University College Dublin

Software Engineering Project ii

Final Report

Lecturer:
Professor Micheal Whealen

Team ***Olympians:***

14208884 - Theshan Ronello
14210146 - Nadeesha Lakmal
14208883 - Kalana Chandrasiri
14208914 - Danindu Gammanpila

Contents

1 Introduction

The Project begins with the motivation of allocating limited number of resources among group of individuals. The goal is to allocate projects among these individuals satisfying their best preference if not the one of the preferences they mentioned. This is a common problem faced in resource allocation in all the industries. As there are limited number of resources and lot of individuals are competing to use these resources.

To achieve the best possible fairness to each individual we need an efficient method of resource allocation. The most primitive option is to use a brute force search for look at all the possible options and allocate. But this method is too time consuming. So we look at more advanced *stochastic search* algorithms. The algorithms that we use are the **Simulated Annealing** and **Genetic Algorithm**.

This project uses a sample data set that contain a set of individuals and their preference list of projects. If the individual has a pre-assigned project then there is a column saying [yes/no] to pre-assigned. If "Yes" then user may only have one project on the preference list. If "No" then he/she can mention list of preferred projects. The maximum size of the list is 10.

We use *stochastic search algorithms* to give these individuals the most fair allocation possible.

Simulated Annealing: This algorithm uses probabilistic techniques for finding the global minimum. Simulated Annealing uses the analogy of gradually cooling a metal from a high temperature where it exist as a liquid, to where it forms a crystalline structure.

Genetic Algorithm: This algorithm mimic the process of natural selection found in evolution theories, *Survival of the fittest* is ensured. It selects a random population and evaluate for the fitness. Eventually it strives to find an optimal solution.

2 Specifications

specifications at a glance

MAIN GOAL: **Provide one to one mapping of Student to Project**

- Given a spread sheet provide one to one mapping of student to project
- Spread sheet is a tab delimited file
- Pre-assigned projects must be allocated correctly to the referred students
- Each students should be allocated a project so that overall disappointment is minimum
- Allocations must be validated
- Each student must be given a project that is in their preference list
- System should be immune to cheats such as if one student only specify one preference he/she must be identified and treated accordingly.
- Score system should be implemented to measure the disappointments with points
- Graphical User Interface for viewing the results *[additional]*
- READ-ME File explaining how to run the project
- Interim report delivered at (29-06-2016)
- The Final Project and the Final Report delivered at (17-06-2016)

3 System Diagrams

The Software development life cycle for this project is explained and documented with the help of different System diagrams.

1. **Work Breakdown Structure[WBS]** : The WBS is used to breakdown the main tasks of the project and divide it among the group members.

2. **Use Case Diagram** : Use case diagram shows the users interaction with the system. The main system functions is shown in a closed environment and the real time users are displayed outside the system boundary.

