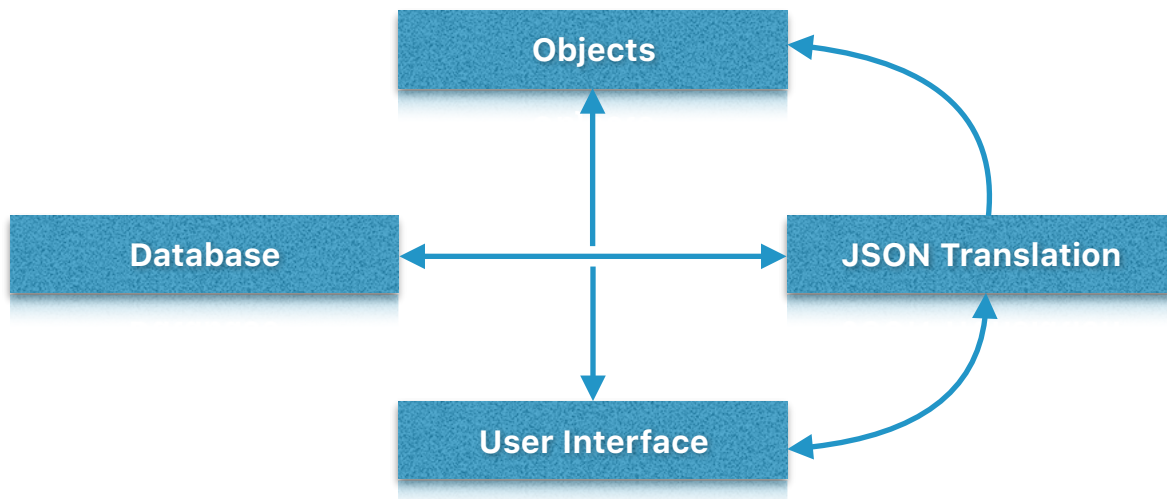


# Debate Forum

## Incremental & Regression Testing

### 2

### CLASSIFICATION OF COMPONENTS



Module	Input	Output	Dependent Components
Database	JSON Translation	JSON Translation	JSON Translation • UI
JSON Translation	Database • UI	Objects • UI	Objects • UI • Database
Objects	JSON Translation • UI	JSON Translation • UI	JSON Translation • UI
UI	<b>User</b> • JSON Translation • Objects	<b>User</b> • Objects • JSON Translation	Database • Objects • JSON Translation

We used the **Bottom-up** and **Top-down** testing forms. 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. We also started at the UI, finding bugs when attempting to use the app that flowed down through the modules. These would not have been detected in Bottom-up testing.

<b>Product</b>	Debate Forum • Object Management Module
<b>Author</b>	Cody Tyson

Incremental			
Defect #	Description	Severity	Solution
1	Object Module did not have any incremental test cases due to the fact that this module was mostly complete last sprint.		
Regression			
0	Object Module did not have any regression test cases due to the fact that this module was mostly complete last sprint.		

<b>Product</b>	Debate Forum • JSON Translation Module
<b>Author</b>	Naveen Ganessin • Alex Rosenberg

Incremental			
Defect #	Description	Severity	Solution
1	There is a race condition in how SQL queries retrieve data in PostQuery	2	The database needs to implement a queue system for requests and the UI needs sanity checks
2	postToDebate does not insert anything in the database due to missing SQL query	1	A DFSQL object should be created and used to execute the appropriate query
3	postID's are not inserted into the DebateComment table, meaning they do not show up in the Debate ArrayList of posts when requested	1	DFSQL should be used to make a second insert query after inserting post information into the Comment table
4	getPosts does not return correct information for certain valid inputs. No posts are returned	1	Callback delegate needs to be assigned and return values checked
5	Debates with the same title can be created right now which should not be allowed	2	Translation should check all other debates for duplication before adding a new debate
Regression			
1	The database queuing system that fixes PostQuery could lead to network timeouts	2	The UI should make multiple attempts to retrieve data upon initial failure
2	An incorrect value for Post.flagged is inserted into the database when running postToDebate	3	PostQuery should treat flagged as an integer, not a boolean
3	The same DFSQL object is used to make two insert queries. They have the wrong row and value arrays, which will not end up inserted in the database	3	A new DFSQL object and two new arrays must be used for the query to insert data into the DebateComment table

Incremental			
Defect #	Description	Severity	Solution
4	UserQuery returns 2 errors when the username is invalid. This means that error message in UI pops up twice	2	Reduce to just one error message
5	Debate class has key missing attributes like debateID and the startTime and EndTime of a debate	1	These need to be added to the debate class in order to allow translation to use them in debate generation

<b>Product</b>	Debate Forum • Database Module
<b>Author</b>	Michael Schloss

Incremental			
Defect #	Description	Severity	Solution
1	Giving multiple execute calls in rapid succession overwrites the delegate callback and produces fall-through	2	Built DFWebServerDispatch to process one item at a time to prevent callback and web server overload
	Given improper DFSQLConjunctionClause values in WhereStruct produced invalid SQL statements	2	Added checks to WhereStruct initializer to make sure expected data is what's being delivered
Regression			
1	WhereStruct initializer can now throw exceptions	1	All code using WhereStruct will have to adopt a try-catch block
2	DFSQL now allows appended statements	3	Multiple SQL statements that must be called can now be pushed into 1 statement

<b>Product</b>	Debate Forum • User Interface Module
<b>Author</b>	Roy Ramstad

Incremental			
Defect #	Description	Severity	Solution
1	When a login fails, two "failure" notifications are fired, resulting in two error dialogues to appear when only one should	3	Ask for UserQuery to fire "failure" notification once on a failed login
2	User interface doesn't deny a post with zero characters to be submitted	3	Insert a case to check if the length of the string in the text field is greater than zero
3	Reported comment doesn't turn red immediately after reporting	3	Within the ActionPerformed method of the comment, setForeground(Color.RED)
Regression			
2	<b>UI did not have any regression testing because the UI can no longer break underlying code due to updated coding styles and compartmentalization</b>		