Iteration #2 - Identifying Structures to Support Primary Functionality

Step #2 - Establish Iteration Goal by Selecting Drivers

The focus of this step is to select the system's primary use cases that are used to support primary functionality. The following are the selected drivers:

- UC-2
- UC-3
- UC-5
- UC-6
- UC-7

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

The focus of the elements that will be refined in this iteration will be based on the two reference architectures selected within Iteration #1, and they are the following:

- Rich Internet Applications (RIA)
- Web Applications

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

Design Decisions and Locations	Rationale
Create a Domain Object for user functionality and access to other functions (Hi-score checks, level checks, etc.)	The domain object will be unique and distinct in terms of ID purposes, but they are used to access the main portion of the software. Without these domain objects, there will not be an efficient way to monitor login and create a hi-score table respective to user scores.
Create a Data Mapper, which is a Database Access Layer to connect user credentials to the Login Database and the Hi-score database	The domain objects created need to be used to access a database which will contain the storage and information of the respective objects hi-score and level information if the credentials are verified and exist within the database.
Use Swing Framework and Java Web Start Framework	The Swing framework offers functionality and support for application development that supports the needs of the use cases designed, as this software requires menus, button toggles and movement of objects across a ‘game board”. As it is also a web-based browser game, the implementation of the Java Web Start Framework would be critical to this.

Step #5 - Instantiate Architectural Elements, Allocate Responsibilities

Design Decisions and Locations	Rationale
Creating a Domain Object for user functionality	This Domain Object is designed to uniquely identify users to accept credentials and allow access to functionalities like login, hi-score checks, etc.
Data mapper designed to connect user credentials to Domain Objects.	The Domain Objects created will be used to be mapped to the respective databases that contain the relevant information once verified, this addresses CON-3 and CON-6.
Using the **Java Swing Framework** to design client application functionality and combo functionality.	Implementing Java Swing Framework allows for combo functionalities that only allow users to move the objects within 1 block moves (up, down, left, right and diagonal). Can also implement other Java technologies to give UI functionalities.
Application is deployed using **Java Web Start Framework** to create software with systems relevant to this iteration.	Java Web Start Framework will make use of the reference architectures design to have a module that will enable user to access the software via web-browser and do any of the functionalities listed.

