Iteration #3 - Addressing Quality Attribute Scenario Drive (QA-3)

Step #2: Establish iteration Goal By Selecting Drivers

For the third iteration, we are focusing on QA-3 quality attribute, which is Availability. The system should be able to handle login attempts for a minimum threshold of at-least 20+ users without causing disruptions, if there is, it should be able to continue operation within under a minute at most.

Step #3: Choose One or More Elements of the System to Refine

The elements that will be refined are the nodes identified from the diagram created in Iteration #1, are which:

Login System
Hi-Score Tracking System

Step #4 - Choose One or More Design Concepts That Satisfy the Selected Drivers

Design Decisions and Locations	Rationale
Introduce a **monitor** tactic to detect possible faults that could occur, by checking to see the state of health of other components.	By monitoring the important elements within the system, it will make sure the chance of failure is seen as early as possible and create a solution to avoid an extended period of downtime. Impacting QA-3 directly.
Use of **Software Upgrades** to recover from the faults and push out upgrades/updates to maintain integrity of software.	When required outside of the normal updates, if there is a failure detected by the monitor, the system can already be prepared with a solution that can be implemented via a software upgrade to make the executables run smoothly.
Using **Shadow** tactic to maintain uptime and availability of software while the upgrade is in process.	This will allow the component that has issues to be running in “shadow mode” while it is being upgraded from the failure into an active role after. It further allows availability of the component to run while the upgrades will fix the cause of failure.
Making use of a **Predictive Model** to help monitor the state of health of the system.	This will ensure that the system will be operating under the normal operation conditions, and if there are any issues that occur it will act on conditions that may cause future faults.
Adding **Degradation** to avoid total system failure.	This will maintain the integrity of the critical components of the system to avoid total failure by dropping less critical components to keep the main critical components up and free of failures. This impacts the login system and hi-score system.

Step #5 - Instantiate Architectural Elements, Allocate Responsibilities

Design Decisions and Locations	Rationale
Deploy a monitor on the Login System	The login system is a very crucial component that needs to be monitored to avoid failures, as users can access the software only by logging into the system using their credentials. If the login system is constantly monitored for chances of failures, it will make sure UC-3 is constantly in check while also performing QA-3 (Availability).
Implement a Predictive Model and Monitor in the Login System component	In the chance that the failure is detected within the login system, using a predictive model will allow for correcting these failures by taking action that will prevent it from happening.
Implement Software Upgrades that will balance failure detections	Because of the use of the Monitor and Predictive model in case any failure is detected and is not able to be prevented on other system components - the use of the Software Upgrade will be used to correct the issues right away without major impact to the software.
Use Degradation to balance and minimize the effects of failure on the system.	Alongside the use of Software Upgrades, Degradation is used as another method to maintain the integrity of the critical components and dropping functions that are not as critical to make sure the software does not fail completely and go into downtime.

