Assignment 2

Description:
A restaurant chain has reached out to your team to build a reservation system.

Here are the details:

- Two categories of users / customers: guest user or registered user.
- Users should be able to search for a table and reserve.
    - User doesn't need to login to the system to reserve a table. If registered users, they can login.
    - User enters name, phone, email, date and time (date picker), and # of guests for dining and system presents available tables.
    - Tables have maximum capacity limit i.e., 2, 4, 6, or 8.
    - Different combinations are allowed, and owner accommodates the seating, for example: someone requests 8 guests and table for 8 is not available but 2 + 6, or 4+4 is available. System should combine the tables and notify owner they need to combine tables. In this case System reserves both tables.
- If a guest user i.e., not a registered user, system should prompt user to register (Optional) before finalizing the reservation.
- Registered users will have these fields:
    - Name, mailing address, billing address (checkbox if same as mailing address), Preferred Diner # (system generated), Earned points (based on $ spent i.e., $1 is 1 point), preferred payment method (cash, credit, check).
- System should track high traffic days / weekends and a hold fee is required i.e. July 4th will require valid credit card on system to reserve the table.
    - Notify user no show will have minimum $10 charge.

Assumptions:
If you make any assumptions to provide a good user experience, please list them.

- All holidays count as high traffic days
- If the restaurant is reserved 50% or more before and after a reservation it is considered high traffic
- All reservations will have a duration of 1 hour

Answer these questions:

1. Based on your understanding of the above requirements, list functional and non-functional requirements for this project. (10 points)

Functional:

1. Graphical Interface for users to interact with the system to perform actions such as signing in/up and making reservations.
2. Authorization and authentication to handle registered and unregistered users.

3. Algorithm to handle whether the reservation can be made for a party of a given amount of people.
4. Traffic monitoring system to track high-traffic days to include reservation fees and no-show charges. The website also needs to validate credit card information with the system database.
5. User's expenditure input system to handle points earned.

Non-Functional:

1. Ability of the system to accurately reserve the right amount of tables, people, times, and even compute combinations of table reservations.
2. Database for saving reservations and users' information.
3. Security: hashing passwords, and saving personal information securely (phone number, billing address, etc)
4. Efficiency (loading, querying, etc)
5. Reliability (error handling, making sure the system will not crash unexpectedly)
6. Ease of use (Since the system has simple functionality, it should be easy for the user to use the software)

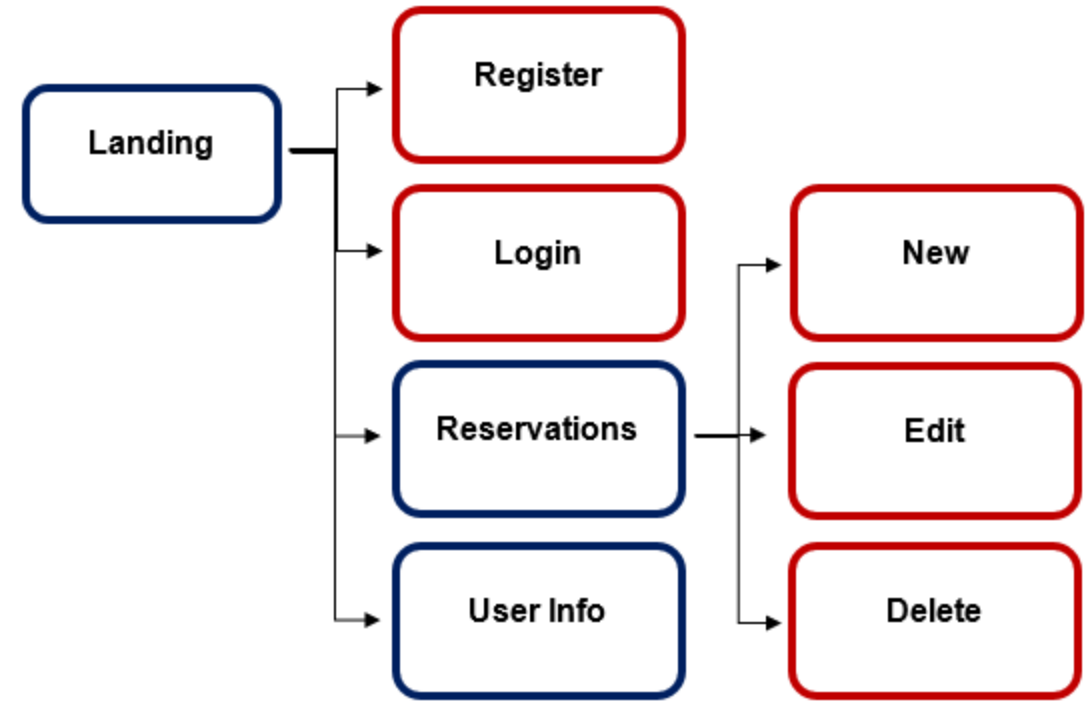
2. Use Case diagrams for each functional requirement. (15)
   - **Graphical Interface**

