Project Plan

GP Genie

\*Paste your sections below Friday morning by 8 AM.

<Insert Table of Contents>

Problem Description

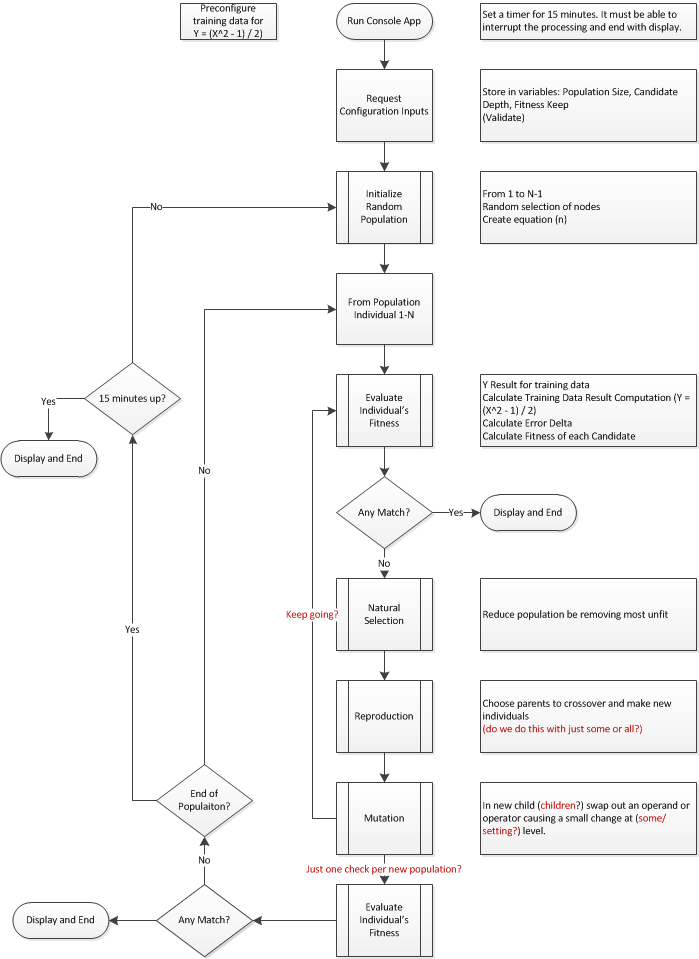
2. Requirements Analysis and Preliminary System Design

*SMM: Numbers here are representative only - use numbers or not as you like. I changed my mind since my e-mail - one main section is fine for #2 with 4 subsections or more depending on what Roger has.*

2.1 System Requirements

(Roger will be providing requirements list in a traceability matrix attached I believe)

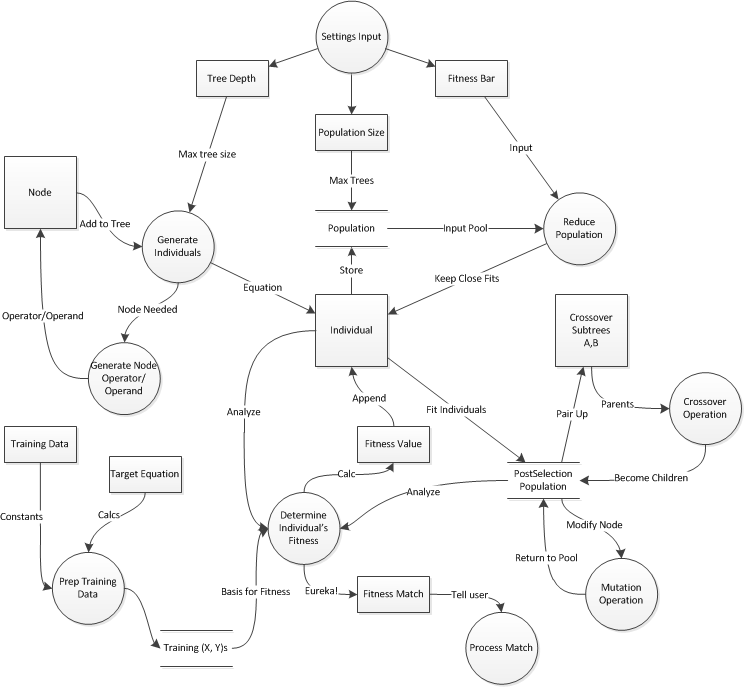
2.2 Initial Requirements Clarification Flowchart