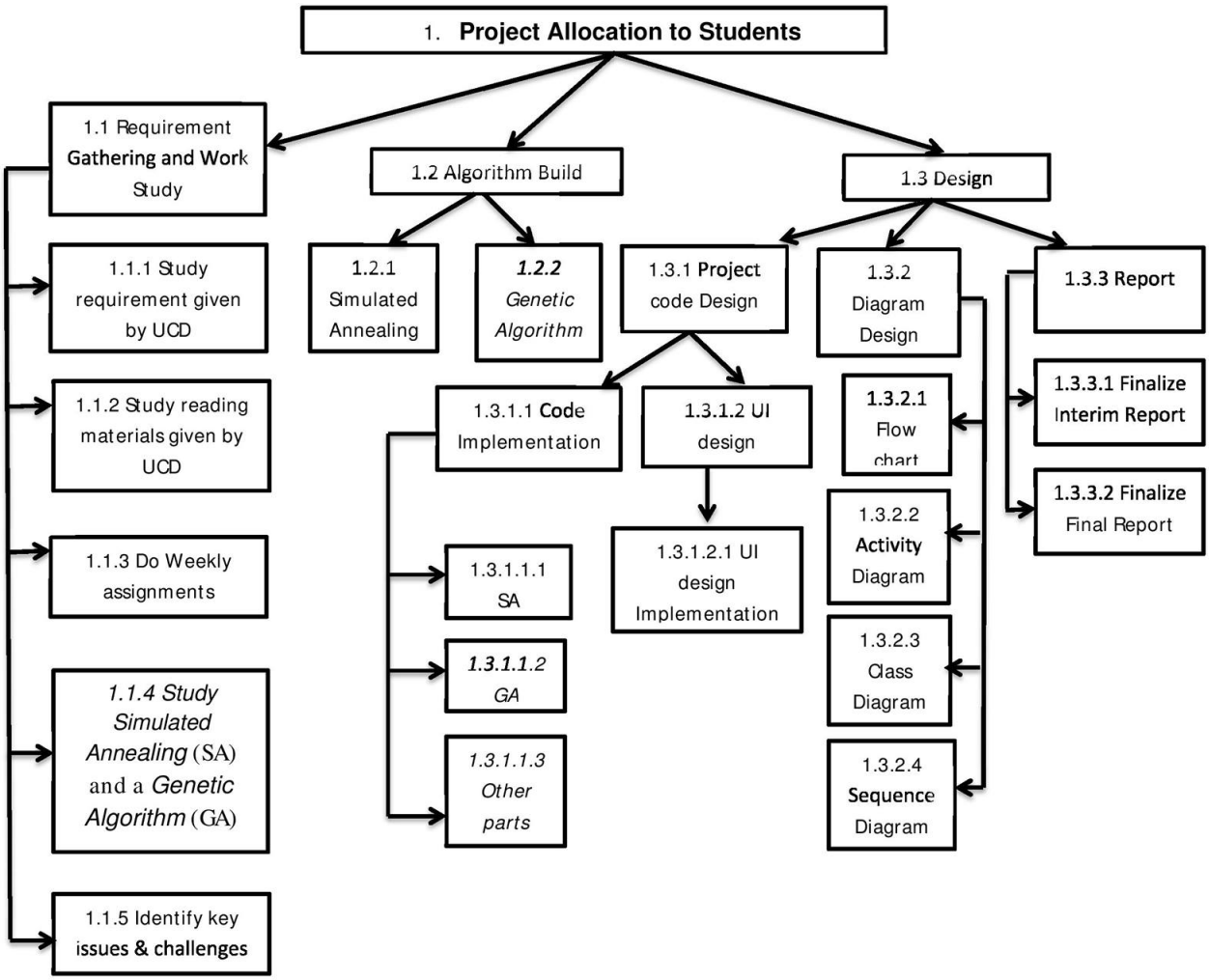
3. **Activity Diagram** : The Activity diagram goes in depth of how the main functionality of the system is achieved. It explores each activity of the system and the flow from one activity to another.

4. **Flow Chart** : The flow chart describes the basic flow of the system. This gives developer perspective of the system, the control flow of the algorithms and the overall system is depicted here.

