Quality Assurance Document for PIM GUI

Contents

[Role Designation 2](#_Toc54106149)

[Database 3](#_Toc54106150)

[Application Design and Installation 3](#_Toc54106151)

# 

# Role Designation

This is the quality assurance documentation for the creation of the Graphical User Interface for the Personal Information Management section of the project.

The members of the PIM group include:

* Nisa Shahril
* Jacob Beynon
* Lewis Impey
* Ryan Murunga

After a team meeting, the roles of each member have been decided based upon the skills that they have and what they would feel the most comfortable doing:

Jacob Beynon – Jacob will be in charge of the actual creation of the GUI for the group. Since Jacob is already experienced in using python and enjoys coding, we collectively decided that he would be the best for that role. We decided to use PyCharm and TKINTER for the creation of the GUI as we already need to learn python as a part of this module so it would make the most sense to use the same language that we are learning.

Lewis Impey – Lewis will take the role of creating the quality assurance documentation for the team and the creation of the design for the GUI. Lewis was chosen to do the quality assurance documentation and the design as both of these go hand in hand as the quality assurance documentation states what the GUI should have and the design is just a visual representation of that.

Nisa Shahril – Nisa will take on the role of organizing the team’s collaboration tools such as the Trello board and creating the PowerPoint for the group seminar. We decided upon this as a group because we believe that Nisa is the best at organization within the group and would be the best at keeping the team on track and reminding everyone as to what they should be doing at any given time.

Ryan Murunga – Ryan will be taken the responsibility of writing the terms and conditions (LSEPI) documentation for the GUI as we believe that he would be the best fit for writing documentation such as this.

# Database

The database that we will be using for the GUI will be SQLite as SQLite was provided to use with a demonstration as to how it works and is the most appropriate for the GUI that we will be building – **ADD TO THIS LATER**

# Application Design and Installation

Before getting into the specifics of what functionality the GUI should have, it is important that a baseline be created that the program follows to make sure that it is not only consistent but also to make sure that the program is readable and takes into consideration any disabilities or conditions some people may have.

Firstly, the text on the GUI should be big enough for anybody to read, including people that may have visual impairments. The text should also follow a consistent trend with the font that is used throughout the program to avoid any disorientation due to a high volume of different font types.

As for the colours that will be used on the GUI, it would be best if the colours of the buttons and textboxes are consistent so that there is no confusion about if something is a textbox or not from somebody that may not be as tech savvy as others. When choosing the colours for the GUI, colour-blind people should also be considered and the chosen colours should be visible to the vast majority of people.

When the program is first loaded, the user should be greeted by a form that includes a title explaining what the program is and the names and student numbers of the people in the team, along with the logo for the company that the program is being made for. The form should also include username and password headings that are accompanied by a textbox. The user should then be able to input their log-in information. For a little bit of added security, the password characters as they are entered should display the '\*' character as to prevent anyone from seeing the password that is being entered. Below the log-in button should be a link to the terms and conditions of using the program, along with a checkbox that the user will need to check to continue. It should be noted that the log-in button should be unavailable to click if the checkbox for the terms and conditions is not checked. If the log-in information that the user entered is incorrect, then text should appear making the user aware of their invalid log-in attempt.

After logging in, the log-in form should close, and the Personnel Records form should open which should allow the user to view all of the personnel records that will be brought into the program by the database that we have created. This form should have the title "Manage Personnel Records" All of the information (first name, last name, telephone number, etc) should be annotated correctly. This form should also have a few different functions that interact with this database (The database should be sorted into ascending order of unique employee IDs).

Firstly, the form should include a subtitle called "Search" and an empty textbox and a button labelled "Search". The textbox should be used to input the name of the person that you would like to search for and when the search button is clicked, the database should show only the people corresponding to that name's information (first and last names, telephone number, etc). The form should also contain a button labelled "Reset" which when clicked should reset the search into the original order that it was in before the search occurred, with the text box also being cleared of what was searched for.

There should also be a button on the form labelled "Delete" which should be unavailable by default to click, that is until the user clicks on the record within the database area that they would like to delete. Clicking the delete button should then remove the selected record from the database.

While it is not exactly shown throughout the program, the database should be updated when things are added and removed and should retain the records from different sessions.

