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

1.1. 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

1.2.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.

1.3. 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

2. **Research and Development**
    - Select Developing Tools, IDEs
    - Select Best GUI Libraries

3. **Requirement Analysis**
    - Work Breakdown Structure
    - Project Gantt Chart
    - Developing Plan

4. **UI, UX Prototyping**
    - Wireframes
    - High Fidelity
    - UX and UI Review

5. **Architecture**
    - Flow Charts
    - Use – Case Diagram
    - Class Diagram
    - High Level Architecture

6. **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

7. **Quality Assurance**
    - Create Test Scenario
    - Unit Testing
    - Component Testing
    - Integration Testing
    - User Testing

8. **Documentation**
    - Daily Scrum Notes
    - Weekly Report
    - Interim Report
    - Final Report
9. **Submission**

1.4.Test Reports and Test Cases

Testing Scope

- In Scope
  Functional Testing for the following modules are in Scope of Testing.
  - Genetic Algorithms
  - Simulated Annealing Algorithms
  - Genetic Algorithms and Simulated Annealing Algorithms
- Out of Scope

  Performance Testing was not done for this application.

Testing Approach

Testing that verifies the implementation of software elements in isolation.

- Manual testing
  - Black box testing - Acceptance Testing and System Testing
  - White box testing – Unit Testing and Integration Testing.

Unit testing

We have created the manual test cases and proceed with the test steps.

Test Case ID	Test Case	Description	Expected Outcome
TC_001	Click on "Load Dataset' button in Automated Project Allocation System window.	System will show Button call "Load Dataset" with text field next to it.	Once you click on the button, System pop out the new window. Search the tsv file and open it. Then in the text field it will show the data set name and the file path.
TC_002	Click on "View Dataset" button Automated Project Allocation System window.	In the system it will display the all data set in a table formate according to the data file.	System will display the student's project allocation data with preferences.
TC_003	Click on "Back" button.	once you click on back button it will navigate to the home window.(Automated Project Allocation System)	System navigate to the Home window.(Automated Project Allocation System)

Read more - https://goo.gl/45NO5z

Defect Analysis

We have created the manual test cases and proceed with the test steps.

Defect ID	Defect Description	Out Come	Priority	Issue Raised By	Assigned User	Status
PA001	SA algorithm in single run will replace initial solution values by the best solution.	In single execution the values for best and initial will be exchanged.	High	Namal	Srimal	Fixed
PA002	Data set not getting randomize in the beginning	It will display only the default values (basic tsv format)	Medium	Poorni	Kavindu	Fixed

Read more - https://goo.gl/6QNxu2

Code Efficiency

Analyze coding standards and verify whether the classes have properly implemented.

Types of testing performed

- Smoke Testing
  This testing was done whenever a Build is received (deployed into Test Environment) for Testing to make sure the major functionalities are working fine, Build can be accepted and Testing can start.
- System Integration Testing
  This is the Testing performed on the Application under test, to verify the entire application works as per the requirements.
- Regression Testing
  This testing ensures that existing functionalities works fine after defect fix and new enhancements are added to the existing application.

Lessons Learnt

- Issue: Smoke testing test cases required to be executed manually each time.
- Solution: Smoke test cases were automated and the scripts were run, which ran fast and saved time.

1.5. Work Breakdown Structure

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

2. 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

- Use case diagram – To start off with the designing phase, we designed the use case diagram to the project in hand.
- 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.

