Criterion B: Design

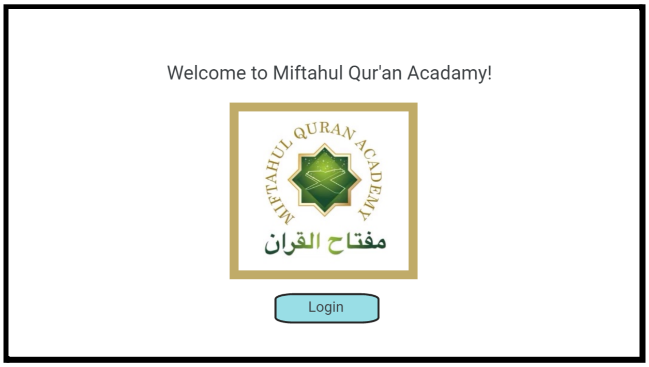
Outline

1. **Design of Panels** - Used to plan out how the product will work and function
2. **Relationship between GUI classes** - Shows how all GUI classes are going to be interconnected with each other
3. **UML diagrams -** How object classes are going to be organized; versions one and two
4. **Progression of The Program and Important Algorithms -** Important methods involved in the planning of the product and flowcharts of how the product will function
5. **File Formats -** Outline of how the files are going to be organized
6. **Test Plan -** Plan for testing the program
7. **Extensibility of Product** - Making the product user-friendly and code listing

Design Of Panels (Graphical User Interface)

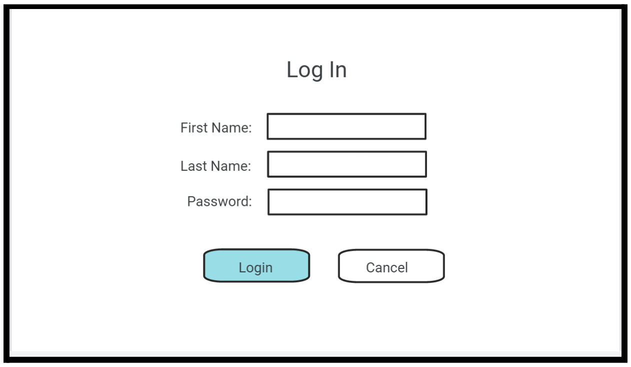
Main Screen

This is the screen that's shown when the user opens the program. Consists of a title, madrasa logo and button.



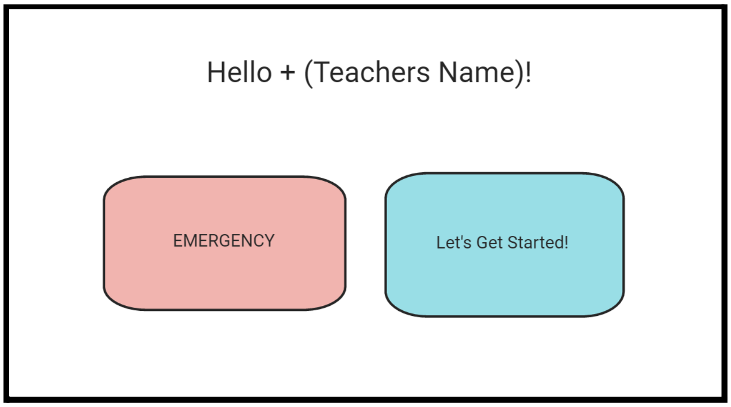
Log in Screen

This is the screen that's shown after the user presses the login button on the main screen.



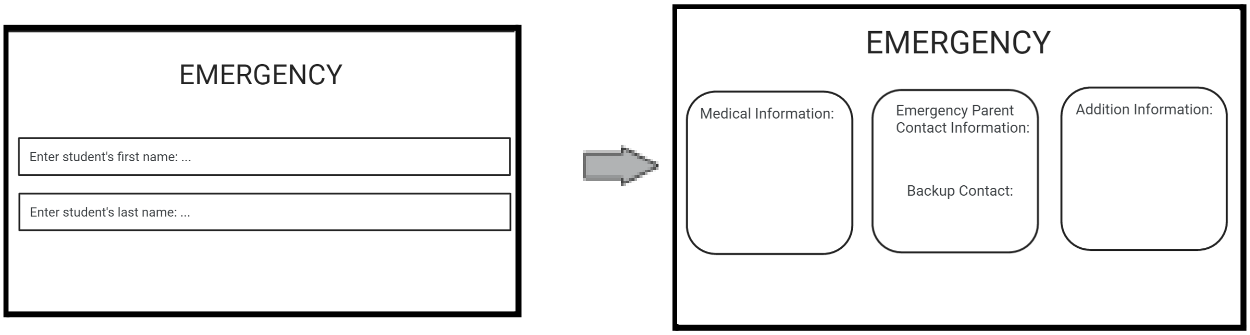
Emergency Or Not Screen

After the user is logged in, they will be shown two buttons asking them if it is an emergency or if they would like to get started.



