TA Management System

1 Introduction

This project involves the development of a system to manage the records of teaching assistants (TAs) within a department. The system ensures that only eligible TAs—those who are currently registered students—are maintained in the records. The objective is to create a streamlined and efficient process for managing TA information.

2 File Filtering System

The TA records are stored in a file named `TAs.txt`. Each record in the file includes the following information:

Student_Id   First_Name   Status   Working_Hours   Year_Hired

The system is designed to ensure that the records only contain current students, removing any records of TAs who have already graduated (classified as ”Alum”).

2.1 System Workflow

The system processes the TA file by performing the following steps:

1. Read the initial file and count the total number of records.
2. Identify and remove any records where the status is ”Alum”.
3. Update the file with the corrected list of TAs.

2.2 Example Workflow

Initial TA File:

```
7
8281954 Giulio Cesare Grad 11 2018
2783632 Aymen Alum 11 2018
5352622 Daksh UGrad 8 2019
7603086 Mia UGrad 5 2020
9345631 Nancy Alum 13 2014
1703358 John Grad 12 2015
2342686 Sofiane Grad 12 2015
```

Corrected TA File:

```
5
8281954 Giulio Cesare Grad 11 2018
5352622 Daksh UGrad 8 2019
7603086 Mia UGrad 5 2020
1703358 John Grad 12 2015
2342686 Sofiane Grad 12 2015
```

3 TA Class Implementation

To manage the TA records, a C++ class named `TA` is implemented. This class encapsulates the TA data and provides methods to check if a TA is an alumnus and to print the TA's information.

3.1 Class Definition

The `TA` class is defined as follows:

Listing 1: TA Class Implementation

```
#pragma once
#include <iostream>
#include <string>

class TA
{
private:
        int Student_Id;
        std::string First_Name;
        std::string Status;
        int Working_Hours;
        int Year_Hired;
public:
        // Default constructor
        TA()
        {
                Student_Id = -1;
                First_Name = "-";
                Status = "-";
                Working_Hours = -1;
                Year_Hired = -1;
        }

        // Parameterized constructor (List Initialization)
        TA(int si, std::string fn, std::string s, int wh, int yh)
        : Student_Id{ si }, First_Name{ fn }, Status{ s },
        Working_Hours{ wh }, Year_Hired{ yh }
        {}

        // Return true if TA is an alumni
        bool Is_Alum()
        {
                return (Status == ”Alum”) ? true : false;
        }

        // Return a string containing the following:
        // ”Student_Id First_Name Status Working_Hours Year_Hired”
        std::string print()
        {
                return (Is_Alum()) ? (””) :
                (std::to_string(Student_Id) + ”␣” + First_Name + ”␣” +
                Status + ”␣” + std::to_string(Working_Hours) + ”␣” +
                std::to_string(Year_Hired));
        }

};
```

3.2 Class Explanation

- **Private Members:** The class contains private members to store the TA's details: `Student_Id`, `First_Name`, `Status`, `Working_Hours`, and `Year_Hired`.
- **Constructors:**
  - `TA()`: A default constructor that initializes the TA attributes with placeholder values.
  - `TA(int si, std::string fn, std::string s, int wh, int yh)`: A parameterized constructor that initializes the TA attributes with the provided values.
- **Methods:**
  - `bool Is_Alum()`: This method returns `true` if the TA is classified as an alumnus, based on the `Status` attribute.
  - `std::string print()`: This method returns a string containing the TA's details in the format: `Student_Id First_Name Status Working_Hours Year_Hired`. If the TA is an alumnus, it returns an empty string.

4 Conclusion

The TA Management System successfully processes and updates TA records by filtering out ineligible individuals based on their status. The `TA` class plays a crucial role in managing and representing the TA data efficiently, ensuring that the department maintains an accurate and up-to-date list of eligible teaching assistants.