**COMP S350F**

**Testing Plan Document**

**Group 29**

**Members:**

**LEE Shu Kuen Joe 13050317**

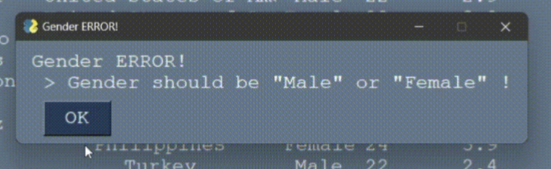
**TSANG Kwok Yiu Kwok 13080961**

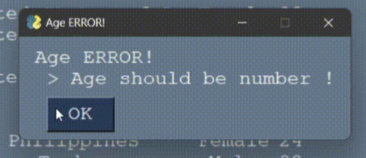
**LAI Pak Lun Patrick 13035707**

**LU Yuk Tong Tony 13439007**

**Validation Testing**

Both login and record systems have data validation done by restricting each text box to be of a specific data type. For example, a GPA score should only be in floating point numbers of a specified length, genders need to be entered as “Male” or “Female” specifically and ages can only be an integer value amongst other things. Formats, length, range of data, and presence of fields are also checked. Here is an example of checking errors in the field of “Gender” and “Age.”



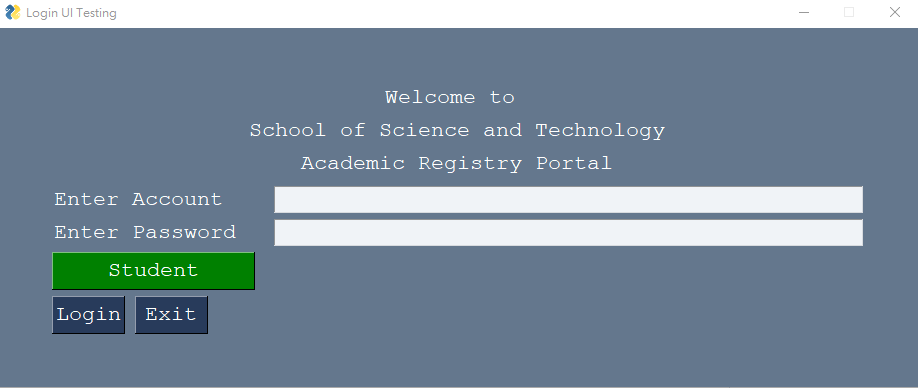


**Client Requirement**

The client requirements contain four main functions: Login Function, Sorting Function, Searching Function and Editing Function. Those functions are executed properly.

**1. Login Function**

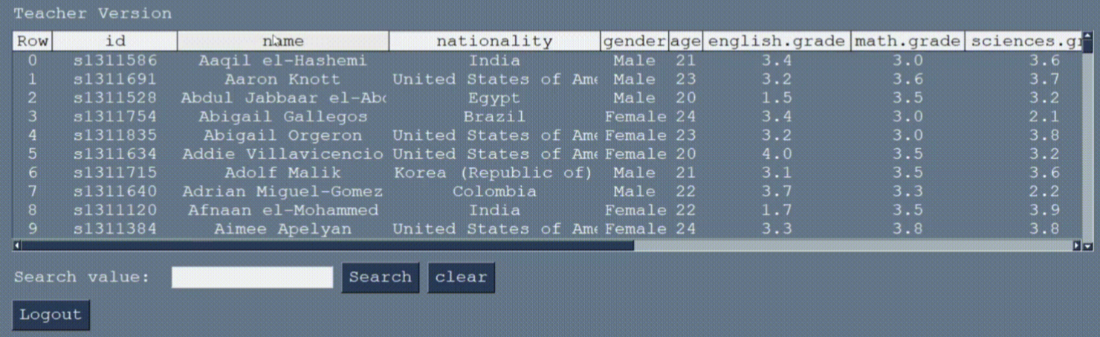
This function demonstrates the ability for users to log in and provide access privileges to the proper role. For example, a student cannot edit a record, but a teacher/IT can.



|  |  |  |
| --- | --- | --- |
|  | **Implementation** | **Expected output** |
| **Login system** | Empty input and click Login button | Fail, output incorrect Account/Password and request user to enter the correct credentials. |
| Using incorrect account and password to login | Fail, output incorrect Account/Password and request user to enter the correct credentials. |
| Using different accounts to login to other pages/roles  Student -> Teacher  Teacher -> Student | Fail, output incorrect authentication and request user to enter the correct credentials. |
| input correct account and password then login | Success, go to the Record page |
| Click the exit button | Turn off the system |

**2. Sorting Function**

All users are allowed to sort the records. The example is demonstrated as a name sorted in ascending order. This is done by clicking the category on the white bar on top and will sort ascending order first, then descending order if clicked again.



|  |  |  |
| --- | --- | --- |
|  | **Implementation** | **Expected output** |
| **Sorting Function** | Click "name” column | Sort ascending order with “name” |
| Click "name” column twice | Sort descending order with “name” |
| Click data row | Nothing happens |

**3. Searching Function**

All users are allowed to search for a specific data entry by typing in keywords or ids. For example, searching for the student sxxxxxxxx or a certain nationality.



|  |  |  |
| --- | --- | --- |
|  | **Implementation** | **Expected output** |
| **Searching Function** | Type “Lor” | Show the student which include “Lor” |
| Type character e.g. “&”,“$”,“^” | Invalid input, error window pops out |
| Empty Input | Nothing happens |
|  |  |
|  |  |

