|  |  |  |
| --- | --- | --- |
| **المملكة العربية السعودية**  **وزارة التعليم العالي**  **جامعة الإمام محمد بن سعود الإسلامية**  **كلية علوم الحاسب والمعلومات** | A description...  **Second term 1441/2020** | **KINGDOM OF SAUDI ARABIA**  **Ministry of Higher Education**  **Al-Imam Mohammad University**  **College of Computer & Information Sciences** |
| **Software Engineering (CS- 310)**  **BSCS- Section: 171**  **Project-Phase No: 02**  **Automatic Jobs Candidates Selection System (AJCSS)**  **(Design)**  **Submitted By** | | |
| **Sultan Attaf Al-Salmi (439013826) – Coordinator**  **Abdulaziz Derham Asseri (439011531)**  **Musaad Mubarak Alhammami (439016695)**  **Yasser Ahmed Almuhaidib (439013620)**  **Auodh Mohmmed AL-Qahtani (435032042)**  **Supervisor**  **Professor Sultan S. Alqahtani**  **Date: 21/3/2020** | | |
|  | | |

Table of Contents

[Abstract: iii](#_Toc35955517)

[1. Introduction 4](#_Toc35955518)

[2. High Level and Medium Level Design 4](#_Toc35955519)

[2.1 System Level Diagram 4](#_Toc35955520)

[2.2 Class Diagram 5](#_Toc35955521)

[2.3 Class Method Descriptions 6](#_Toc35955522)

[3. Detailed Design 10](#_Toc35955523)

[4. User Interface Design 12](#_Toc35955524)

[5. Conclusion 16](#_Toc35955525)

# Abstract:

This design document shows the architectural design of the AJCSS. It contains graphical representation about classes and functions, and the representation of requirements is shown in a way that reflects the purpose of the system. Also, it gives an example of how the software will be like.

