Criterion B: Design

INPUT AND OUTPUT

Student Information Tab

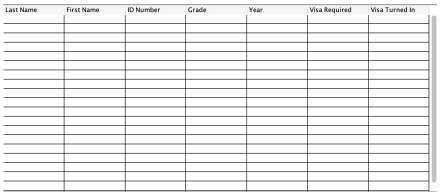
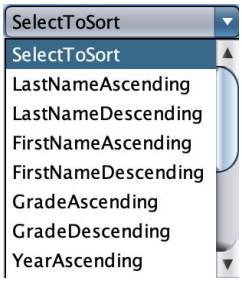
Input	Data Type	Example
Last Name	String	Son
First Name	String	Alex
ID Number	int	20442
Grade	String (Parsed to int)	10
Year	int	2019

Visa Information Tab

Input	Data Type	Example
Full Name	String (ComboBox Selection)	Alex
Passport Nationality	String	Thailand
Passport Number	String	G95742336
Passport Expiry Date	int	20210814
Visa Required?	Boolean	True
Visa TurnedIn?	Boolean	False
Visa Number	String	66343091
Visa Expiry Date	int	20230901

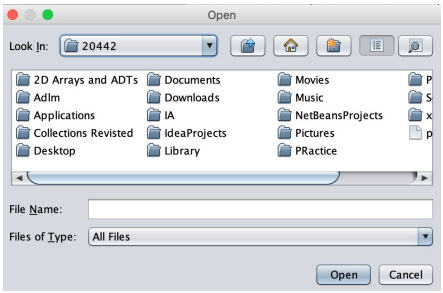
*some visa numbers have alphabets

Display Tab

Output	**Data Type**	**Example**
Display Table	ArrayList <Student> students ArrayList <VisaInformation> visainformation	
Sorts the data as desired when choosing the sorting method from the combobox	Combo Box Selection	

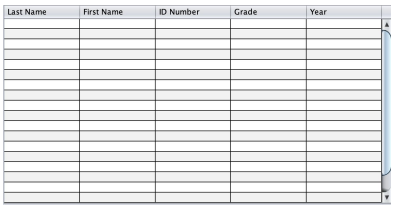
Query Tab

Output	**Data Type**	**Example**
Display Table	ArrayList <VisaInformation> visaRequiredStudents	
Sorting Data	Combo Box Selection	
Show Passport Nationality	String	Thailand
Show Passport Number	String	G95742336
Show Passport Expiry Date	int	20210814
Show Visa Number	String	66343091
Show Visa Expiry Date	int	20230901

Show Visa TurnedIn?	Boolean	False
Image Upload	Image	

Data Management Page

Input	**Data Type**	**Example**
Row number	int	2

Output	**Data Type**	**Example**
Display Table	ArrayList<Student> ArrayList<VisaInformation>	

The rest of input for both in student information and visa information sub-tabs are the same. But these inputs would overwrite the old information.

