

Deliverable #2

Group #4

Attia, Abrar - attiaa1 - 400017188

Ansell, Evan - ansellea - 1415992

Fayez, Susan - fayezs - 001404420

Yin, Hao - yinh1 - 400016540

Yang, Zhiwen - yangz18 - 400023048

2018-03-09

1 Introduction

1.1 Purpose

The purpose of this document is to provide a thorough description of the whole system with different kinds of system diagrams. The document will also include a summary on the architecture design of the system. In addition, the following document will be used to build and develop the classes required for the Forester application. The intended audience of the document is mainly the software developers and project managers.

1.2 System Description

The Forester system is a plant identification system implemented by Blackboard Architecture. It accepts three user types: average users, researchers and administrators. They have different levels of permission to view and manipulate data from the data source. The system stores plant data and identifies designated matches according to the input of users. The users enter a number of different plant characteristics which are checked by the relevant experts. Then the system fetches data from the data source and displays output to the users.

The whole system contains four entities(Plant Data, Search History, Registered Users, Modifications) which hold all the data of the system. With the help of controller classes(I/O Controller, Identification Experts, Security, Modification Controller), users are able to login, edit account information, send input, receive output and edit plant data. The system also consists of eleven boundary classes: Identify Plant, Results, View Search History, View Data, Login, Login Error, Change Password, Submit Modifications, Manage Modifications, Researcher and Administrator. They communicate with entity classes by controllers to realize data transmission back and forth.

1.3 Overview

The remaining sections of the document contains two different system diagrams which are the use case diagram and the analysis class diagram. The overall architectural design of the application is also discussed later on and focuses on two aspects which are the system architecture and the subsystems. This is then followed by class responsibility collaboration cards for the boundary classes, control classes, and entity classes. The document then ends with a division of labour sheet.

2 Use Case Diagram

1. The User wants to identify a plant. They can either answer the ID questions to identify a specific plant or enter a location to identify the plants native to that area. The system produces the results.
2. The User wants to view the results of a past search, the system displays it.
3. The User wants to clear their search history. The system prompts them to confirm the deletion.
4. The Researcher or the Administrator wants to log in to the system. The system validates their credentials.
5. The Researcher wants to view the database. The system displays it.
6. The Researcher wants to submit a Database Update Request. The type of request can be either an Edit, Addition, or Deletion.
7. The Researcher wants to view their contributions. The system displays them.
8. The Researcher or the Administrator wants to update their password. The system prompts them to confirm the update.
9. The Administrator wants to manage the Database Update Requests. They can either display the requests, confirm requests, or deny requests.

Figure 1 displays the Use Case Diagram for the Forester system.

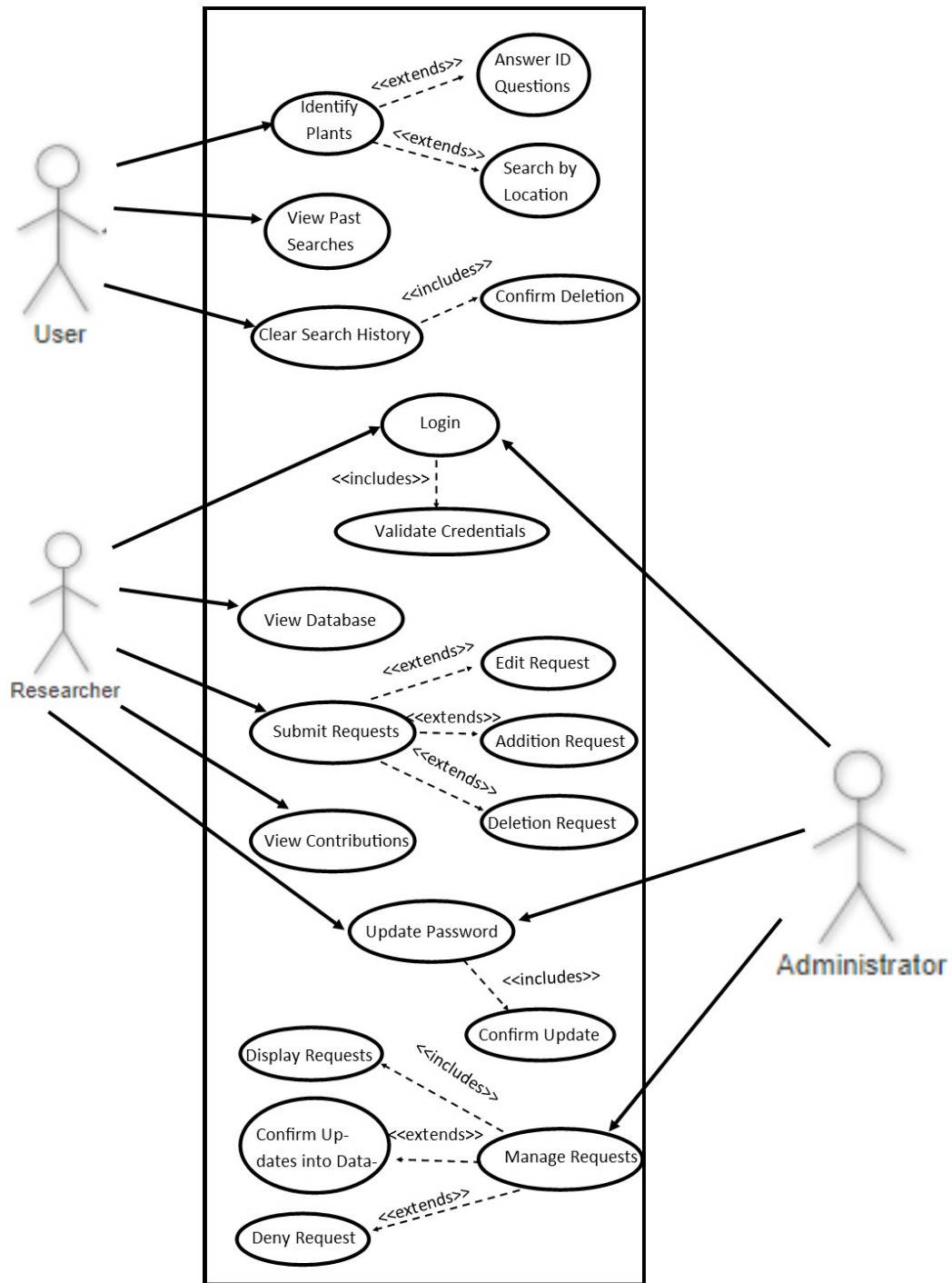


Figure 1: Use Case Diagram

3 Analysis Class Diagram

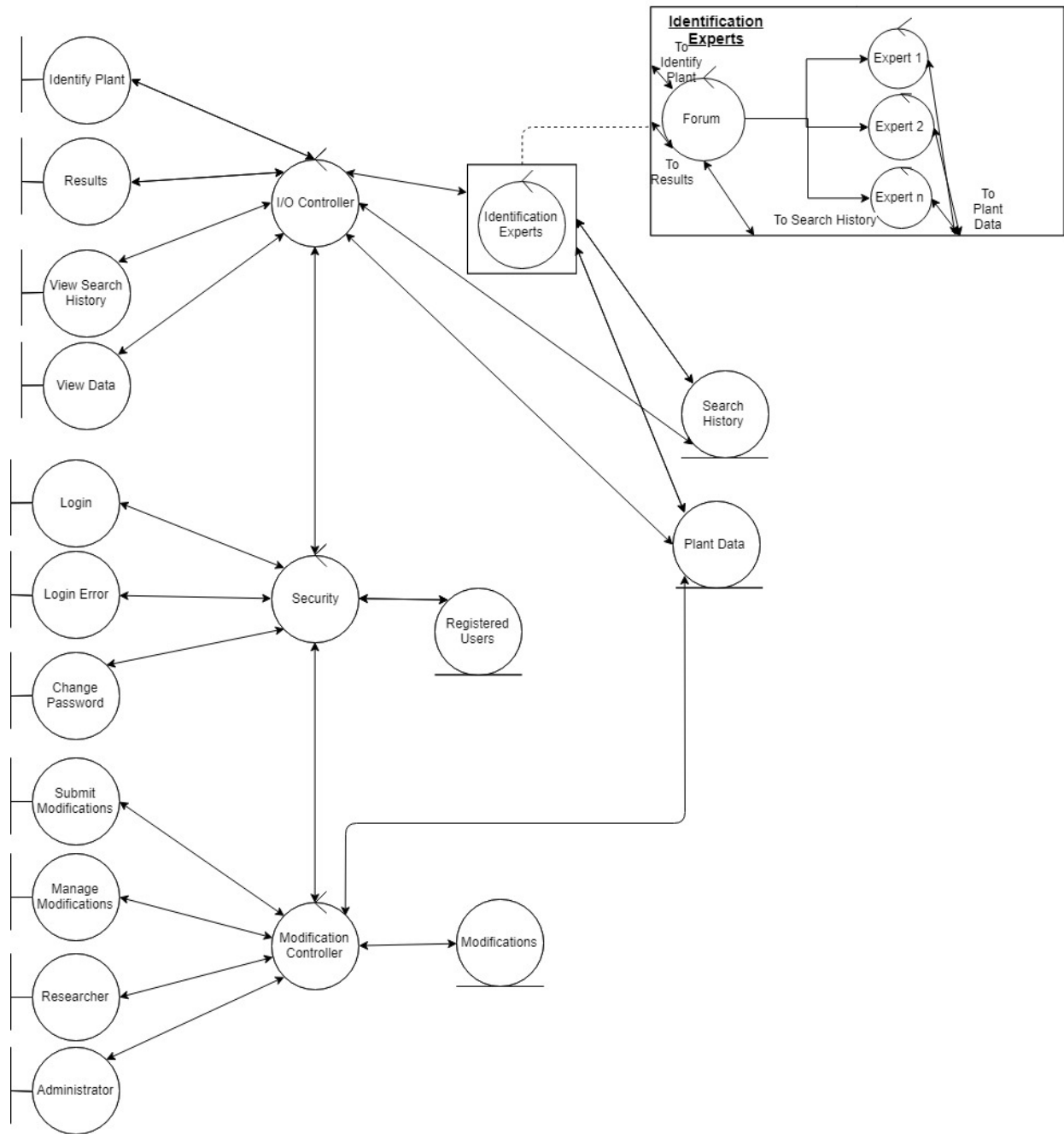


Figure 2: Analysis Class Diagram

4 Architectural Design

4.1 System Architecture

The overall architecture of the Forester system is Blackboard Architecture. To begin, the main functionality of the system is to identify plants based on specified characteristics. These characteristics are entered by the user as observations and the blackboard system is used to compare the observations to the data facts which exist in the expert specialists. In addition, Forester is mainly a blackboard architecture since each expert acts as a knowledge source and works independently from one another. Each expert provides a partial solution to what plant has been described and a final solution is determined by combining the results of each knowledge source. Moreover, the plant data class would act as the data store in the architecture which further justifies the choice of blackboard architecture for the system. The Forester system also includes other subsystems that may be excluded from the blackboard architecture; however, they are very minor in comparison to the core functionality of the system which is to use experts to identify plants based on inputted specifications. Below is a Structural Architecture Diagram of the system which assists in understanding why the overall architecture of the system is a blackboard architecture. The blue box of the analysis class diagram is identified as the Identification subsystem which features all the classes that are attributed to the blackboard architecture. The remaining subsystems are much more minor in comparison to the Identification subsystem and do not have a large impact on the overall architecture.

Figure 3 shows the different subsystems within Forester.

4.2 Subsystems

The subsystems of Forester are displayed in Figure 1 - Structural Architecture Diagram. As shown, the Forester system is divided into 3 major subsystems with the largest being the Identification subsystem (blue box). The identification subsystem deals with the main functionality of the application which is to allow users to identify plants based on inputted characteristics and comparing them to the experts in order to produce a result, as well as viewing search history and data. This subsystem is then divided into 2 smaller subsystems which include the Identification Agents subsystem (light blue classes) and the Data Retention and Searching subsystem (yellow classes). The Identification Agents subsystem features the classes that take user input, analyze it using the experts, and return a result. Whereas, the Data Retention and Searching subsystem focuses on allowing the user to view their past searches and the data on file. In addition, another subsystem of Forester is the Authentication subsystem (pink box) which focuses on the system security and user login operations. The last subsystem is the High Level Modification subsystem (green box) which controls the abilities of different types of users including the researchers and the administration. The researchers and admin need to be approved by the security controller in the Authentication subsystem in order to move onto the High Level Modification subsystem and to make alterations to the plant data. Moreover, the Authentication subsystem works alongside the Identification subsystem and the High Level Modification subsystem by allowing users to login and gain access to the different functions of the application.

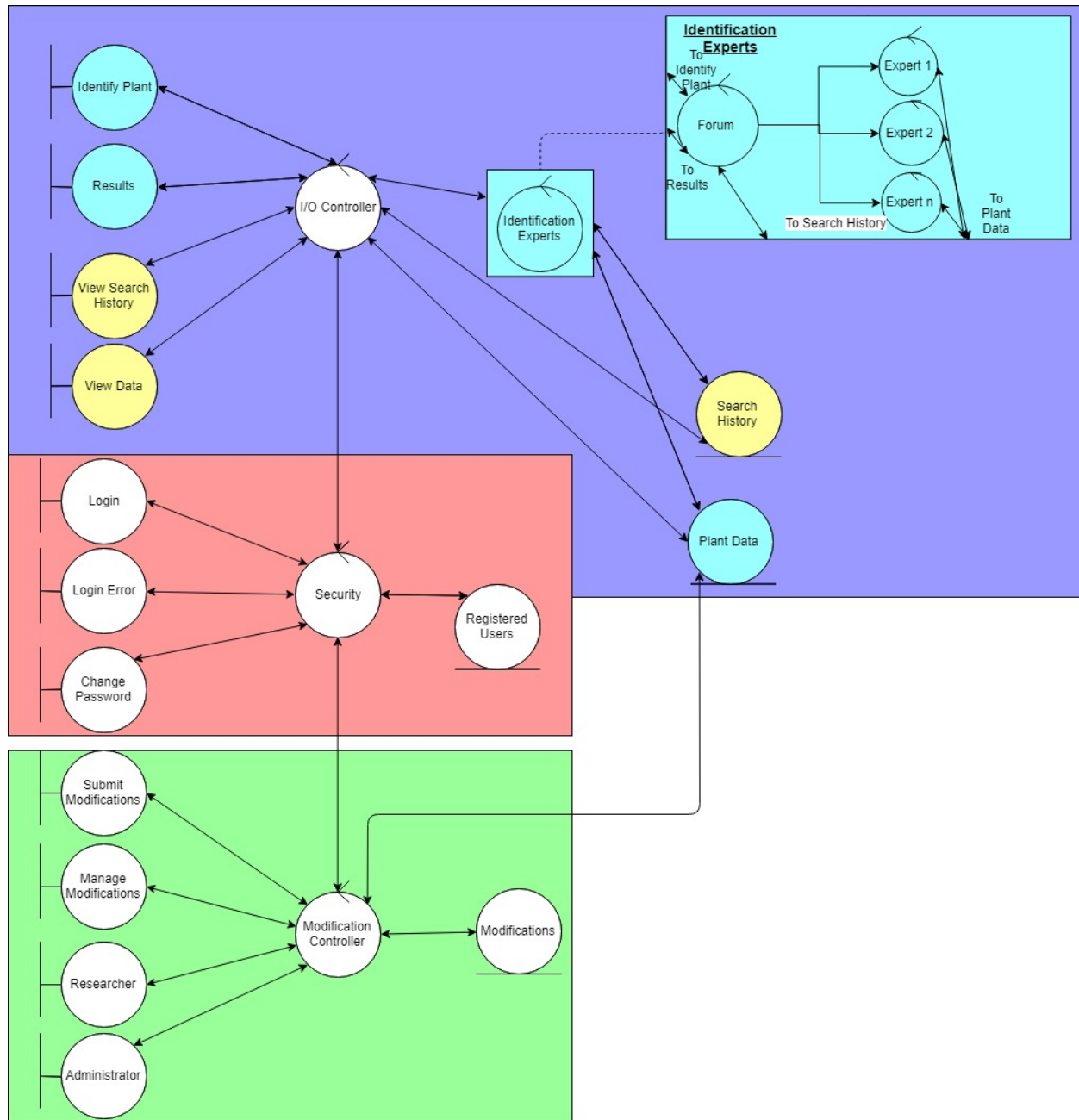


Figure 3: Structural Architecture Diagram - Subsystems

5 Class Responsibility Collaboration (CRC) Cards

5.1 Boundary Classes

Class Name: Identify Plant	
Responsibility:	Collaborators:
Receive user input	
Send input data to experts	I/O Controller
Send location data to experts	I/O Controller
Begin login process	I/O Controller

Class Name: Results	
Responsibility:	Collaborators:
Receive result data from experts	I/O Controller
Display result	
Return to Identify Plant page	I/O Controller

Class Name: View Search History	
Responsibility:	Collaborators:
Request search data	I/O Controller
Display search history	
Sort and filter displayed data	I/O Controller
Return to Identify Plant page	I/O Controller

Class Name: View Data	
Request plant data	I/O Controller
Display plant data	
Sort and filter displayed data	I/O Controller
Return to Researcher page	I/O Controller

Class Name: Login	
Responsibility:	Collaborators:
Receive user input	
Send input data to Security controller for validation	Security
Return to Identify Plant	Security

Class Name: Login Error	
Responsibility:	Collaborators:
Display error message	
Return to Login	Security
Return to Identify Plant	Security

Class Name: Change Password	
Responsibility:	Collaborators:
Receive user input (password)	
Send input data to Security controller	Security
Return to previous page (Researcher or Administrator)	Security

Class Name: Submit Modifications	
Responsibility:	Collaborators:
Receive user input	
Send user input to Modification Controller	Modification Controller
Return to Researcher page	Modification Controller

Class Name: Manage Modifications	
Responsibility:	Collaborators:
Request Modification data	Modification Controller
Display Modification date	
Write modified data to main Plant Data	Modification Controller
Return to Administrator page	Modification Controller

Class Name: Researcher	
Responsibility:	Collaborators:
View data	Modification controller
Request modifications	Modification controller
Change password	Modification controller
Log out and return to Login page	Modification controller

Class Name: Administrator	
Responsibility:	Collaborators:
Manage modifications	Modification controller
Change password	Modification controller
Log out and return to Login page	Modification controller

5.2 Controller Classes

Class Name: I/O Controller	
Responsibility:	Collaborators:
Accept plant data from Identify Plant	Identify Plant
Send plant data to Identification Experts	Forum
Accept plant result data from Identification Experts	Forum
Send plant results from the Identification Experts to Results and open the Results page	Results
Receive sort and filter data	View Search History, View Data
Read search history data	Search History
Read plant data	Plant Data
Modify history or plant data based on specified sort and filters	
Receive request for Identify Plant page and open it	Security, Results, View Search History, Identify Plant
Receive request for Search History page and open it	Security, View Search History
Receive request for View Data page and open it	Security, View Data
Request for Login page to be opened	Security
Request for Researcher page to be opened	Security

Class Name: Security	
Responsibility:	Collaborators:
Receive username and password	Login
Access registered data	Registered Users
Search for username	Registered Users
Validate password	Registered Users
Open Login Error page if username/password incorrect	Login Error
Accept new password for current user	Change Password
Write new password to Registered Users for current user	Registered Users
Receive request for Login page and open it	I/O Controller, Modification Controller, Login Error, Login
Receive request for Change Password page and open it	Modification Controller, Change Password
Relay request for View Data page to be opened	Modification Controller, I/O Controller
Relay request for Researcher page to be opened	I/O Controller, Modification Controller
Request for Researcher page to be opened when a researcher logs in	Modification Controller
Request for Administrator page to be opened when an administrator logs in	Modification Controller

Class Name: Modification Controller	
Responsibility:	Collaborators:
Receive modification data	Submit Modifications
Write modification data	Modifications
Receive request for data of current modifications	Manage Modifications
Read modification data	Modifications
Send modification data to Manage Modifications	Manage Modifications
Receive request to merge current modifications into Plant Data	Manage Modifications
Write modification data to Plant Data	Plant Data
Receive request for Researcher page and open it	Security, Researcher
Receive request for Administrator page and open it	Security, Administrator
Receive request for Submit Modifications page and open it	Researcher, Submit Modifications
Receive request for Manage Modifications page and open it	Administrator, Manage Modifications
Request for Login page to be opened	Researcher, Administrator, Security
Request for View Data page to be opened	Researcher, Security

Class Name: Forum	
Responsibility:	Collaborators:
Receive plant data	I/O Controller
Send plant data to relevant experts	Expert
Send plant data search to Search History	Search history
Receive results from each expert	Expert
Determine the best result based on the experts' results	
Send the best result to Results	I/O Controller
If only location data given from user, send all results from the location-only expert to Results	I/O Controller
Send error result	I/O Controller

Class Name: Expert	
Responsibility:	Collaborators:
Receive plant data	Forum
Access plant data	Plant Data
Search through plant data	Plant Data
Return plants matching given plant data	Forum

5.3 Entity Classes

Class Name: Plant Data	
Responsibility:	Collaborators:
Hold plant data	
Provide read access to plant data	Expert, I/O Controller
Provide write access to plant data	Modification Controller

Class Name: Search History	
Responsibility:	Collaborators:
Hold search history data	
Provide read access to search data	I/O Controller
Provide write access to search data	Forum

Class Name: Registered Users	
Responsibility:	Collaborators:
Hold username, password, and classification of users	
Provide read access to user data	Security
Provide write access to user data	Security

Class Name: Modifications	
Responsibility:	Collaborators:
Hold data of modifications to be made to plant data	
Provide read access to modification data	Modification Controller
Provide write access modification data	Modification controller

A Division of Labour

Purpose - Allen
System Description - Hao
Overview - Allen
Use Case Diagram - Susan
Analysis Class Diagram - Evan, Hao, Abrar
System Architecture - Abrar
Subsystems - Abrar
CRC cards - Everyone (Evan)
Division of Labour - Abrar