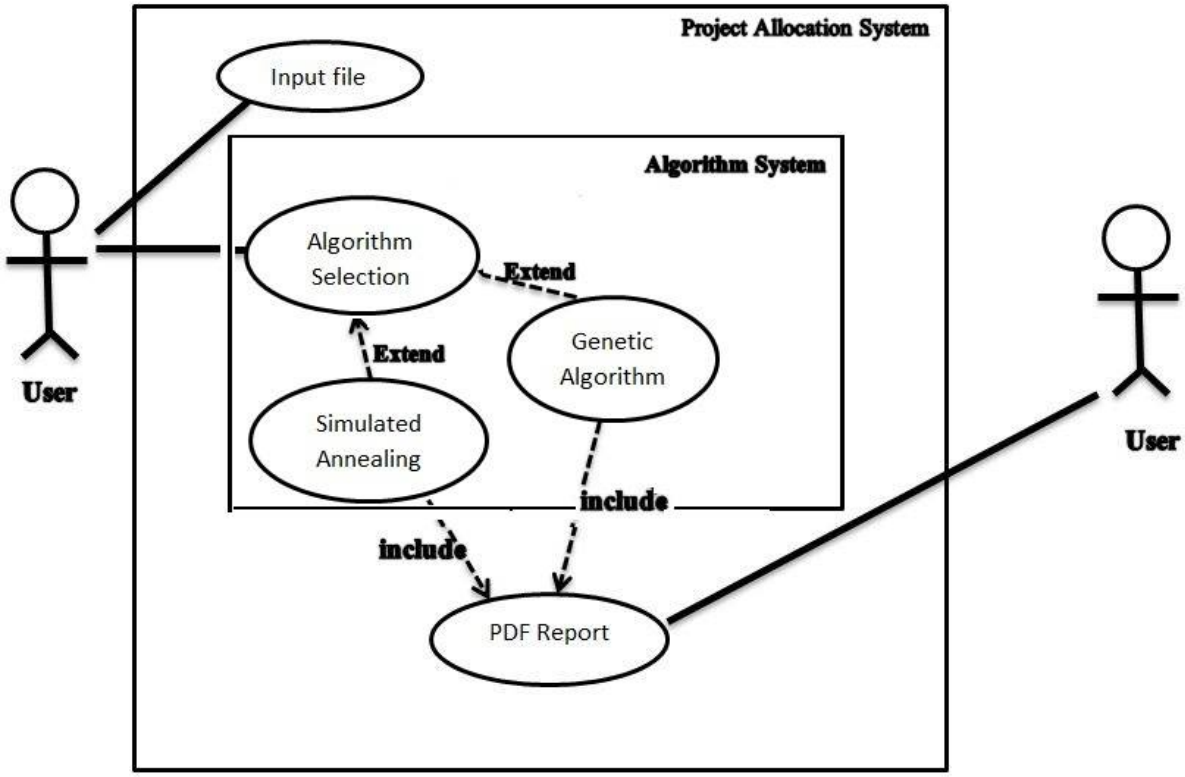
5. **Class Diagram** : Class diagram shows the system architecture at a more detailed level. The classes used in object oriented design and implementation. The attributes and behaviors of these classes and the communication among these classes are clearly elaborated within this class diagram.