UML Diagram

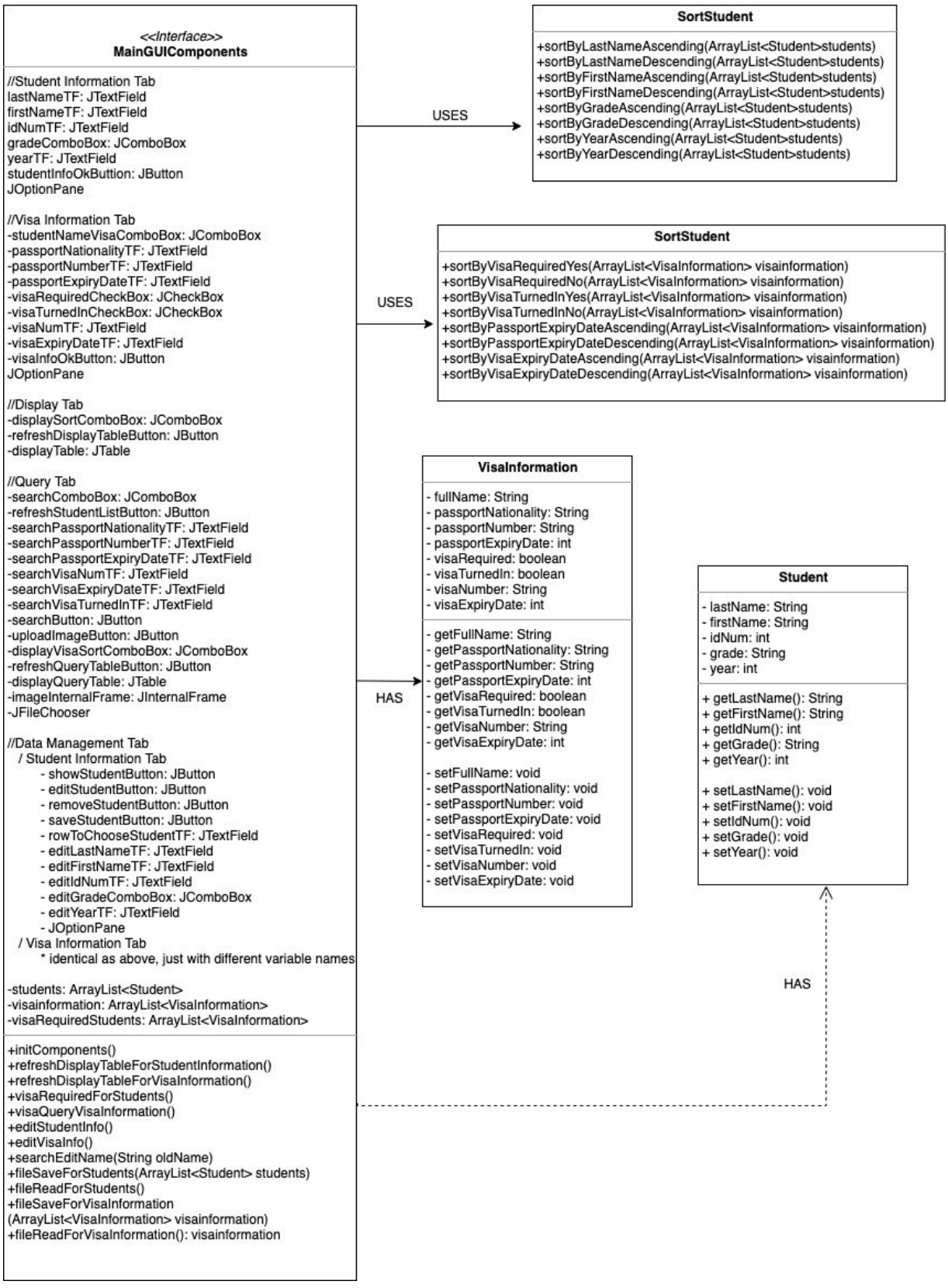


Figure 1: UML diagram, visually portraying the structure of the database. JLabels that are only used for the user interface are not listed in the MainGUI interface as they did not take any role in coding.

