**Automated Project Allocation System**

**(APAS)**

Software Engineering Project II – COMP3006L

**Final Report**

**Team – 6droids**



**14208971 Tharkana D Kodagoda**

14208893 Sahitha Nelanga H De Silva

14208910 H W Srimal Priyanga Fonseka

14209059 Dilina Namal Weerasinghe

14209074 P W Poorni Yasodara

14209759 Kavindu Yudeesha Lakshan Narathota

**ABSTRACT**

**School of Computer Science and Informatics**

**University College Dublin**

This is the Final Project Report submitted for the Software Engineering Project II by **6droids.**

* “APAS” is a software solution which helps the user to assign projects to students in a way that it capitalize the link between students and their preferred projects.
* This project minimizes the possibility of getting an undesirable project for student by providing provision for system to assign the most preferred 1st, 2nd or 3rd project of students.
* In this project the user require to feed data such as Student Name, Pre-Arranged Project if they have done with a supervisor in advance and their preferences from 1 to 10.
* By providing a spreadsheet organized as above, generates a 1 to 1 mapping of students to project, giving each student with their highest preferences as for as possible. There is a need of defining the “valid” mapping and “best” mapping as many students will not get their top preferences.
  + Valid Mapping
    - is one each students gets out of their expressed preferences
    - gives priority to the pre-assigned projects given the spreadsheet
* Best Mapping
  + - Minimize the disappointment of students by giving the best to their preferences

Eg: A student who gets preference 1 has zero disappointment

A student who gets preference 2 has one unit of disappointment

Etc…

**Table of content**

1. **Project Specifications**
   1. **Requirements**

**Programming Language - Java**



Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

Source – Wikipedia

**IDE – NetBeans**



NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

Source – Wikipedia

* 1. **Functionalities**

**Searching for the “Best” Valid Mapping: Two Approaches**

By considering all valid mappings of projects to students, and return the mapping with least total disappointment as a solution.

In our system we have two stochastic techniques which are alternate approaches implemented by us as a part of our system.

* Simulated Annealing (SA)
* Generate random solutions
* Experiment with temperature schedules
* Define Energy Function
* Genetic Algorithm (GA)
* Generate random solutions
* Experiment with Population Sizes, Mating and Culling Policies
* Define Fitness Function

**Invalid Mapping**

Giving every student one of their preferred projects will not be possible at all given times. Especially if a student express only one preference. In such situations the student will be assigned with the left over project that has not been assign to anyone else.

This works in a way that our system assigns a random left over project to a student who states only one preference than it is to a student who has 10.

**Define Solution**

For the SA solution, run the algorithm 10 times and return the “best” solution. For the GA solution, run the algorithm for N generations and return the solution with the highest fitness. Report to the user whether the solution is a valid solution. If it is not, report to the user which student or students have been assigned projects they did not ask for. In every case, offer an assessment of the solution quality.

* 1. **Work – Package Overview**

When implementing “APAS” system below mentioned main tasks were identified where they were further divided into multiple tasks.

1. **Project Management**

* Assign Key Roles
* Project Plan

1. **Research and Development**

* Select Developing Tools, IDEs
* Select Best GUI Libraries

1. **Requirement Analysis**

* Work Breakdown Structure
* Project Gantt Chart
* Developing Plan

1. **UI, UX Prototyping**

* Wireframes
* High Fidelity
* UX and UI Review

1. **Architecture**

* Flow Charts
* Use – Case Diagram
* Class Diagram
* High Level Architecture

1. **Development**

* File Handling
* Develop Basic Mapping
* Implement Simulated Anneal
* Implement Genetic Algorithm
* Define Invalid Mapping
* SA Solution and Report Validity
* GA Solution and Report Validity
* Bug Resolving