*** The diagrams are included in the below pages ***

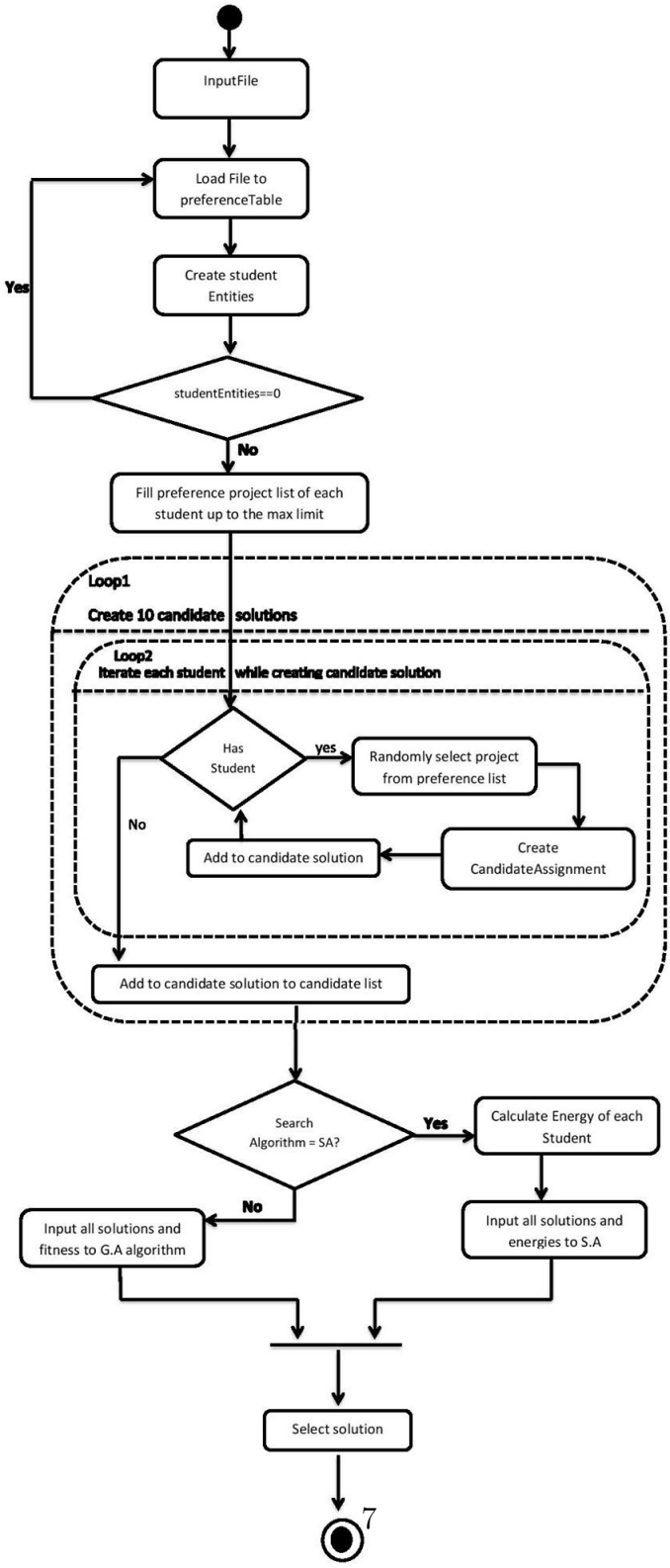
3.1 Work Break Down Structure[WBS]