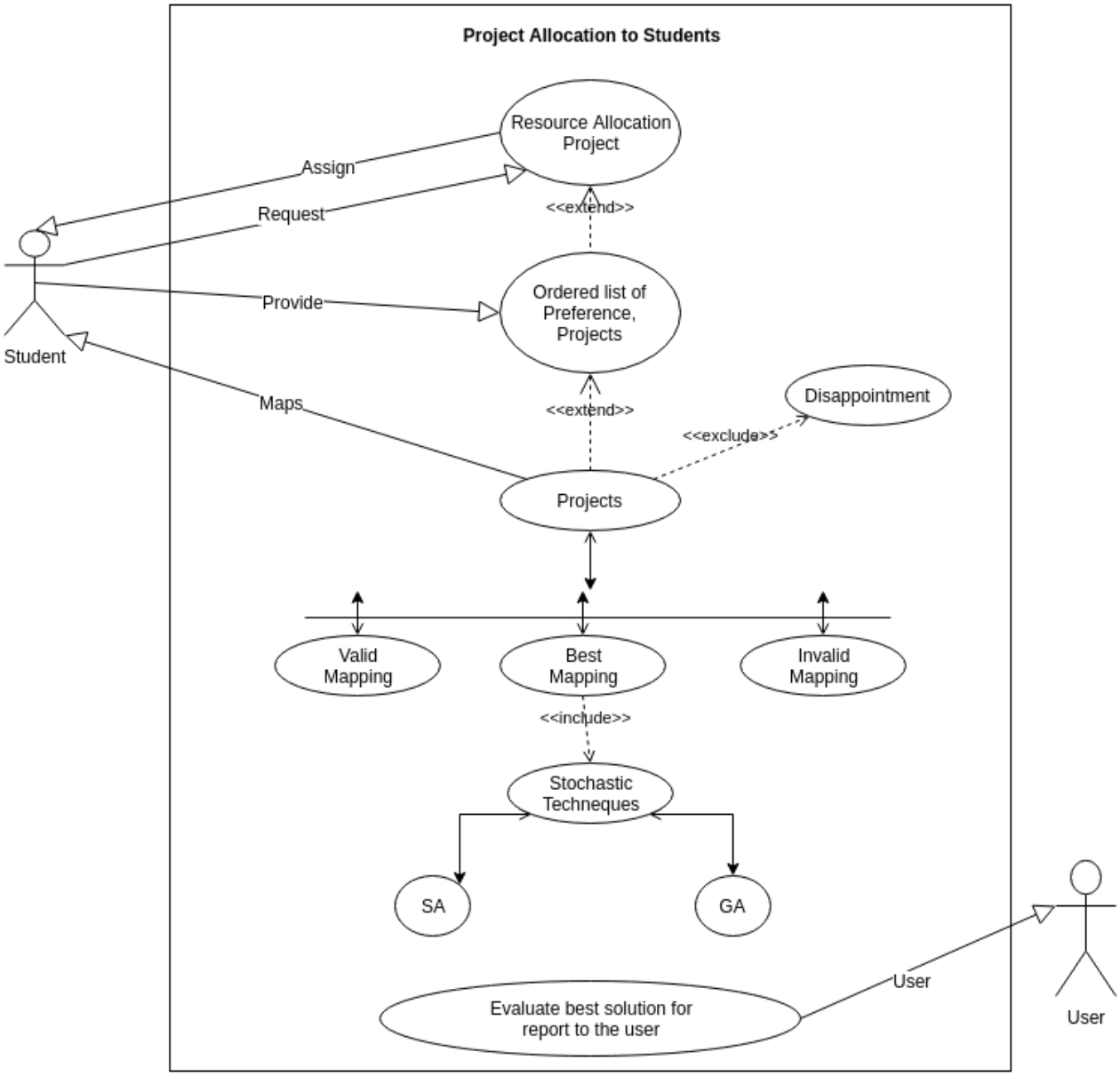
❖ **Development**

- 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**

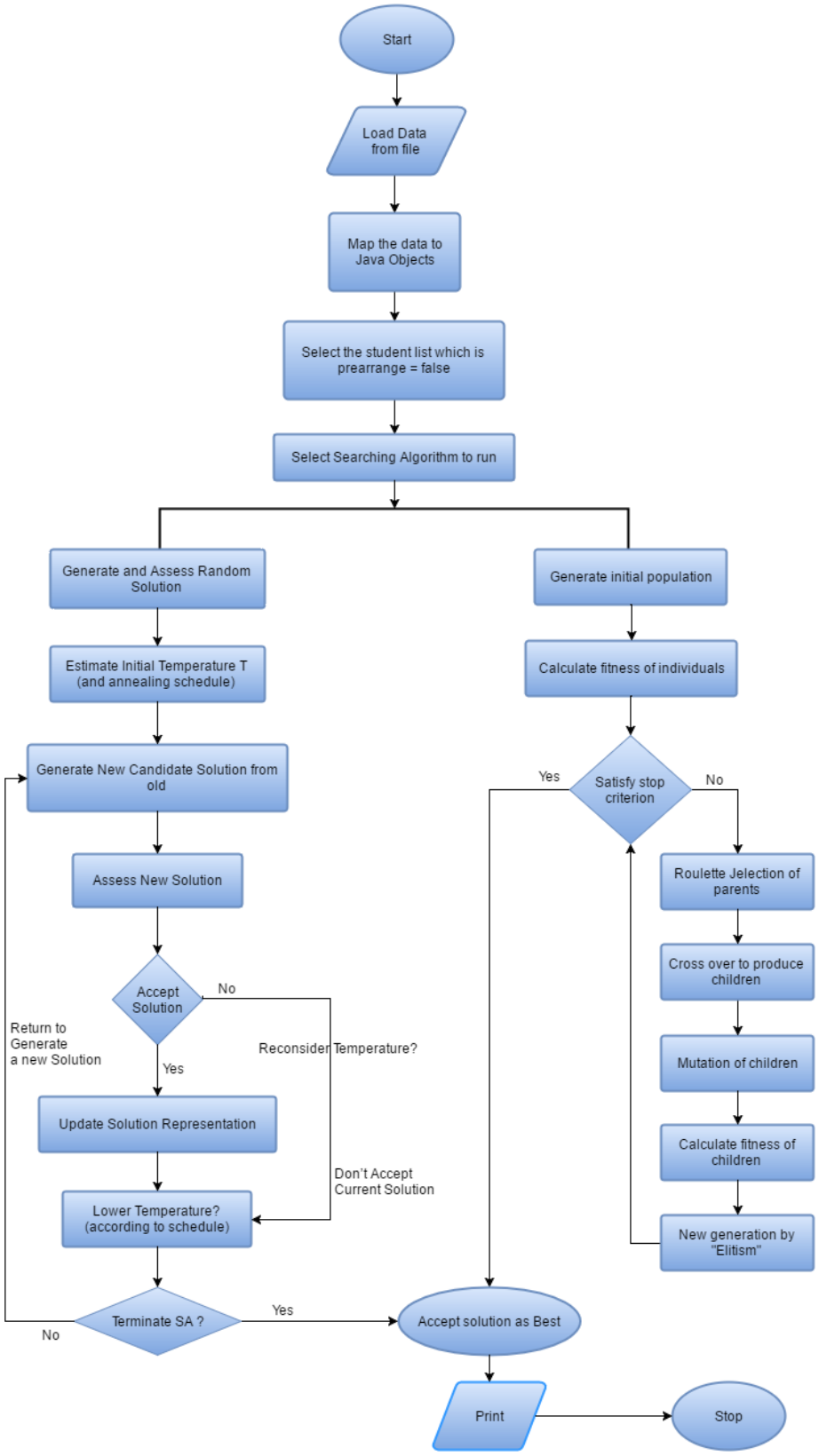
After developing the project to a nearly complete state, we have created and came up with Test scenarios to test the software with testing procedures. Which are Unit Testing, Component Testing, Integration Testing and User Testing.

