

Priti Lalloobhai			
<b><u>REQUIREMENT-ID:</u></b>	<b><u>UI CROSS-REFERENCE:</u></b>	<b><u>TYPE (F/NF)</u></b>	<b><u>Package/Class/Method</u></b>
<b>RQ-01</b>	The system displays the leaderboard to all players after each game.	<b>F</b>	summary.html – Leaderboard
<b>RQ-02</b>	The system retrieves leaderboard data from the database after game ends.	<b>F</b>	summary.js – handleLeaderboardData()
<b>RQ-03</b>	The system uses WebSocket to receive leaderboard updates from the Java backend.	<b>F</b>	summary.js – WebSocket onmessage
<b>RQ-04</b>	The system formats and displays leaderboard data using HTML/JS only (no game logic in JS).	<b>F</b>	summary.js – renderLeaderboard()
<b>RQ-05</b>	JSON must be used for all communication between frontend and backend.	<b>F</b>	summary.js – message parsing
<b>RQ-06</b>	The Summary UI section must be part of a single-page HTML interface.	<b>NF</b>	summary.js – message parsing
<b>RQ-07</b>	The Summary page must interface with Page Manager to be shown/hidden appropriately.	<b>F</b>	summary.js – showSummary() function
<b>RQ-08</b>	Each Summary client must have a unique identifier.	<b>F</b>	summary.js – showSummary() function
<b>RQ-09</b>	The leaderboard must auto-update when new game results are available.	<b>F</b>	summary.js – WebSocket onmessage
<b>RQ-10</b>	The leaderboard shall display player statistics such as number of wins and losses.	<b>F</b>	summary.html – Table columns