LAB 3 – R-IDE PROTOTYPE TEST PLAN

Daniel Koontz

Old Dominion University

CS411W: Professional Workforce Development

Prof. Janet Brunelle

April 04, 2023

Version 1

Test Category: Automation / Algorithms		**Description:** Auto-Update CMakeLists.txt		
Test Case:		**Case Name:** Update CMakeLists.txt	**Version:** 1.0	**Written By:** Daniel Koontz
Requirements Fulfilled: 3.1.2.1 3.1.2.1.*		**Purpose:** Ensure that R-IDE correctly updates the CMakeLists.txt file while the user edits and updates the other files within the package		
Setup Conditions: 1. VSCode, Microsoft ROS, and R-IDE are installed and launched 2. The selected workspace folder contains a ROS package recognizable by R-IDE 3. The selected ROS package contains a correctly formatted CMakeLists.txt file				
Test Case Activity		**Pass/Fail**	**Comments**	**Expected Result**
1	From the R-IDE Creation Wizard, name and create a ROS message file in the desired package.			The message path should exist under `add_message_files`. `find_package` should contain `message_generation`. `generate_messages` should be uncommented.
2	From the R-IDE Creation Wizard, name and create a ROS service file in the desired package.			The service path should exist under `add_service_files`. `find_package` should contain `message_generation`. `generate_messages` should be uncommented.
3	From the R-IDE Creation Wizard, select new ros node and select a python node. Give the node a name and select any other attributes of the node. Click next to complete creating the wizard.			The `catkin_install_python ` file should be uncommented and contain the path to the python file.

4	From the R-IDE Creation Wizard, select new ros node and select a cpp node.Give the node a name and select any other attributes of the node. Click next to complete creating the wizard. Enter the command palette (Ctrl + shift + P) and select \`R-IDE: Add an executable\`. Select the package and the newly created source file and give the node an appropriate name			There should exist in the CMakeLists.txt an \`add_executable\` call with the given name and source files. Above it should be the commented name of the node and below should be a \`target_link_libraries\` call with the name.
5	From the R-IDE Creation Wizard, select new ros node and select a cpp node.Give the node a name and select any other attributes of the node. Click next to complete creating the wizard. Enter the command palette (Ctrl + shift + P) and select \`R-IDE: Add a library\`. Select the package and the newly created source file and give the node an appropriate name			There should exist in the CMakeLists.txt an \`add_library\` call with the given name and source files. Above it should be the commented name of the node and below should be a \`target_link_libraries\` call with the name.
6	Enter the command palette (Ctrl + shift + P) and select \`R-IDE: Add a new package to the find package call\`. Select the desired package and select the updated contents of the \`find_package\` call			The contents under the \`find_package\` call should be updated to reflect the user selection and should all be valid packages available on the system.
7	Delete a .msg file that exists in the \`add_message_files\` call in CMakeLists.txt			The file should no longer be listed
8	Delete a .srv file that exists in the \`add_service_files\` call in CMakeLists.txt			The file should no longer be listed
9	Delete a .py file that exists in the \`catkin_install_python\`\` call in CMakeLists.txt			The file should no longer be listed