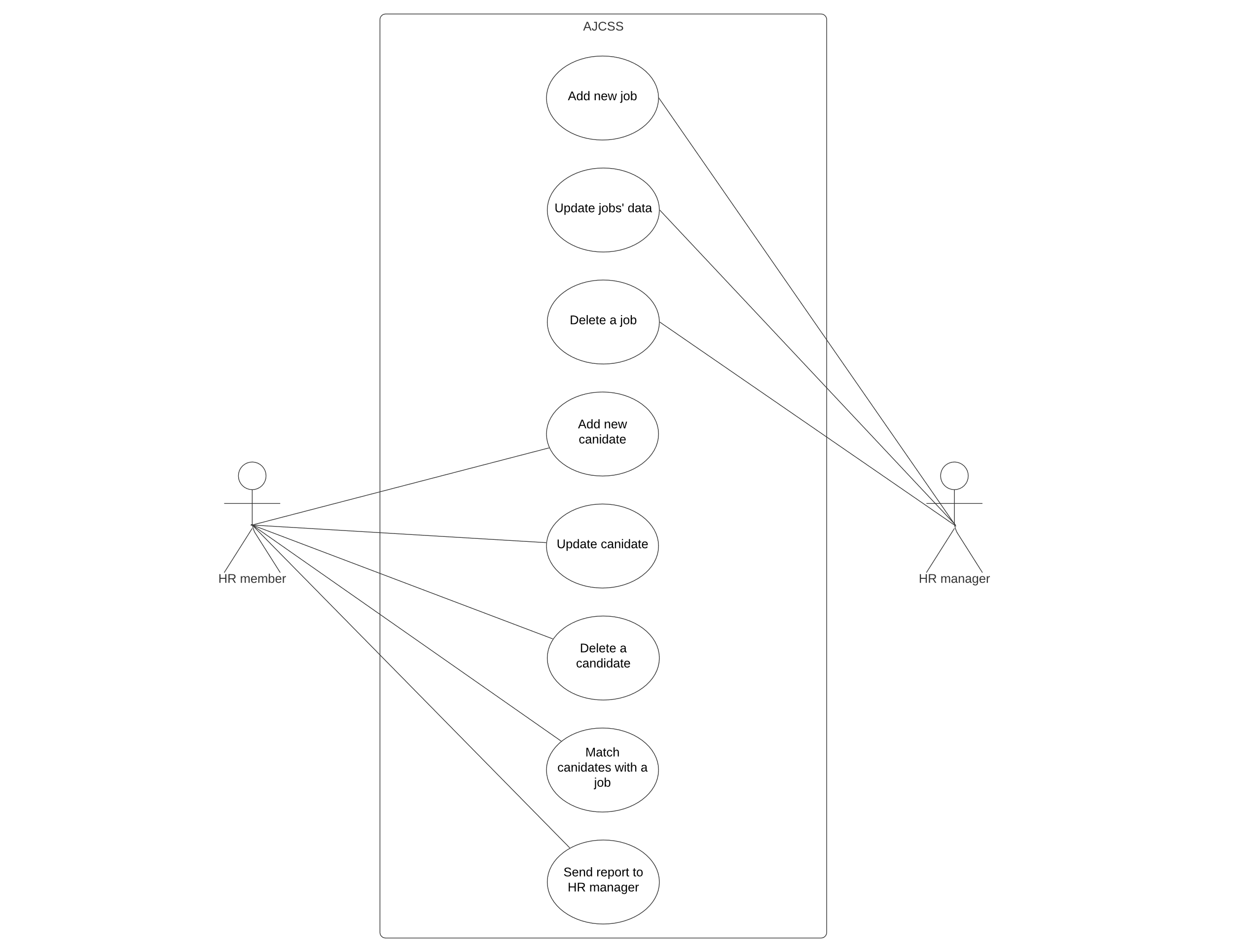
# 1. Introduction

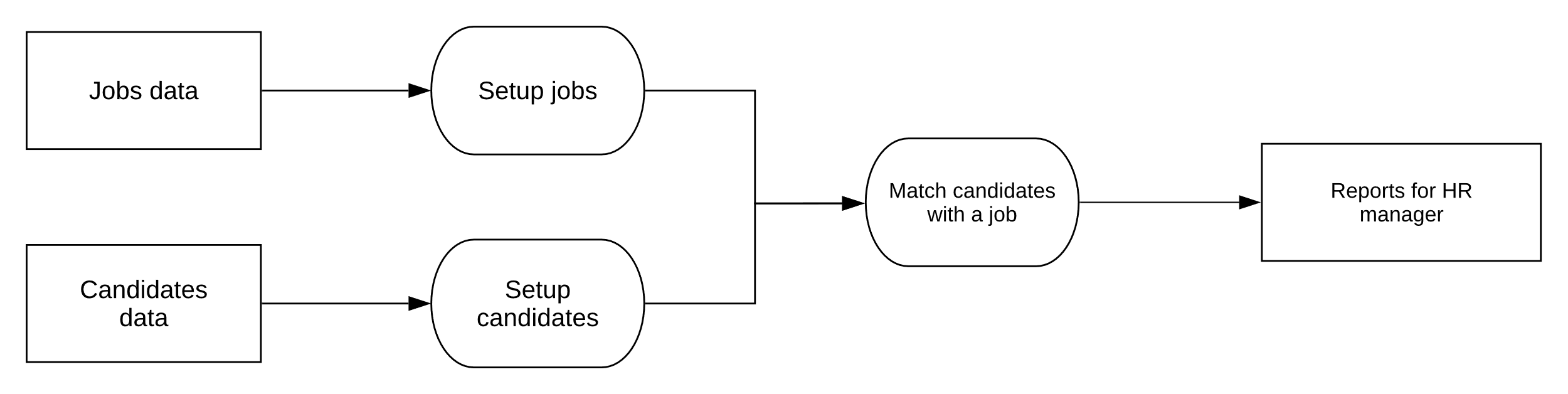
# 

# 2. High Level and Medium Level Design

## 2.1 System Level Diagram

The system level diagrams are: system’s interaction model (Use-case diagram) and system’s architectural pattern (pipe and filter pattern).



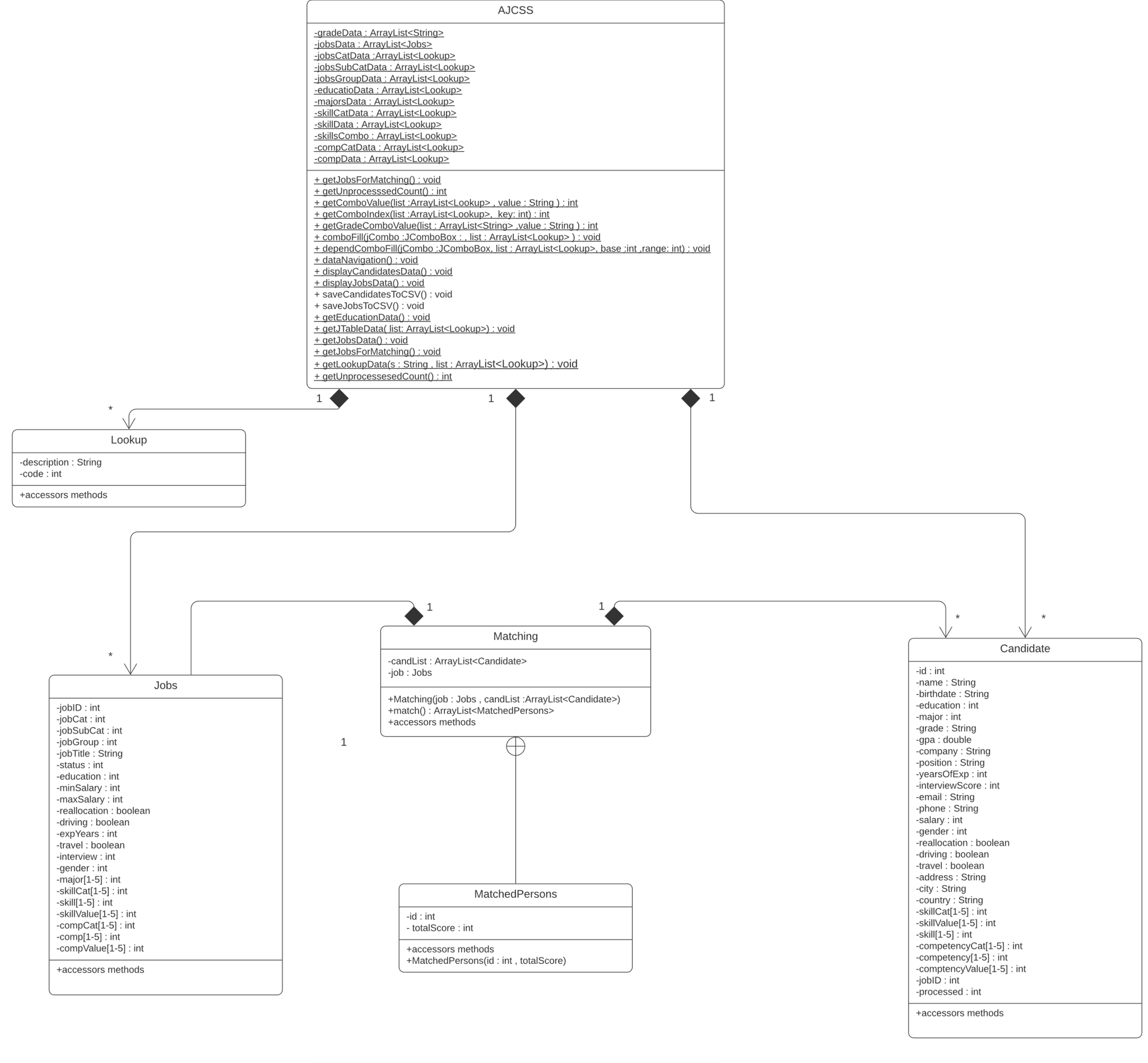


**Fig. 1** Use case diagram that depicts the use cases of HR manger and member actors.

**Fig. 2** pipe and filter pattern that depicts the data flow to the process of finding a match.

## 2.2 Class Diagram

The class diagram shows the structure and relationships of AJCSS software.



**Fig. 3** AJCSS class diagram

## 2.3 Class Method Descriptions

**AJCSS**

The AJCSS class is main class which has the graphical user interface (GUI) components, the methods that initializes all the data and fills the GUI components when the program starts. Also, it has methods which save the progress into CSV files.

