|  |  |  |
| --- | --- | --- |
| **System Overview and**  **Design** | Ibrahim | Define the system  architecture, class diagram, and data flow. Ensure all requirements are  covered and understood before coding starts. |
|  |  |  |
| **Patient Class**  **Implementation** | George | Implement the patient class, defining attributes  like ID, gender, arrival time, and patient type (urgent/normal). |
|  |  |  |
| **QueueManager Class**  **Implementation** | Ibrahim | Implement the QueueManager class, which will  handle dispatching patients to queues, serving patients, and managing queue  priorities. |
|  |  |  |
| **SystemManager Class**  **Implementation** | George | Implement the SystemManager class to manage the  overall system, including handling the patient list, time simulation, and  invoking queue management. |
|  |  |  |
| **Dispatch Patients Logic** | Ibrahim | Code the logic for  transferring patients to the appropriate queue based on their arrival time  and urgency. |
|  |  |  |
| **Serve Patients Logic** | George | Develop the logic for  serving patients, including limiting the number of patients served each time  step and prioritizing urgent patients. |
|  |  |  |
| **Calculate Statistics** | Ibrahim | Implement the logic to  calculate statistics like average waiting time and number of patients served. |
|  |  |  |
| **Exception Handling** | George | Add exception handling to  manage out-of-bounds or invalid operations in patient lists or queues, using try-catch blocks. |
|  |  |  |
| **User Input and Time**  **Advancement** | Ibrahim | Implement user-triggered  time advancement (every minute), and ensure time simulation functions  correctly during the simulation process. |
|  |  |  |
| **Main Program (Driver)** | Both | Implement the main function to bring everything  together, simulate patient queue operations, and output the final statistics. |
|  |  |  |
| **Testing and Debugging** | Both | Test the system with  different patient scenarios (100, 300, 700 patients), fix bugs, and ensure  the system runs without crashes or logical errors. |
|  |  |  |
| **Documentation** | Both | Write the README, system  documentation, and comments within the code for clarity and ease of  understanding. |
|  |  |  |
| **GitHub Version Control** | Both | Regular commits to  GitHub, ensure each team member makes meaningful contributions. Proper commit  messages and code management practices. |