Step #6 - Sketch Views and Record Design Decisions

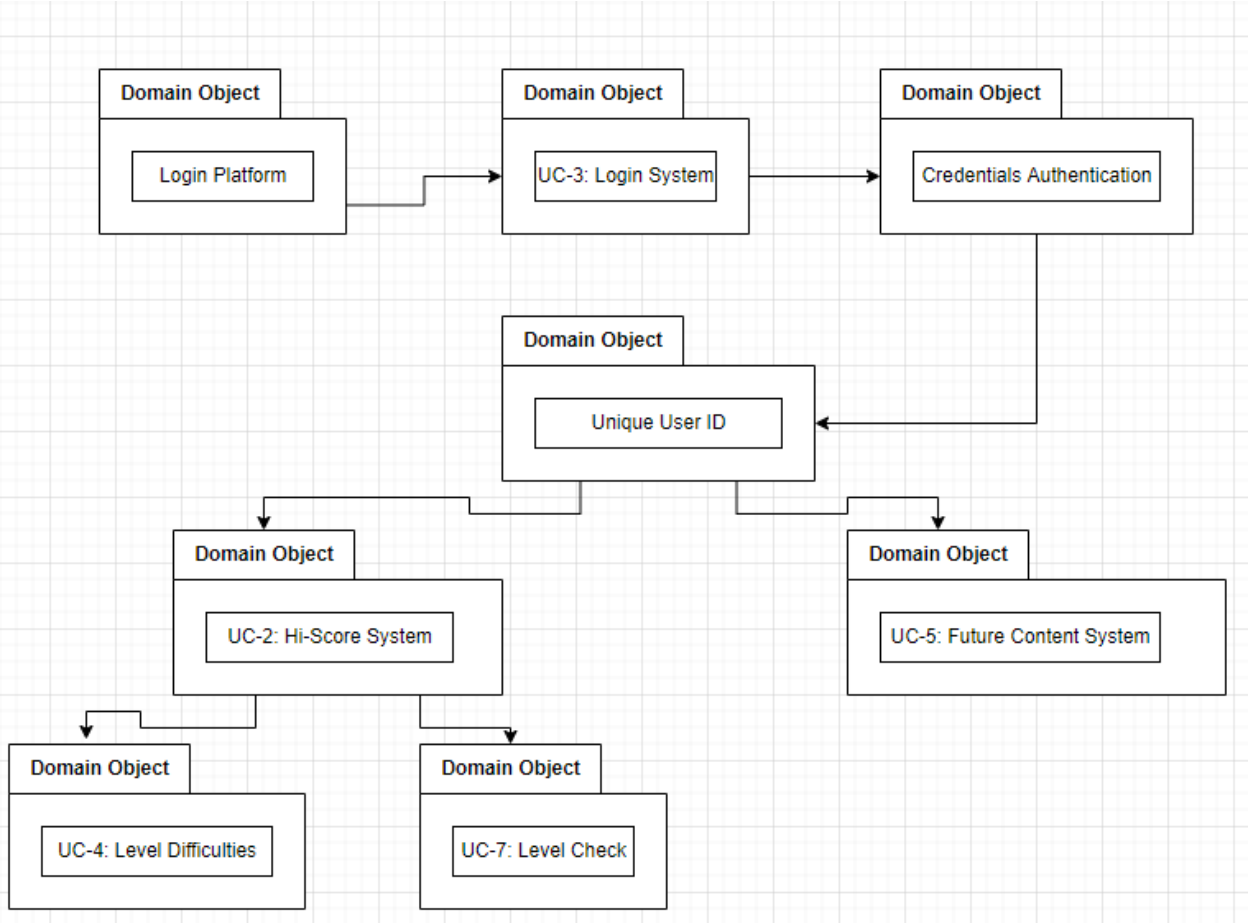

Figure 1: Domain Objects Diagram (User functionality)

