# Release Information

|  |  |
| --- | --- |
| *Release* | *Deployment* |
| *Leader* | *Austin* |
| *Git Repo* | *https://atlas.ion.uwplatt.edu/bitbucket/scm/ise/1-nullreferenceexception* |

Slightly fewer bugs. Finalized commenting, unit testing, and UI for deployment.

# Progress

|  |  |  |  |
| --- | --- | --- | --- |
| **Stories and spike solutions worked on during this release** | | | |
| **Story, Spike** | **Description** | **% Impl.** | **Notes, plan for completion** |

|  |  |  |
| --- | --- | --- |
| **Accomplishments by engineer** | | |
| **Team Member** | **Total Time (hrs)** | **Stories, Spikes implemented** |
| *Mitch Vance* | *5.5* | *Client unit testing, documentation* |
| *Ryan Wolff* | *9* | *Server cleanup, server unit testing, documentation* |
| *Liam Crain* | *5* | *Server/client cleanup, documentation, presentation creation* |
| *Austin FitzGerald* | *4* | *Client cleanup, documentation* |
| TEAM TOTAL | ***23.5*** |  |

|  |  |
| --- | --- |
| **Document updates** | |
| **Document** | **Changes** |
| *GroupNullReferenceExceptionDeploymentReport.docx* | *Created and added to repo* |
| *Final Presentation.pptx* | *Created and added to repo* |
| *Iteration 1 Planning Record.docx* | *Added to repo* |
| *Iteration 2 Planning Record.docx* | *Added to repo* |
| *Iteration 3 Planning Record.docx* | *Added to repo* |
| *Iteration 1 Report.docx* | *Added to repo* |
| *Iteration 2 Report.docx* | *Added to repo* |
| *Iteration 3 Report.docx* | *Added to repo* |
| *Client Class Diagram.png* | *Created and added to repo* |
| *Server Class Diagram.png* | *Created and added to repo* |
| *Database Diagram.png* | *Created and added to repo* |

# Non-implemented

|  |  |
| --- | --- |
| **Issue Number** | **Description (Prob / Resolution)** |
| NRE-14 | Direct messaging will not be implemented. |
| NRE-61 | Chatroom password changing as well as admin-ing will not be implemented. |
| NRE-63 | It is unfeasible to make the forms resizable, will not be implemented. |

# Screen Shots

*We have not added any functionality for deployment – please see previous iteration reports for screenshots.*

# Test Coverage Report

For this program we were able to achieve 4.19% test coverage for our client and 8.82% test coverage for our server. We were unable to achieve 100% test coverage for any of our classes in this program. This is due to several factors which are a result of our program being about data transfer over a network. More specifically since the program’s focus is on data transfer it is extremely convoluted to unit test anything that requires a NetworkStream, which many classes do. Another reason we were unable to achieve 100% coverage was due to the need to connect to the SQL server. Without being able to query the server from our program we are unable to unit test with 100% coverage without a convoluted test setup. We were also unable to achieve 100% test coverage because unit testing UI components was not required, nor would it be feasible.

In all, it was not possible to achieve a high percentage of code covered by unit tests because our project is extremely reliant on a server and database connection. It would not be feasible to implement mock NetworkStreams or sockets – therefore most of our code is not testable. We were successful in creating good unit tests for functions that operate on data directly. Our group believes that this is actually a good thing. We rely on very few data operatory functions, therefore there are less places that our code can fail. These few spots of possible failure have been unit tested thoroughly: in the Server ChatroomList is 64% covered and, in the client UserService is over 90% covered.

# Diagrams





