Assignment 4

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.
  - Users don'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 a maximum capacity limit i.e., 2, 4, 6, or 8.
  - Different combinations are allowed, and the owner accommodates the seating, for example: someone requests 8 guests and a table for 8 is not available but 2 + 6, or 4+4 is available. System should combine the tables and notify the owner they need to combine tables. In this case the System reserves both tables.
- If a guest user i.e., not a registered user, the system should prompt the 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 a valid credit card on the system to reserve the table.
  - Notify the user that no show will have a minimum $10 charge.

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

Answer these questions:

1. (*25 points*) Provide detailed project planning tasks document related to above requirements. Document must include:
   - Project scope
   - Risk analysis and recommendations
   - Implementation plan (discuss development methodology, technologies, etc)
   - Project schedule (What will be delivered and when. Use weeks.)
   - Solution architecture (diagram & discuss each component/module)
   - Testing plan
   - Quality measurement plan
   - Deployment plan (use diagram if needed)
   - Maintenance plan (who will maintain and how customers log defects)

Project Scope

Goals: Create an app for a restaurant chain which will allow them to have their customers reserve tables ahead of time. The app should be able to suggest options and track data for analysis purposes and customer experience.

Deliverables: The app itself, backend information, information that programmers need to maintain the system

Tasks:

> Create a front end that can- Allows users to reserve tables, allow users to register, allow users to enter payment and personal information
>
> Create a backend that can- Keep track of all the users, keep track of high traffic days, triggers to notify users about events, keep track of user's reward points

Costs: Paying our development team, paying for any project management software, costs included in hosting the app

Deadline: 5 months to complete the whole app

Risk Analysis and Recommendations

Risks associated with creating a mobile app: The app will be subject to many updates and patches depending on what devices the app will need to support. The more devices we choose to support, the longer it will take to deploy and more maintenance will be needed after it is done to keep up with device updates, etc.

Another risk would be data leak. If we decide to store customers' credit cards on file there is a risk that such sensible information can be leaked which would be a big liability to us. The solution would be to use an API such as Square or Stripe.

Recommendations: Create a webapp that anyone can access from a browser. This will lead to much less maintenance in the long run and will allow our team to create an application much faster since we are only focusing on one direction for this to work.

Implementation Plan

We will first complete the modeling for the entire app. This means designing out the project into the classes/entities that will be involved and defining their attributes as well.
From there, we will begin working on the front end first. By getting the UI finished, it means we will have all the functionality of the app done from the user's perspective, but the data will not be saved just yet. The front end will be created using Javascript. Depending on what is best, we may use React to create it, which is just a JS framework.
After finishing the front end, we will work on creating the backend, first just the modeling and creating all the tables so that the data has somewhere to go. We will create the backend using PHP and MySQL.

The final step will be connecting the two. We must make sure that changes in the front end are reflected in the back end. So when a user reserves a table, it causes the backend to allocate those seats for the user. When they register, it should create an entry for them in the backend, etc. This step we will also focus on refining all aspects of the app and making sure it is ready for the client.

Project Schedule

WEEK 1: Information gathering and data modeling

WEEKS 2-6: Creating the front end, fully fleshing out functionality.

WEEK 6-7: Reviews with the client to make sure the front end is what they asked for, these reviews will also periodically happen but this will be the first major review. We will adjust accordingly.

WEEKS 8: Creating the design for the backend.