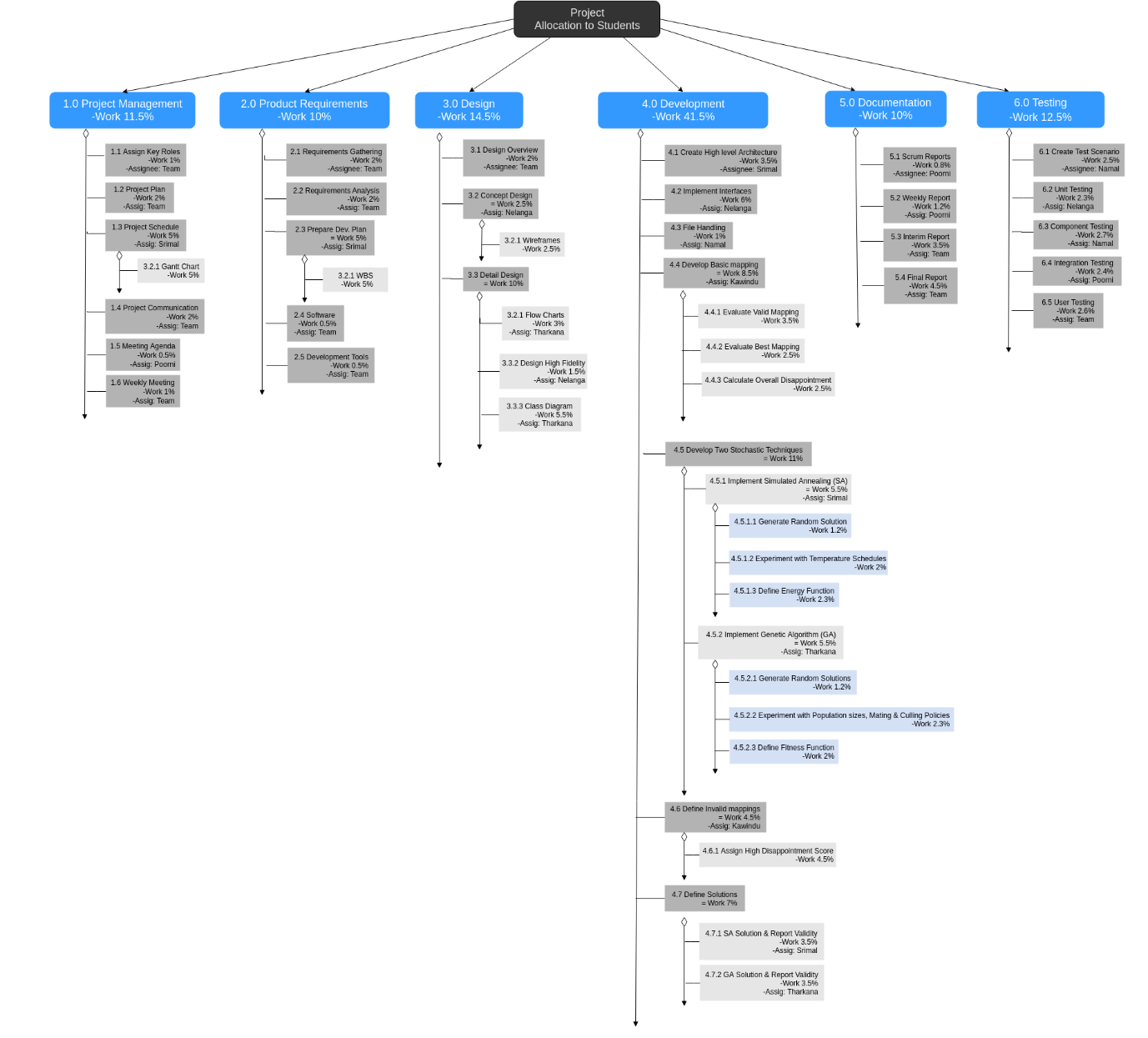
1. **Quality Assurance**

* Create Test Scenario
* Unit Testing
* Component Testing
* Integration Testing
* User Testing

1. **Documentation**

* Daily Scrum Notes
* Weekly Report
* Interim Report
* Final Report

1. **Submission**
   1. **Test Reports and Test Cases**
   2. **Work Breakdown Structure**

****

Read More - <https://goo.gl/1QFi8U>

1. **Software Implementation**

We divided the complete project into smaller divisions to make things easier and to reduce the work load of which each member gets. We call this division structure as '3D Structure'.