Final Prototype

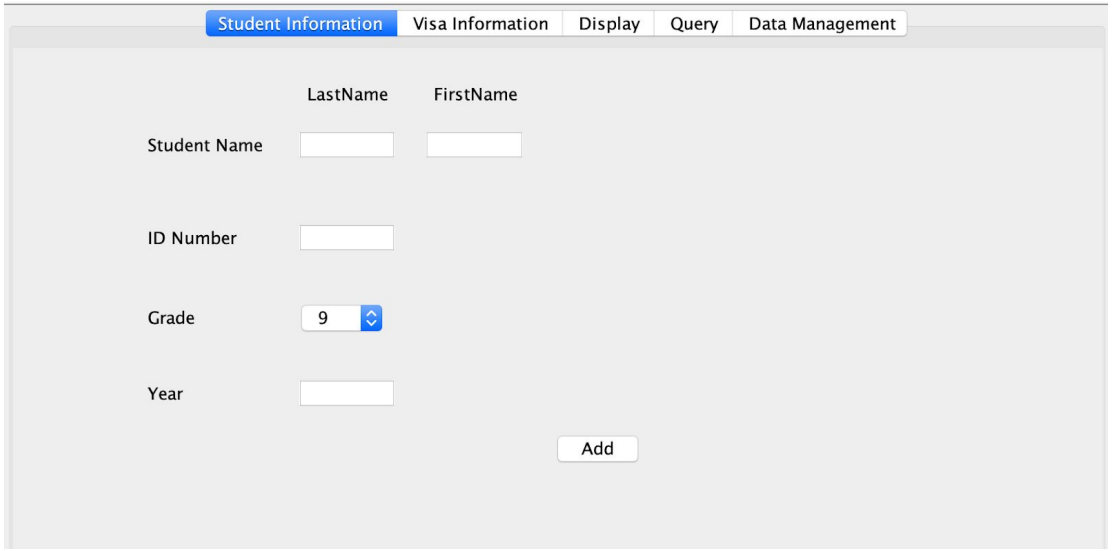


Figure 2: Student Information Tab for data input

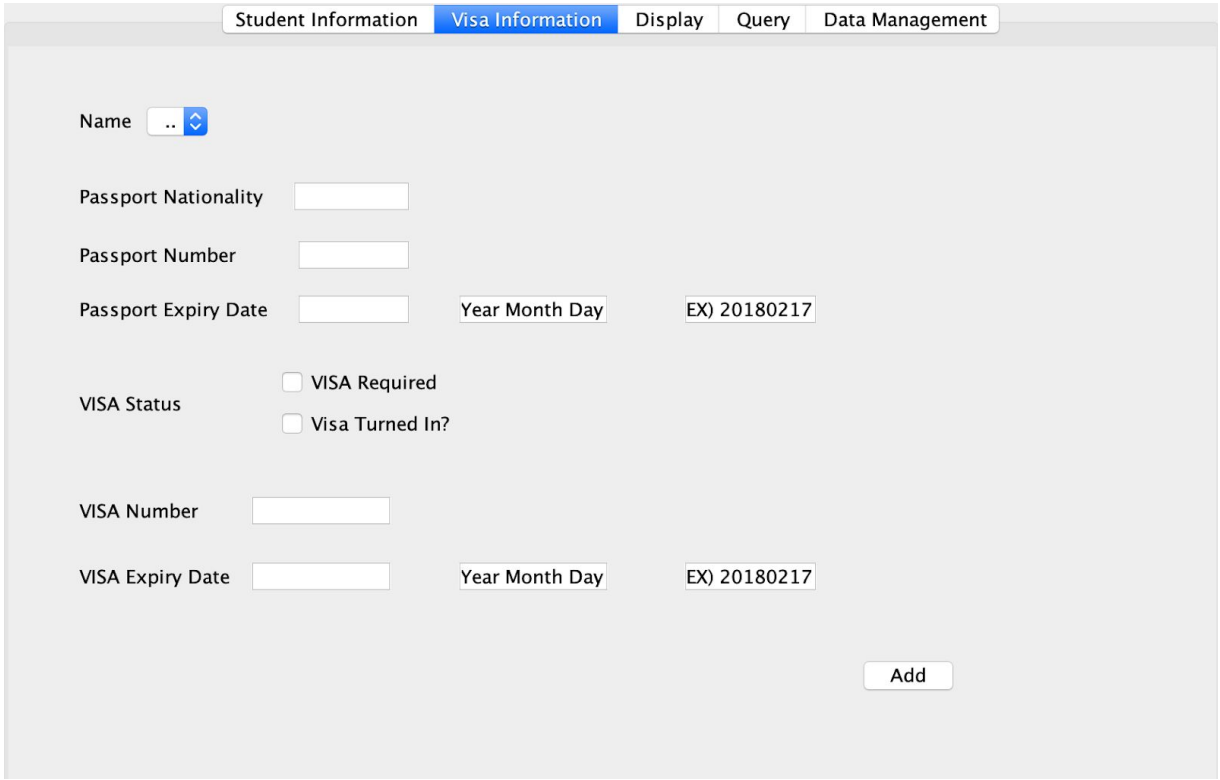


Figure 3: Visa Information Tab for data input

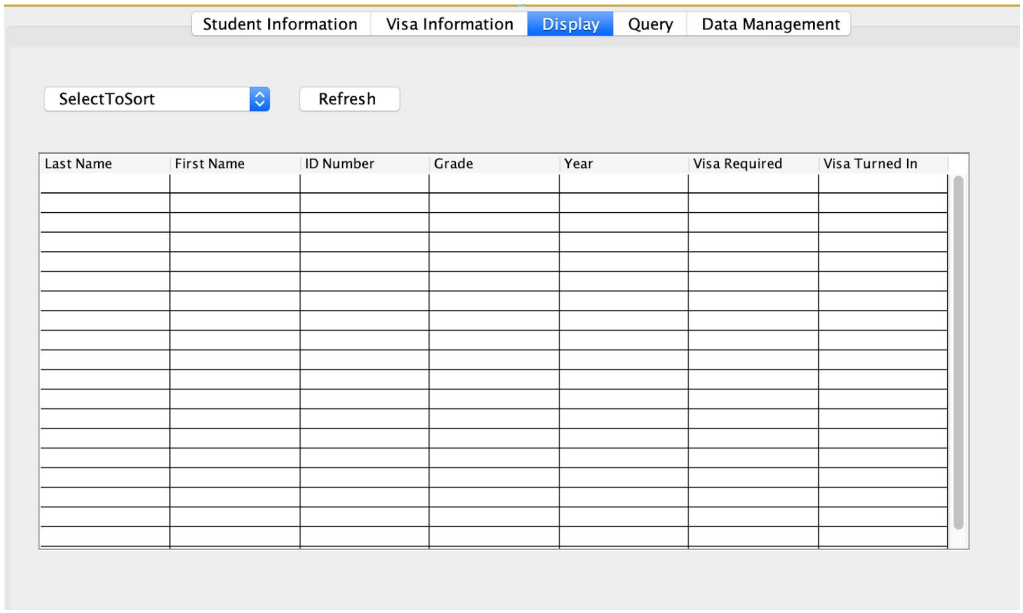