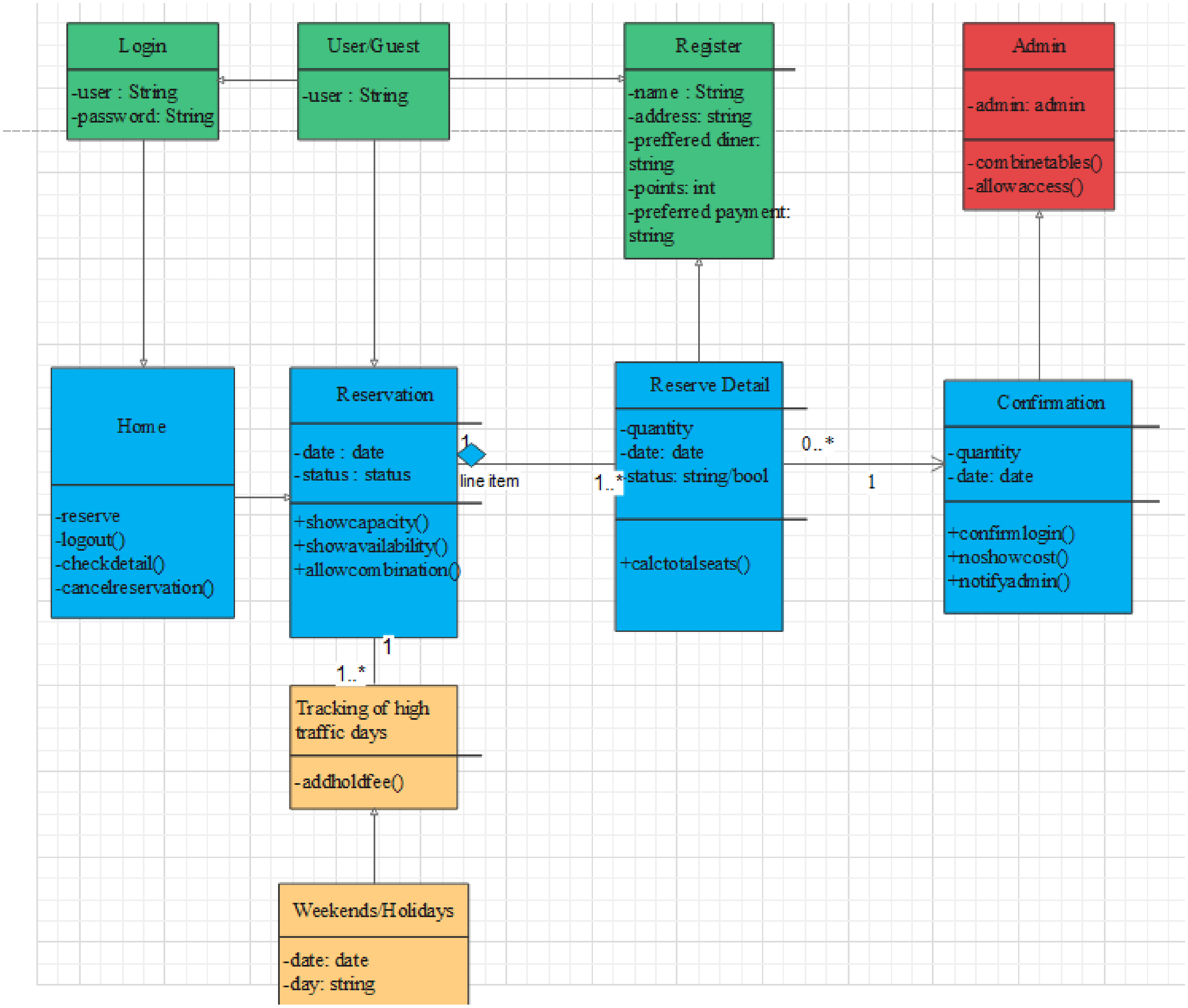
WEEKS 9-13: Creating the backend itself. Make sure it can be accessed and updated.

WEEKS 13-16: Connecting the front end to the backend, making sure that all interactions on the front end are reflected in the backend.

WEEKS 16-19: Styling and making the app as user friendly as possible. The styling we do will be based on what our clients want. This stage will require an agile approach as we will be constantly meeting with the client to ensure the product is how they want it to look.

WEEK 20: Final touches and any adjustments that need to be made will be done.

Solution Architecture



Testing Plan

We plan to create the application through test driven development. This means before we implement functions, we will first test them out through unit testing. So we will have unit tests for every aspect of our front end, the goal is to have 100% code coverage from our tests. Our unit tests will be automated in that they will be running continuously, and if there are any test failures, we can see it immediately and fix accordingly.

Quality Measurement Plan

For our project we plan to keep an eye on a couple of things noteworthy. Overall we want our app to be practical or user friendly and with that comes with code quality, we want to keep our code readable and minimal as possible to avoid bugs and not only will it give us an easier time in the development process but it would also give us positive performance times as well as scalability. To ensure that we have built a quality product we will perform product testing, maintenance and look for ways to constantly build upon our product

Deployment Plan

The number one important constraint is time. Time needs to be accounted for when developing our app. We need to have a clear goal in mind that is not only realistic but also efficient in resources. Next is the risks involved in the development process and the end product after timeline. We should be aware and document these risks that could impact our deadline and deployment. Next is to separate the tasks appropriately to certain members or teams. Doing so properly will ensure efficient progress throughout the development and testing procedure. Probably one of the most important things will be our resources. What will we need in order to carry out our project. We should document all of the hardware and software we need to be using to complete our app; this goes hand in hand with budget costs and overhead. Overall good communication within the team is needed, communicating with each other on what our assignments and deadlines are is very important to meeting time constraints.

Maintenance Plan

We will have two approaches to maintenance:

> We maintain the app- This would mean that we would be in charge of taking care of the app from here on out. We would handle the hosting of the application as well as any updates and changes that need to be made. The client would pay us a fee and send requests as needed. Depending on the request we will charge accordingly. We will be performing corrective software maintenance to fix any bugs and customer reported defects that might impact the performance of the system. We will also use preventive software maintenance and adaptive software maintenance to make sure our product keeps working as desired. This might include changing operating systems, cloud storage, hardware etc.
>
> They maintain the app- We would hand over the code and the application and leave it up to the client to maintain and manage the app. This would mean that the client already has some kind of in-house development staff that can maintain it. We will always be open for requests but it would require a fee for us to even get a consultation, and any work would be a fee as well.

NOTE: you completed some of this in previous assignments. Now put it all together in this assignment.

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

REQUIRED:

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

Group Member Name	What is your contribution?	Discussion Notes
1. Kenneth Easo	Created the Doc and filled out The risks and implementation, recommendations and project schedules	Researched project risk management as well as looked up recommendations on project practices and set a realistic time frame if this were a huge project.
2. Bryan Tieu	Created Solution Architecture, helped with writing the plan and implementation	I looked up a solution architecture diagram and ultimately created a similar diagram using the examples on the draw application. I also looked up what are common practices in the planning and implementation procedures in development phases.
3. Jorge Ngimbi	Helped formulate the risk analyzes and the maintenance plan.	Searched for the best software maintenance plan that best fits our project and the risk of storing customers credit card information.

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).