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | getUnprocesssedCount |
| **Visibility** | Public |
| **Return type** | Int |
| **Parameter, types** | None |
| **Description** | This method counts how many unprocessed candidate are there. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | getComboValue |
| **Visibility** | Public |
| **Return type** | Int |
| **Parameter, types** | list : ArrayList<Lookup> ,  value : String |
| **Description** | This method gets the value of the selected item in a combo box. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | getComboIndex |
| **Visibility** | Public |
| **Return type** | Int |
| **Parameter, types** | list : ArrayList<Lookup> ,  key : int |
| **Description** | This Method gets the index of the selected item in a combo box. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | comboFill |
| **Visibility** | Public |
| **Return type** | void |
| **Parameter, types** | jCombo : JComboBox ,  list : ArrayList<Lookup> |
| **Description** | This method fills a combo box from an arraylist. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | dataInitialization |
| **Visibility** | Public |
| **Return type** | void |
| **Parameter, types** | none |
| **Description** | This method initializes all the data at the start of the program from CSV files. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | getJobsData |
| **Visibility** | Public |
| **Return type** | void |
| **Parameter, types** | none |
| **Description** | This method fetches the jobs data from CSV file. Then stores the fetched data into an arraylist in program. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | getLookupData |
| **Visibility** | Public |
| **Return type** | void |
| **Parameter, types** | s : String ,  list : ArrayList<Lookup> |
| **Description** | This method fetches the lookup data from CSV file. Then stores the fetched data into an arraylist in program. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | SaveCandidatesToCSV |
| **Visibility** | Public |
| **Return type** | void |
| **Parameter, types** | none |
| **Description** | This method saves the  candidate’s data into a CSV file. |

|  |  |
| --- | --- |
| **Class** | AJCSS |
| **Method** | SaveJobsToCSV |
| **Visibility** | Public |
| **Return type** | void |
| **Parameter, types** | none |
| **Description** | This method saves the job’s data into a CSV file. |

**Matching**

This Matching class is designed to process the matching of the selection criteria by using a method which finds a match for unprocessed candidates with a vacant job.

|  |  |
| --- | --- |
| **Class** | Matching |
| **Method** | match |
| **Visibility** | Public |
| **Return type** | ArrayList<MatchedPersons> |
| **Parameter, types** | none |
| **Description** | This method looks for a match between selected unprocessed candidates and a vacant job. |

**Jobs**

The Jobs class is designed to store jobs’ data that contains essential data for processing. As well update a job’s data. This class contains accessors methods.

**Lookup**

The Lookup class is designed to reference jobs’ data to coded values. A job’s data is written with a description; thus, it will reference the job’s data description to a coded integer value to ease the operation and process. This class contains accessors methods.

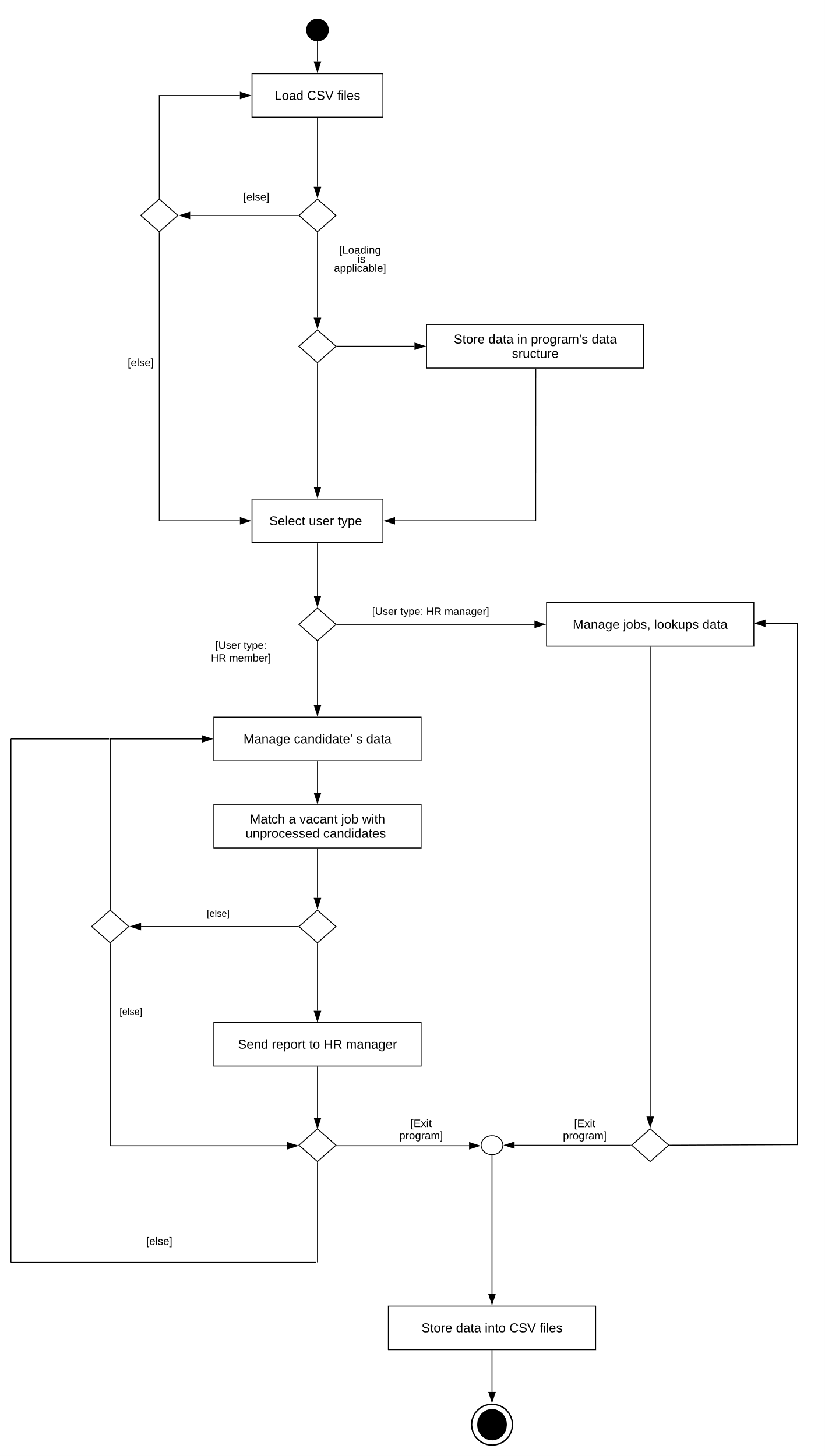
**Candidate**

The Candidate class is designed to store candidates’ data that contains essential data for processing. As well update a candidate’s data. This class contains accessors methods.

