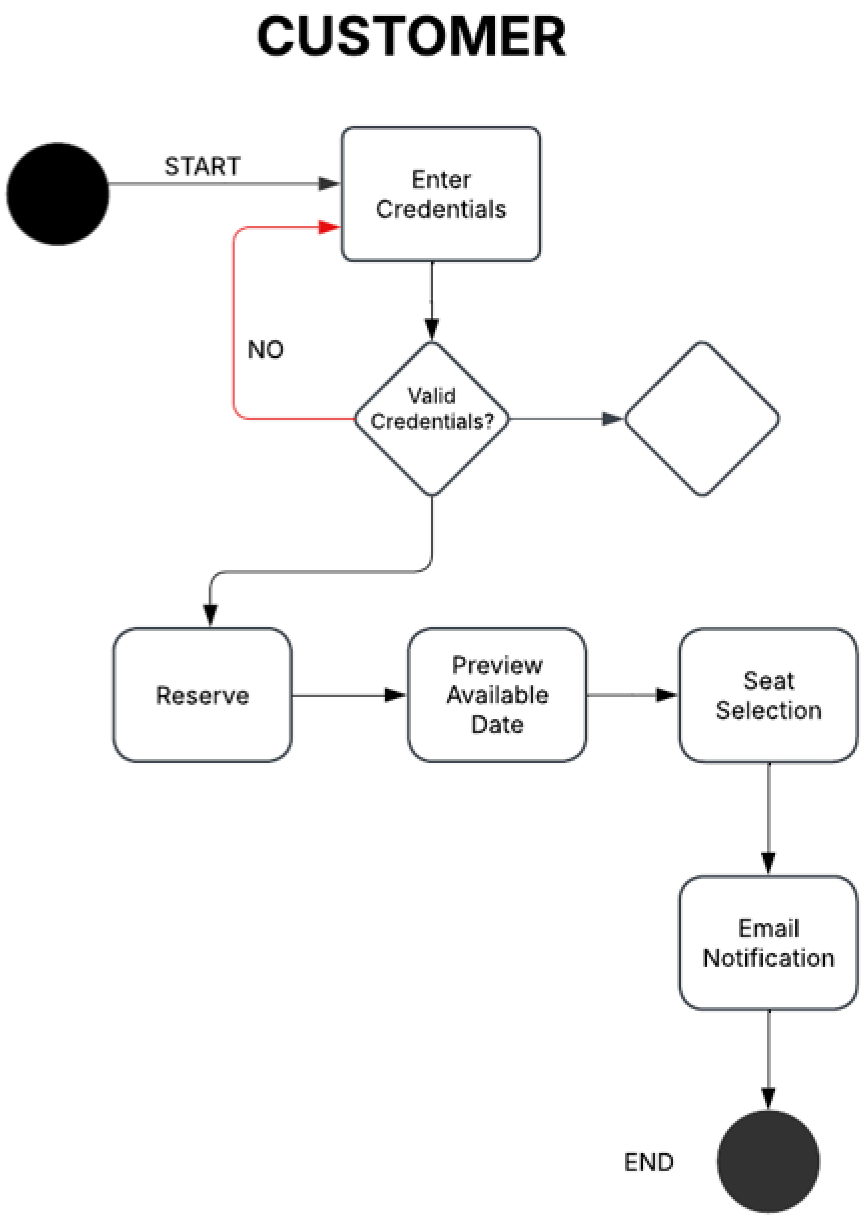
Level 2 DFD - Admin/Staff

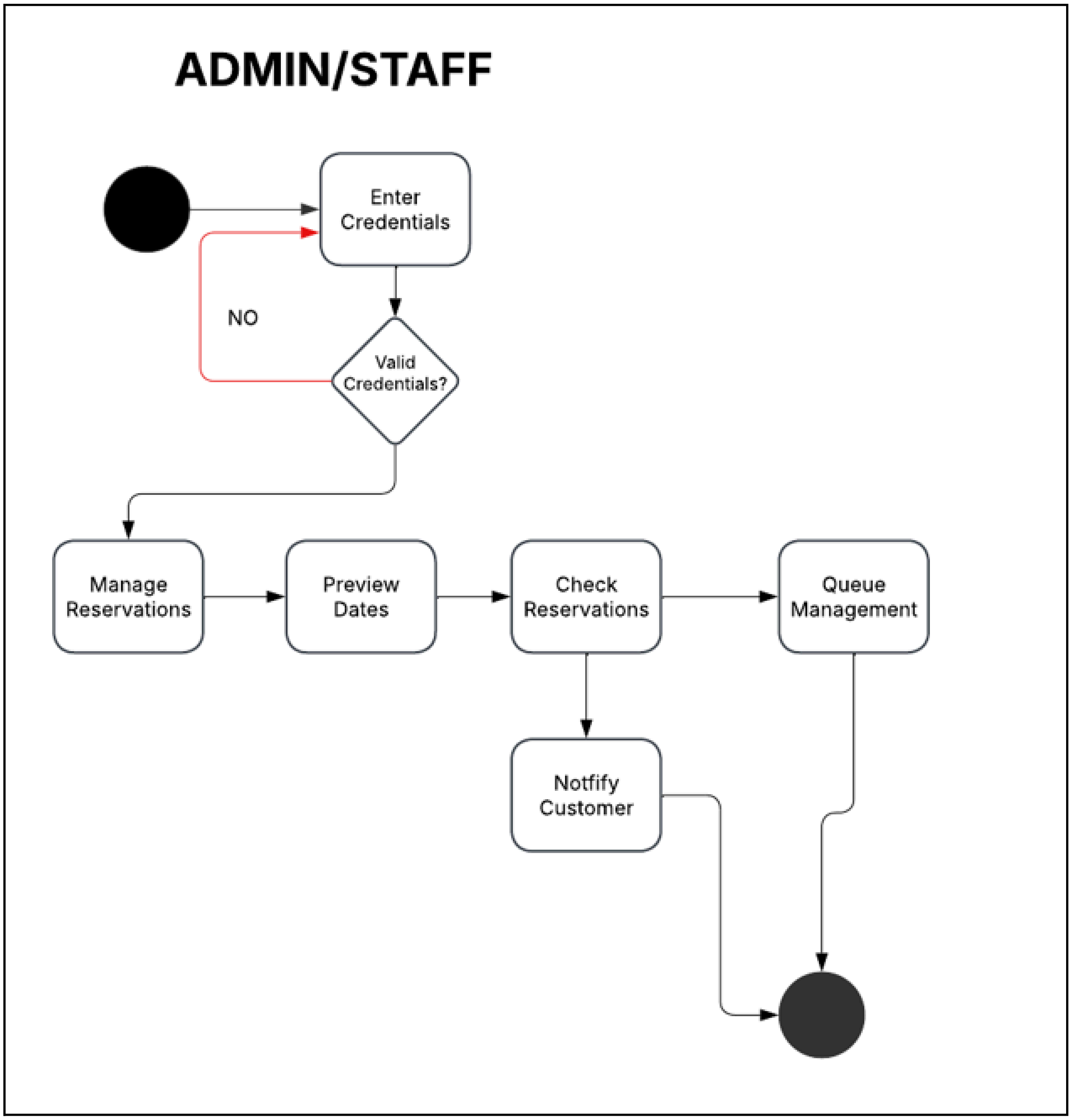

Admin/Staff
Login Information
System Access
2.0
Login
User details Verification
Verified User
D1
Account
Check the Reservation Details
Review Details
2.1
Review New Reservation
Get the Reservation Details
Confirm the Review Details
D2
Reservation
Approve the Review Reservation Details
Approved the Reservation Details
2.2
Approve Reservation
Get the Approve Reservation Details
Confirm The Approved Reservation Details
2.0
Reservation Management
Request for Reject the Reservation Details
Reject the Reservation Details
2.3
Reject Reservation
Get the Reject Reservation Details
Confirm the Reject Reservation Details
Modify the Reservation Details
Modified Update Reservation Details
2.4
Modify Reservation
Get the Modify Reservation Details
Update the Modified Reservation Details
Assigns a Table to a Customer Reservation once Approved.
Confirmed Customer Reservation Talbe
2.1
Assign Table to Approved Reservation