Pages

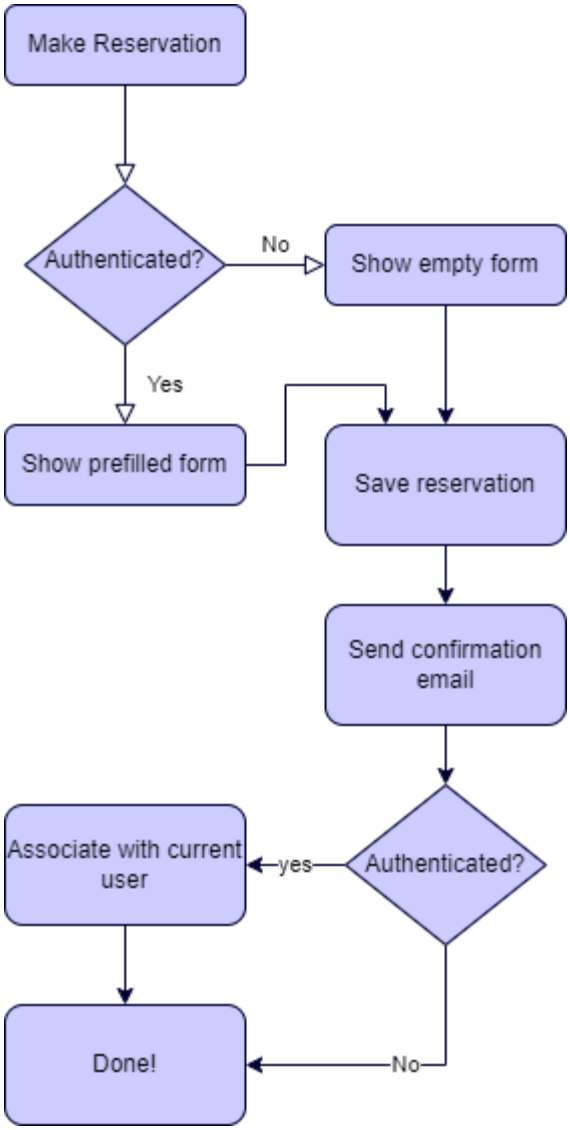
- **Landing**
  - **User Info**
  - **Reservations**
    - **New reservation**
    - **Edit Reservation**
    - **Delete Reservation**
  - **Login**
  - **Register**

LEGEND:

- Show page
- Form



- **Authorization and Authentication**



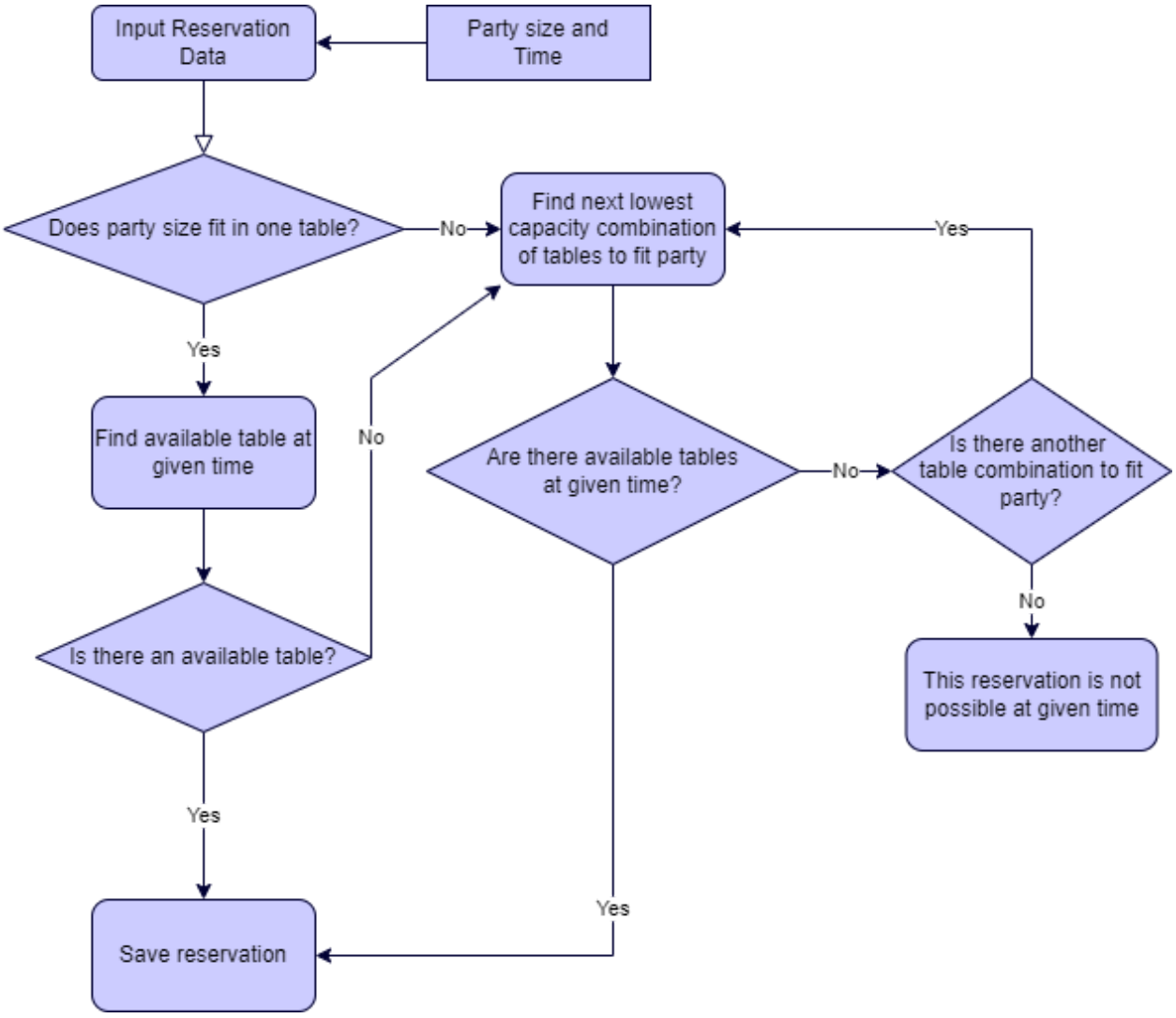
- **Database**

User		
Field	**Type**	**Reference**
Name	String	
Password	String	
Email	String	
Mailing Address	String	
Billing Address	String	
Preferred Dinner	Int	
Preferred Payment Method	String	
Points	Int	
Reservations	Array	Reservation

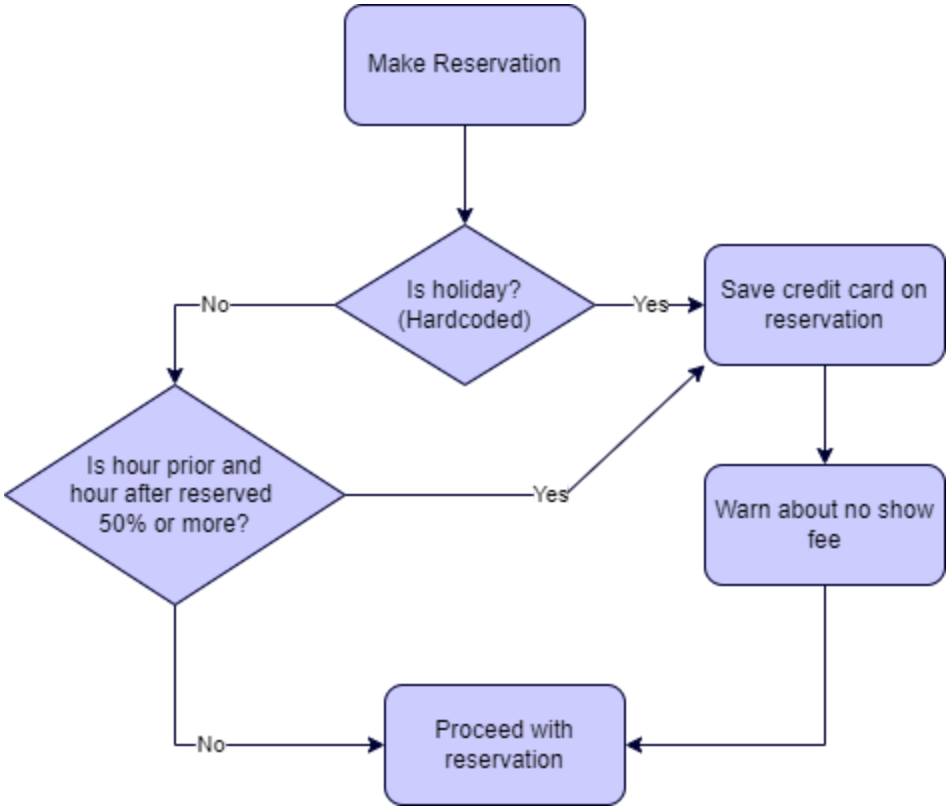
Reservation		
Field	**Type**	**Reference**
Start Time	String	
End Time	String	
Tables	Array	Table
User	Object	User
Party Size	Int	