# 3. Detailed Design

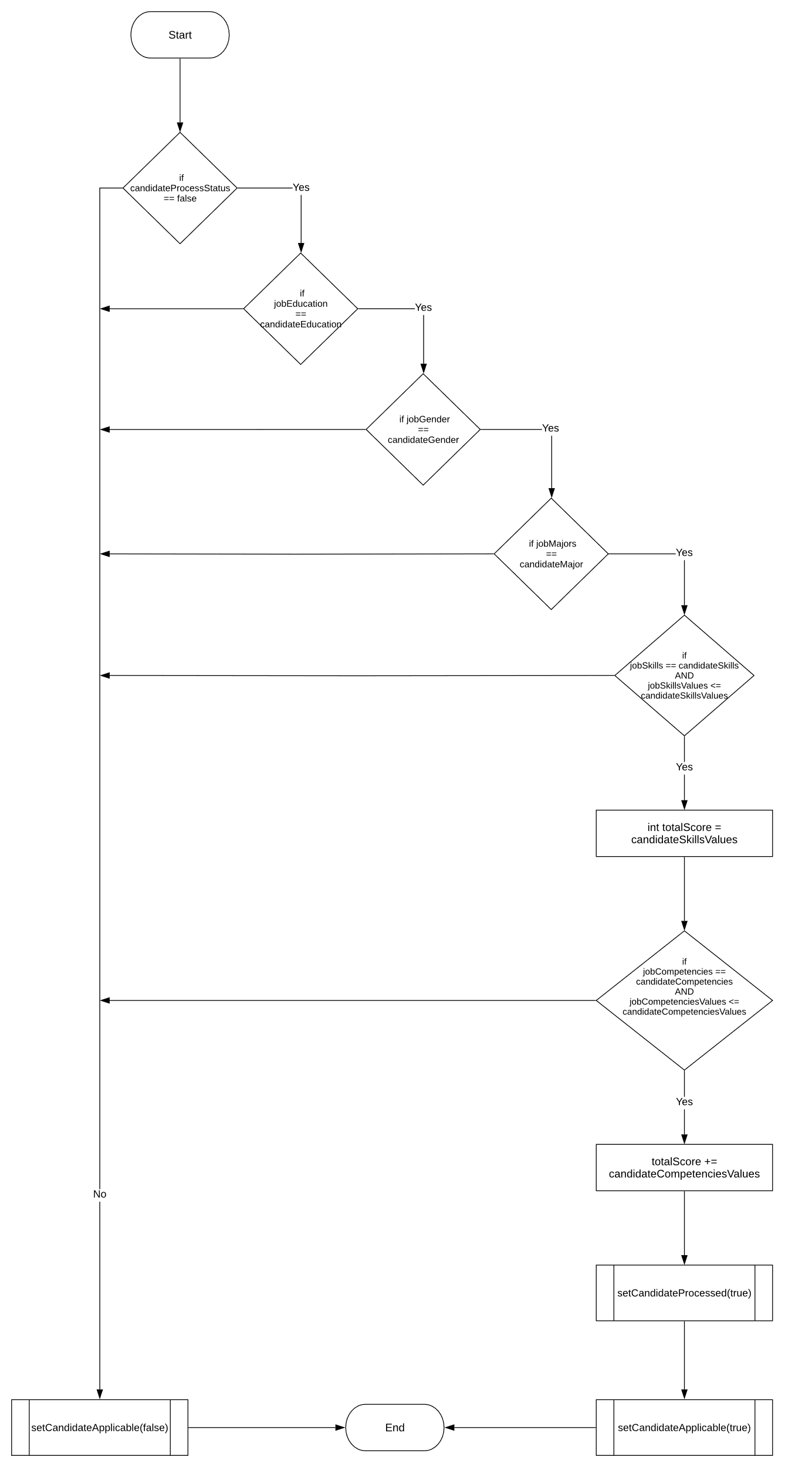
This section contains diagrams depicting low-level details of pieces of the AJCSS software.

Figure 2 depicts an activity chart at the start of the program. Activity includes loading the files into the program, storing data in program data structures, selecting user type, sending reports and storing data into files.



**Fig. 4** AJCSS activity chart

Figure 3 is a flowchart showing the method Mathching:match() which finds a match for unprocessed candidates with a vacant job.



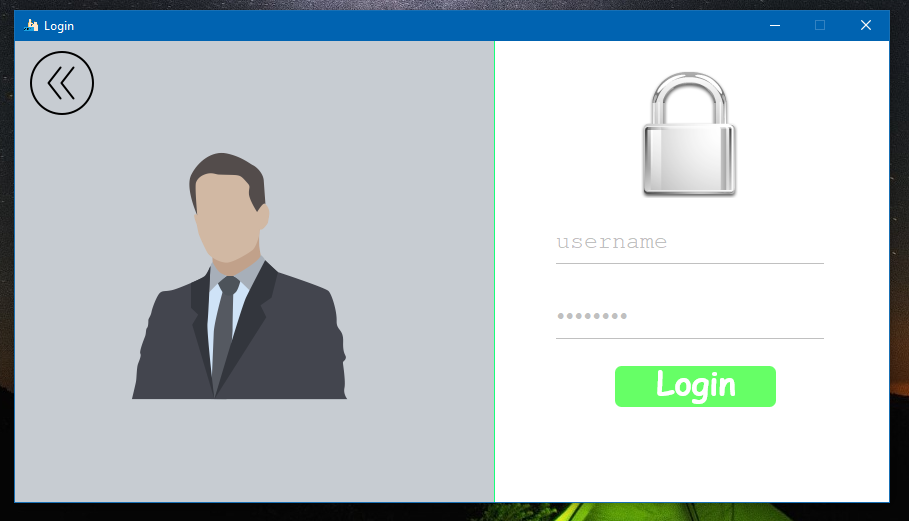
**Fig. 5** The match() method from the Matching class.