Add Customer Reservation Table
Customer Reservation Table Added
D3
Seating
Reassigns a Reservation to a new Table when needed
Confirmed Csutomer Reassign Reservation Table
2.2
Reassign Table
Update Customers Reassign Reservation Table
Confirmed Customer Reassign Reservation Table
2.0
Seating and Queue Management
Records Walk-in Customers for a Table
Confirmed Customer Walk-in Details
2.3
Manage Walk-in Queue
Add Walk-in Customer Details
Confirmed Customer Walk-in Details
Update Customers Seating Status
Confirmed Update Customers Seating Status
2.4
Update Seating Status
Update Customers Seating Status
Confirmed Update Customers Seating Status
Sends Booking Confirmation
Confirmation Notification Stored
2.1
Send Confirmation
Save Notification Data
Confirmation Notification Stored
2.0
Notification Center
Sends Cancellation Notification
Confirm Cancellation Notification
2.2
Send Cancellation Notice
Save Cancellation Notification
Cancellation Notification Stored
D4
Notification
Sends Reminder Rservation Details
Confirm Reminder Notification Stored
2.3
Send Reminder
Save Reminder Notification
Reminder Notification Stored
Get the Daily Data Reports
Daily Data Reports
2.1
Generate Daily Report
Save Daily Data Reports
Fetch the Daily Data Reports
2.0
Report Generation
Get the Weekly Data Reports
Weekly Daily Data Reports
2.2
Generate Weekly Report
Save Weekly Data Reports
Fetch the Weekly Data Reports
D5
Reports
Get the Monthly Data Reports
Monthly Daily Data Reports
2.3
Generate Monthly Report
Save Monthly Data Reports
Fetch the Monthly Data Reports
Get the Generate Reports
Export/Save Generate Reports
2.3
Export/Save Report
Get the Generate Reports
Confirm Generate Reports

ERD


Customers
PK CustomerID
Name
Email
Password
Phonenumber
Admin/Staff
PK StaffID
Name
Role
Email
Password
Reservation
PK ReservationID
FK CustomerID
Date
Time
NumberOfGuest
Status
Report
PK ReportID
StaffID
DateGenerated
Type
Filepath or Content
Seating
PK SeatID
FK ReservationID
Capacity
Availability
TableName
Notification
PK NotificationID
FK CustomerID
FK ReservationID
Message
DateSent