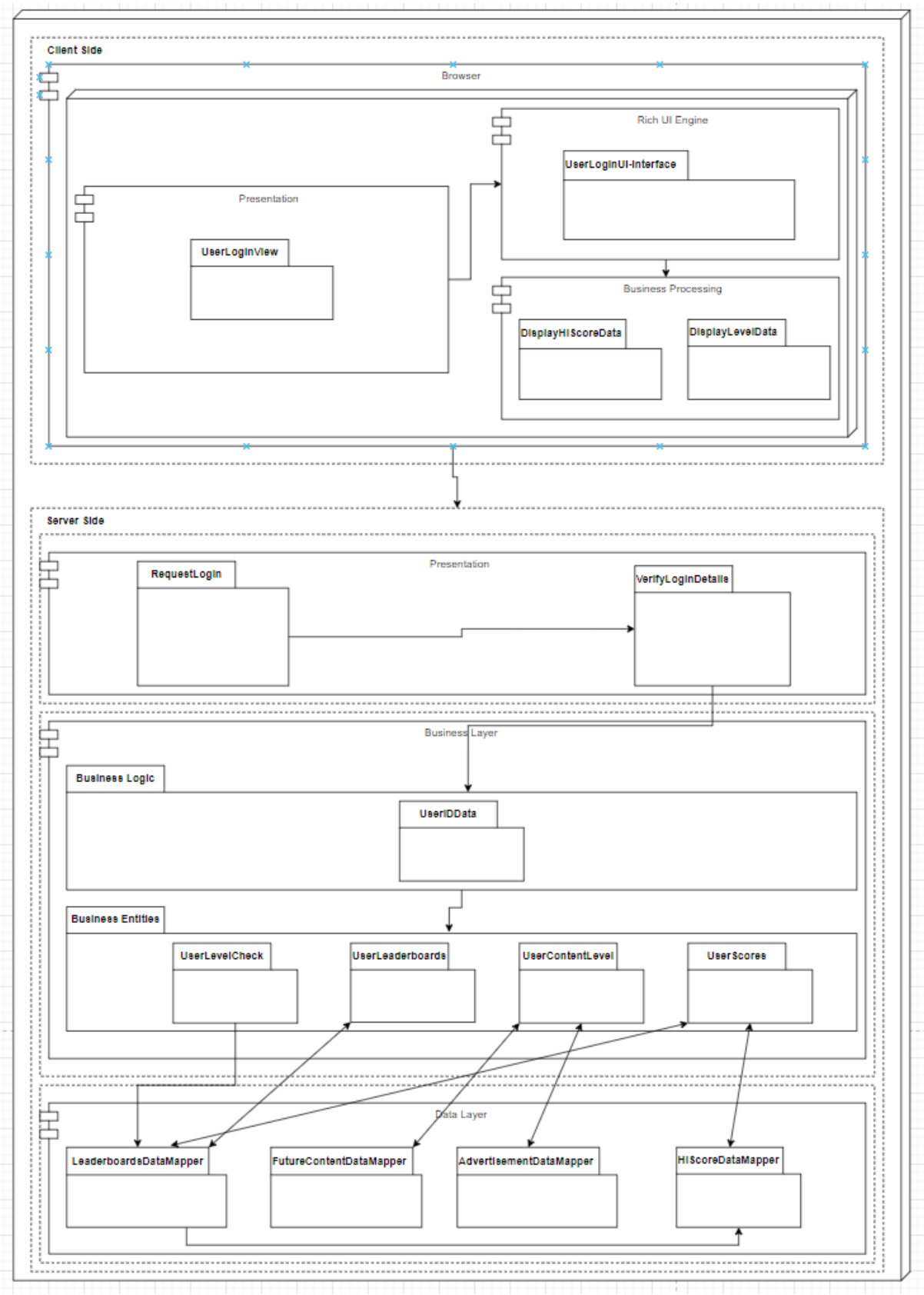

Figure 2: Modules for Data Mapper and Domain Objects for Use Cases

Element Name	Responsibility
UserLoginView	Shows user the login page, where they can register or login.
UserLoginUI-Interface	User is shown the visible interface to login into the system, they can also register, and it will take the user to another webpage to complete the registration.
DisplayHiScoreData	Responsible for showing user ID their hi-score data based on completed levels.
DisplayLevelData	Responsible for showing user ID their current level and levels cleared until this point.
RequestLogin	Requests login validation based on user credentials entered, it will either accept or reject the credentials based on the details from the login database.
VerifyLoginDetails	Login details are verified through the login database by parsing the login details from the previous element.
UserIDData	All available user data (levels, hi-score, content level, etc.) is shown to the user with the ID provided.
UserLevelCheck	User can view their current level and what levels they have cleared already.
UserLeaderboards	User can view their hi-score based on the leaderboards (global rankings).
UserContentLevel	User can view their current content features (limited edition levels, icons, etc.).
UserScores	User can view their current score and the score obtained at levels cleared.
LeaderboardsDataMapper	Responsible for events related to the leaderboards.
FutureContentDataMapper	Responsible for events related to the Future Content System.
AdvertisementDataMapper	Responsible for events related to the Advertisement System.
HiScoreDataMapper	Responsible for events related to the Hi-Score System.