# 4. User Interface Design

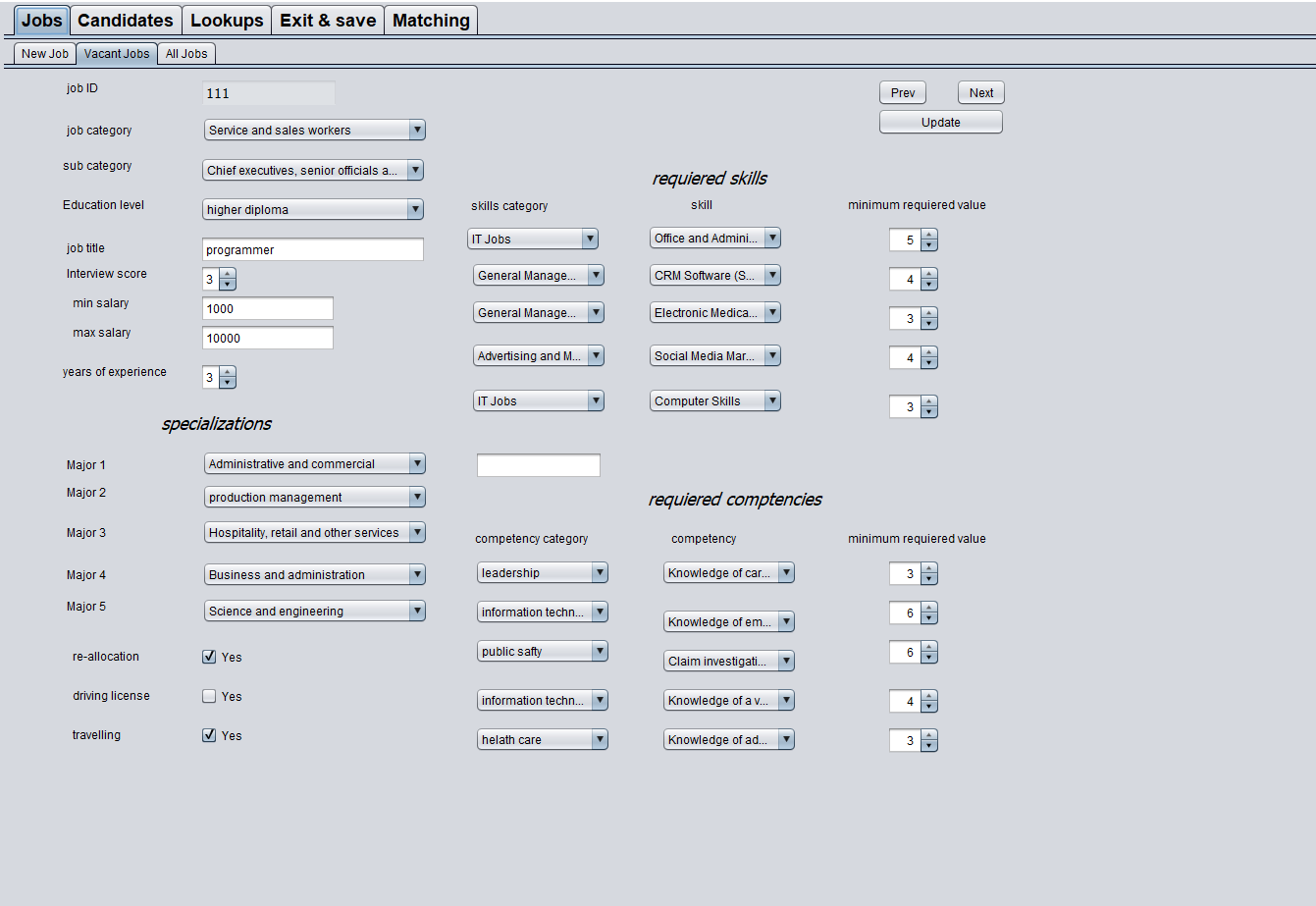
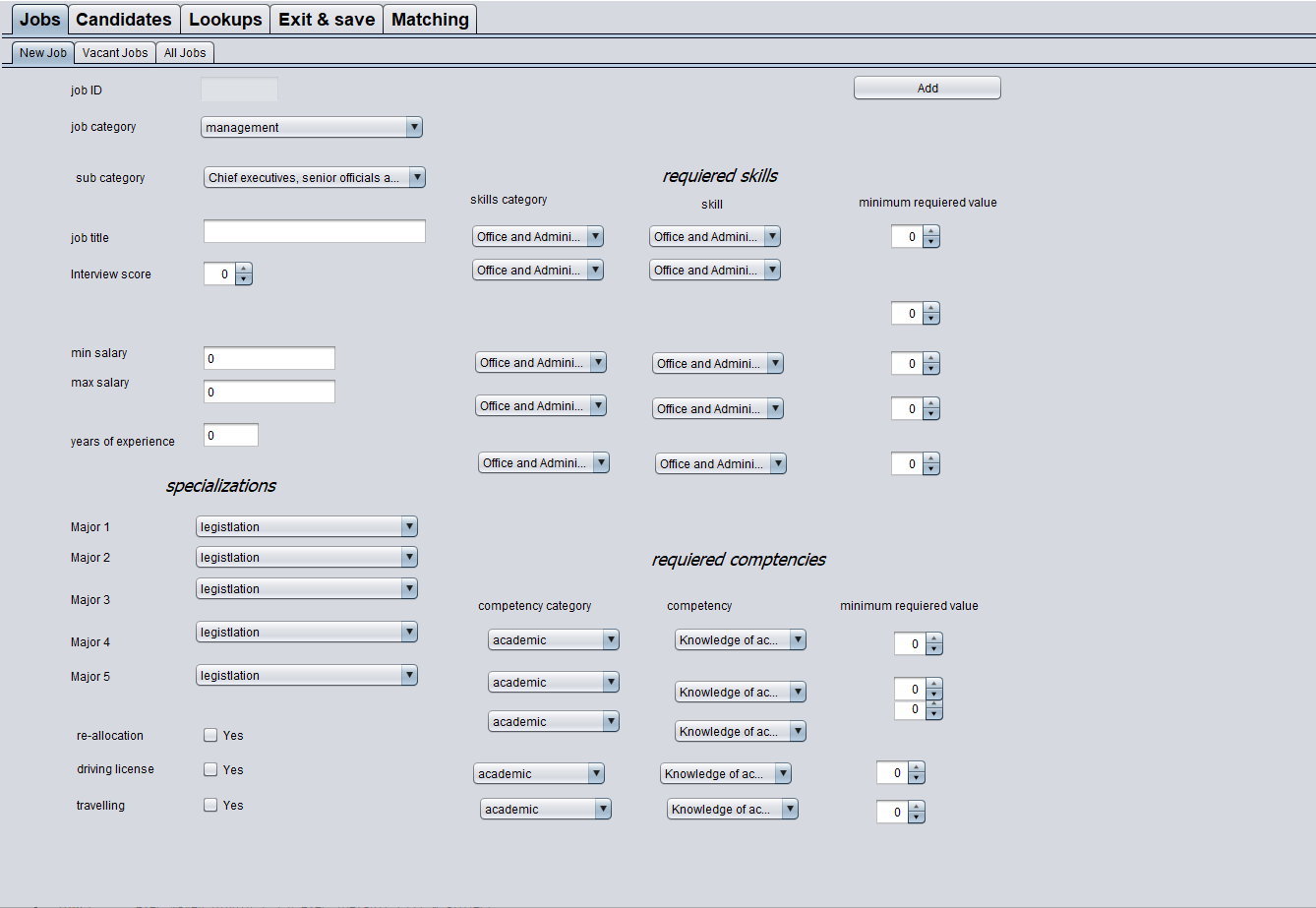
This section contains screen shots of initial versions for the AJCSS user interface.



