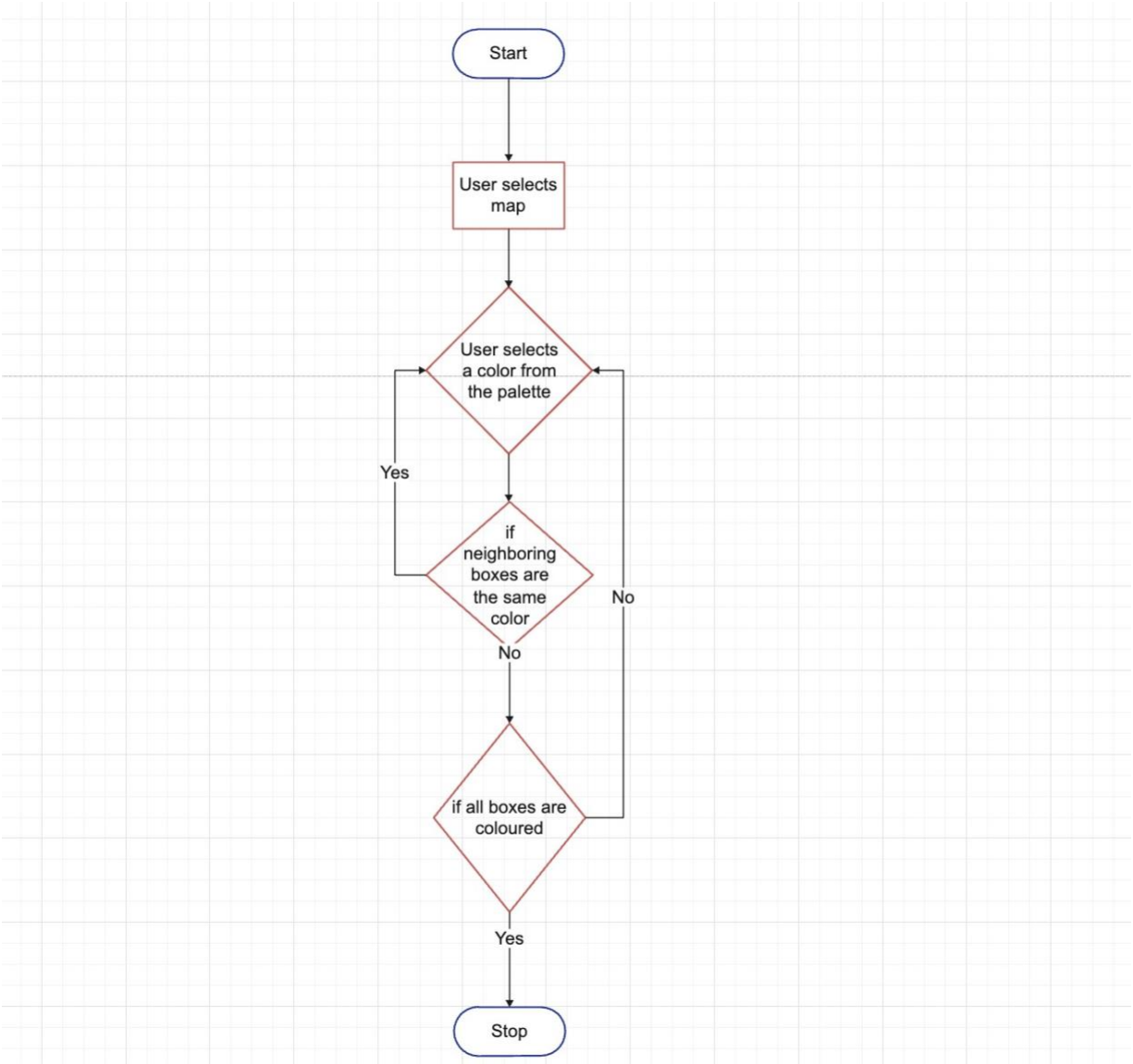
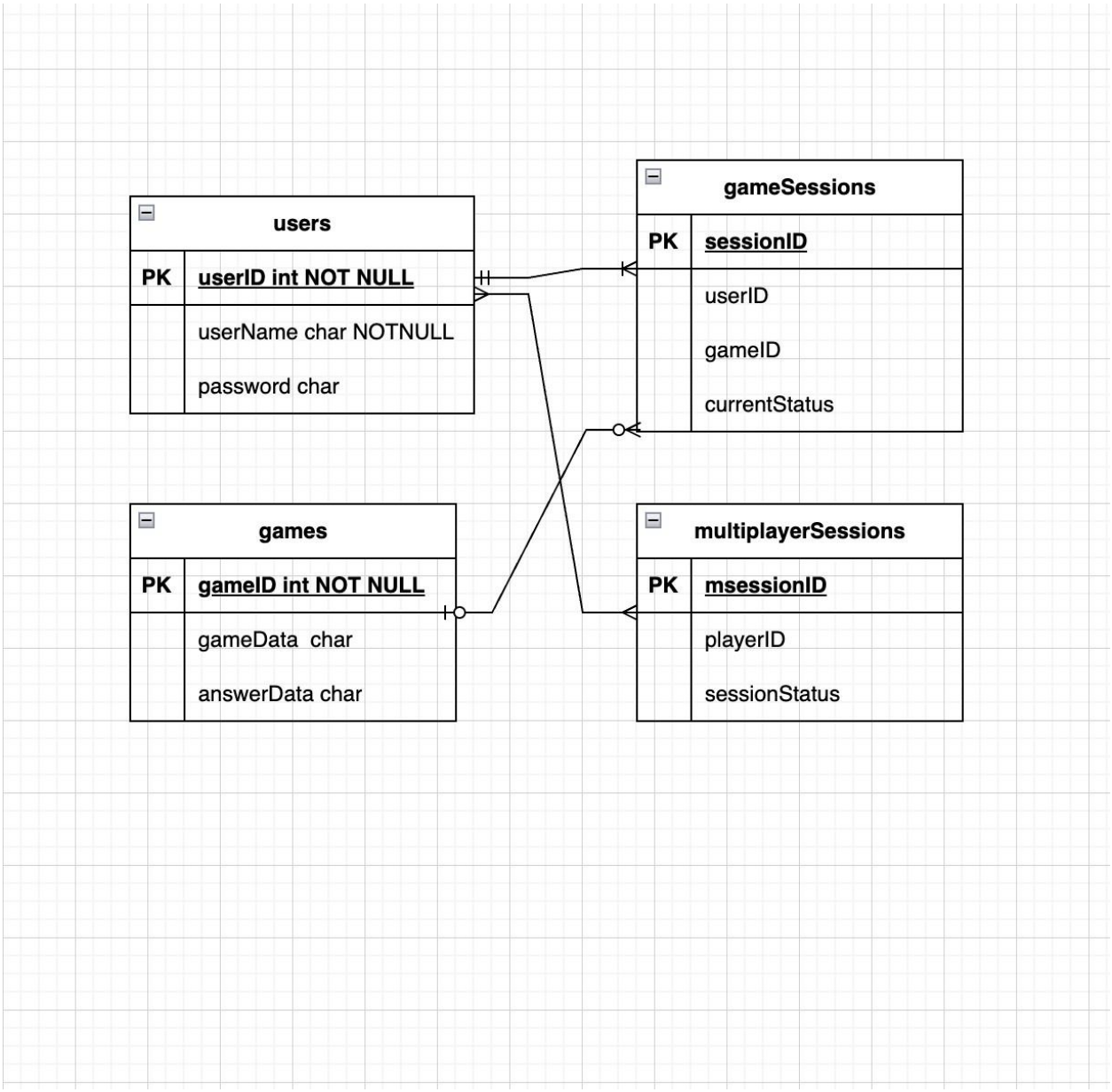
Design Specific Document