Activity Diagram

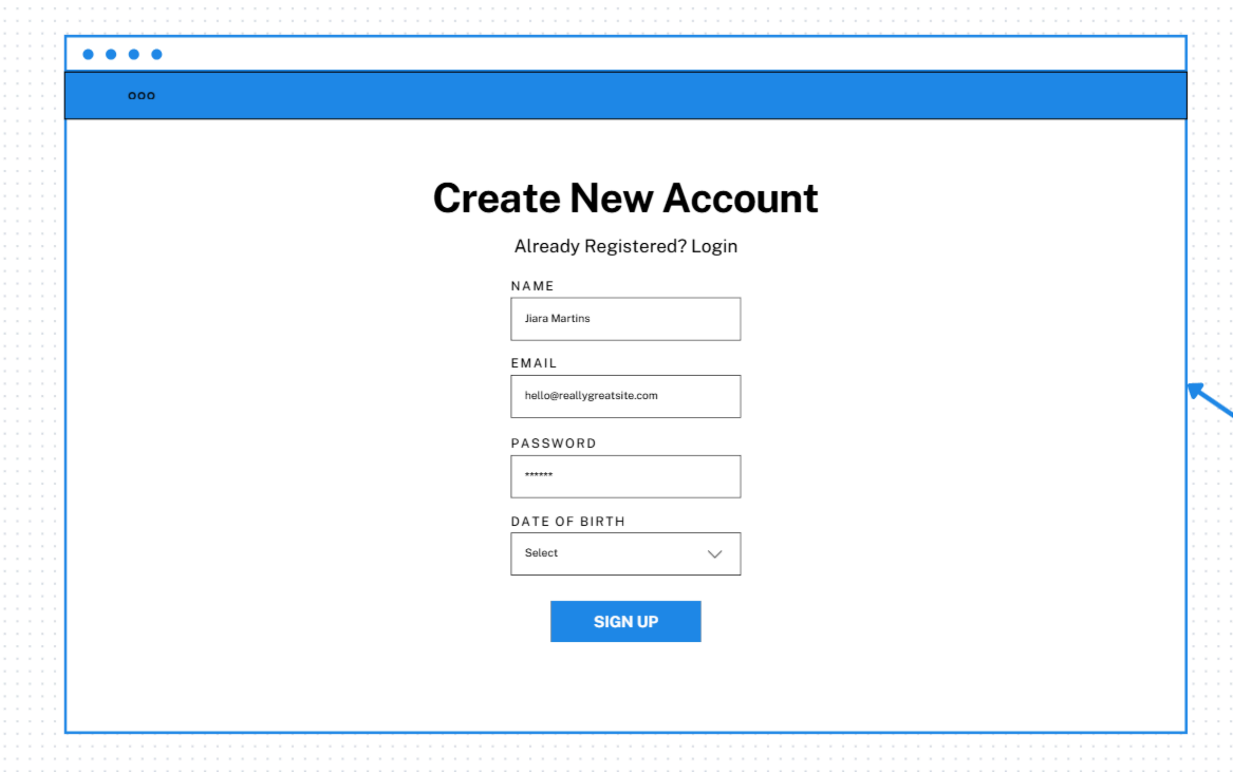



ADMIN/STAFF
Enter Credentials
NO
Valid Credentials?
Manage Reservations
Preview Dates
Check Reservations
Queue Management
Notfify Customer

SYSTEM DESIGN

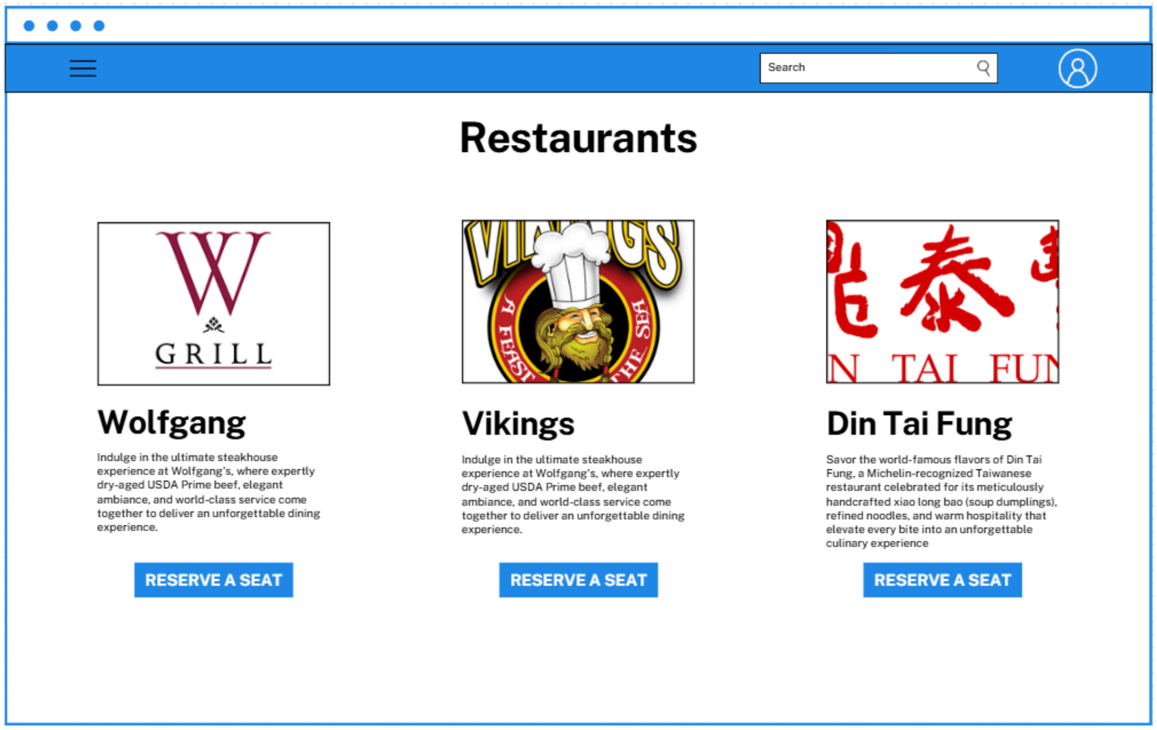
This should include all the system wireframe of the **core features** of the project along with detailed explanations.