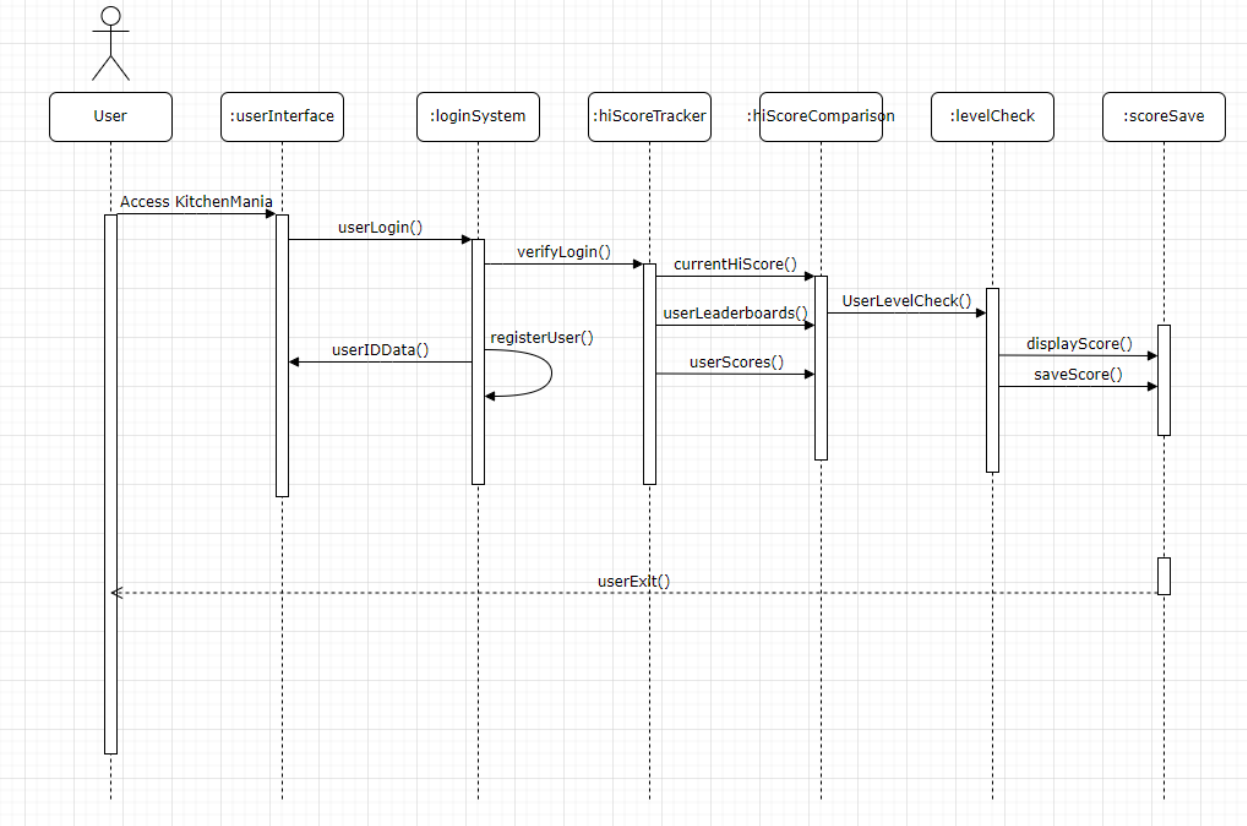

Figure 3: Sequence Diagram for Use Cases (UC-2, UC-3, and UC-7)

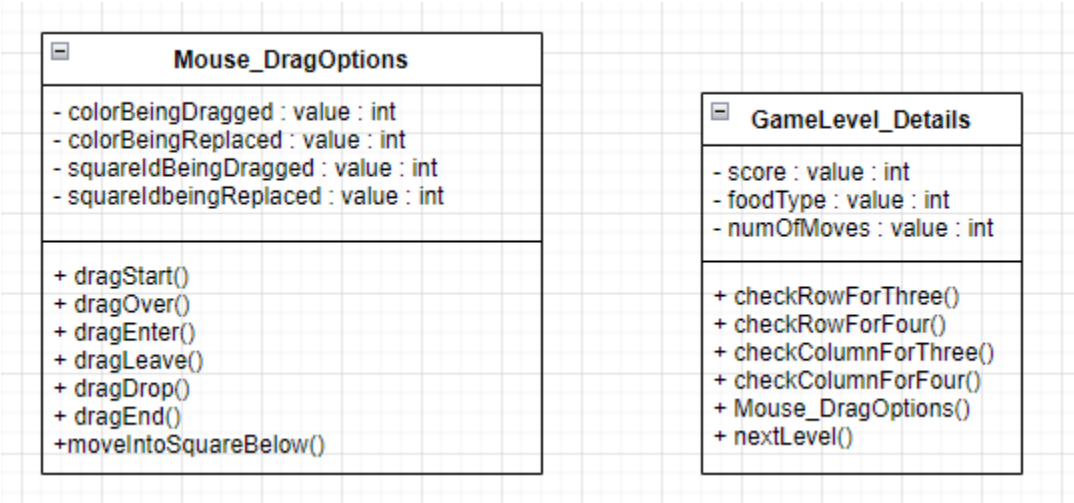

Figure 4: UML Class Diagram for Java Swing Framework

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

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions
		UC-2	Modules in the architectures have been created to support this use case, specifically through Domain Objects and Data Mapping.
		UC-3	Modules in the architectures have been created to support this use case, specifically through Domain Objects and Data Mapping.
	UC-5		Elements for this use case have been identified as part of the modules created for UC-2 and UC-3.
		UC-6	Modules in the architectures have been created to support this use case, specifically through Java Swing Framework.
		UC-7	Modules in the architectures have been created to support this use case, specifically through Java Swing Framework, Domain Objects and Data Mapping.

	QA-1		Elements that support UC-2 and UC-3 have been identified.
QA-2			**No relevant decisions made.**
	QA-3		Elements that support UC-2 and UC-3 have been identified.
		QA-4	Elements that support UC-6 have been identified that allow combo features that can be used.
QA-5			**No relevant decisions made.**
	QA-6		Elements that support UC-2, UC-3 UC-7 have been identified that allow user login credentials to be unique.
		CON-1	Modules available to meet constraint have been fully identified.
CON-2			**No relevant decisions made.**

	CON-3		Modules available to meet requirements of constraint through Domain Objects and Data mapping.
		CON-6	Modules available to meet constraint have been fully identified through design decisions.
	CRN-2		Modules have been identified to allow use of Java Technologies to incorporate team knowledge on functionality.
		CRN-3	Modules have been created and team members have allocated work given to them regarding the design decisions in this iteration.