Step #6 - Sketch Views and Record Design Decisions

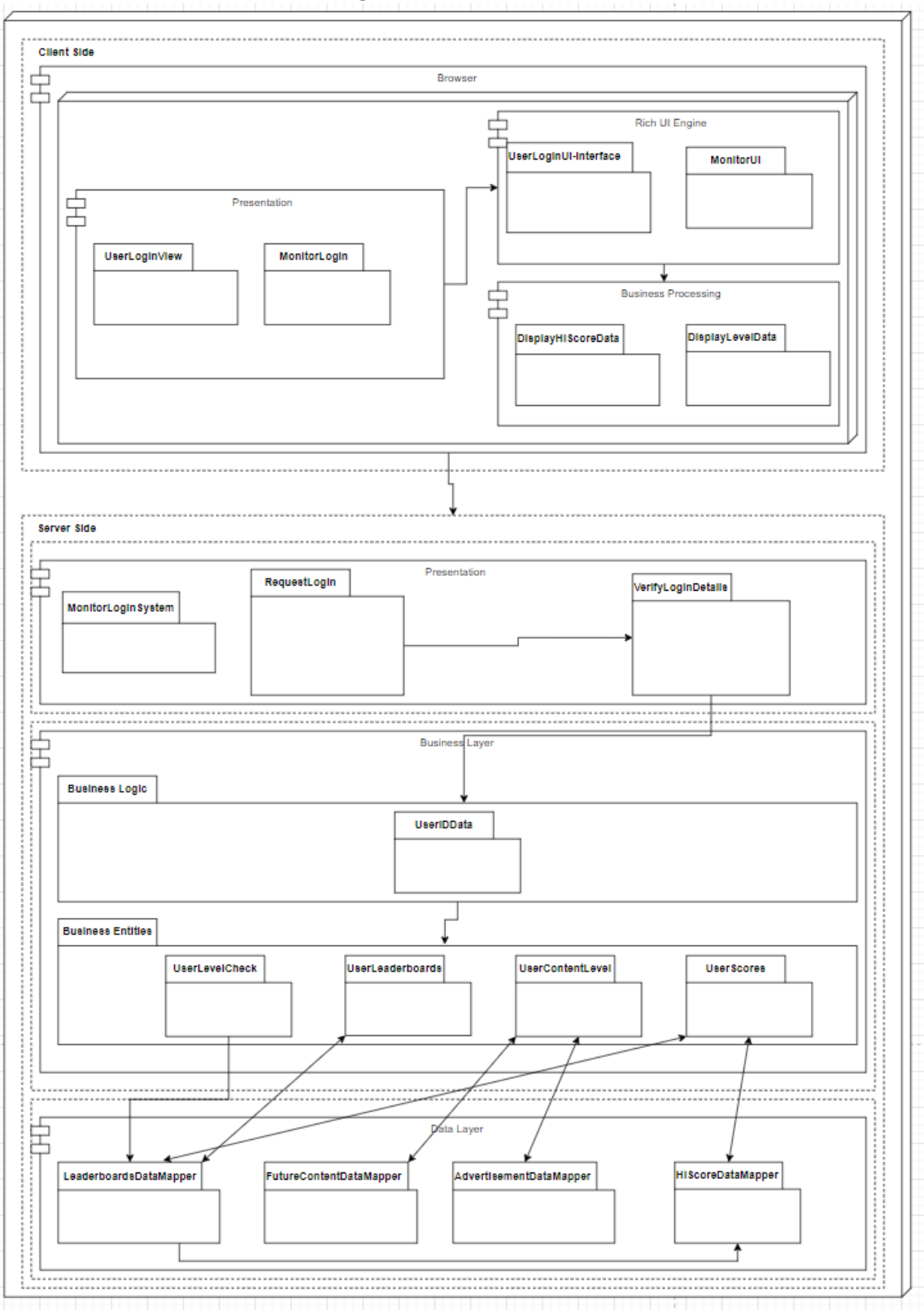

Figure 1: Refined Architecture Diagram with Monitor Tactics (Client Side: Presentation & Rich UI – Server Side: Presentation)

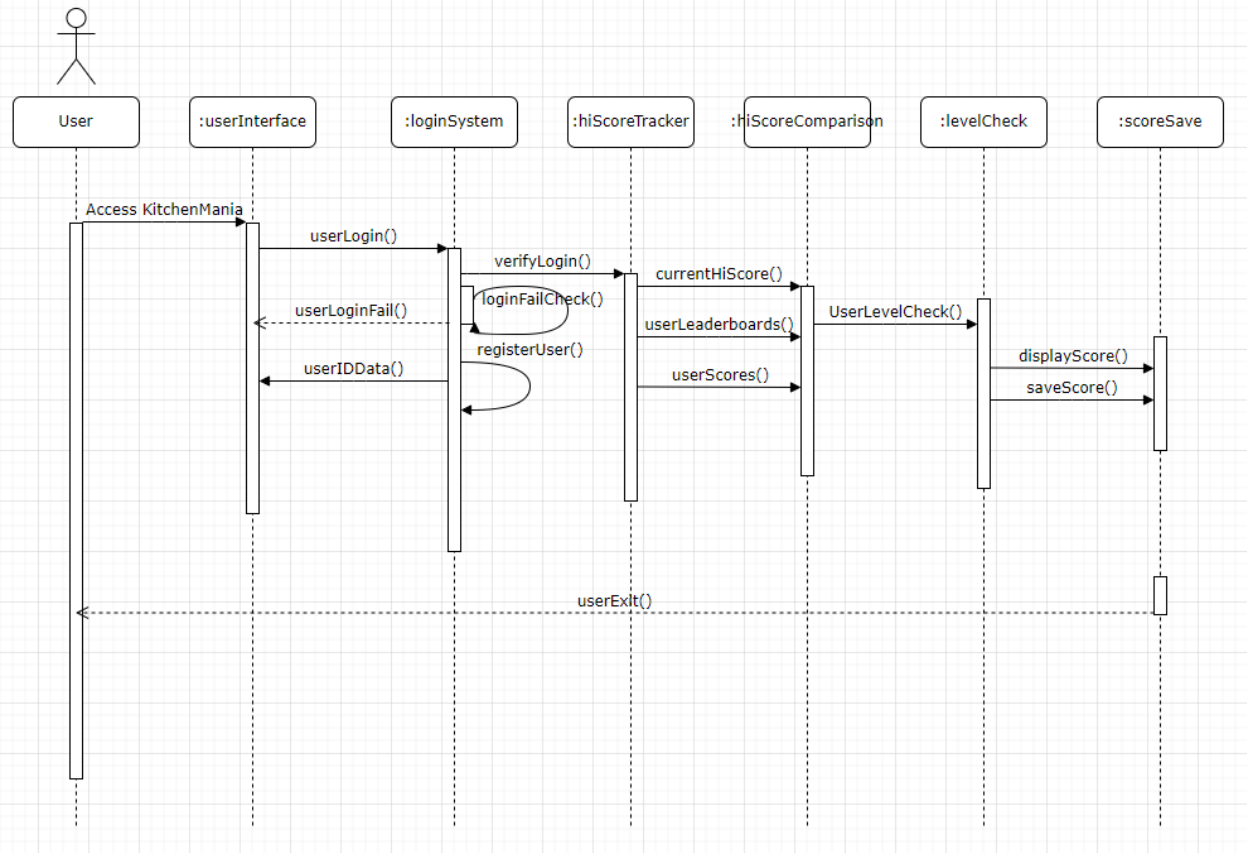

Figure 2: Refined Sequence Diagram from Iteration #2 (Contains Predictive Model within Login System)

Step #7 - Perform Analysis of Current Design and Review Iteration

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions
	QA-1		The introduction of the design concepts within the Availability tactic has an impact on the performance of the system due to ensuring that the failures are monitored and will be dealt with without dropping performance of the software unless required for big software upgrades.
QA-2			**No relevant decisions made.**
	QA-3		By creating a predictive model and using a monitor, the Login System will be made sure to avoid any possibility of complete failure by looking for signs of pre-failure and adjusting to those right away to create a solution to combat the issues.
QA-5			**No relevant decisions made.**
	QA-6		**No relevant decisions made.**

		CON-2	Monitoring of the login system and using the predictive model ensures that the software avoids failure and has back-up in place if it does.
	CON-3		Current design decisions have an impact on login ability to ensure that it can handle the threshold requirement as they are all directed towards QA-3 (Availability), as such this constraint ties directly with the ability of the system to handle the number of users accessing the system at once.
	CRN-2		**No relevant decisions made.**