LOGIN MODULE



- The user creation portion of the proposed system will allow customers to easily register for restaurant appointments by providing essential information such as their name and email, along with secure password creation and verification. The design will prioritize responsiveness and accessibility, ensuring a seamless and inclusive experience across all devices.

APPOINTMENT SCHEDULING MODULE



Restaurants

Wolfgang

Indulge in the ultimate steakhouse experience at Wolfgang's, where expertly dry-aged USDA Prime beef, elegant ambiance, and world-class service come together to deliver an unforgettable dining experience.

RESERVE A SEAT

Vikings

Indulge in the ultimate steakhouse experience at Wolfgang's, where expertly dry-aged USDA Prime beef, elegant ambiance, and world-class service come together to deliver an unforgettable dining experience.

RESERVE A SEAT

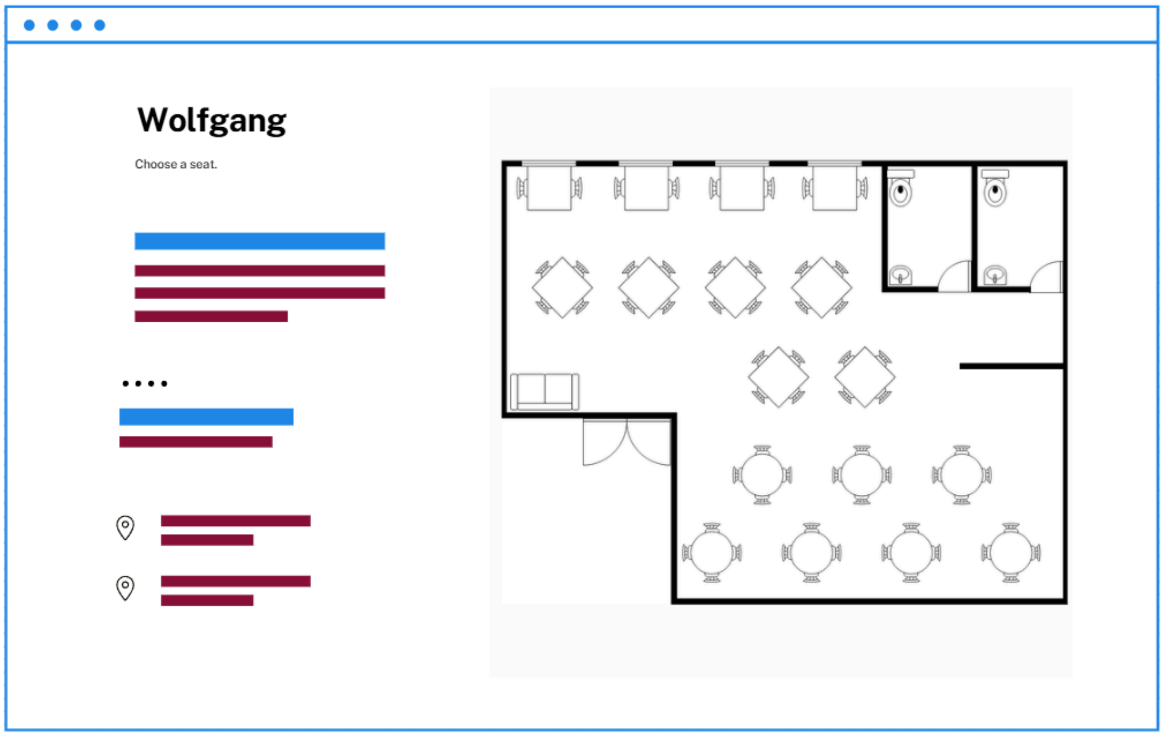
Din Tai Fung

Savor the world-famous flavors of Din Tai Fung, a Michelin-recognized Taiwanese restaurant celebrated for its meticulously handcrafted xiao long bao (soup dumplings), refined noodles, and warm hospitality that elevate every bite into an unforgettable culinary experience

RESERVE A SEAT

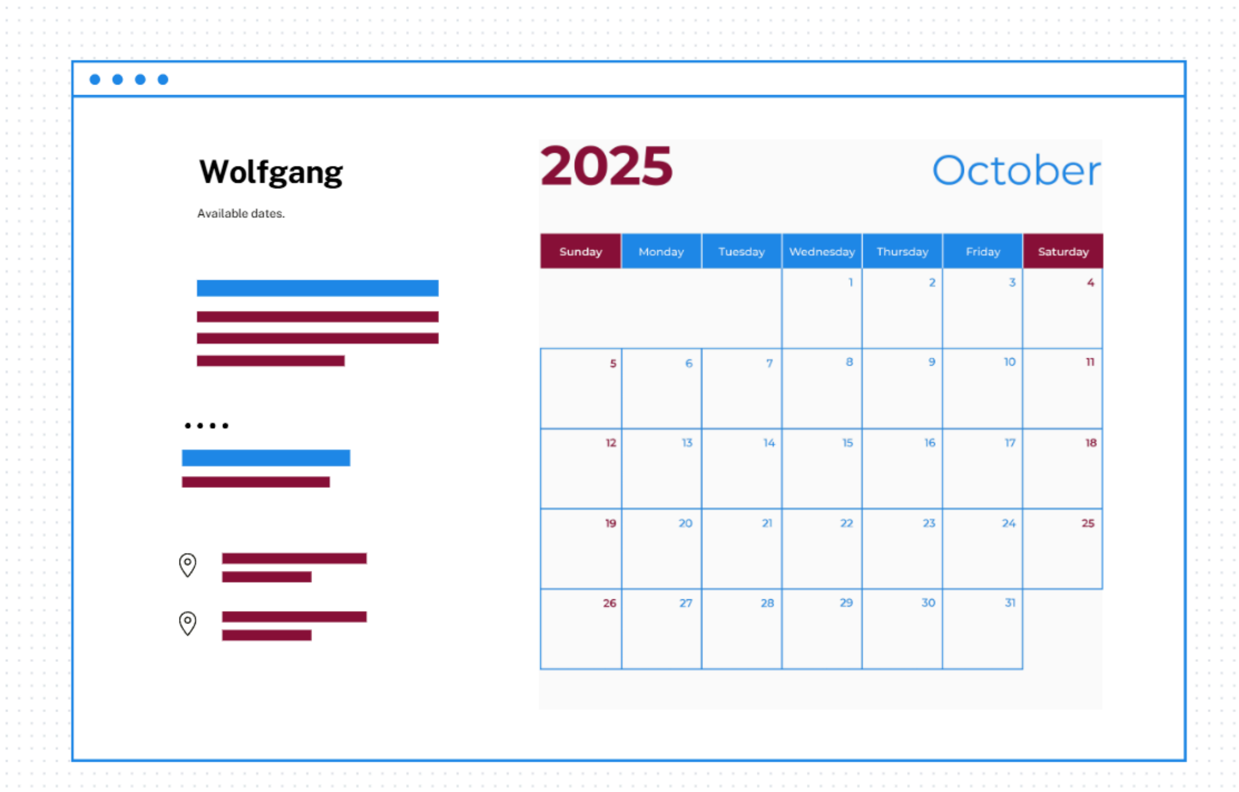
- Users have the option to select from a diverse range of restaurants when making a seat reservation, enhancing their dining experience with multiple culinary choices. By clicking the "Reserve a Seat" button, they can easily choose their preferred date, time, and available seating arrangements, ensuring a seamless booking process.

