**Assignment 1**

**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.

You are asked to recommend a software development process for this system.

**Answer these questions:**

1. Is the waterfall model a good choice for developing this system? Provide detailed reasoning. (5 points)

Waterfall is not the best choice for this system. The waterfall method takes a long time to produce working software and get feedback from the users. It leads to too much time spent planning without time spent implementing thus unforeseen design challenges are bound to arise. In order for a team to design and build software in an efficient and cost effective way they must be quick to react to changes to requirements or due to feedback from users. Often times most project requirements will not be clear until the team tries to implement them and the model does not accommodate flexibility very well.

1. Is iterative development leading to a single release of the system a good choice for developing this system? Provide detailed reasoning. (5 points)

Iterative development is better than something like waterfall but working towards only a single release is why this method is also not the best one to use for this system. Iterative development is the cyclical process of analyzing, designing, implementing, and testing. This style of development has advantages to it in the agility of the team to make changes and decreased costs over a more rigid development process like waterfall. However, iterating towards only a single release takes away much of this benefit as the information gained with each iteration will be limited to internal testing and feedback not feedback from the actual users of the software.

1. Is agile development with releases of small increments of software a good choice for developing this system? Provide detailed reasoning. (5 points)

Agile is a good development choice for this system for a number of reasons. The big advantages of agile lie in the way that software development can quickly adapt to and embrace changes. With Agile, you are much more quickly able to get a minimum viable product and thus start testing and getting real world user feedback. Getting this crucial feedback early in the process of implementing and developing software decreases both the cost and time required. Cost is decreased because the number of man hours spent redesigning and implementing features will fall dramatically. Agility enables the software to adapt quickly as changes occur.

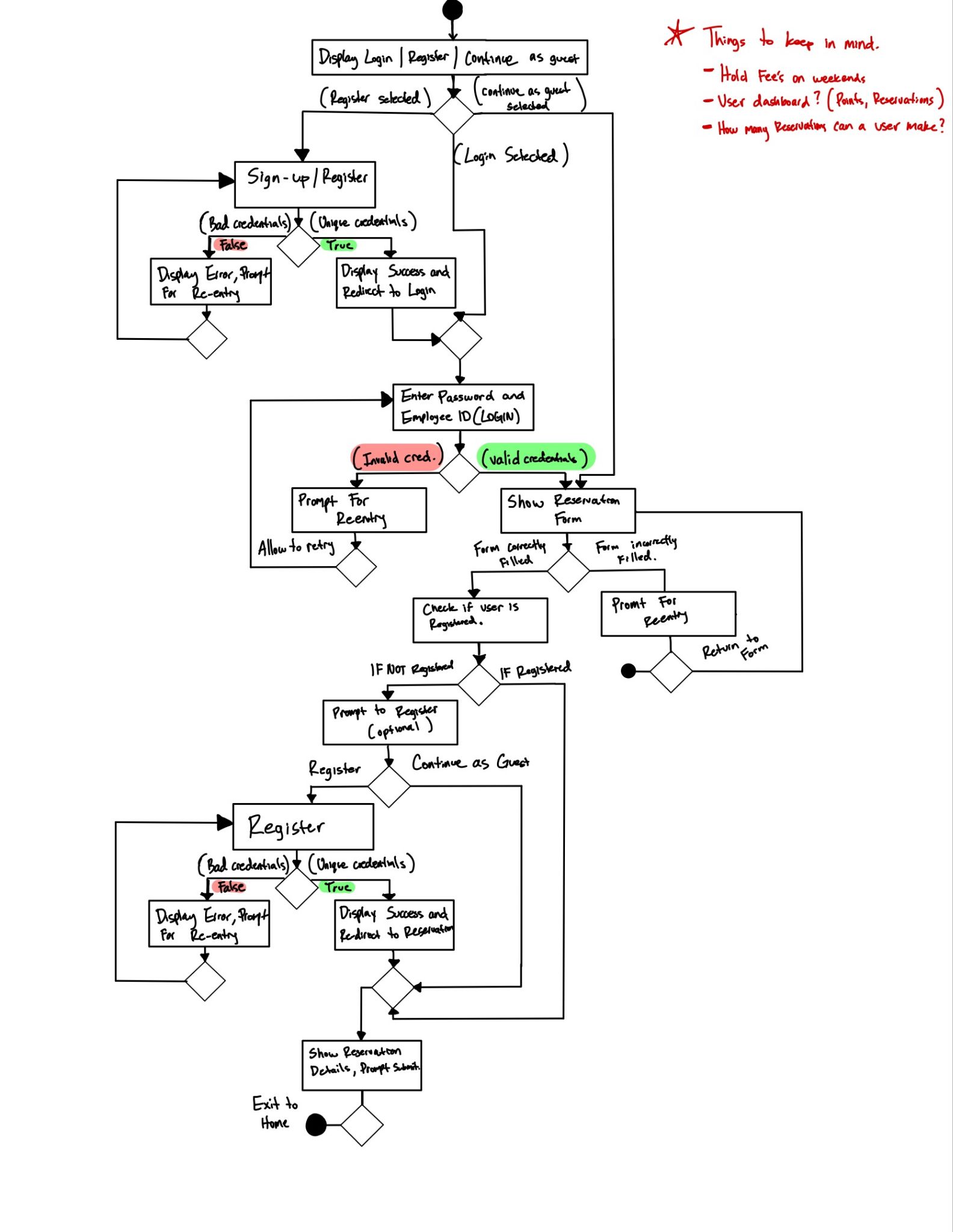
1. Discuss your first thoughts on how you will approach the solution? You should discuss in detail with reasoning. (5 points)

Approaching this problem, the first logical step is to break the application down into more manageable components while keeping things at a high enough level to understand and plan around the complexities and constraints of the system. This could involve designing and illustrating database schema and flowcharts for the application, as well as establishing a robust architecture to serve as the foundation for the application.

Breaking down the system into smaller components will also let us easily divide up the work. User sessions, user sign up, booking calendar, user auth, etc are all tasks that can be divided among group members. Starting with the base functionality and adding more of the requested features after we establish a minimum viable product.

Our group discussed creating this system with a Django backend web server and a React frontend.

Below is a flowchart illustrating an initial view of the system flow based on these design requirements.



**REQUIRED:**

Fill in this table, provide as much details possible:

| Group Member Name | What is your contribution? | Discussion Notes |
| --- | --- | --- |
| 1. Bryan Smith | Worked on questions 1-4 after meeting with group | We will likely be using django or a node backend which will be exciting do design with |
| 1. David Carlon | Created the Flow chart/ UML diagram. Talked with partners to make a very boilerplate answer which was further elaborated by partners. | Very good conversations were made over what technologies we will use. I am excited to be working on this project with my partners. |
| 1. Dhruv Shah | Proofread and edited Questions 1-4 after group members finished and added more depth to the answers where I could. | I am Eager to learn any new technologies/framework that we have decided to use for development and work with my teammates. |

**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).