Table		
Field	**Type**	**Reference**
Capacity	Int	

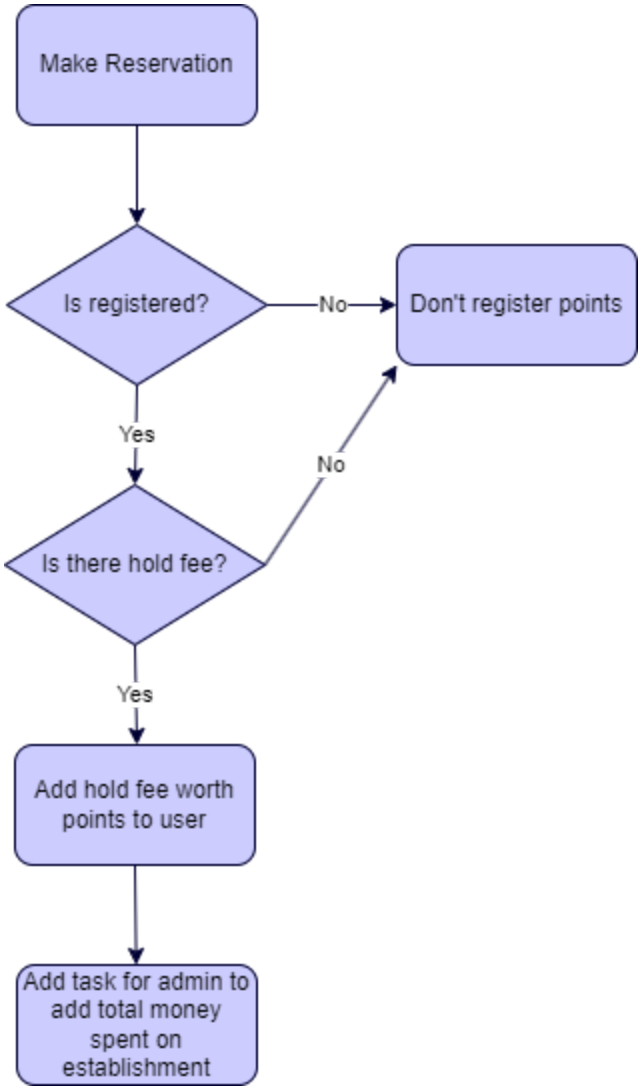
- **Reservation handler**



- **Traffic monitoring**



- **User's point system**



REQUIRED:

Fill in this table, and provide as many details as possible:

Group Member Name	What is your contribution?	Discussion Notes
1. Bryan Alvarez	Discussed, analyzed and determined functional and non-functional requirements. Created Diagrams	
2. Caleb Rogers	Discussed, analyzed and determined functional and non-functional	Functional requirements would include the more “surface” level details and

	requirements. Created Diagrams	immediate features. Non-functional requirements are the backend/server side necessities that the website needs to operate and run.
3.		

What to turn in:
- Only soft copy uploaded on or before the due date.
- No extensions.
- To get full credit provide details and diagrams (when appropriate).