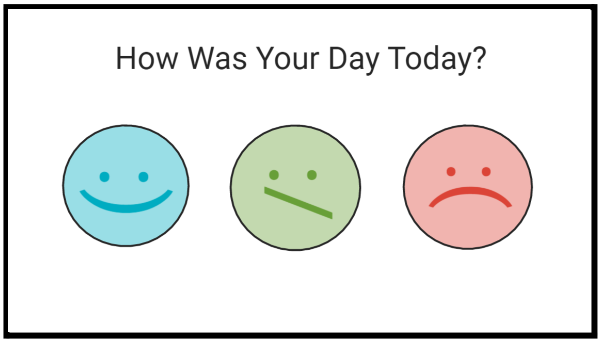
Emergency Option

This is chosen in case of an emergency such as if a student has gotten hurt or if a student is experiencing health conditions.

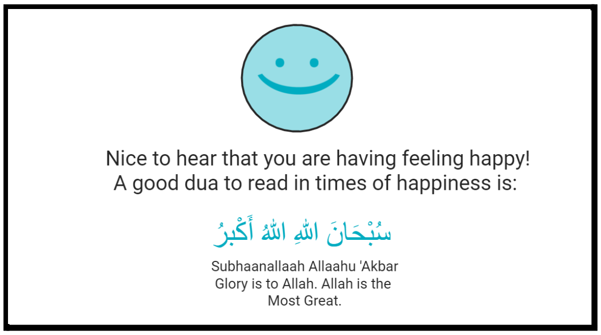

EMERGENCY
Enter student's first name: ...
Enter student's last name: ...
EMERGENCY
Medical Information:
Emergency Parent Contact Information:
Backup Contact:
Addition Information:

Let's Get Started Option

If the let's get started option is chosen, the program goes on to show more options. Each emoji (or image) is clickable.

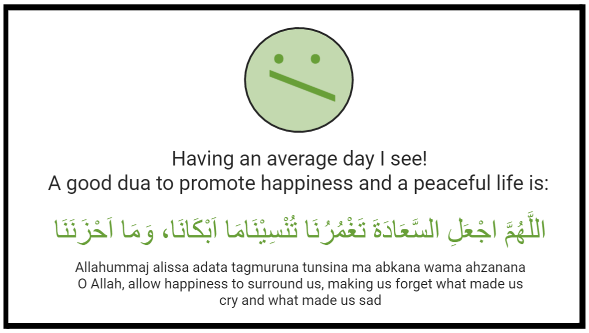

How Was Your Day Today?

Displayed in response to whichever option is chosen. When I am referring to a “dua”, I am referring to a religious saying/request to God.


Nice to hear that you are having feeling happy!
A good dua to read in times of happiness is:

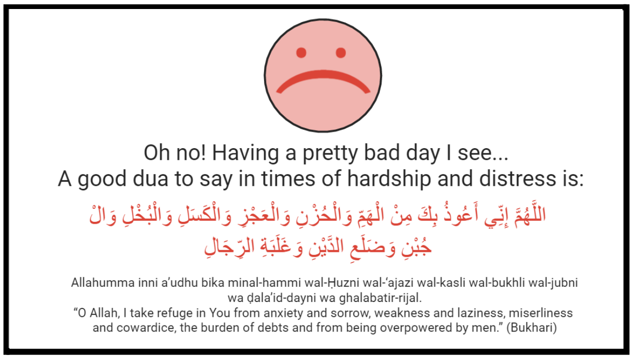
سُبْحَانَ اللهِ اللهُ أَكْبَرُ

Subhaanallaah Allaahu 'Akbar
Glory is to Allah. Allah is the
Most Great.