2.1.Use-Case Diagram



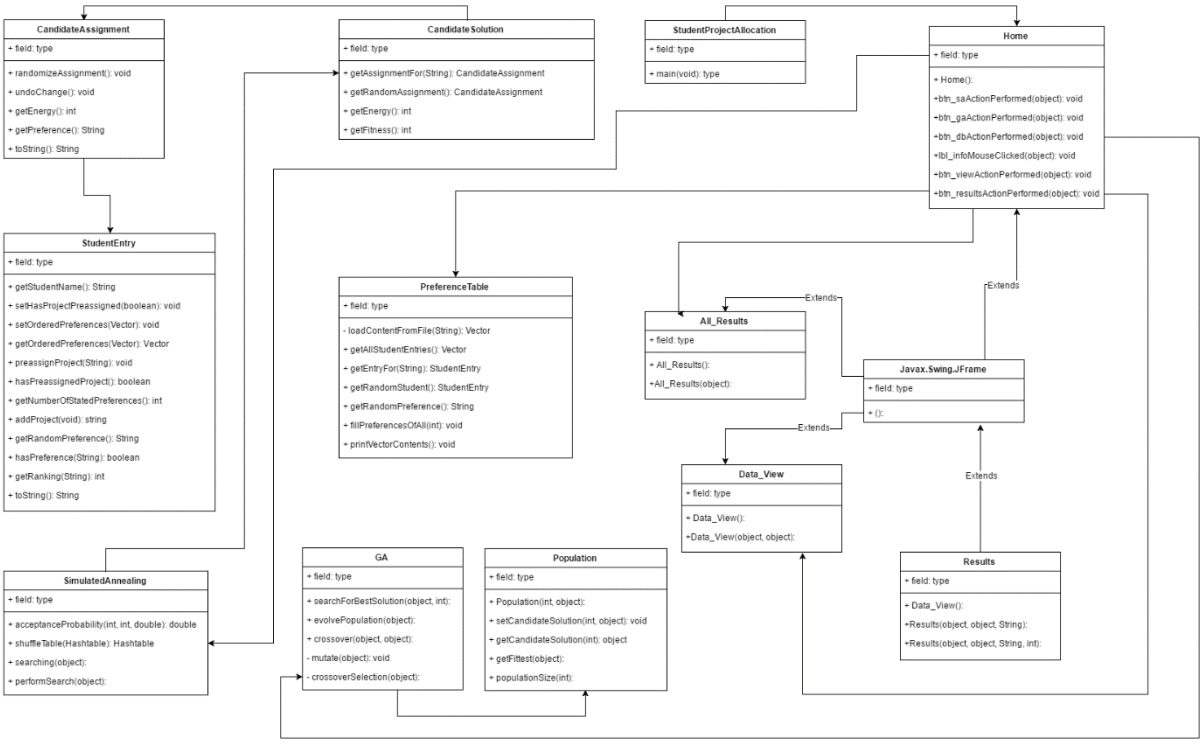
Read More - https://goo.gl/45RT0K

2.2.Flow Chart Diagram



Read More - https://goo.gl/0C9w6h

2.3.Class Diagram



Read More - https://goo.gl/qwHEEq

2.4. Wireframes

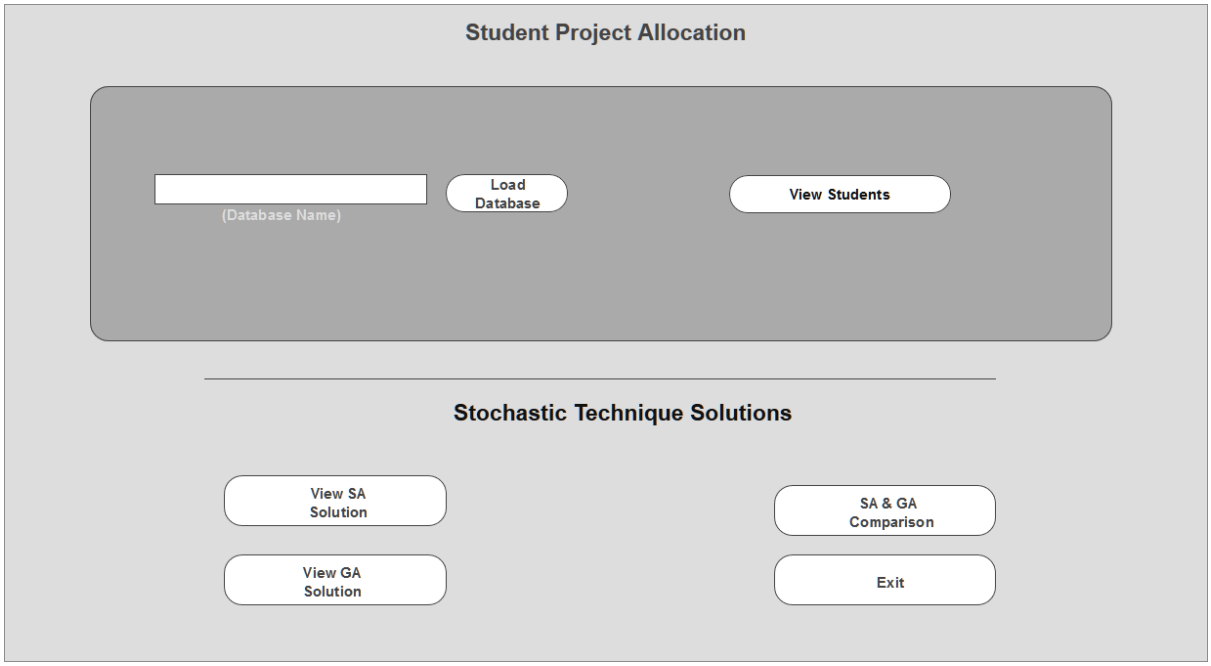


Figure 1 - Home

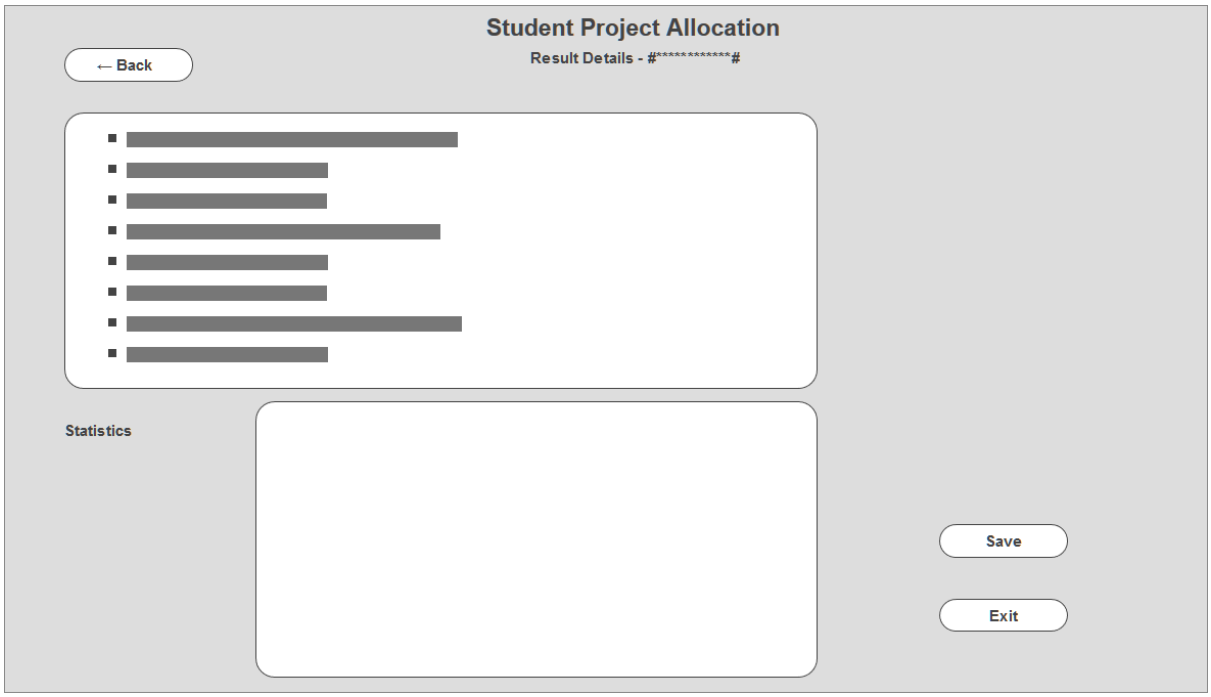


Figure 2 - Results

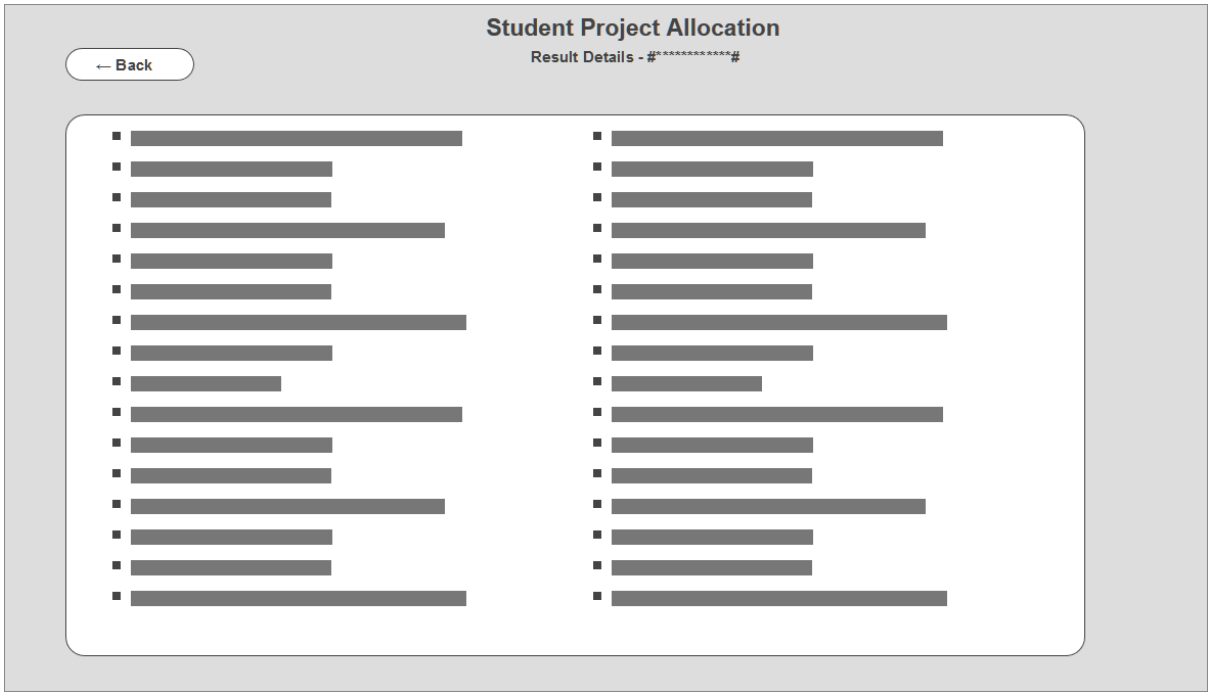


Figure 3 - Comparison

2.5. User Interfaces

Home Screen

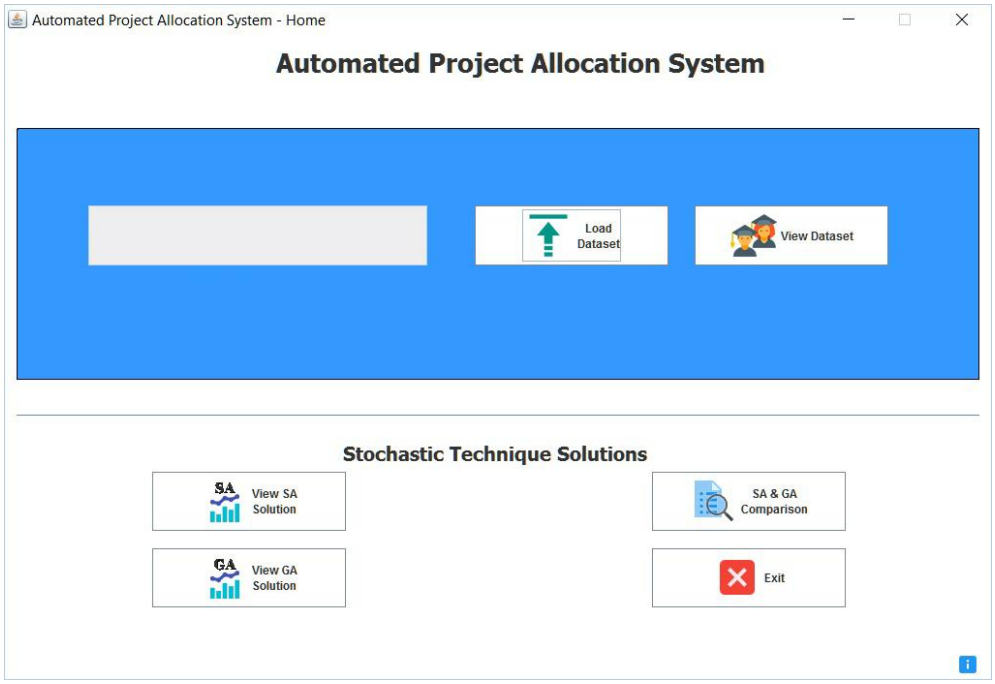


Figure 1 - Home

Data Set Screen

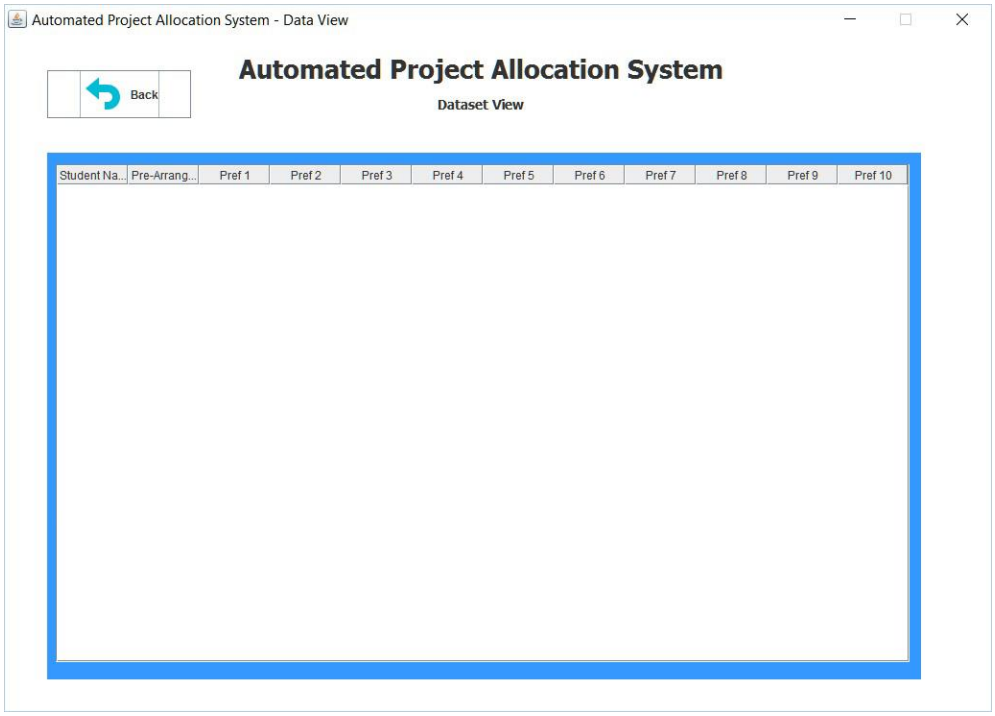


Figure 2 - Data Set Screen

Simulated Annealing Results

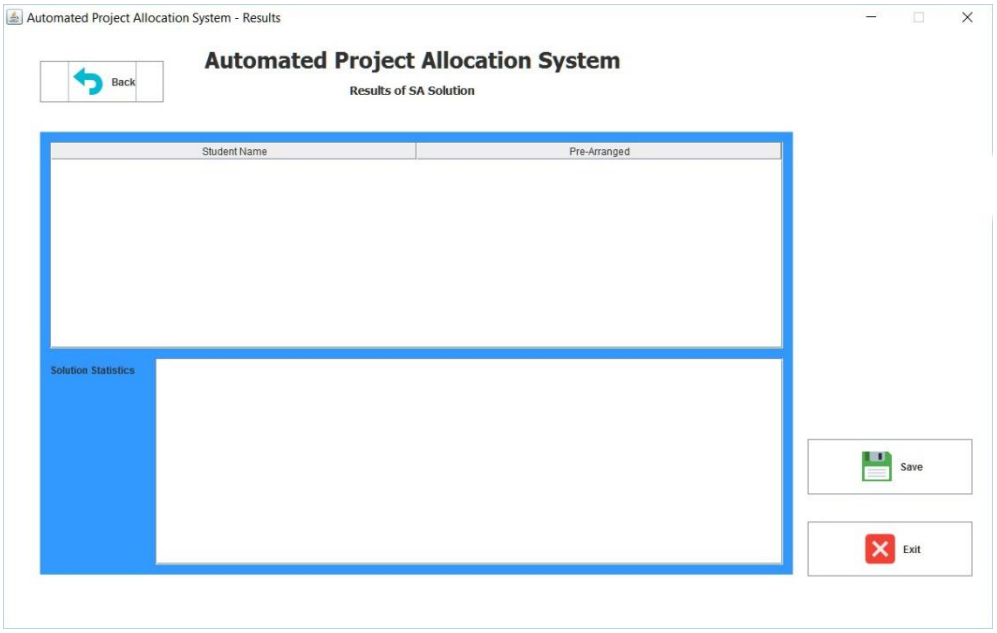


Figure 3 - Simulated Annealing Results

Generic Algorithm Results

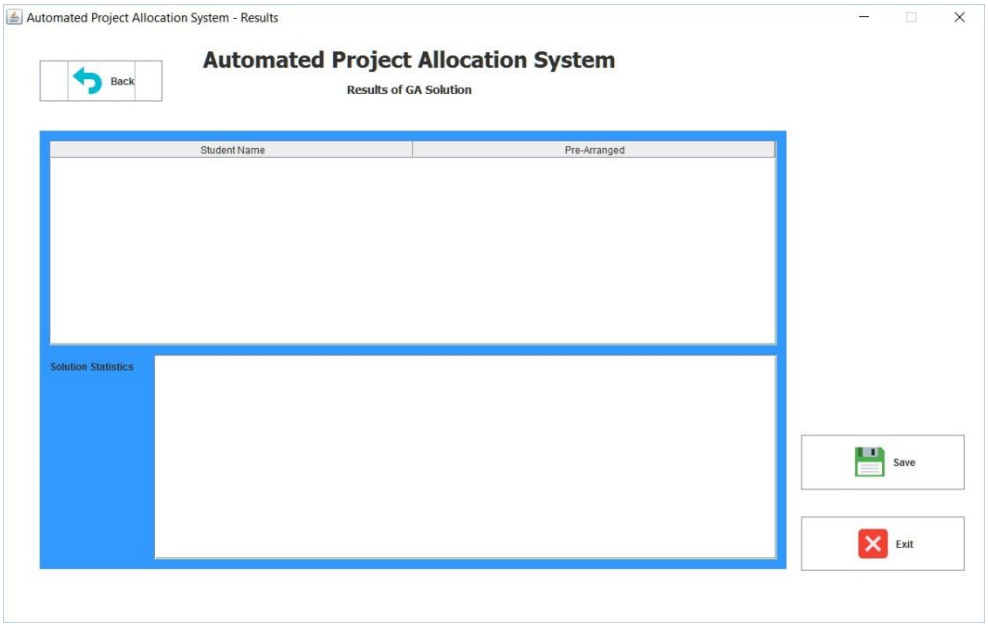


Figure 4 - Generic Algorithm Results

Results Comparison

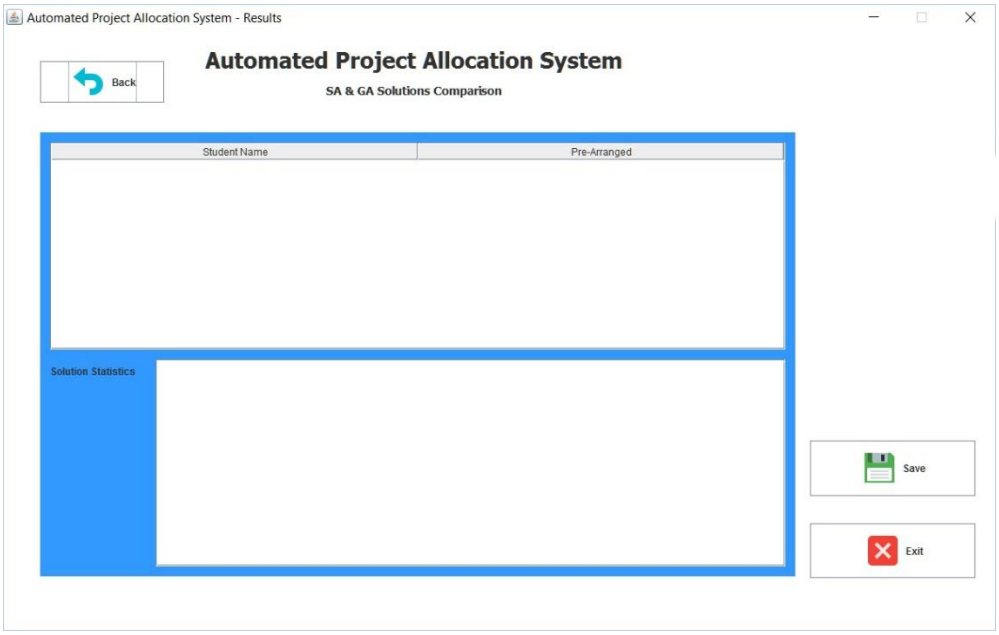


Figure 5 - Result Comparison

3. 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.

4. Additional Features

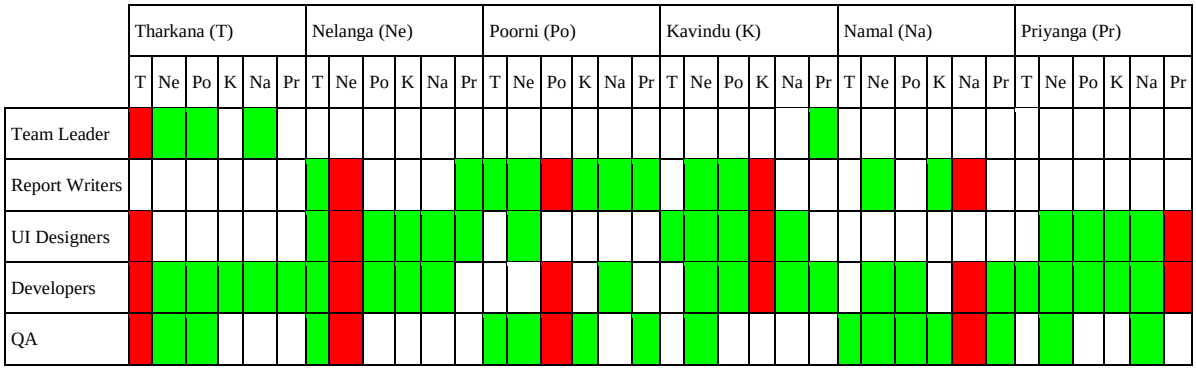
We thought of developing “APAS” by adding some features where it becomes more user friendly.

- 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.
- User can view the loaded data from the “View data set” interface before its being processed by the system.
- User can observe SA Result set and GA Result set in two different interfaces
- In addition to the previous feature, user can compare both result sets in the comparison interface
- User can save all 3 result sets in to files using the system.

5. Team Analysis

5.1. Team Roles

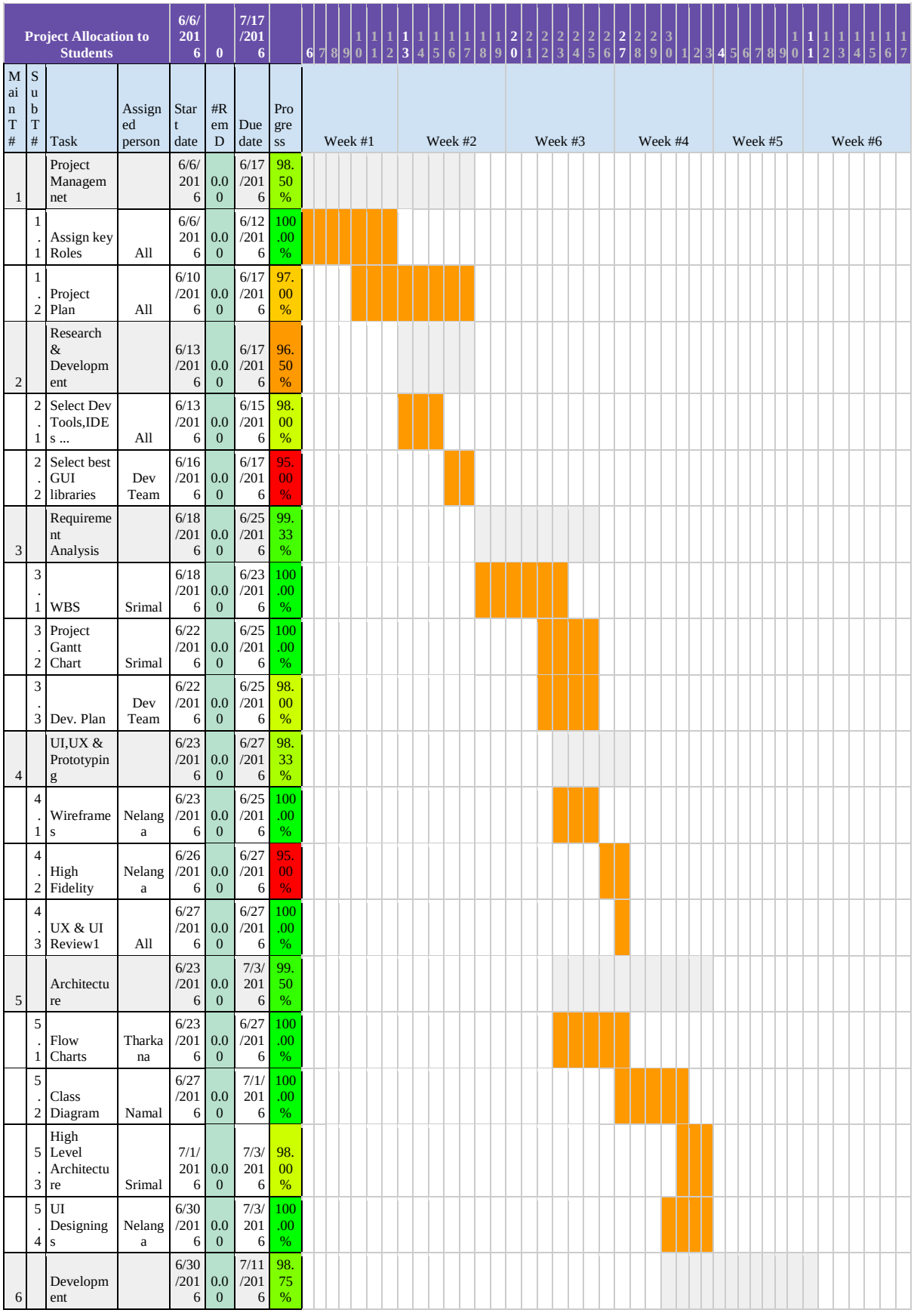
First thing we did in the project was delegating roles between team members. So we created a spreadsheet in Google Drive to vote team members for their expertise. We voted team members for their areas of expertise and at the same time vote ourselves for our strong areas. Through that we selected the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



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

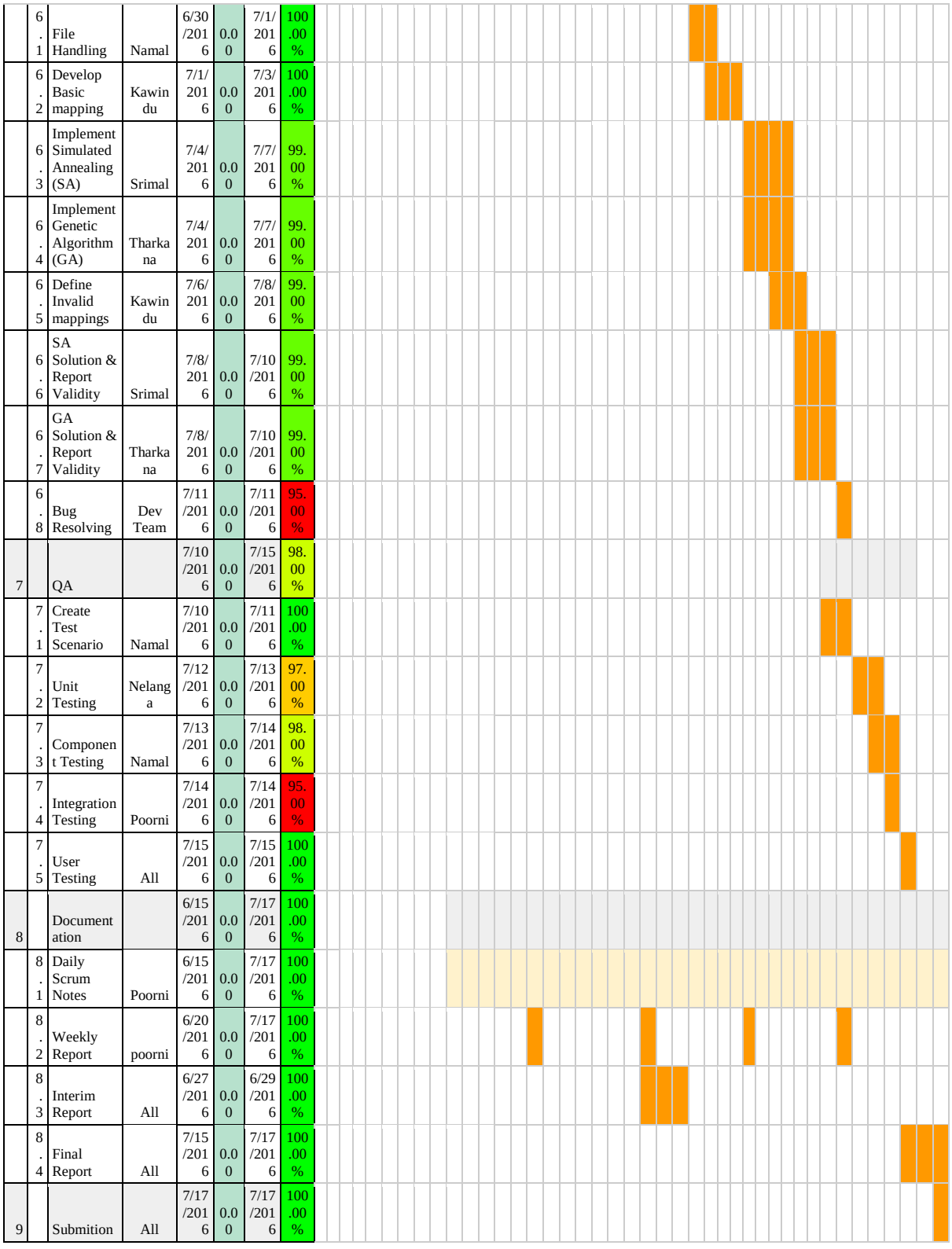
Dev Team	Tharkana, Priyanga, Kavindu, Nelanga
Designing Team	Nelanga, Kavindu
Documentation Team (Reporting, QA)	Poorni, Namal

5.3.Project Sprint



Project Allocation to Students				6/6/2016	0	7/17/2016	
Main T #	Sub T #	Task	Assigned person	Start date	#Rem D	Due date	Progress
1		Project Managemnet		6/6/2016	0.00	6/17/2016	98.50 %
	1.1	Assign key Roles	All	6/6/2016	0.00	6/12/2016	100.00 %
	1.2	Project Plan	All	6/10/2016	0.00	6/17/2016	97.00 %
2		Research & Development		6/13/2016	0.00	6/17/2016	96.50 %
	2.1	Select Dev Tools,IDEs ...	All	6/13/2016	0.00	6/15/2016	98.00 %
	2.2	Select best GUI libraries	Dev Team	6/16/2016	0.00	6/17/2016	95.00 %
3		Requirement Analysis		6/18/2016	0.00	6/25/2016	99.33 %
	3.1	WBS	Srimal	6/18/2016	0.00	6/23/2016	100.00 %
	3.2	Project Gantt Chart	Srimal	6/22/2016	0.00	6/25/2016	100.00 %
	3.3	Dev. Plan	Dev Team	6/22/2016	0.00	6/25/2016	98.00 %
4		UI,UX & Prototyping		6/23/2016	0.00	6/27/2016	98.33 %
	4.1	Wireframes	Nelanga	6/23/2016	0.00	6/25/2016	100.00 %
	4.2	High Fidelity	Nelanga	6/26/2016	0.00	6/27/2016	95.00 %
	4.3	UX & UI Review1	All	6/27/2016	0.00	6/27/2016	100.00 %
5		Architecture		6/23/2016	0.00	7/3/2016	99.50 %
	5.1	Flow Charts	Tharkana	6/23/2016	0.00	6/27/2016	100.00 %
	5.2	Class Diagram	Namal	6/27/2016	0.00	7/1/2016	100.00 %
	5.3	High Level Architecture	Srimal	7/1/2016	0.00	7/3/2016	98.00 %
	5.4	UI Designings	Nelanga	6/30/2016	0.00	7/3/2016	100.00 %
6		Development		6/30/2016	0.00	7/11/2016	98.75 %

	6.1	File Handling	Namal	6/30/2016	0.00	7/1/2016	100.00%
	6.2	Develop Basic mapping	Kawindu	7/1/2016	0.00	7/3/2016	100.00%
	6.3	Implement Simulated Annealing (SA)	Srimal	7/4/2016	0.00	7/7/2016	99.00%
	6.4	Implement Genetic Algorithm (GA)	Tharkana	7/4/2016	0.00	7/7/2016	99.00%
	6.5	Define Invalid mappings	Kawindu	7/6/2016	0.00	7/8/2016	99.00%
	6.6	SA Solution & Report Validity	Srimal	7/8/2016	0.00	7/10/2016	99.00%
	6.7	GA Solution & Report Validity	Tharkana	7/8/2016	0.00	7/10/2016	99.00%
	6.8	Bug Resolving	Dev Team	7/11/2016	0.00	7/11/2016	95.00%
7		QA		7/10/2016	0.00	7/15/2016	98.00%
	7.1	Create Test Scenario	Namal	7/10/2016	0.00	7/11/2016	100.00%
	7.2	Unit Testing	Nelanga	7/12/2016	0.00	7/13/2016	97.00%
	7.3	Component Testing	Namal	7/13/2016	0.00	7/14/2016	98.00%
	7.4	Integration Testing	Poorni	7/14/2016	0.00	7/14/2016	95.00%
	7.5	User Testing	All	7/15/2016	0.00	7/15/2016	100.00%
8		Documentation		6/15/2016	0.00	7/17/2016	100.00%
	8.1	Daily Scrum Notes	Poorni	6/15/2016	0.00	7/17/2016	100.00%
	8.2	Weekly Report	poorni	6/20/2016	0.00	7/17/2016	100.00%
	8.3	Interim Report	All	6/27/2016	0.00	6/29/2016	100.00%
	8.4	Final Report	All	7/15/2016	0.00	7/17/2016	100.00%
9		Submition	All	7/17/2016	0.00	7/17/2016	100.00%



Read more - https://goo.gl/YG5o09

6. 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.

Hope this project meet the required specifications to the point and our specially added features will help the user as well.

There's a lot more that we could say about our project, issues we faced and also how we managed them, but we believe this short report will give you a good understanding about our project.