Figure 4: Display Tab to show all student information as well as their visa necessity

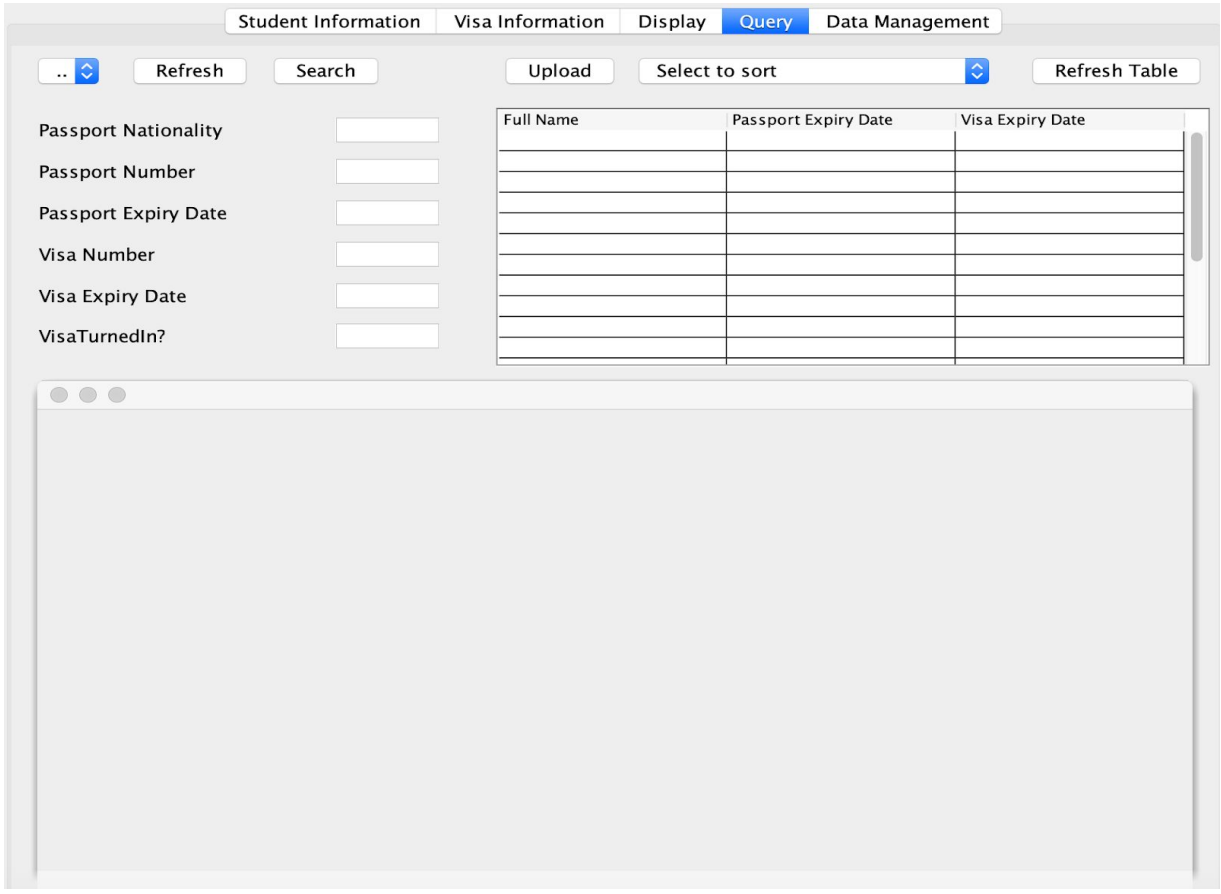


Figure 5: Query Tab for students who need visa. Upload image of the student passport copy.