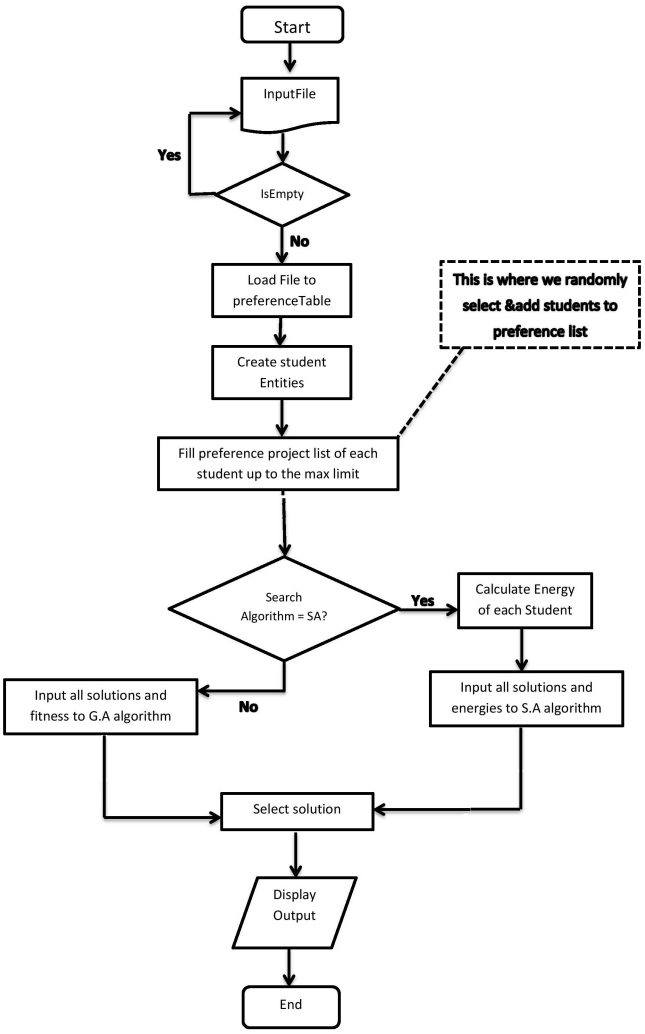
3.2 Use Case Diagram



3.3 Activity Diagram

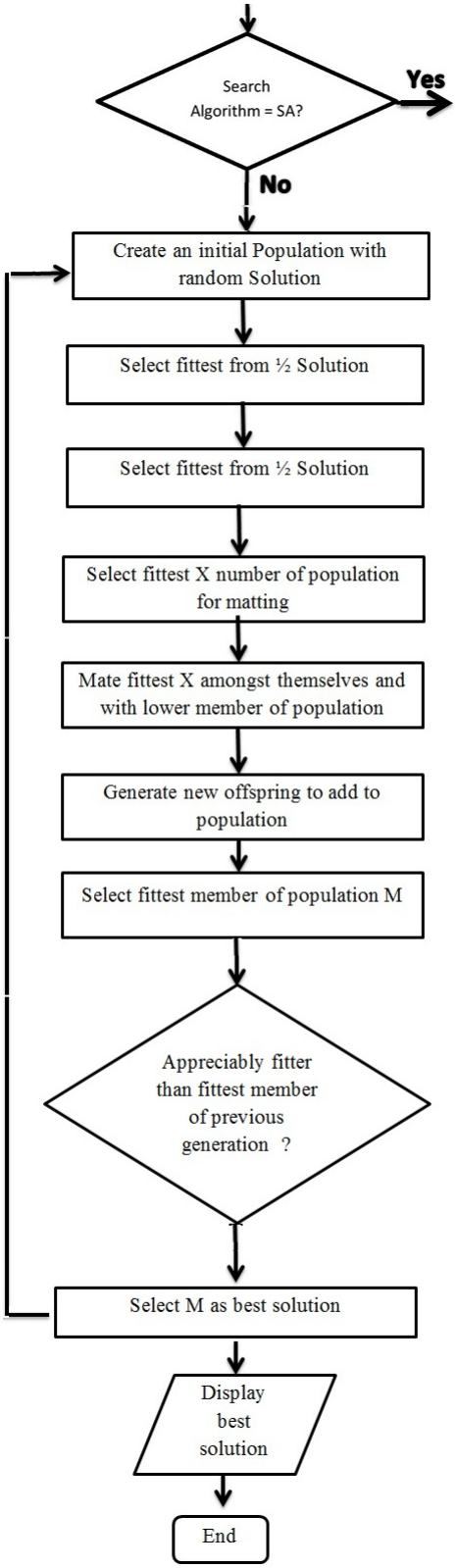


3.4 Flow Chart



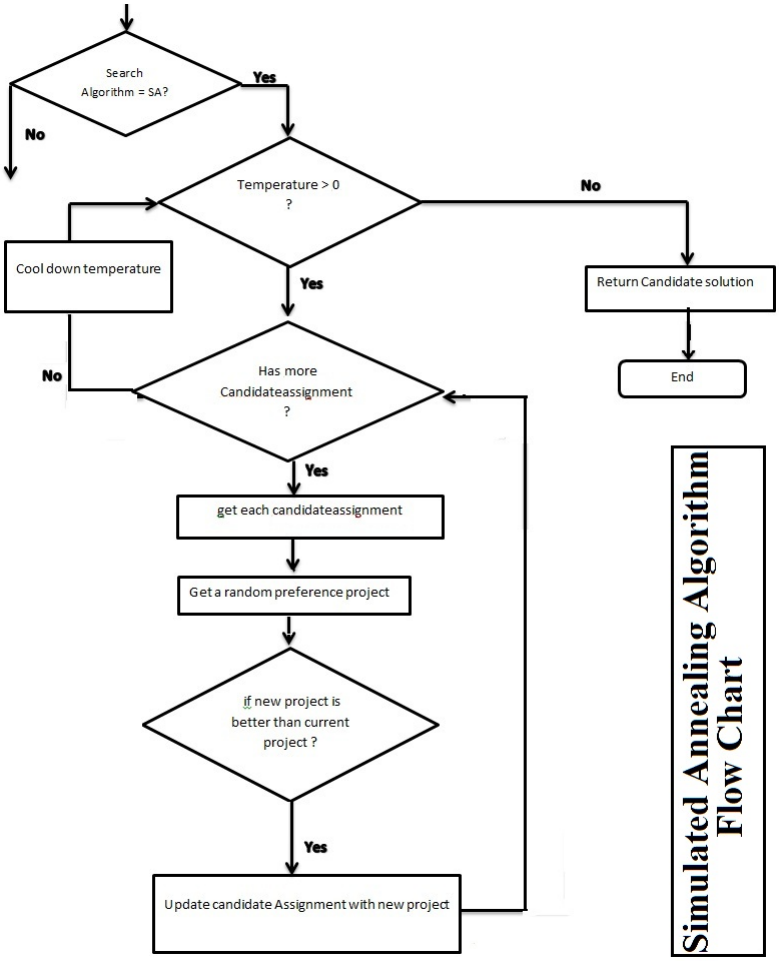
**** The elaborated Algorithm flow charts are included below ****