1. **Project Description:** The Color Mapping project is an online gaming pla6orm where users can engage in creatively coloring various shapes on a grid, adhering to specific rules concerning color selection, namely that neighboring boxes must have different colors. Access to the game requires users to log in or register. The main objective is to offer an immersive and stimulating experience suitable for players of all ages.

   Flow chart:



ERD Diagram:



2. **Methods used:**
   a) A simple method for solving the puzzle could be a greedy algorithm:

   Start with an empty grid.
   Iterate over each box in the grid.
   For each box, check the colors of its neighboring boxes.
   Color the current box with the first available color not used by its neighbors.
   Repeat until all boxes are colored.

b) Another approach is back propagation:
   The algorithm selects an uncolored region and tries all available colors for that region. If a color is valid it's assigned to the current region, and the function recursively calls itself. If the recursive call returns true, indicating that the map is successfully colored, the algorithm returns true. If no valid coloring solution is found for the current region and color, the algorithm backtracks by undoing the color assignment and continues trying other colors or regions. If no valid coloring solution is found for the entire map, the algorithm returns false.

c) selectColor(): for coloring the box
d) changeMap(): for displaying the next map
e) startGame() and stopGame(): for starting and stopping the game
f) saveSession(): stores session for each game

3. **Market space and selling points:**

a) Creativity: Allows users to express their creativity by coloring different shapes on a grid.
b) Challenge: Provides a challenge by requiring users to follow specific rules related to color selection.
c) Variety: Offers dynamically changing maps with varying shapes and sizes, enhancing the diversity of the coloring experience.
d) Engagement: Features game controls and map selection to ensure continuous engagement.

4. **Product features:**
   a) Signup: User must sign up to play the game.
   b) Signup: user must login.
   c) Color Palate: Offers a palate of colors for users to choose from.
   d) Difficulty level: Allows users to choose from a difficulty level, each with its unique size.
   e) Rule Enforcement: Ensures that neighboring boxes are not colored the same.
   f) User Progress Tracking: Tracks user scores and saves it for future sessions.
   g) Reset game: user can reset the game.
   h) Exit game: user can exit the game.
   i) Logout: user must be able to logout.
   j) Timer based: timer starts whenever the user starts the game.
   k) Display the solution: user should be able to view the solution after the timer’s up.

5. **Deployment:**
    a) Once the flask project is done, set up Heroku by logging in, creating a new app, and linking it to the Git repository.
    b) Add the PostgreSQL add-on to the Heroku app for database support.
    c) Configure any necessary environment variables for the app, such as database connection strings.

    d) Deploy the Flask application to Heroku by pushing the code to Heroku's remote repository using Git.

6. **Milestones:**
    a) M1 (Setup and Initial Design): Establish project structure, design database schema, and set up basic Flask and React applications.
    b) M2 (Main Algorithm): Implement the coloring logic with UI.
    c) M3 (User Progress): Store user progress, account management, and leaderboard.
    d) M4 (Testing and Deployment): testing and deploying the project.