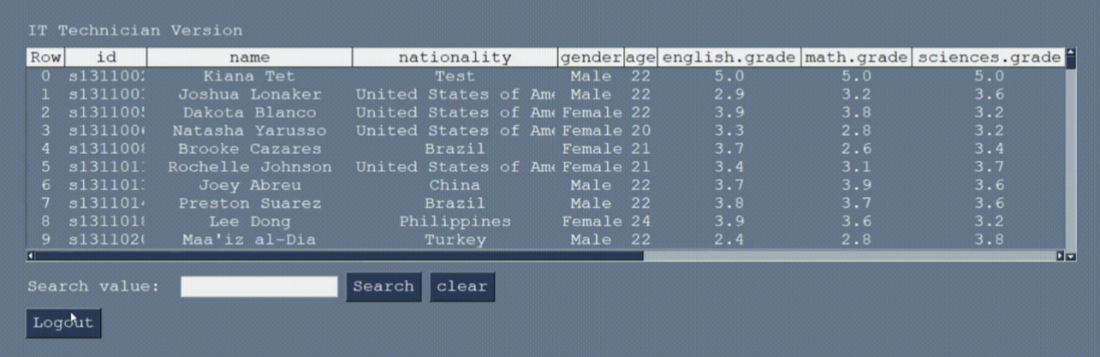
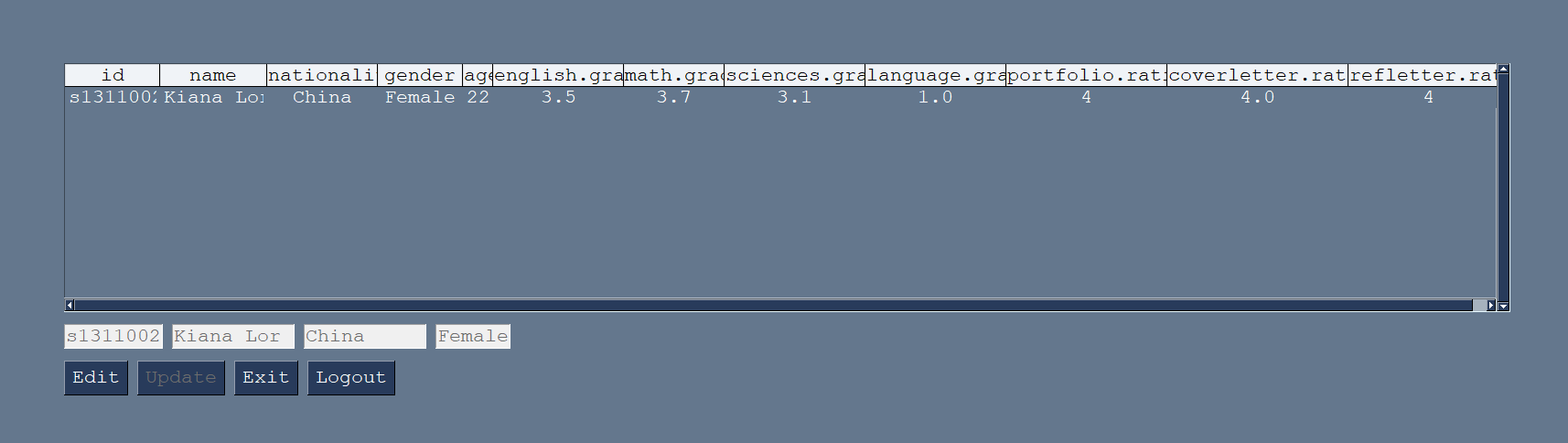
**4. Editing Function**

Specified users like Teachers and IT Technician can update a data entry by writing in the text box provided. The data in the field then overwrites the original data and the data entry is updated and added in the database. The text boxes' order corresponds to the data's layout displayed along the white bar up top. An example of “Kiana Tet” is shown for updating the nationality to “Test”.

一張含有 文字, 軟體, 螢幕擷取畫面, 多媒體軟體 的圖片

自動產生的描述

(Original Record)

(Modified Record)

|  |  |  |
| --- | --- | --- |
|  | **Implementation** | **Expected output** |
| **Editing Function** | Click the data in the table | Show that row data |
| **Students** edit their personal information | Success, Update the change |
| **Students** input symbol in their personal information | Fail, personal information can only be character or integer |
| **Students** edit their academic records | Fail, Student cannot edit their academic records |
| **Teacher** edits student’s personal information | Fail, Teacher cannot edit student’s personal information |
| **Teacher** edits academic records | Success, Update the change |
| **IT Technicians** student’s personal information | Success, Update the change |
| **IT Technicians** edits academic records | Success, Update the change |
| **IT Technicians** edits  ->Student  ->Teacher  account or password | Success, Update the change |
| click logout button | go to the Login page |

**Defect Testing**

|  |
| --- |
| **Layout reuse rule bug** |
| When a user logs out of a teacher or IT Technician account and logs back into the Teacher or IT Technician account again, it will crash the whole system.  We assumed it was caused by the presence of a loop in the GUI that keeps detecting the events, then not breaking the loop when logging out, and then breaking the layout reuse rule when logging in again.  According to this idea, changes should be made to the logout event. When the user clicks logout, it will break the loop first, and call the Login GUI. However, it was normal for the student account to log in after logging out, but the bug still existed in the teacher and IT technician. |

|  |
| --- |
| **Filtered record edit bug** |
| After a filtered record is modified, the clear button is clicked and functioned to return data to an unfiltered record, the record will return to its original state.  We believe the occurrence of this bug is because the original data frame was complicated and filtered when we did the searching. |