* Designing Team (Wireframes, UI)
* Developing Team (Flow Charts. Development)
* Documentation, Testing and QA Team (Reports, QA)
* **Research and Development**
* Development Tools -
* GitHub - We have used GitHub to use the coding collaborately to maintain time lines progress and to share the work progress of the users.
* Google Drive - We used the Google Drive to share documents and Images (Reports and Charts).
* Trello - Trello is used to manage the project and assign the work for members
* Development IDE -
* NetBeans - We used the Oracle NetBeans IDE for development as we are using Java SE. NetBeans is a very user friendly IDE for developing Java.
* **Requirement Analysis**
* Work Breakdown Structure (WBS) - First of all we have had a discussion to identify the project domain to plan a project plan and the obstacles we may get.
* Gantt Chart - After the WBS, we made a Gantt Chart to make project development deadlines for development parts.
* **UI, UX and Prototyping**
* Wireframes - The Designing Team has made a project UI schematic as a basic UI development.
* High Fidelity - After discussing and editing the wireframes for the requirement, the design team came up with a developed the wireframe to the actual design view.
* UI & UX Review - After coming up with a wireframe and High Fidelity designs we had make changes that we thought unnecessary. And also made changes for better UX
* **Architecture**
* Flow Charts - After planning the project we have developed a Flow Chart to make the program work flow.
* Class Diagram - After making the Flow Chart, Then moved on to make the Class diagram which describe the structure of the system by showing its classes, their attributes, operations (or methods), and the relationships among objects.
* High Level Architecture - Divided into separate modules to work differently. Data Input, SA Solution Module, GA Solution Module and Results View Module. Each module work separately to fulfill its intended work.
* UI Designing - For UI Designing we have used JAVA Swing and AWT libraries to make and develop the UI part of the project.
* **Development**
* File Handling - As an optional and additional feature, we have added a feature to load a data set (database) instead of hard coding it into the software.
* Develop Basic Mapping - An algorithm is used to map the projects to students to their most preferable selection.
* Implement Simulated Annealing (SA) - Simulated annealing is a probabilistic technique for finding a good (not necessarily perfect) solution to an optimization problem which is done by,
* Generate Random Solution
* Experiment with Temperature Schedules
* Define Energy Function
* Implement Genetic Algorithm (GA) - Genetic algorithm is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems which are,
* Generate Random Solutions
* Experiment with Population sizes, Mating & Culling Policies
* Define Fitness Function
* Define Invalid Mappings - Invalid mapping is when two students are assigned to a single project. This is resolved by Simulated Annealing (SA) and Genetic Algorithm (GA).
* SA Solutions & Report Validity - When selected this option to solve the invalid mapping SA Solution algorithm runs 10 times to finds the best suitable solution. When this shows the solution we review and validates it as the best.
* GA Solution & Report Validity - When selected this option, this solves as same as the SA Solution but, runs n times to view the best suitable solution for us to validate as the best solution.
* Bug Resolving - After going through the test scenarios, we encountered some code errors and logic errors. We have resolved it by recoding to go through again with test scenarios.
* **Quality Assurance**
* After developing the project to a nearly complete state, we have created and came up with Test scenarios to test the software with testing procedures. Which are Unit Testing, Component Testing, Integration Testing and User Testing.
  1. **Flow Chart Diagram**