3.5 Genetic Algorithm Flow Chart

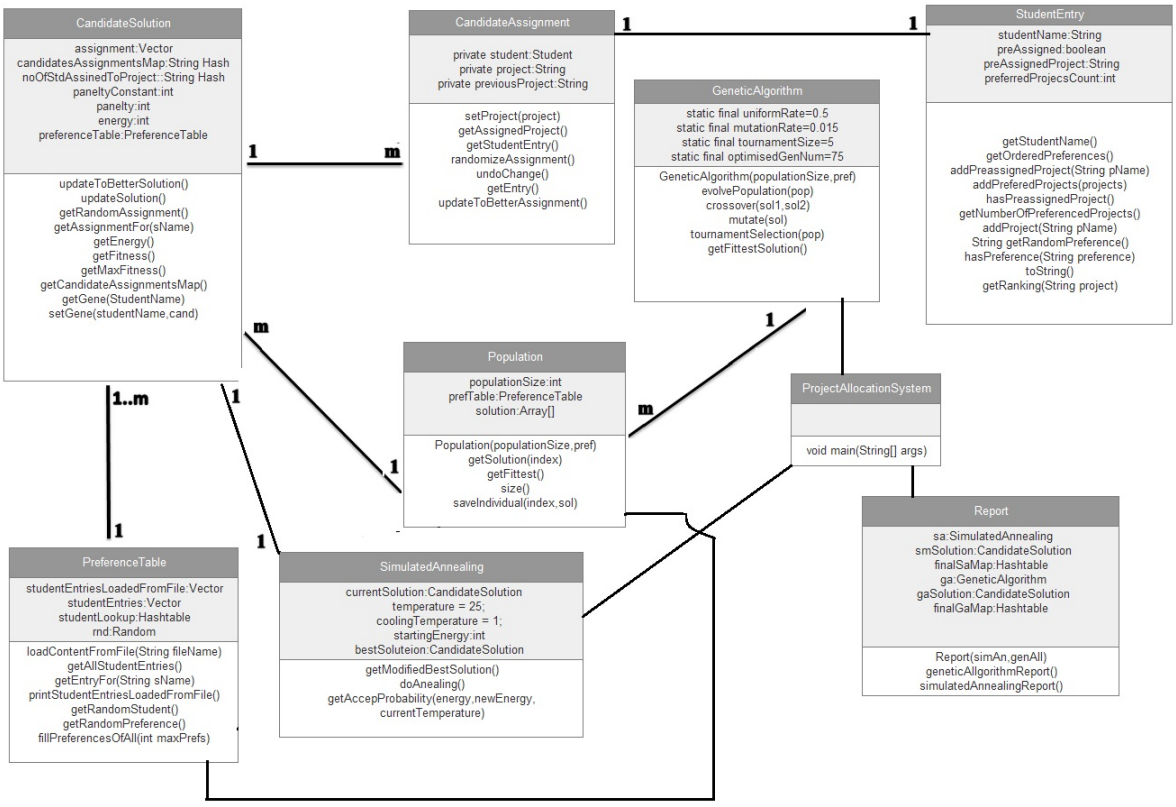


Genetic Algorithm Flow Chart

3.6 Simulated Annealing Flow Chart



3.7 Class Diagram



4 Project Analysis

4.1 What we have achieved

In the interim reporting we mentioned that we have divided the project into 3 essential parts. Here we have specified what we have achieved in order.

1. **Project Skeleton** : The assignments have been completed as the base model or the backbone. We have included some more additional helper functions to these classes for the aid of the main two algorithms.
Assignments : Completed
Additional helper Functions for the algorithms [*example :* updateToBetterSolution in the CandidateSolution class]

2. **Algorithms** : The two algorithms Simulated Annealing and Genetic Algorithm is the brain of the whole system. We have given much thought and man power to develop a bug free version of this algorithm.
Algorithm 1 : Simulated Annealing, The algorithm is implemented fully and tested.
Algorithm 2 : Genetic Algorithm, This algorithm is also implemented and tested.

3. **Output Report GUI and Validations** : The main output is the project allocation to the given student list in the .tsv file. This is displayed in all the console, the GUI and in the report. Other than that some main comparisons and validity evaluations have been done in the Analysis Reporting part. And also there are added extra functionality to the GUI itself.
Output : This System console output contains the final project allocations based on each algorithm separately. This is the most primitive and base level output that we generated.
Report : A Report to User is this part. This is an Analysis report that we prepared including allocation tables, energy and penalty calculations, and validity comparisons of the algorithms.
GUI : The Graphical User Interface that we developed as an additional feature for this project lets you choose the .tsv file then execute and get results for each algorithm and view the report and export it as a pdf file.

4.2 Additional Features

We added some extra functionality to this project

1. **The Algorithms and Reporting** : We have successfully build the two algorithms with extensive testing over the Weeks. As the presentation of results is the key performance measure here, we divide into three redundant parts. That is the above mentioned[console, report and GUI].

After the execution of the algorithm a pdf file is generated containing the results and evaluation of those results algorithm wise.