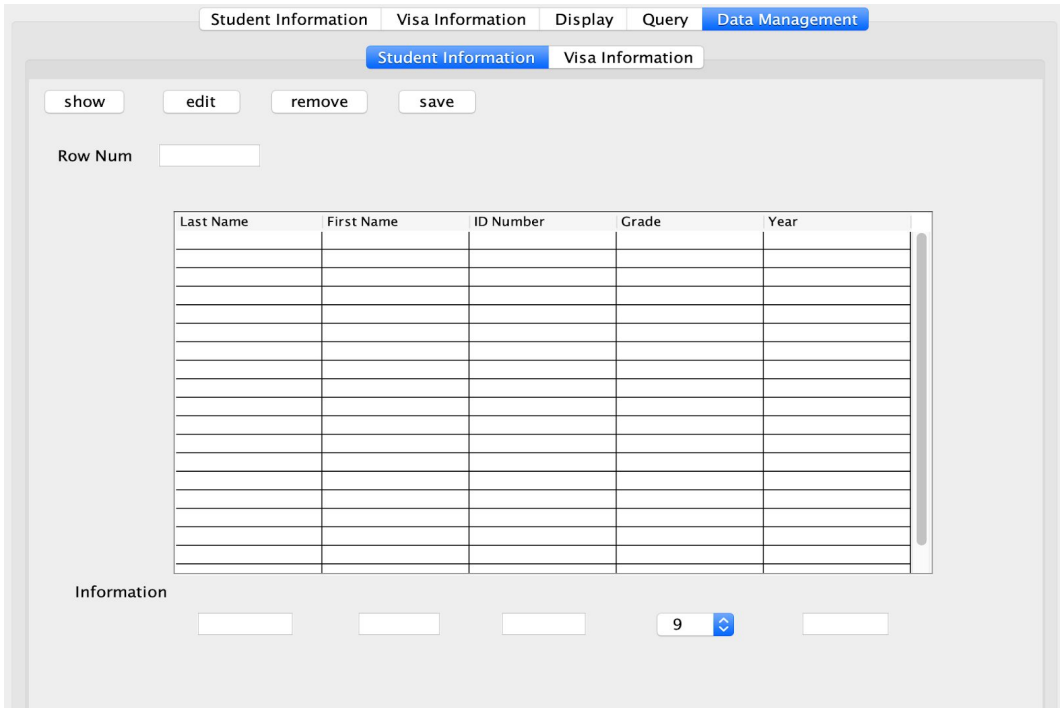


Figure 6: Data Management Tab to edit and remove student information.