**Fig. 6** This window lets you select the user type.

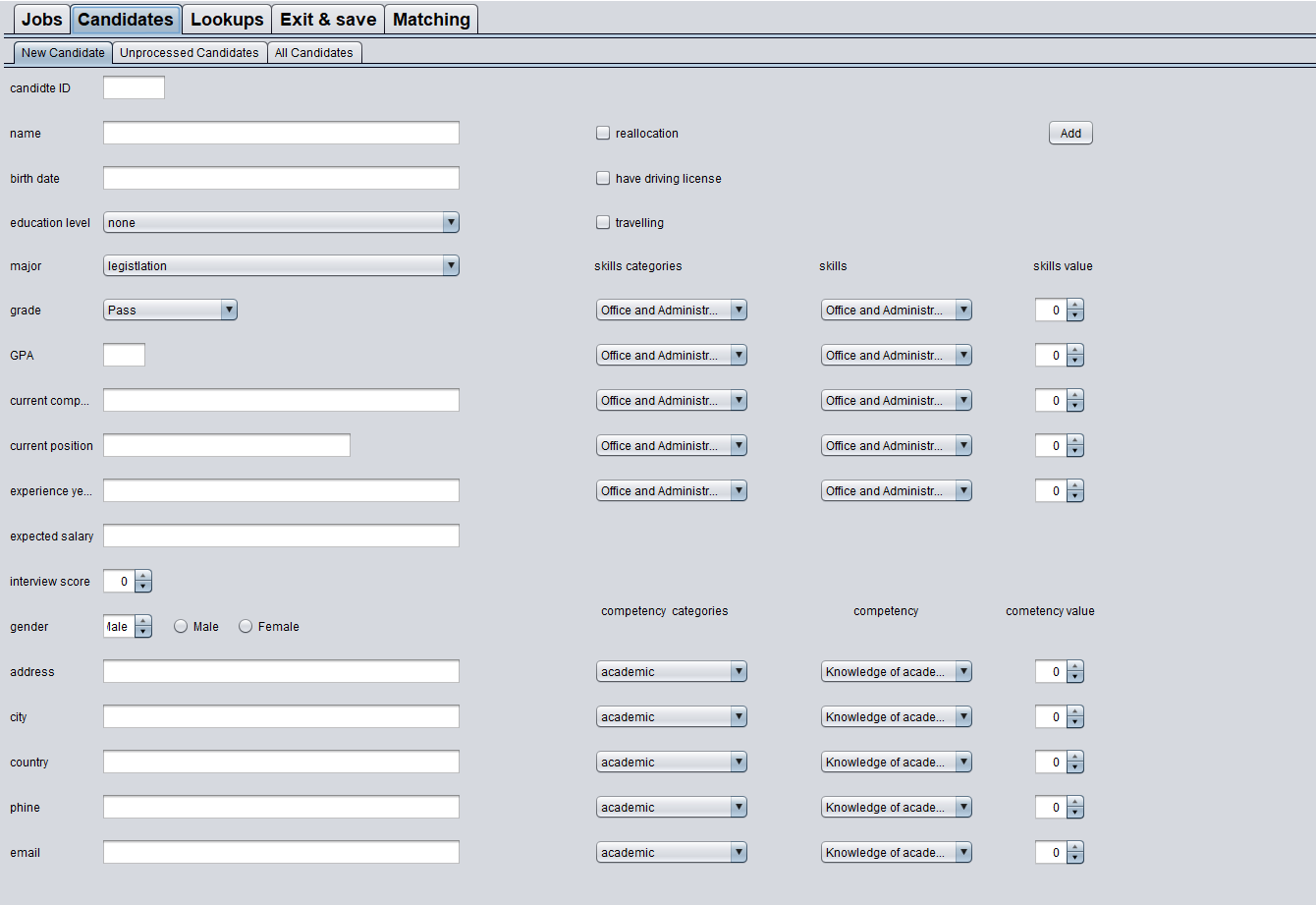


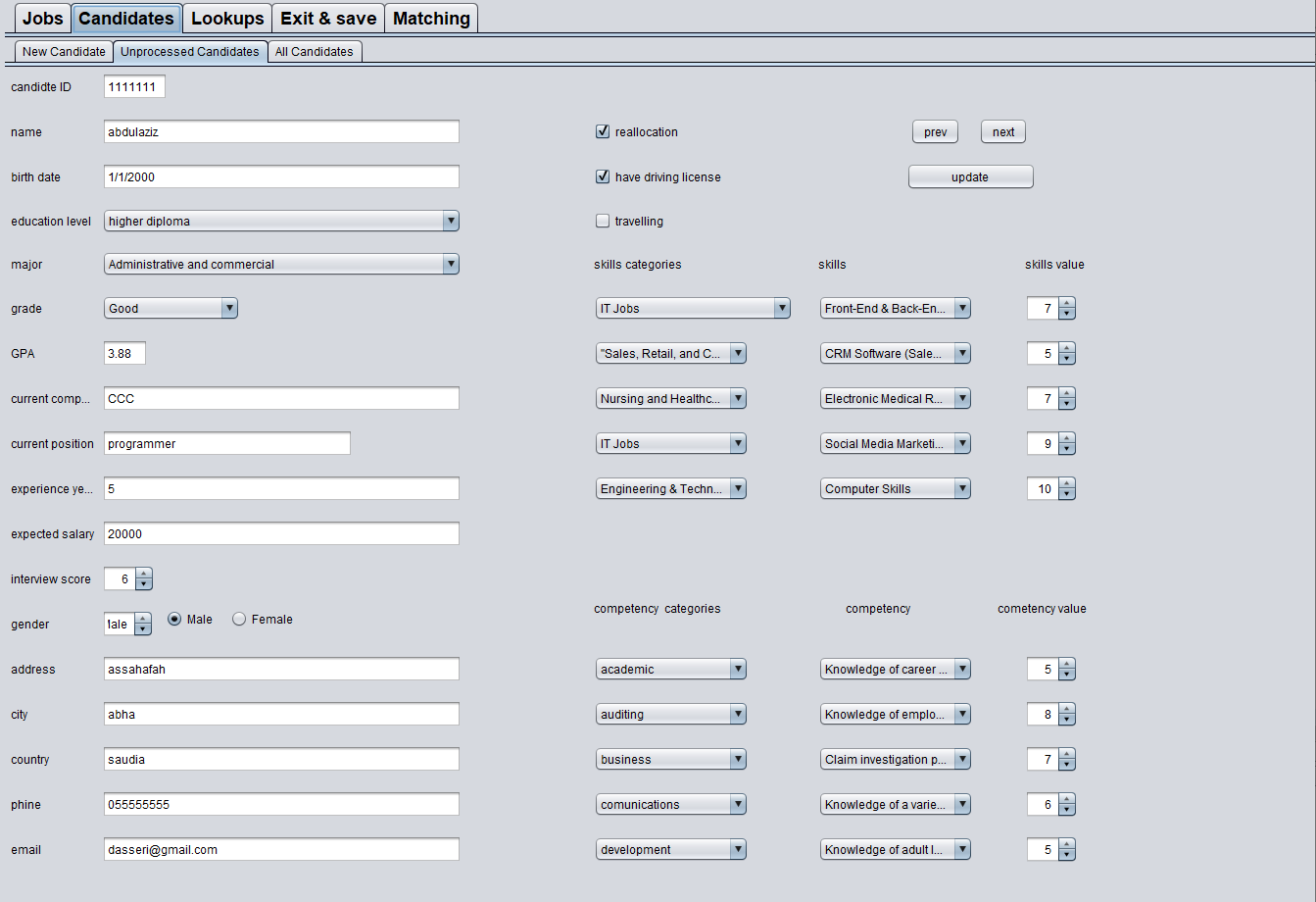
**Fig. 7** This window is for HR manager user type which asks for admin credentials.



**Fig. 9** A screen shot of the jobs-vacant jobs section which shows you all vacant jobs.

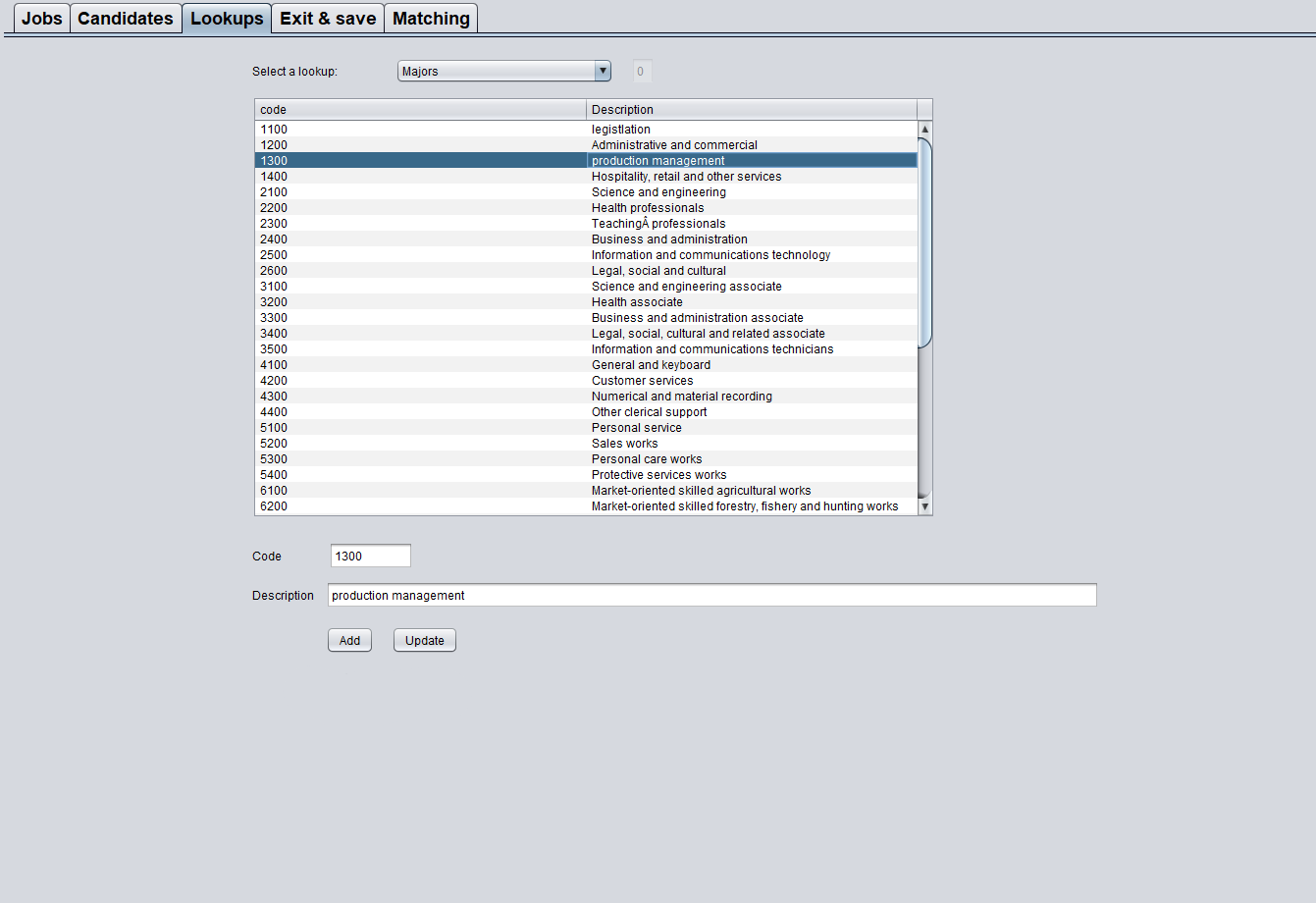
**Fig. 8** A screen shot of the jobs-new job section which let you setup a new job.