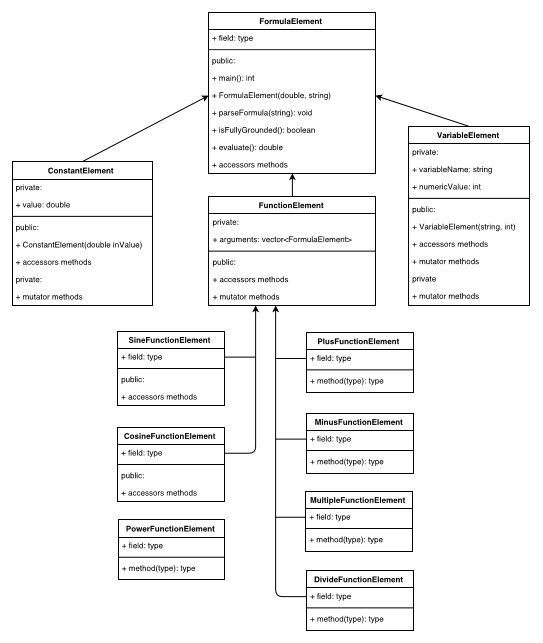
Read More - <https://goo.gl/iVVJp9>

* 1. **Use-Case Diagram**

****

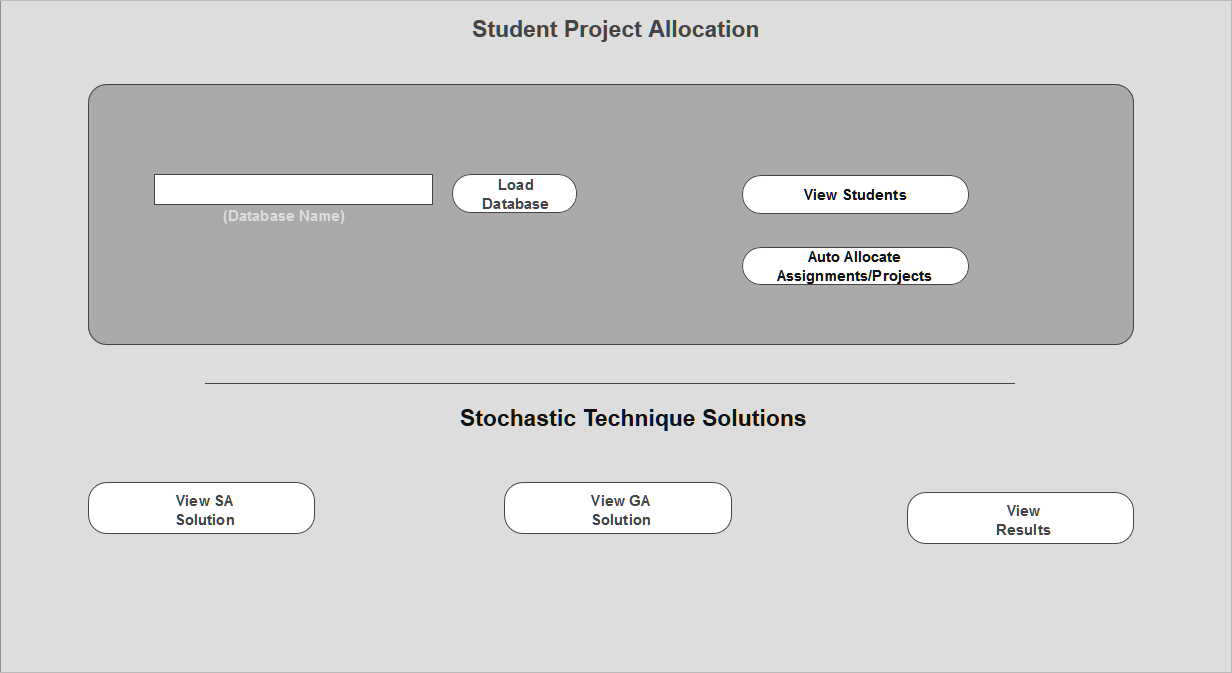
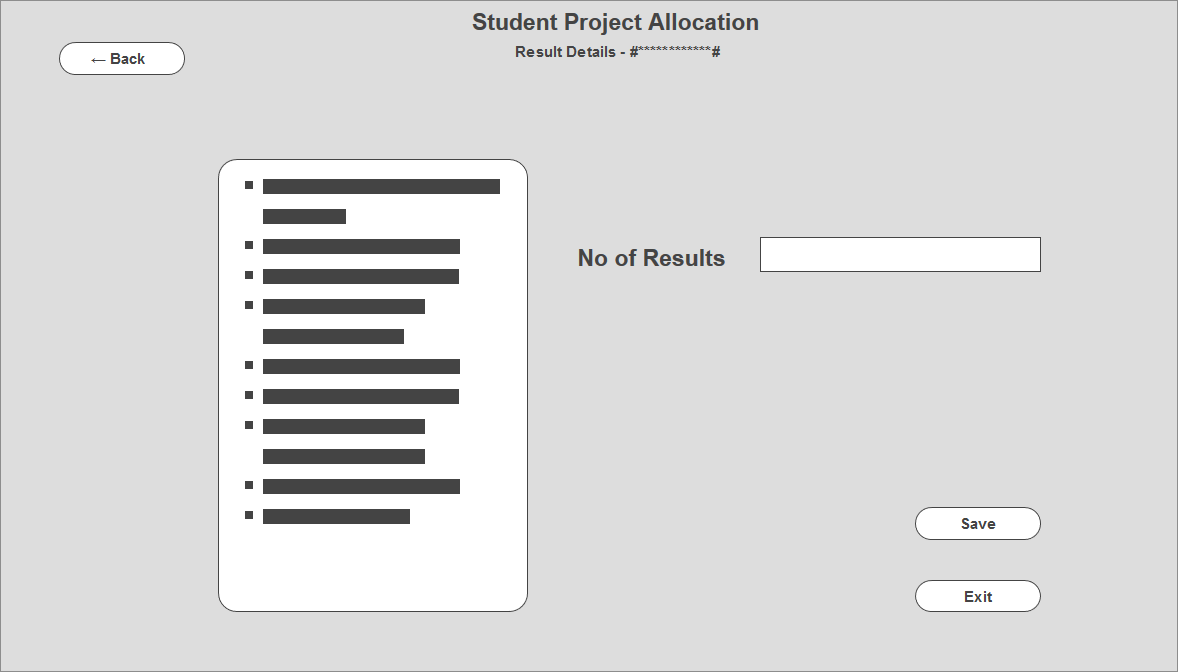
Read More - <https://goo.gl/45RT0K>