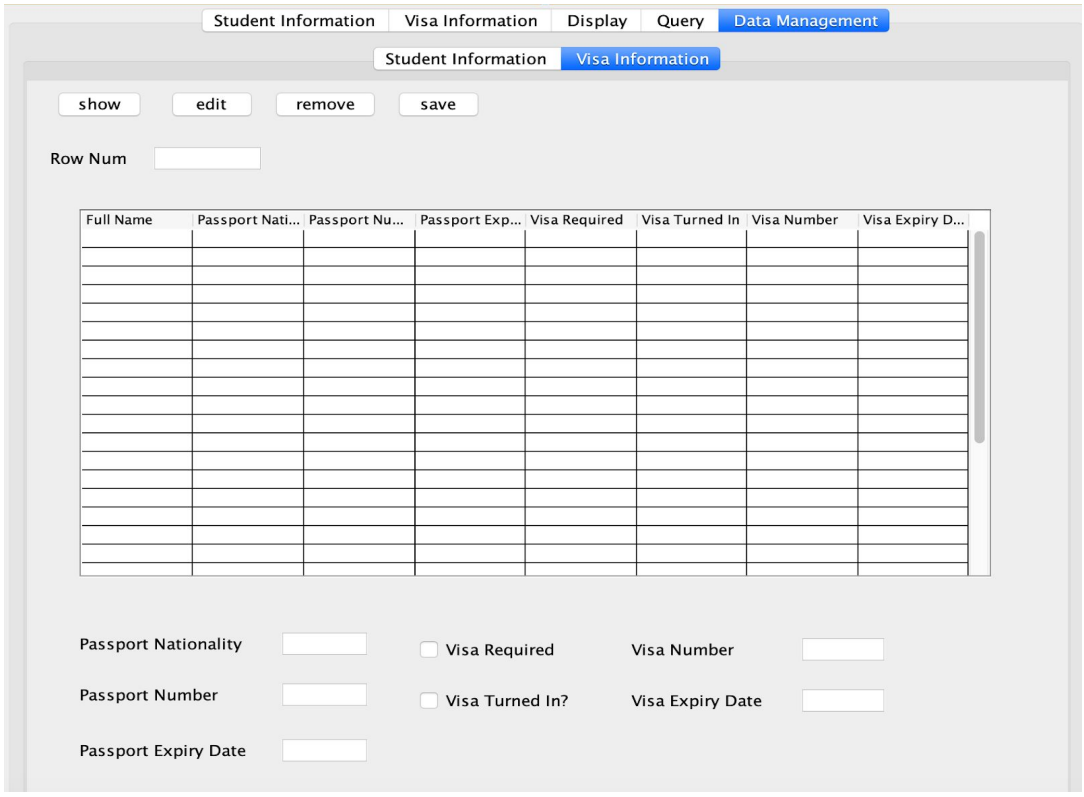


Figure 7: Data Management Tab to edit and remove visa information.

Chronological Development Plan

1. Brainstorm a possible database that I could make to contribute to the school (1 week) Something that has not been concretely constructed yet.
    a. Pick a client depending on the database I am thinking of making it.
    b. Email if the database I am making will be beneficial and meaningful to the client.
2. Initial Client Interview (1 day)
    a. Discuss the possible features that can be added to the database.
    b. Discuss the goal of the database
    c. List of client's desires
3. Create preliminary prototype for discussion (1 day)
4. Second Interview (1 day)
    a. Show and discuss the Initial Prototype of the Database
        i. Addition of Data management Page
        ii. Query Page separation
        iii. Minor Input formatting changes
5. Design MAIN GUI (1 week)
    a. Student Information
    b. Visa Information
    c. Display
    d. Query
    e. Data Management (edit and remove inputted data)
        i. Student Information
        ii. Visa Information
6. Create template classes (Student, Visa Passport) (1 day)
7. Create Sorting Class (SortStudent, SortVisaInformation) (3 days)
8. Work on Programming / Internal Mechanism of the Project (4 weeks)
    a. Key Work
        i. Input Images (passport copy) using filechooser and sort them by their "file names" and upload automatically when a student name is selected.
        ii. Edit and remove information by linking ArrayLists and looping through.
        iii. GUI Interface Backwand Coding ("Add" Button pressed, etc)
9. Create Error Handling Methods (JOptionPane) for any possible input or editing errors.
10. Implement file saving and file reading methods into program (3 days)
11. Execute testing plan to test program. (1 week)
    a. Make necessary changes in error
    b. Reflect on criteria of success
12. Final Interview client (1 day)
    a. Create a video instruction on how to use program
    b. Receive feedback and discuss for extensibilities and recommendations.

Testing Plan:

Addresses all criteria for success

Action To Be Tested	Test Method
Allow the client to save and read all the input information	Revert the local text save files for verification in any addition or change in student and visa information. Run the program multiple times to test if the information is read each time.
Upload Image of a passport page when a certain student is selected after automatic search of the directory folder with load of image files.	Test if and only if png or jpg file can be uploaded. Try uploading a pdf and doc file to ensure error in such occasions. Try inputting multiple images at the same time.
Data management for edit and remove input data	Check if changes made in the Student Arraylist would also be applied to the VisaInformation Arraylist. Check if the changes are applied in the local save files.
Login Page for the user	Type in the username and password with different capitalization.
A Pop-Up notification for any error, verification, or providing necessary information.	Click the “add” button when not all information is typed in. Click the “edit” or “remove” button without typing in which row to edit.
Search through the database to view every necessary information regarding visa	Check if the information matches with what is loaded in local text save files.
All information with the exception of visa and passport number will be sorted in any order	Test all strings that are different by one character. Test all integer values that are within the proximity of 1. Test values with exactly the same value as well.
Students will be filtered automatically if they need a Visa for travelling.	Check if any students without any visa information or have “visaRequired” checkbox boolean result as a false are in the combobox in the “Query” Tab. If so, error has been detected.

Total Word Count: 223 words