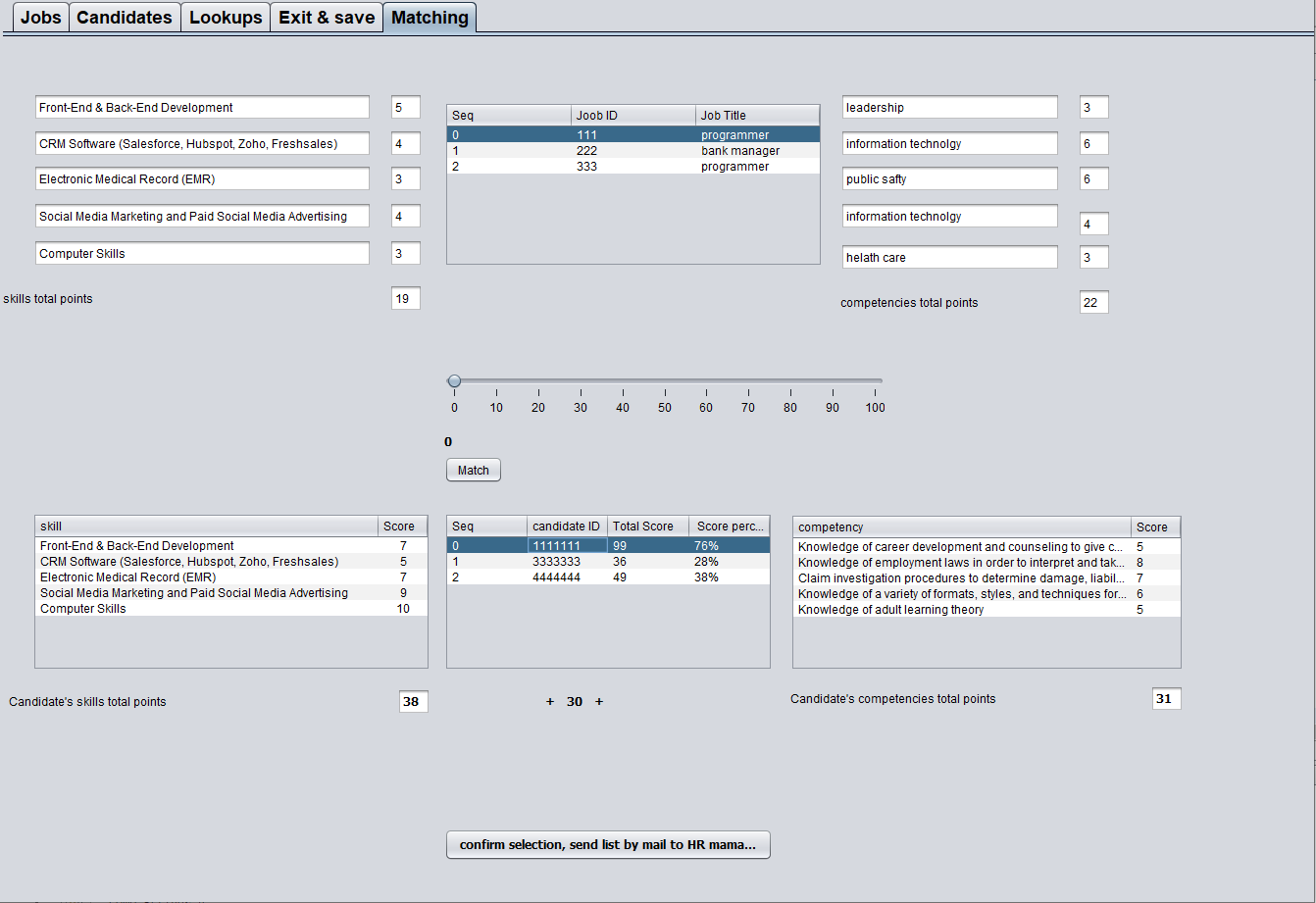




**Fig. 11** A screen shot of the candidates- unprocessed candidate section which shows you all unprocessed candidates.

**Fig. 10** A screen shot of the candidates- new candidate section which lets you setup the a new candidate.





**Fig. 12** A screen shot of the lookups section which shows you all lookups that were loaded from CSV file.

**Fig. 13** A screen shot of the matching section which lets you select unprocessed candidates to match them with a vacant job.

# 5. Conclusion