|  |  |  |
| --- | --- | --- |
| **Priority** | **Requirement** | **Description / Rationale** |
| **Must Have** | Sending data in JSON format through TCP | By using a neutral data format clients on different platforms can exchange data with the server. |
|  | Add User / Create Rental / Return Rental functions | Core system functions to manage user account creation and storage as well as rent and return games the system cannot operate without these. |
|  | In-memory data storage | Stores all runtime data for example users, games, rentals while the program is running |
|  | Basic dashboard (read-only client) | Gives the user the ability view the current state of games, and rentals and user details. |
| **Should Have** | Handling errors and Input validation | Provides clear feedback on errors and prevents invalid entries such as missing IDs or names. |
|  | Thread-safe data access | Avoids situations where issues can occur when multiple users are connecting to the server at once. |
| **Could Have** | Authentication / login system | Require login credentials for clients and would simulate a real environment. |
|  | Game stock, due dates, and late fees | Could add a more detailed to how the real-world application would run in an environment for an actual game renting service |
| **Won’t Have (for now)** | Web or mobile interface | Prototype focuses on command-line clients only so potentially adding in an interface could increase user satisfaction |
|  | Payment or authentication system | Out of scope for the current prototype phase but would provide a more real like environment if added to an actual game rental application. |