SEATING MANAGEMENT MODULE



- Once a restaurant is selected, users can access an interactive seating arrangement diagram that displays the layout of the dining area, including available tables and their proximity to amenities like the entrance or restrooms. This feature allows users to choose their preferred seating location based on factors such as privacy, ambiance, and views, ensuring a more personalized dining experience.

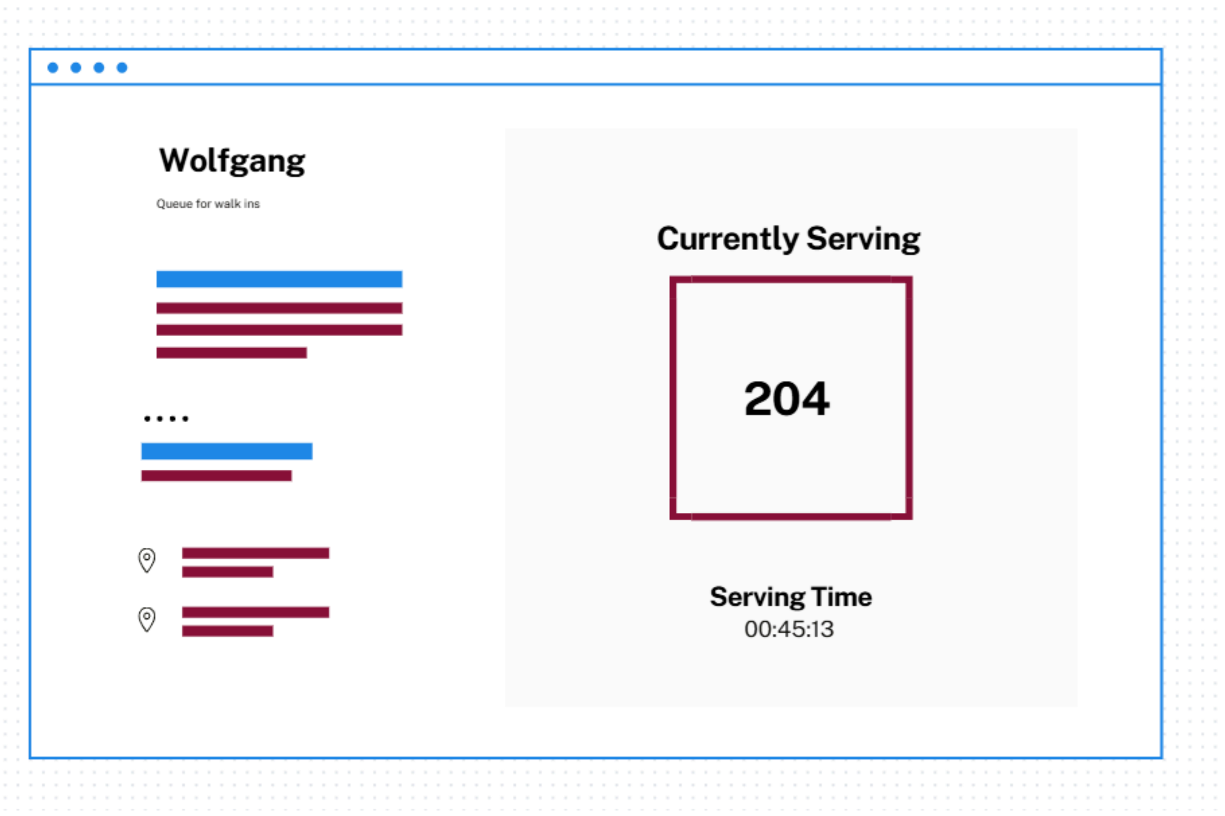
REAL-TIME CALENDAR VIEWING MODULE



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

- Customers will have access to a detailed calendar feature that displays available reservation dates and times for their chosen restaurant, allowing them to easily plan their visits. Additionally, the calendar will indicate which dates are already reserved, ensuring customers can make informed decisions when selecting their preferred dining experience.

QUEUE MANAGEMENT MODULE



- Queue management for walk-ins allows customers to view the current token number being served, providing them with real-time updates on their standing in line. Additionally, it displays the estimated wait time before the next token will be called, helping patrons better plan their visit and manage their expectations.

Part III – Reflection & Critique:

1. **Write a 1–2-page reflection on the importance of aligning requirements with system models.**

In systems development, making sure that requirements are aligned with system models is one of the most important steps in building a successful product. Requirements tell us what the system needs to do, while system models show how it will do it. When these two are in sync, it helps everyone involved, from developers to stakeholders, stay on the same page and work toward the same goals.

One big advantage of this alignment is that it makes everything easier to trace. If a requirement changes, you can quickly see what parts of the system model are affected. This helps avoid confusion and makes it easier to manage updates or improvements. It also ensures that nothing important gets left out during development.

Another reason this alignment matters is communication. System models can turn complex or abstract requirements into something more visual and understandable. This is especially helpful when working with people who aren't very technical. When everyone can clearly see how the system is supposed to work, it's easier to make decisions and avoid misunderstandings.

