**3. Overall Strategy and Approach**

**3.1. Testing Strategy**

EGR101 Simulation System Testing will include testing of all functionalities that the scope (section 2) identified. System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, workflows, functionality access, testing of internal & external interfaces.

The testing types section following this one will be discussing what needs to be tested. However this section will describe how the types will be tested.

**3.1.1 Emulation Testing**

* **Testing Objective**: The Arduino emulation functionality, its simulated pin connections, voltage at per component, and current of the system according to the specific requirements in the SRS.
* **Technique**: Java application that provides an instance of the emulation, and provides simulated voltages, connections, etc. warnings should be output if invalid connections and input voltage pairs are present during execution, otherwise if valid connections and input voltage pairs are present during execution the application should validate specific test cases.
* **Completion Criteria**: When all available test cases for emulation have been completed successfully

**3.1.2 Virtual Programmable Board Testing**

* **Testing Objective**: The simulated Arduino hardware should perform actions based specifically on instructions compiled from the Arduino language according to the specific requirements in the SRS.
* **Technique**: Java application that provides a group of test cases with Arduino code and component configurations which will execute and validate expected behavior.
* **Completion Criteria**: When all available test cases for Virtual Programmable board testing have been completed successfully

**3.1.3 Communication Testing**

* **Testing Objective**: Valid communication between Unity and the emulated Arduino according to the specific requirements in the SRS.
* **Technique**: Java application that executes a group of test cases that begin an instance of the application and capture TCP logging information from the communication between Unity and Java.
* **Completion Criteria**: When all available test cases for Communication testing have been completed successfully

**3.1.4 Simulation Testing**

* **Testing Objective**: correct sensor readings and actions performed based on information captured from emulated Arduino connection in the 3-dimensional simulated environment.
* **Technique**: Python application that executes Unity executable simulation on a test oriented course and logs information regarding sensor information and performed actions.
* **Completion Criteria**: When all available test cases for simulation testing have been completed successfully

**3.1.5 Bot Design Testing**

* **Testing Objective**: design changes utilizing the bot design interface should change Arduino emulator component list.
* **Technique**: Unity based test that sends information to the Java Server and logs whether components, or connections were applied successfully to the emulator.
* **Completion Criteria**: When all available test cases for Bot Design testing have been completed successfully

**3.2 System Testing Criteria**

In order to start system testing, certain requirements must be met for testing readiness. The readiness can be classified into usability testing, functional testing, and data and documentation testing.

**3.3 Testing Types**

**3.3.1 Usability Testing**

User interface attributes, cosmetic presentation and the content will be tested for accuracy and general usability. The goal of Usability Testing is to ensure that the User Interface is comfortable to use and provides the user with consistent and appropriate access and navigation through the functions of the application.

3.2.3.7 The system shall restart the simulation if the restart button is selected

3.2.3.8 The system shall end the simulation if the end button is selected.

3.2.3.9 The system shall pause the simulation if the pause button is selected.

3.2.3.10 The system shall play the simulation if the play button is selected.

3.1.3.3 The system shall record the pin locations that are connected by a wire in a file.

3.1.3.4 The system shall have a button to save the current wire configuration.

3.1.3.5 The system shall save the current configuration of wires when the “save” button is pressed.

3.1.3.6 The system shall have a button to exit the Wiring and Design Interface.

3.1.3.7 The system shall exit to the main view screen when the exit button is pressed.

3.2.3.1 The system shall have a File button that shows file operations when clicked on.

3.2.3.2 The system shall have a save button appear when the File button is clicked

3.2.3.3 The system shall save the Arduino script when the save button is clicked.

3.2.3.4 The system shall have an open button appear when the File Button is clicked.

3.2.3.5 The system shall open a new Arduino script when selected

3.2.3.6 The system shall have a “save configuration” button appear when the File button is clicked.

3.2.3.7 The system shall save a configuration file when the “save configuration” button is clicked.

* + - 1. The system shall produce a configuration file that contains the wiring setup and the Arduino code when the “save configuration” button is pressed.

**3.3.2 Functional Testing**

The objective of this test is to ensure that each element of the component meets the functional requirements of the EGR101 course as outlined in the:

* Software Requirements Specifications Document
* EGR101 Instructors rules or conditions

3.4.3.1 The system shall allow for pin connections between the Arduino pins and components

3.4.3.2 The system shall simulate the Arduino clock by counting in microseconds after executing

3.4.3.3 The system shall allow for calls from Arduino code to delay programmable interaction with the components

3.4.3.4 The system shall match the behavior defined from compiled Arduino code

3.4.3.5 The system shall allow for digital writing to pins by providing voltage via simulated Pulse Width Modulation (PWM)

3.4.3.6 The system shall allow for analog writing to pins by providing a voltage.

3.4.3.7 The system shall provide traditionally used libraries like Servo.h and Serial.h to interface with.

3.4.3.8 On execution the system shall push voltage to each pin sequentially beginning at the IO and power ports of the Arduino.

3.4.3.9 On execution the components shall work if and only if they have adequate ground connection, and enough voltage to satisfy the potential of the component

### 3.1.3.1 The system shall allow the user to select a wire.

### 3.1.3.2 The system shall allow the user to move the wire to connect two pin locations.

3.2.3.1 The system shall prompt the user to select a course, the course selected will be displayed and executed on.

3.2.3.2 The emulation thread shall send each components behavior to the simulation UI via TCP connection per cycle.

3.2.3.3 The simulation UI shall send each components sensor data to the Arduino Emulation via TCP connection per update iteration.