Having an average day I see!
A good dua to promote happiness and a peaceful life is:

اللَّهُمَّ اجْعَلِ السَّعَادَةَ تَغْمُرُنَا تُنْسِيْنَامَا اَبْكَانَا، وَمَا اَحْزَنَنَا

Allahummaj alissa adata tagmuruna tunsina ma abkana wama ahzanana
O Allah, allow happiness to surround us, making us forget what made us
cry and what made us sad

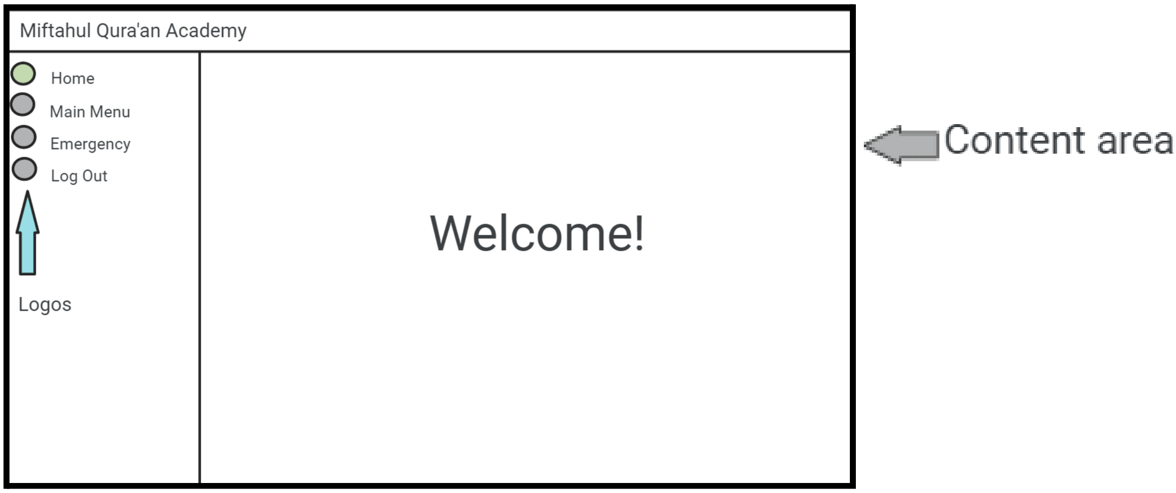

Oh no! Having a pretty bad day I see...
A good dua to say in times of hardship and distress is:

اللَّهُمَّ إِنِّي أَعُوذُ بِكَ مِنْ الْهَمِّ وَالْحُزْنِ وَالْعَجْزِ وَالْكَسَلِ وَالْبُخْلِ وَالْ
جُبْنِ وَضَلَعِ الدَّيْنِ وَغَلَبَةِ الرِّجَالِ

Allahumma inni a’udhu bika minal-hammi wal-Ḥuzni wal-‘ajazi wal-kasli wal-bukhli wal-jubni
wa ḍala’id-dayni wa ghalabatir-rijal.
“O Allah, I take refuge in You from anxiety and sorrow, weakness and laziness, miserliness
and cowardice, the burden of debts and from being overpowered by men.” (Bukhari)

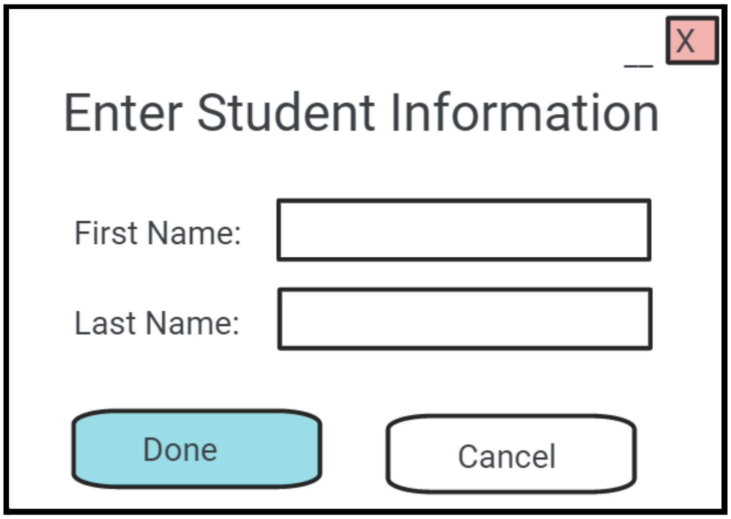
Main View

This is the screen that is displayed as the user has now entered the program. The user can navigate between four tabs on the left. Emergency navigates to “Emergency Option” shown above. Log out navigates to “Main Screen” shown above and the home tab just displays “Welcome!”. Main Menu navigates to “Tab: Main Menu” shown below. All views after here are located in the content area.

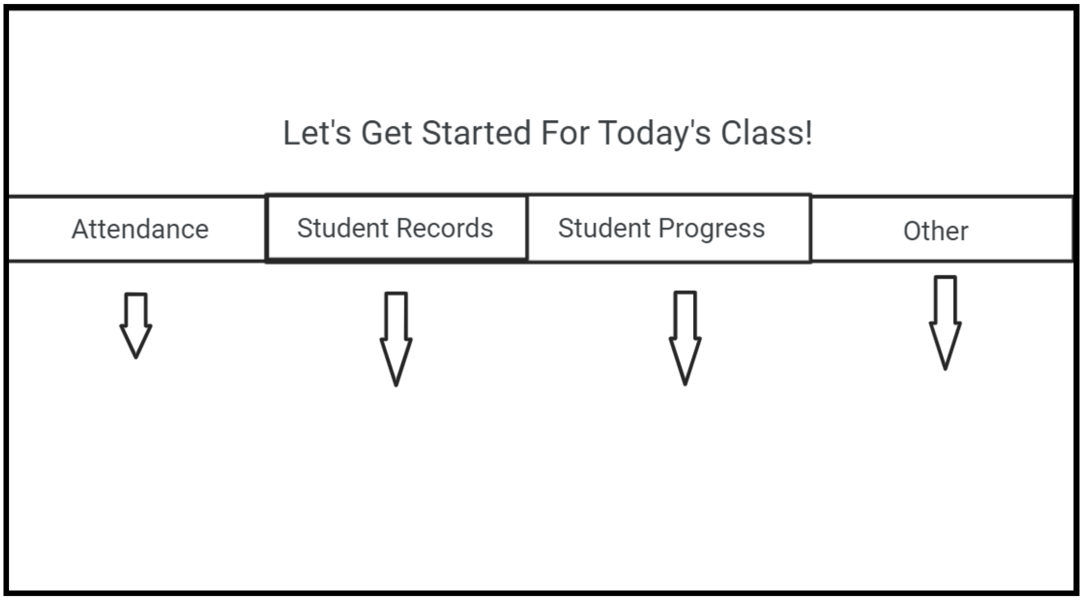


Student Information Dialog

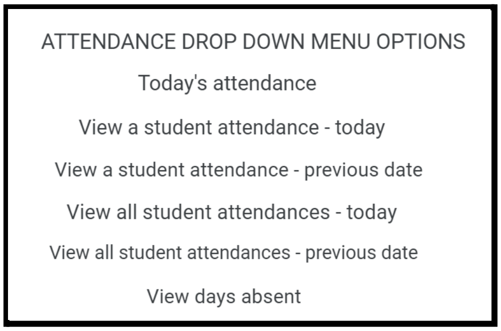
This pops up many times throughout the program and is referenced in the menus below such as view/edit student records.

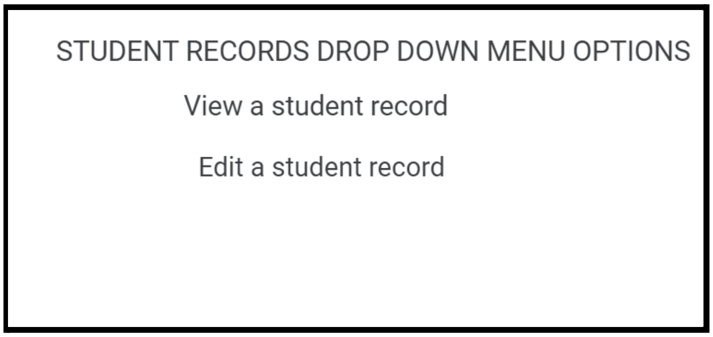


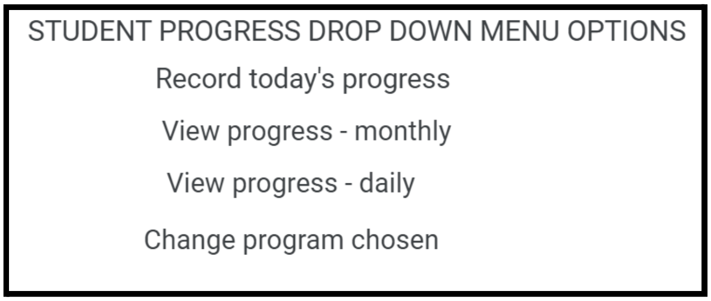
Tab: Main Menu
This brings up the menu page where the user can navigate between the different functions of the program. It uses a drop-down menu to accomplish this.



The drop-down menu options for each function are:


ATTENDANCE DROP DOWN MENU OPTIONS
Today's attendance
View a student attendance - today
View a student attendance - previous date
View all student attendances - today
View all student attendances - previous date
View days absent





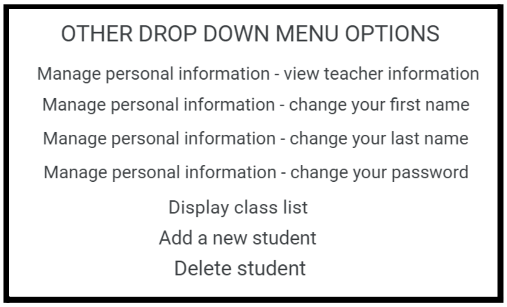
STUDENT PROGRESS DROP DOWN MENU OPTIONS

Record today's progress

View progress - monthly

View progress - daily

Change program chosen



OTHER DROP DOWN MENU OPTIONS

Manage personal information - view teacher information

Manage personal information - change your first name

Manage personal information - change your last name

Manage personal information - change your password

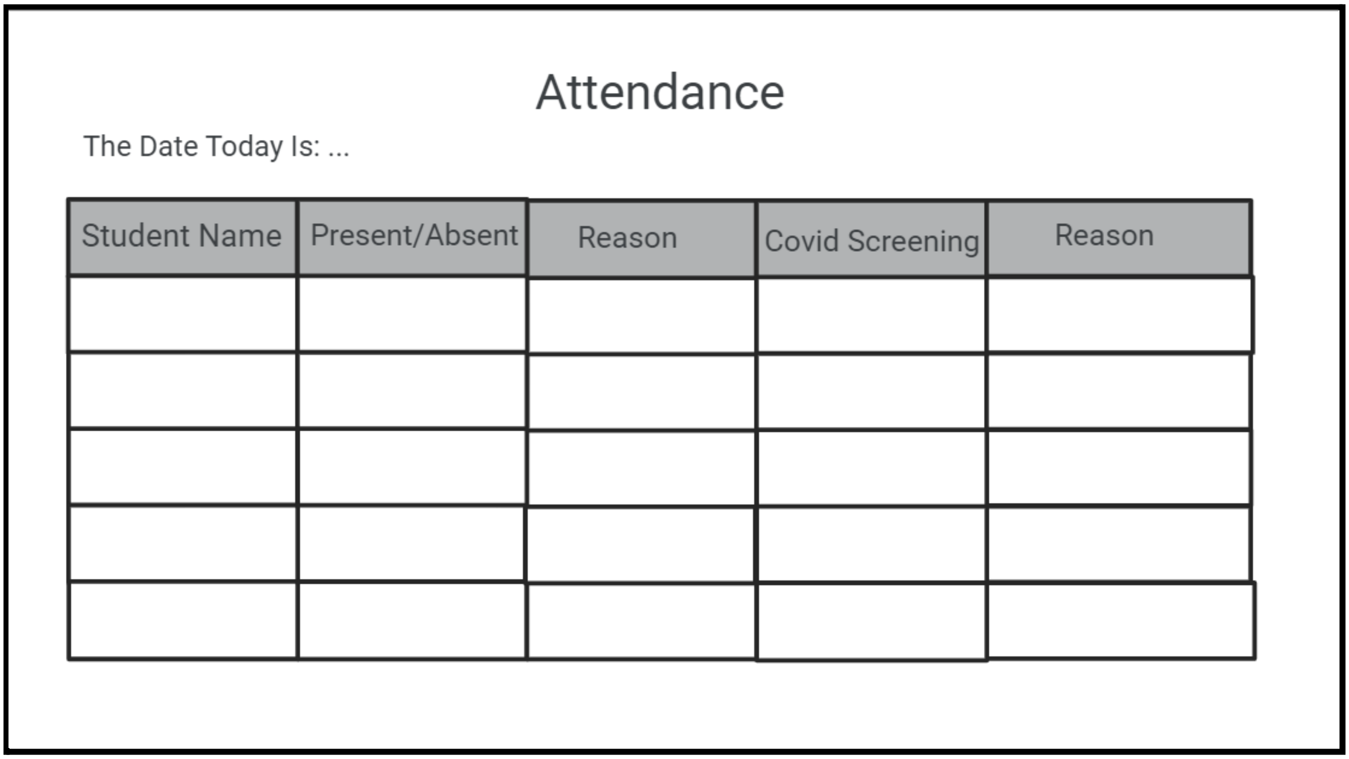
Display class list

Add a new student

Delete student

Attendance Drop Down Menu Options - Today's Attendance, View all student's attendances - today, View all student's attendances - previous date

If recording today's attendance is chosen for the attendance option, the user can input the attendance for the day in a grid. The present/absent and covid screening columns are both checkboxes and the reason for their absence or incompleteness is inputted in the reason column.



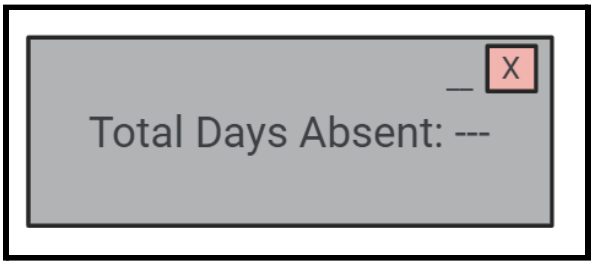
Attendance

The Date Today Is: ...

Student Name	Present/Absent	Reason	Covid Screening	Reason

Main Menu Options - View days absent

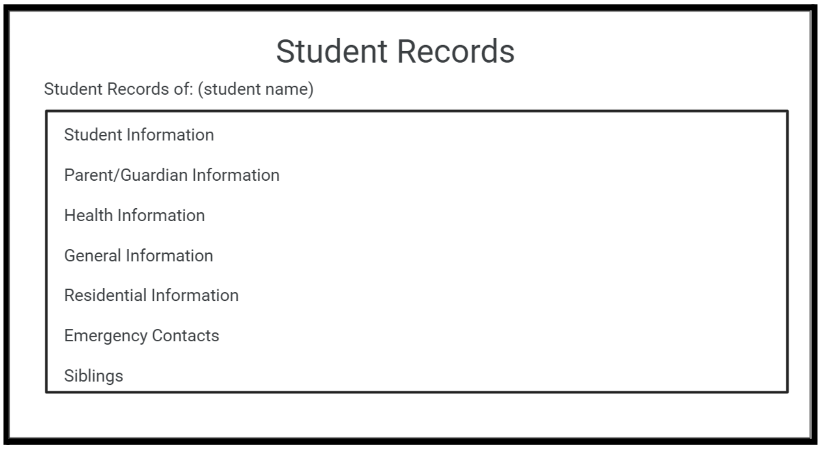
Student information dialogue pops up, then total days absent is calculated and displayed in a dialogue.



Total Days Absent: ---

Main Menu Options - View a student record

Student information dialogue is displayed then all information of the student entered organized by the categories gets displayed.



Student Records

Student Records of: (student name)

Student Information

Parent/Guardian Information

Health Information

General Information

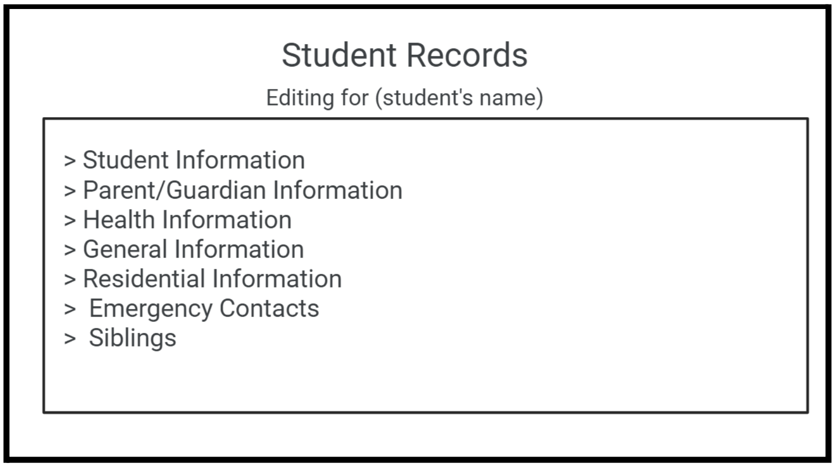
Residential Information

Emergency Contacts

Siblings

Main Menu Options - Edit a student record

Student information dialogue is displayed then information can be edited in each category.



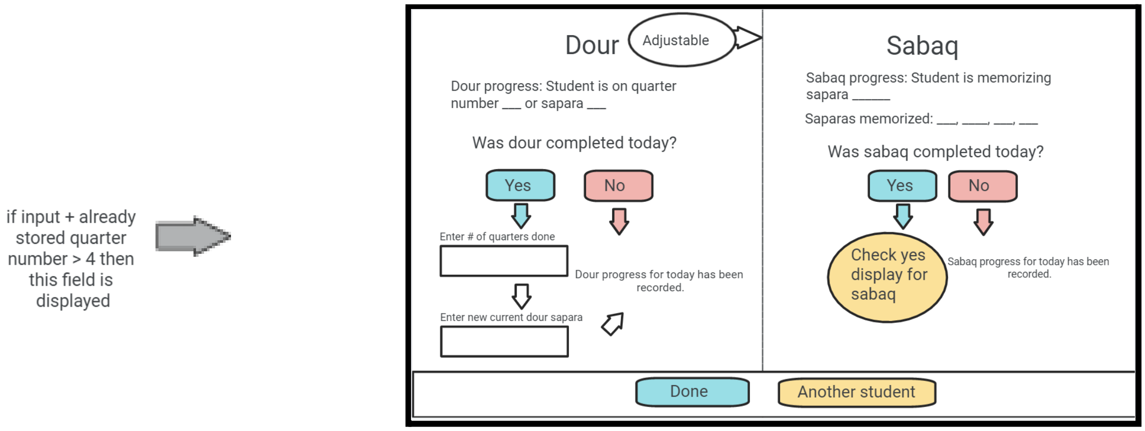
Student Records

Editing for (student's name)

> Student Information
> Parent/Guardian Information
> Health Information
> General Information
> Residential Information
> Emergency Contacts
> Siblings

Main Menu Options - Record today's progress

First student information dialogue, then this screen is shown. View varies depending on the student entered, if the program of the student is "hafiz", they would not be shown the Sabaq half of the page, but for other students, the entire page is shown. This process is explained in appendix VIII.



if input + already stored quarter number > 4 then this field is displayed

Dour

Adjustable

Dour progress: Student is on quarter number ___ or sapara ___

Was dour completed today?

Yes

No

Enter # of quarters done

Dour progress for today has been recorded.

Enter new current dour sapara

Sabaq

Sabaq progress: Student is memorizing sapara _____

Saparas memorized: ___, ___, ___, ___

Was sabaq completed today?

Yes

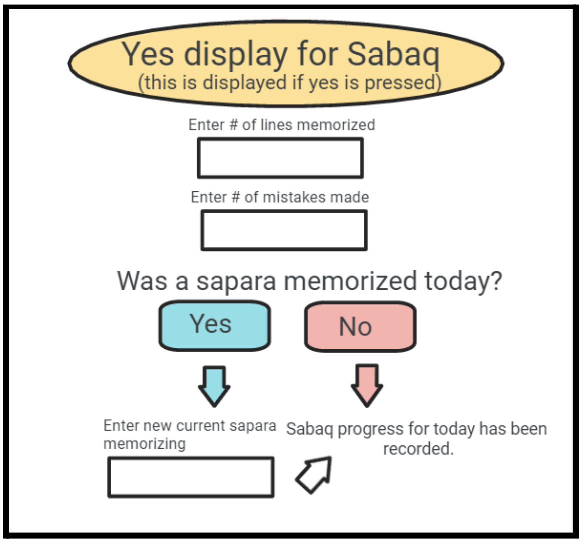
No

Check yes display for sabaq

Sabaq progress for today has been recorded.