* 1. **Class Diagram**



Read More - <https://goo.gl/pbOnzp>

* 1. **Wireframes**



* 1. **User Interface**

**Home Screen**

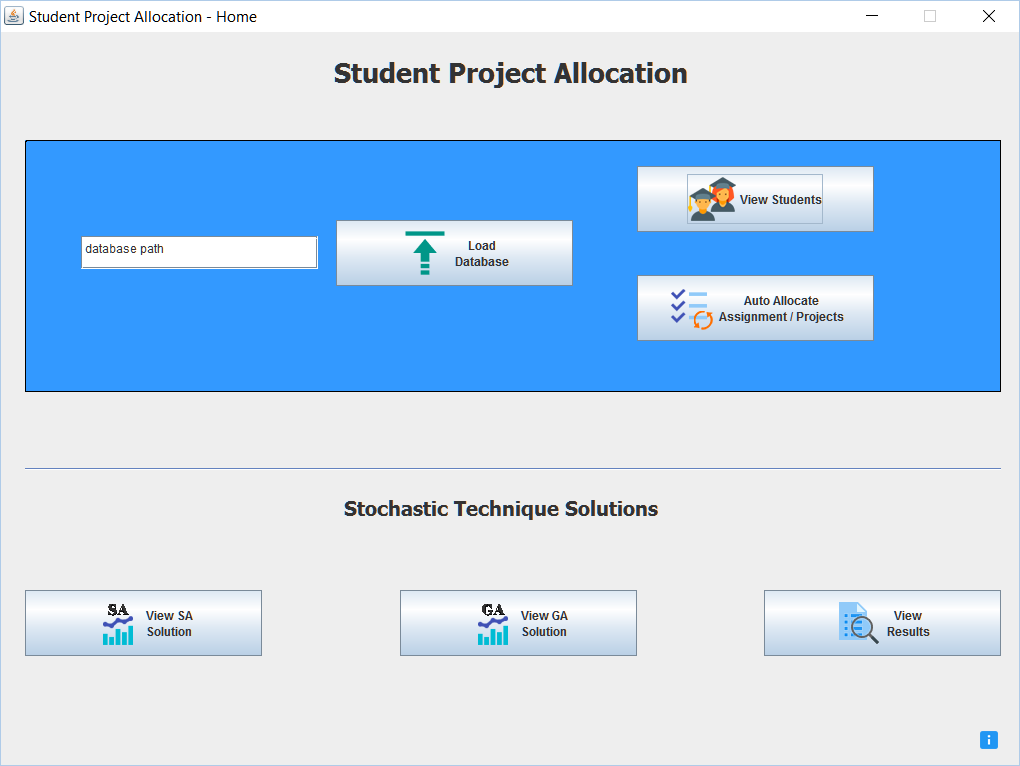
****

Figure 1 - Home Screen

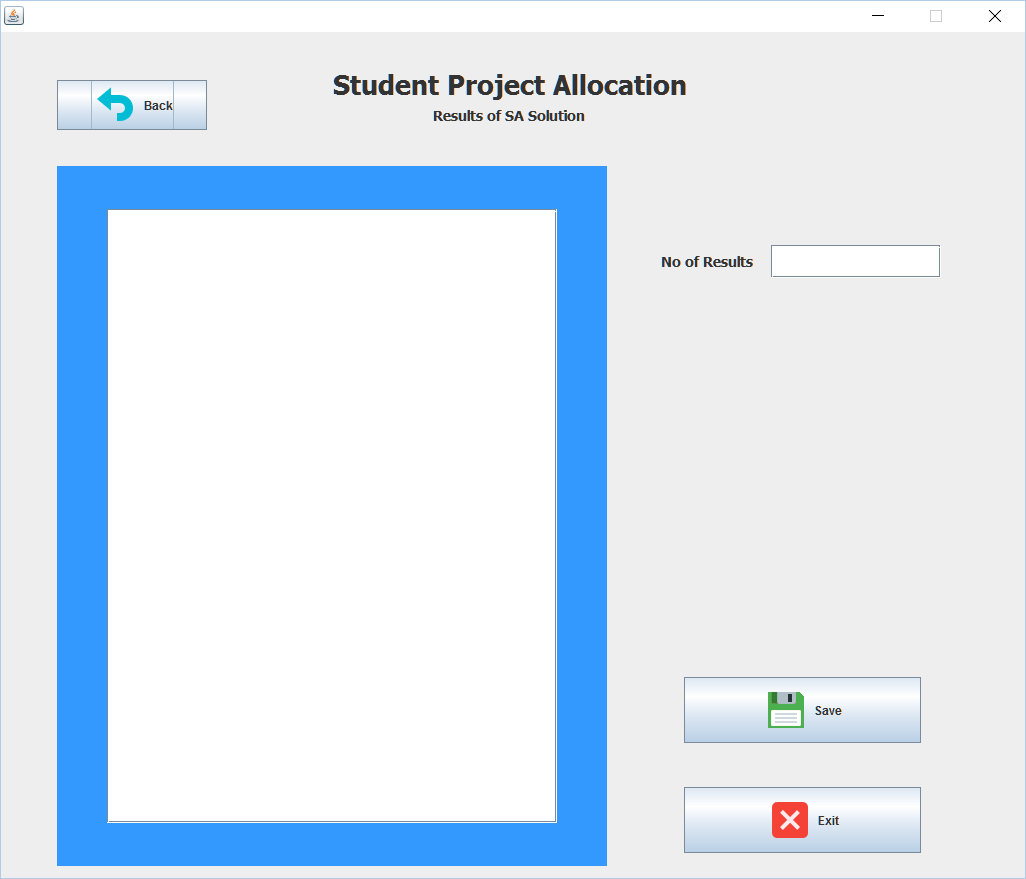
**Results Screen**

Figure 2 - Results Screen

1. **Analysis of Success/Failure**

From the day one we worked hard on our project. Sailed through so many difficult tasks and with dedication, team spirit we manage to overcome all barriers, difficulties to make it a success. Even though there were few short comings we made it a success.

**Successes**

* Implementation of Simulated Annealing
  + - * Generate Random Solution
      * Experiment with Temperature Schedules
      * Define Energy Function
* Implementation of Genetic Algorithm
  + - * Generate Random Solution
      * Experiment with Population Size, Mating and Culling Policies
      * Define Fitness Function

By successfully implementing above two techniques we were able to get the solution with lowest energy, solution with highest fitness and were able to report to user whether the solution is valid or not. If it’s an invalid solution, it will report to the user which student or students have been assigned projects they did not ask for. An assessment of the solution quality will be offered in every case.

1. **Additional Features**

We thought of developing “APAS” by adding some features where it becomes more user friendly.

* We have added a feature to load a data set (database) instead of hard coding it into the software. This gives the opportunity to user to choose the data set that user wants.

1. **Team Analysis**
   1. **Team Roles**

