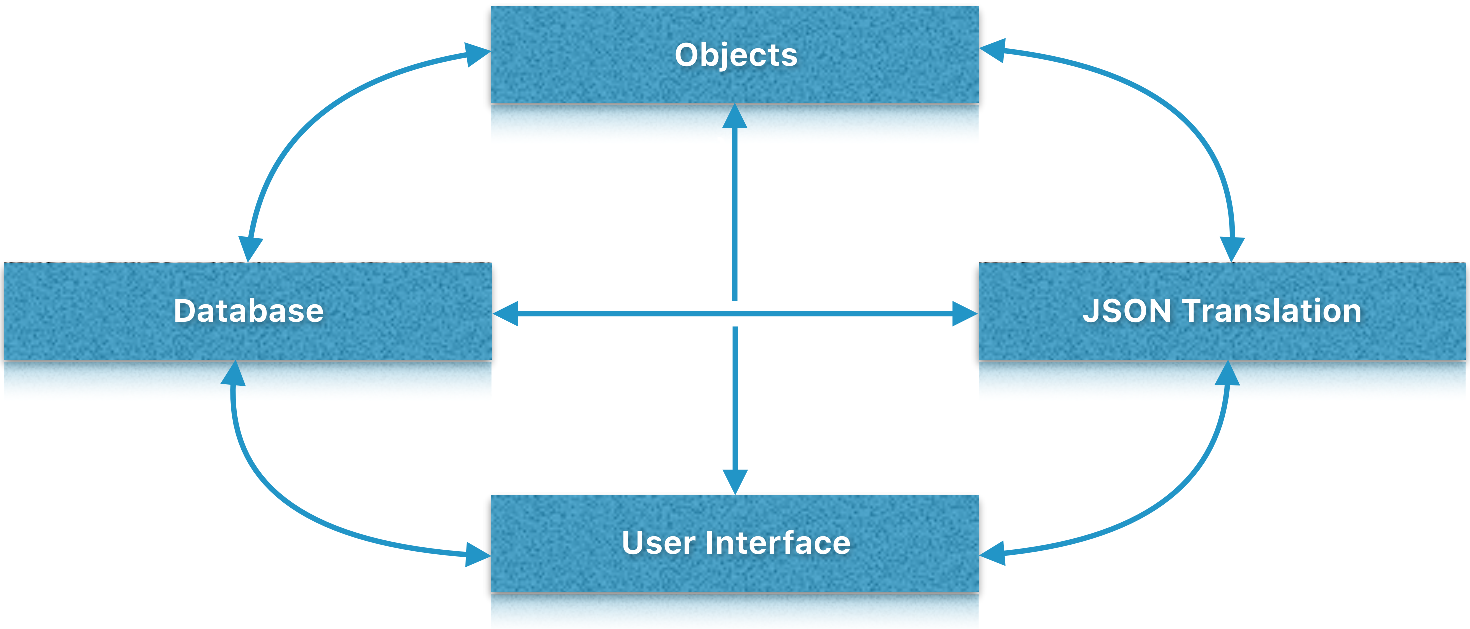
Cody Tyson • **Roy Ramstad** • Naveen Ganessin • **Alex Rosenberg** • Michael Schloss

Debate Forum

Incremental & Regression Testing

Classification of Components



Module	Input	Output	Dependent Components
Database	JSON Translation	JSON Translation	UI • JSON Translation
JSON Translation	Database	Objects • UI	Objects • UI • Database
Objects	JSON Translation	UI	UI • JSON Translation
UI	**User** • JSON Translation	**User** • JSON Translation	Database • Objects • JSON Translation

We used the **Bottom-up** testing form. Our components interact with each other fairly linearly the majority of the time, as such it's relatively simple and efficient to start at the bottom and keep adding components to the top to test them incrementally

CS 408
Team 7

Product	Debate Forum • Object Management Module
Author	Cody Tyson • Alex Rosenburg

