**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 good user experience, please list it.

**Answer these questions:**

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

**Functional**

* user should be able to register
* user should be able to search for available tables and make a reservation
* user must provide name, phone, email, date, time, and number of guests so that system can return the available table if any
* system should not allow user to reserve a table with the number of guests higher than the limit
* system should be able to store registered customer info in database
* generate a daily, weekly, monthly, and yearly report sales
* register customers shall earn points for every dollar spent
* hold a fee on July 4th and require a valid credit card on file
* notify user about the minimum $10 fee for no showing up to reservation
* system should suggest user to register before confirming reservation if user is not registered
* notify user before and during high traffic days

**Non-Functional**

* system should be able to fetch data from database in less than 1 seconds.
* User Interface must be friendly and easy to navigate.
* Handle high traffic on busy days
* Encrypt customer information and protect from cyber attacks
* Different browsers and/or operating should be compatible with the system
* Daily backup should be quick

1. Use Case diagrams for each functional requirement. (15)

user

User should be able to register/login

User should be able to enter personal information and # of guests

System should present available tables depending on guest # and combinations of tables available

If registering, system should save the user info. If logging in system should prefill data based on user.

If not registered, user should be prompted to register

If registered, points should be added to the users account

Should be able to view weekly, monthly, and yearly sales reports

Should be able to notify all the users when there are high traffic days before and during and validate credit card info on those days and notify a $10 no show fee as well

Manager

System

**REQUIRED:**

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

|  |  |  |
| --- | --- | --- |
| Group Member Name | What is your contribution? | Discussion Notes |
| 1. Kenneth Easo | Conducted research about the project and other projects alike to gather information and make inferences on what is required for the app, as well as created diagrams for the requirements. | Although there were a lot of information out there we picked and chose the ones that had the most information that we can work off of without going too in depth or off topic. |
| 1. Bryan Tieu | Provided valid points on the reasoning of which information is valid in our search for what is required for our app and provided information on the diagrams | While researching I noted that our requirements aren’t very complex and made some points across what we needed to stay on top of. |
| 1. Jorge Ngimbi | Started to research processes and collaborated with the group to create the diagram and fill in the information. | When researching I had to make a note that my team members and I should be on the same page. |

Github: <https://github.com/BryanTieu/4351Project>

**What to turn in:**

- Only soft copy uploaded on or before due date.

- No extensions.

- To get full credit provide details and diagrams (when appropriate).