First thing we did in the project was delegating roles between team members. So we created a spreadsheet in Google Drive to vote team members for their expertise. We voted team members for their areas of expertise and at the same time vote ourselves for our strong areas. Through that we selected the best for their strengths accordingly.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Tharkana (T) | | | | | | Nelanga (Ne) | | | | | | Poorni (Po) | | | | | | Kavindu (K) | | | | | | Namal (Na) | | | | | | Priyanga (Pr) | | | | | |
|  | T | Ne | Po | K | Na | Pr | T | Ne | Po | K | Na | Pr | T | Ne | Po | K | Na | Pr | T | Ne | Po | K | Na | Pr | T | Ne | Po | K | Na | Pr | T | Ne | Po | K | Na | Pr |
| Team Leader |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Report Writers |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UI Designers |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Developers |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| QA |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Task** | **Assigned Persons** | **Follow up** |
| *Team Leader* | Tharkana |  |
| *Report Writers* | Poorni / Namal | Kavindu |
| *UI Designers* | Nelanga / Kavindu | Team |
| *Developers* | Tharkana / Priyanga / Kavindu / Nelanga |  |
| *QA* | Poorni / Namal | Team |

|  |  |
| --- | --- |
| Dev Team | Tharkana, Priyanga, Kavindu, Nelanga |
| Designing Team | Nelanga, Kavindu |
| Documentation Team (Reporting, QA) | Poorni, Namal |

* 1. **Project Sprint**

1. **Conclusion**

Great team spirit and endless nights of hard work strengthened our roots from day one to make this project a success.

The days spent with the project kept us all busy with a lot of tasks we accomplished in a timely and in an effective way. We shared all difficulties and the spirit of being such an amazing team made us better by day finally to make this project a success.