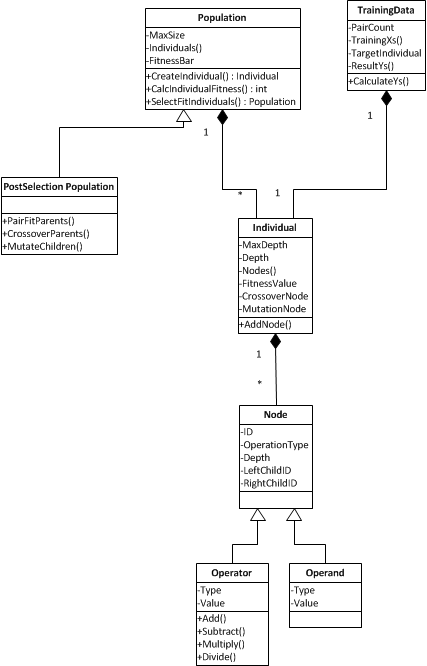
2.3 Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Description/Process** | **Possible Values** |
| Crossover NodeIDs A, B | Integer | Randomized Node ID of each of two Individuals’ trees that will be the crossover target in each |  |
| Crossover SubTrees A.B | EquationTree | Subset of each of the crossover individuals’ trees which will be placed at the Crossover Node ID in the other tree |  |
| Crossover TreeIDs A, B | Integer | Randomized IDs of two remaining trees in the PostSelection Population list; trees will be selected and paired for crossover with each other until non remail |  |
| Crossover Trees AMod(n), BMod(n) | EquationTree Array(Nx2) | Randomly selected Pairs of PostSelection trees which will be/have been targets for the crossover section of the other in the Crossover operation |  |
| Fitness Bar | Integer | User input percent of the population which will be kept after Natural Selection | 1 - 99 |
| Individual Fitness Value | Integer | Sum of the Tree Y Deltas from 1 to quantity of training Ys | 0 to any with 0 being an exact match to the target equation result |
| Individual ID | Integer | Unique identifier for each candidate in a population | 100-1000 |
| Individual Tree(n) | EquationTree | Collection and sequence of Nodes making up a candidate equation | Example:  (x \* x + 20 -6 / 4/ 9 \* 114) / (2 \* x + 5 – 6 / 2 – x) |
| Most Fit Equation | Integer | ID of the tree after crossover and mutation whose resulting fitness value is the lowest |  |
| Mutation NodeID(n) | Integer | Randomized NodeID generated for each tree in Crossover Trees population |  |
| Mutation Operand Node | Integer | Random replacement operand for a node in a crossover tree |  |
| Mutation Operator Node | Integer | Random replacement operator for a node in a crossover tree |  |
| Node ID | Integer | Unique identifier of a specific tree node | 1-100? |
| Population | List | List of Individual Trees for fitness evaluation |  |
| Population Size | Integer | Quantity of individual equations in the population | 100-1000 |
| PostSelection Population | List | Even numbered collection of Individual tree equations remaining after Natural Selection |  |
| Program Duration | Integer | Minutes input by user for the length of time the program has to run | Default: 15 |
| Target Equation | EquationTree | The target for individuals’ fitness when being evaluated  (Not sure this is needed in the program as long as training data is provided as input) | ( (x^2) - 1 / 2) |
| Training Data | Nx2 Array | Collection of the training X and Y pairs for the target equation | (x1, y1) – (xN, yN) |
| Training Data Size | Integer | Number of X and Y pairs in the training data that will be used to iterate over the training data with each Individual equation |  |
| Tree Depth | Integer | Limit to number of levels deep a candidate tree can be | 1-30? |
| Tree Node Count | Integer | Number of total nodes in an Individual’s tree used for iterating over nodes to locate crossover and mutation locations |  |
| Tree Operand Max | Integer | User input for the highest an operand value can be | 1 - 100 |
| Tree Operand Min | Integer | User input for the lowest an operand value can be | -100 - 0 |
| Tree Operand Node | TreeNode | Operator or Operand node of the candidate equation tree | x, integers -100 – 100? |
| Tree Operator Node | TreeNode | Operator or Operand node of the candidate equation tree | + - \* / |
| Tree Y Delta (n) | Integer | Absolute value of the difference between the Tree Y Result (n) and the Training Data Y (n) | Any positive integer |
| Tree Y Result(n) | Integer | Result of running the training Xs through the new equation tree individual | Any. Rounding will be fine given the wide range of training data Xs we will use |

2.3 Data Flow Diagram



2.4 Preliminary Class Diagram



Weekly SCM files and folders for traceability

Work Plan

**Roles & Responsibilities**

The following is a list of roles and who is responsible for each role.

|  |  |
| --- | --- |
| **Roles** | **Name** |
| Project Management & Communications Liaison | Ujin Han |
| Process Leader | Ujin Han  Justin Florkiah |
| Development Leader | Li Wang |
| Technical Leader | Roger Peterson  Felista Mpanga |
| Measurement Leader | Susan Mairs  Ujin Han |
| Capture Requirements | Susan Mairs |
| Coding | Li Wang  Roger Peterson |
| Code Review | Susan Mairs |
| Testing | Felista Mpanga |

For the roles with multiple team members listed, they will share the responsibilities of that particular section of the project. As the project goes on and more responsibilities come up, those roles will be added. During the project, if a certain team member is struggling with the given task, others with knowledge and availability will assist in its completion.

**Milestones & Schedule**

To complete the project with changes in requirements by December 14th, the team has following milestones and dates set.

|  |  |
| --- | --- |
| **Milestone** | **Due Date** |
| Requirements & preliminary system design finalized; Project Plan completed and submitted | October 26, 2013 |
| All complements needed for program design finalized | October 29, 2013 |
| Working software | November 19, 2013 |
| Testing software | November 26, 2013 |
| Revised working software with new requirements | December 7, 2013 |
| PowerPoint presentation completed | December 12, 2013 |
| Project completion and deliverables submitted | December 14, 2013 |

The team has a weekly meeting on Tuesday evenings at 6 pm until the end of the semester to check-in and discusses the progress of the project. These meetings will last from an hour to two hours depending on the subject of the meeting at the time. Additional meetings will be scheduled as the team sees fit.