INCREMENTAL			
Defect #	**Description**	**Severity**	**Solution**
1	There may be a null pointer exception in the standard User constructor. This is because it depends on UserQuery (in JSON translation) returning non-null from its constructor, which it sometimes does not, causing a null pointer exception.	**1**	Null check inside standard User constructor before using UserQuery function results
2	There may be a null pointer exception in the multi-field User constructor. This is because it depends on UserQuery (in JSON translation) returning non-null from its constructor, which it sometimes does not, causing a null pointer exception when accessors and mutators are used.	**1**	Null check inside multi-field User constructor before return
Regression			
1	Standard User constructor may return non-null User that does not have valid field data due to failed UserQuery	**2**	Check that the username field of the returned User object is non-null
2	Multi-field User constructor may return non-null User that does not have valid field data due to failed UserQuery, causing accessors and mutators to return null	**2**	Check that the return value of accessors and mutators are valid prior to using their returned values
3	Cached objects may not be properly saved into local storage if saved in an unexpected order by UI	**3**	UI should use abstracted `saveSession()` method instead of low level
4	Cached Objects may not be properly loaded back into memory if read out of order by UI	**3**	UI should use abstracted `loadUser()` and `loadDebate()` methods

Product	Debate Forum • JSON Translation Module
Author	Naveen Ganessin

INCREMENTAL			
Defect #	**Description**	**Severity**	**Solution**
1	Translation should wait for response from the database before sending info to UI	**1**	UI should have a function that translation could call to let it know that data has been received and ready to access
2	There is no way to let calling functions know when the information requested is downloaded. Right now it would simply return control to them without any data.	**1**	A notification center class will be used to post data when it is available and the calling class can then listen for this object to be made available and grab it when it does become available.
Regression			
1	The UI would always get null Users and false booleans for `verifyLogin()`. This is due to a change in the database side where asynchronous calls would return control to the program immediately. This is so that the main UI thread does not appear to freeze while waiting for data to download from the database. This fix on the database side broke JSON translation functionality resulting in null objects.	**1**	UI should have a function that translation could call to let it know that data has been received and ready to access
2	While fixing the issue for announcing the arrival of information, all the JSON translation methods requesting information have no return values. Therefore the calling functions have to aware of this.	**2**	Calling Functions will have to listen within the notification center class while waiting for information to be delivered instead of expecting a return value. The JSON translation method will then post this information in the notification center for it to be used.

Product	Debate Forum • Database Module
Author	Michael Schloss

INCREMENTAL			
Defect #	**Description**	**Severity**	**Solution**
1	The uploader and downloader do not check at all if JSON translation supplied them with a null SQL statement. They will attempt to execute and create a URLConnection and feed the null value into the database, which could lead to security issues	1	I will provide a check in the uploader as well as the downloader to make sure if a null SQL statement is delivered, the app does not attempt to upload the null statement. This will be returned to JSON translation as a DFSQLError exception.
	The NotificationCenter would not allow passing of data back and forth between notification receiver and caller. This produced the unintended side-effect of downloaded data being lost.	2	NotificationCenter has been updated to pass an Object through a notification. Only the poster can deliver this Object; only the observer(s) can receive this Object.
Regression			
1	The uploader and downloader, as well as the generic function ‘executeSQLStatement(_:, _:)’ now have a null check to prevent possible security leaks	1	JSON translation will have to change its code to adopt the possibility of an error being thrown at runtime for a null statement. As of this change, the project will not compile.
2	The downloader and uploader were changed to be multi-threaded. The reason for this was because the UI would hang and eventually lock up completely if URLConnection was persisted on the main thread.	1	JSON translation does not wait for the thread to return, rather it tries to access the data immediately after placing a call to the database. This is the old behavior and was correct. Now JSON translation will have to wait for the thread to call its delegate method to return data to the UI.

Product	Debate Forum • User Interface Module
Author	Roy Ramstad

INCREMENTAL			
Defect #	**Description**	**Severity**	**Solution**
1	For fields such as username and password, there is not a maximum character limit. This raises a DFSQLError.	**3**	Implement a length limit for certain fields
2	For fields such as username and password, when the length of the fields exceeds the width of the screen, the fields shrink to minimum size	**3**	The method `setResizable(_:)` is available for Components which will restrict the components from being resized
Regression			
1	A whitespace character is allowed as valid input for the username field. This will cause an issue with the database module using the username as the primary key	**2**	Use JFormattedTextField as the UI's input field, and specify the expression which are allowed