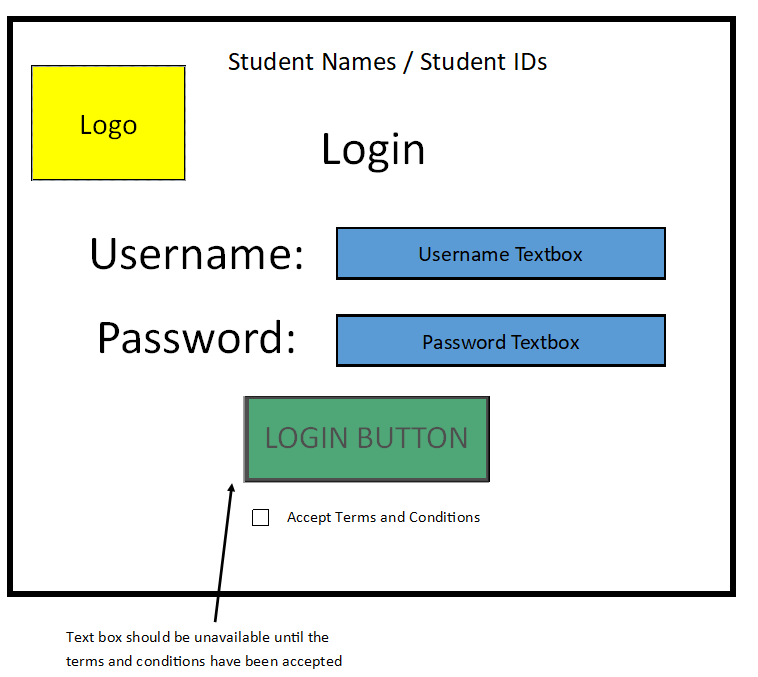
The last button on the form should be labelled "Add" and when clicked should popup a new form.

It should be noted that for additional security measures, if the user has been inactive for a chosen amount of time, the session should timeout and the user should be logged out, to prevent anyone from accessing the Personnel Records while the user is away.

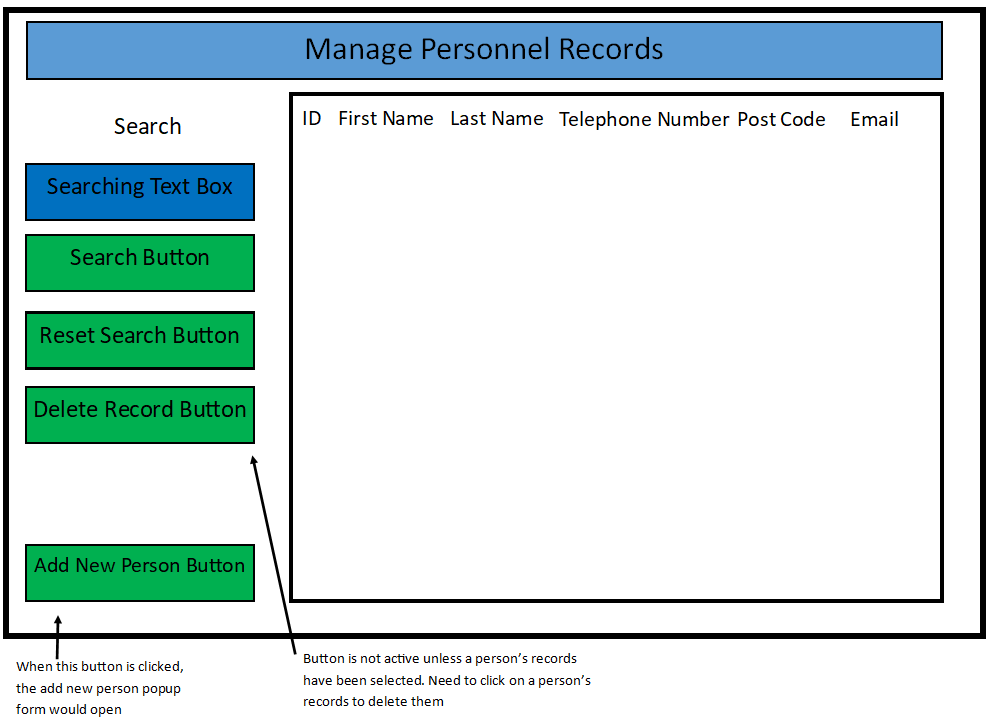
The new form that the "Add" button pops up should be referred to as the Add New Person Form along with a title that says "Add New Person". This form should contain a label with a corresponding empty textbox for each label. These labels and text boxes should refer to information about the new person being added (ID, first name, last name, telephone number, etc). There should then be a button at the bottom of the page labelled "Add" which should add the new person to the database using the new input data. When the "Add" button has been clicked, the New Person Form should close, and the database should have been updated with the new person.

Below are some designs that have been created for the GUI:

Log-in form:



Personnel Records Form:



Add New Person Form:

