**Bio-Dome Software Design Document**

The following documentation details the Classes, Methods and Variables used to create the Bio-Dome Simulator software on behalf of NitroGenics Corp.

**Class Name:** BioDomeControls

**Public Variables:**

temperatureStateIndicator of TYPE ClimateVariables

humidityStateIndicator of TYPE ClimateVariables

windSpeedStateIndicator of TYPE ClimateVariables

temperatureSeverityIndicator of TYPE ClimateSeverityIndicator

humiditySeverityIndicator of TYPE ClimateSeverityIndicator

windSpeedSeverityIndicator of TYPE ClimateSeverityIndicator

vent1 thru vent4 of TYPE ControlUnit

heat1 thru heat4 of TYPE ControlUnit

sprink1 thru sprink2 of TYPE ControlUnit

bioDomeControlUnits of TYPE ControlUnit[]

**Methods:**

// Create an ArrayList to hold the System Controls

// for The Bio-Dome Simulator Tool

public void LoadBioDomeObjectsArray()

// Method sets the Initial state for all the Bio-Dome control units

// and atmospheric state indicators to either 'Closed, 'On, 'Off' or 'Optimal'

public void SetInitialControlStates()

// Method sets the Shut-Down state for all the Bio-Dome control units

// and atmospheric state indicators to 'Undefined'

public void SetShutDownControlStates()

// Method displays the current state of all the Bio-Dome controls

// using the toString method

public void DisplayBioDomeControlDetails(BioDomeSimulator bioDome)

// Method sets the color and background of the given control based on its state

public String GetStateColor(String state)

// Method sets the initial color for all controls in their Off-Line State

public void SetBioDomeOffLineState(BioDomeSimulator bioDome)

// Method sets the initial color for all controls in their On-Line State

public void SetInitialBioDomeOnDisplay(BioDomeSimulator bioDome)

// Loads the ETI Tool Bio-Dome Control Combo boxes with their Value Options

public void LoadETITool(BioDomeSimulator BioSim)

// Loads the TPI Tool Bio-Dome Control Throttle Combo boxes with the available

// time options

public void LoadAutorunThrottle(BioDomeSimulator BioSim)

// Generates a testcase template for the Pre-Test and Expected Results listboxes in

// the Testplan Implementation Tool and the Exploratory Testcase Implementation

// Tool

public void GenerateTestcaseResultsTemplate(BioDomeSimulator bioDome)

// Loads the current control values into the CurrentSimulatorValues rich text box

// in either the Test Implementation Tool or the Exploratory Testcase Implementation

// Tool. The method will display the values that existed prior to the testcase

// being run. When the current testcase has completed and a testcase result is

// returned the rich text box will be cleared in order to be ready for the next testcase.

public void LoadCurrentControlValues(BioDomeSimulator bioDome)

===============================================================

**Class Name:** BioDomeSimulator : Form

**Public Variables:**

bioControls of TYPE BioDomeControls

fullBioDomeControlList of TYPE List<Label>

bioClimateControlList of TYPE List<Label>

bioClimateSeverityList of TYPE List<Label>

bioVentList of TYPE List<Label>

bioHeaterList of TYPE List<Label>

bioSprinklerList of TYPE List<Label>

bioControlsInUseList of TYPE List<Label>

expTestcaseControlList of TYPE List<ComboBox>

expTestcaseClimateValueList of TYPE List<NumericUpDown>

**Methods:**

// Constructor for the BioDomeSimulator Form

public BioDomeSimulator()

// Method handles the functionality that is triggered when the “Turn System On/Off”

// button is pressed.

private void btnBioChangeSystemState\_Click(object sender, EventArgs e)

// Method handles displaying the current state of the system when the button

// btnBioViewSystemState is clicked

private void btnBioViewSystemState\_Click(object sender, EventArgs e)

// Method separates the Testbed from the Simulator when the

// btnTMDeCoupleTestbed is clicked.

private void btnTMDeCoupleTestbed\_Click(object sender, EventArgs e)

// Method closes the Exploratory Testcase Implementation Tool and re-loads the

// login screen when the btnExpCloseTestSessionis clicked.

private void btnExpCloseTestSession\_Click(object sender, EventArgs e)

// Method clears all user inputted values in the Exploratory Testcase Implementation

// Tool when the btnExpClearTestCase is clicked

private void btnExpClearTestCase\_Click(object sender, EventArgs e)

// Method calls the LoadCurrentControlValues and the

// GenerateTestcaseResultsTemplate methods that populate the Pre-Test Simulator // and Expected results listboxes along with loading the control units with their

// potential state options when the btnExpLoadNewTestcase button is clicked.

private void btnExpLoadNewTestcase\_Click(object sender, EventArgs e)

// Method closes the Testplan Implementation Tool and re-loads the login screen

private void btnTpCloseTestSession\_Click(object sender, EventArgs e)

// Method allows the current Test case to be saved to the current Testplan when

// the btnExpSaveTestCase button is clicked

private void btnExpSaveTestCase\_Click(object sender, EventArgs e)

// Method allows for editing of the current test plan when the btnTpEditTestPlan

// button is clicked

private void btnTpEditTestPlan\_Click(object sender, EventArgs e)

// Method exposes the Exploratory Testcase Implementation Tool to the user when the rbTmExecuteSingleTestcase radio button is selected.

private void rbTmExecuteSingleTestcase\_CheckedChanged(object sender, EventArgs e)

// Method exposes the Testplan Implementation Tool to the user when the

// rbTmExecuteFullTestPlan radio button is selected.

private void rbTmExecuteFullTestPlan\_CheckedChanged(object sender, EventArgs e)

// Method forces the program to quit when the btnExitSimulator is selected.

private void btnExitSimulator\_Click(object sender, EventArgs e)

// Method generates a new log file when the btnBdlCreateNewLogfile is clicked.

private void btnBdlCreateNewLogfile\_Click(object sender, EventArgs e)

// Method saves anything captured in the Logging Window to the currently

// selected log file when the btnBdlSaveToLogfile is clicked.

private void btnBdlSaveToLogfile\_Click(object sender, EventArgs e)

// Method prints the test results of the current test run when the

// btnBdlPrintTestReport is selected.

private void btnBdlPrintTestReport\_Click(object sender, EventArgs e)

// Method clear all text from the Logging Window after saving it to the currently

// selected log file or temporary text file if no log file has been opened or

// selected when the btnBdlClearLoggingWindow button is clicked.

private void btnBdlClearLoggingWindow\_Click(object sender, EventArgs e)

// Method prints out the contents of the currently selected log file when

// the btnBdlPrintLogReport is selected.

private void btnBdlPrintLogReport\_Click(object sender, EventArgs e)

// Method loads all the control unit object lists. These object lists will be

// used to change the background color of the graphical representations of the

// Simulator control units

public void LoadClimateAndControlLists()

// Method provides the mechanism to change the foreground and

// background color of any word or sequence of words in the

// Logging Window allowing for a color rich reporting environment

// within the Logging Window text area.

public void Append(RichTextBox rtb, String color, String s)

// Method creates a new test plan in the SQL database by creating a new

// test plan table in the BioDomeTestPlans Database.

private void btnTPCreateNewTestplan\_Click(object sender, EventArgs e)

===============================================================

**Class Name:** ControlUnit

**Private Variables:**

name of TYPE String

state of TYPE String

**Methods:**

// Constructor for the Class ControlUnit

public ControlUnit(String name)

// Getter Method for name

public String GetName()

// Getter Method for state

public String GetState()

// Setter Method for state

public void SetState(String state)

// Overridden ToString method

public override string ToString()

**Class Name:** ClimateVariables

**Private Variables:**

name of TYPE String

state of TYPE String

**Methods:**

// Constructor for the Class ControlUnit

public ClimateVariables (String name)

// Getter Method for name

public String GetName()

// Getter Method for state

public String GetState()

// Setter Method for state

public void SetState(String state)

// Overridden ToString method

public override string ToString()

===============================================================

**Class Name:** ClimateSeverityIndicator

**Private Variables:**

name of TYPE String

state of TYPE String

**Methods:**

// Constructor for the Class ControlUnit

public ClimateSeverityIndicator (String name)

// Getter Method for name

public String GetName()

// Getter Method for state

public String GetState()

// Setter Method for state

public void SetState(String state)

// Overridden ToString method

public override string ToString()