This is an additional feature that we included that this can be used to gain a clear statistical advantage, and easy result extraction.

2. **The GUI** : We build a simple graphical user interface to display the main results of the two algorithms and the report. This is build in a tabbed panel where user have a tab for *Genetic Algorithm*, a tab for *Simulated Annealing* and another tab to view the *Analysis Report*.

The GUI was built with the concept of simplicity in mind, where we kept the use of external graphical libraries as low as possible and tweak the java swing elements to get a user preferable interface.

3. **Coding Tweaks** : In *simulated Annealing* we included an additional function to update the current solution to a better solution other than getting a whole random solution. **updateToBetterSolution** in CandidateSolution class.

4.3 What we couldn't achieve and drawbacks

During the development process of this project we faced multiple obstacles. We came together as a group and came up with timely solutions just like the algorithms we are building.

1. **Platform Selection** : In the initial group meetings we agreed to use *eclipse IDE* as the development platform. But after we had done some implementation of the code and the GUI implementation started. We faced problems on finding a good Java Swing interface designer for eclipse. Although extensions like *WindowBuilder* exists it arose the question whether these add-ons will later be a problem when the lecturer is going to test the code.

So we migrated the whole project to the NetBeans Platform where there is default java Swing support is available and the code is executable without extension dependencies.

3. **Diagram Designing** : As the algorithms are complex and needs extensive study it took lot of time to understand and design different UML diagrams. Also we faced minor difficulties in finding a good UML diagram design tools.

5 Technical Specifications

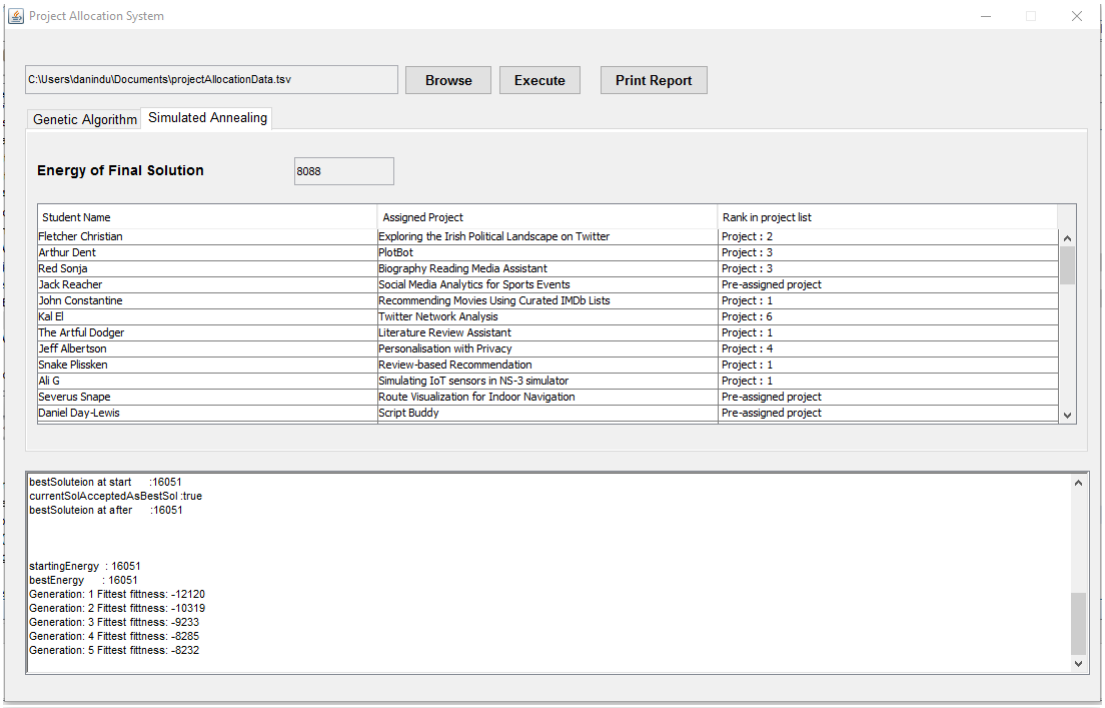
- Base language : **java**
- JDK Version : *jdk-1.7, jdk-1.8*
- Integrated Development Environment [IDE] : **NetBeans 8.+**
- Repository : *Bit Bucket*
- Version Control : **Git**
- Tested Environments : *Microsoft Windows, Mac Os, Ubuntu Linux*