Aligning requirements with models also helps reduce risks. If the system model doesn't match the requirements, there's a higher chance of building something that doesn't meet user needs. That can lead to wasted time, extra costs, and frustration. But when everything is aligned from the start, problems can be caught early, and the final product is more likely to work as intended.

Overall, aligning requirements with system models is a smart and necessary practice. It improves clarity, supports better teamwork, and helps ensure that the system actually does what it's supposed to do. As a student learning about systems development, I see how important this is and plan to apply it in future projects to build better, more reliable systems.

2. **Critically analyze how incomplete, ambiguous, or conflicting requirements could affect the accuracy of system models.**

When building a system, the quality of the requirements plays a huge role in how accurate and useful the system model turns out to be. If the requirements are incomplete, unclear, or even contradict each other, it can cause a lot of problems down the line.

Incomplete requirements are like missing puzzle pieces. If we don't know everything the system is supposed to do, the model will reflect that gap. Developers might make assumptions or leave out important features simply because they weren't told about them. This can lead to a system that doesn't fully meet user needs or fails in unexpected ways.

Ambiguous requirements are just as risky. If a requirement can be interpreted in more than one way, different team members might understand it differently. This leads to confusion and inconsistent design choices. For example, if a requirement says "the system should be fast," what does "fast" actually mean? Without specifics, it's hard to model or measure anything accurately.

Conflicting requirements are even worse. Imagine one requirement says the system should store user data permanently, while another says it should delete data after 30 days. Which one is correct? Conflicts like this force developers to guess or make compromises, which can result in a system that doesn't satisfy anyone.

All of these issues make the system model unreliable. And since models are used to guide development, testing, and even communication with stakeholders, any inaccuracies can snowball into bigger problems, like delays, extra costs, or having to redo work.

To avoid this, it's important to spend time gathering and refining requirements before modeling begins. Talking to stakeholders, asking questions, and reviewing everything carefully can help make sure the requirements are clear, complete, and consistent. That way, the system model can truly reflect what the system is supposed to do.