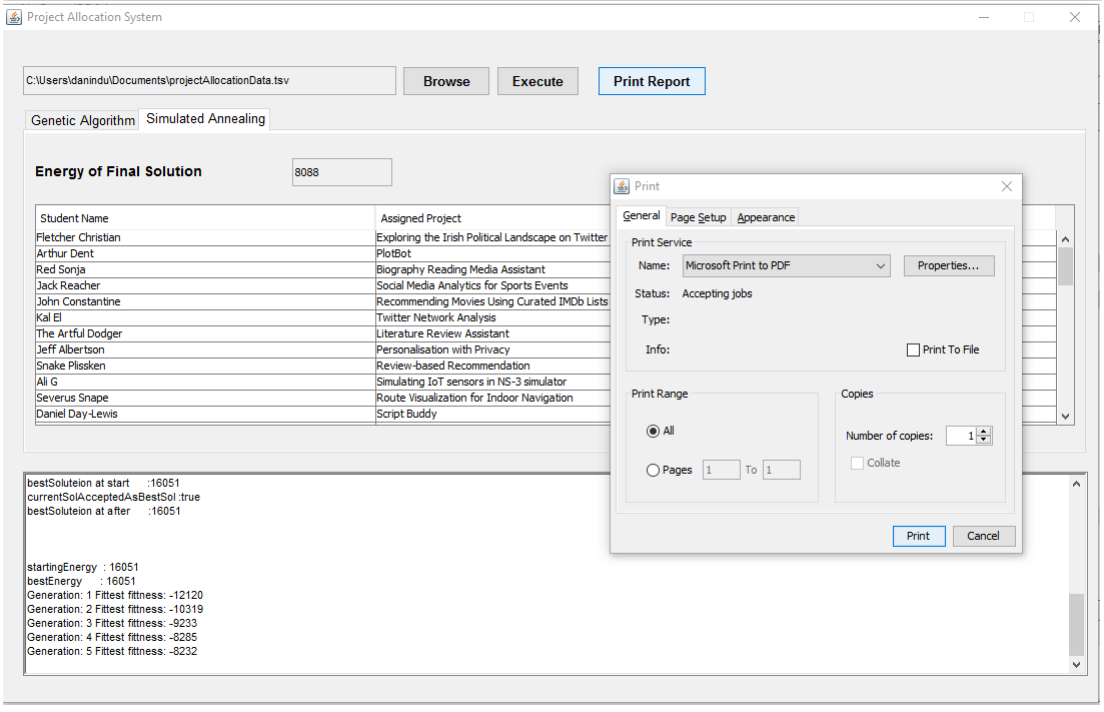
6 Team Analysis

Student Name	**Assigned Work**	**Progress**
Theshan Ronello	Version Control	1.Used Git Version Control base. Along with the Bit Bucket repository for Management 2.Different parts of the system is developed in Git Branches.
Theshan Ronello	Report Management	We Use LaTex environment to develop quality reports.
Theshan Ronello	Contributions	1. Helped in the project backbone, did the Algorithms and the Report
Theshan Ronello	Project Validation	1. Did testing on the overall project testing, the final validation and the upload
Nadeesha Lakmal	Contributions	1. Created the project backbone and did the Algorithms
Nadeesha Lakmal	Project Diagrams	1. Class Diagram 2. Flow Chart
Nadeesha Lakmal	Project Validation	1. Did Individual Class Validations

Kalana Chandrasiri	Requirement Gathering	1. Initial study of the project and work-package creation
Kalana Chandrasiri	Project Diagrams	1. Work Break Down Structure 2. Use-Case Diagram 3. Activity Diagram and helped in the report content creation.
Kalana Chandrasiri	Contributions	1. Contributed to the Overall design of the project participated in the project backbone creation
Danindu Gammanpila	GUI	1. Build the Graphical User Interface
Danindu Gammanpila	Contributions	1. Contributed to different classes mainly the result output optimization
Danindu Gammanpila	Project Validations	1. Validated the GUI Outputs

7 Screen Capture





Thank You!

Team ***Olympians:***

14208884 - Theshan Ronello [*theshan779@gmail.com*]
14210146 - Nadeesha Lakmal
14208883 - Kalana Chandrasiri
14208914 - Danindu Gammanpila