Done

Another student



Yes display for Sabaq
(this is displayed if yes is pressed)

Enter # of lines memorized

Enter # of mistakes made

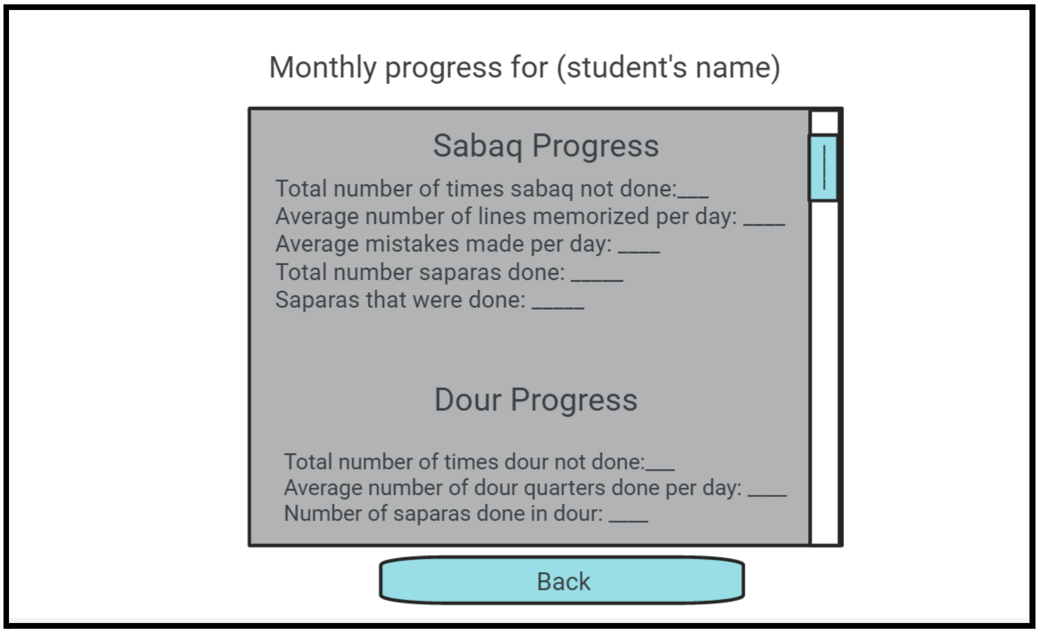
Was a sapara memorized today?

Enter new current sapara memorizing

Sabaq progress for today has been recorded.

Main Menu Options - View progress monthly

Student information dialogue pops up then this screen is displayed. Sabaq progress is not displayed for certain students depending on their program but dour progress is.



Monthly progress for (student's name)

Sabaq Progress

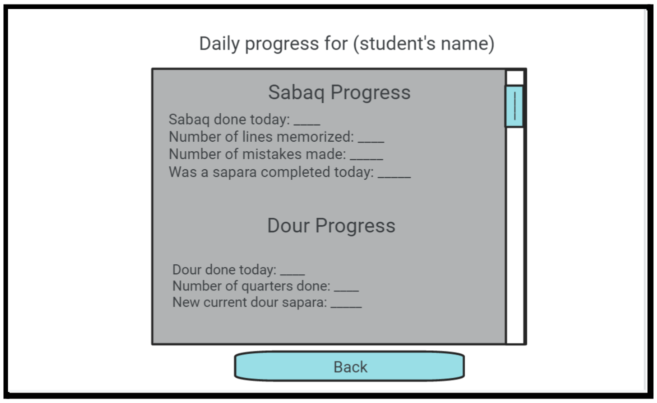
Total number of times sabaq not done:___
Average number of lines memorized per day: ____
Average mistakes made per day: ____
Total number saparas done: _____
Saparas that were done: _____

Dour Progress

Total number of times dour not done:___
Average number of dour quarters done per day: ____
Number of saparas done in dour: ____

Main Menu Options - View progress daily

Student information dialogue pops up, then the progress recorded for the student that day is shown. Sabaq progress is not displayed for students of a certain program.



Daily progress for (student's name)

Sabaq Progress

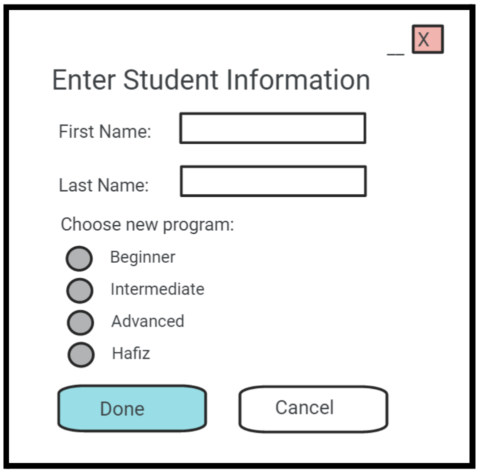
Sabaq done today: ____
Number of lines memorized: ____
Number of mistakes made: ____
Was a sapara completed today: _____

Dour Progress

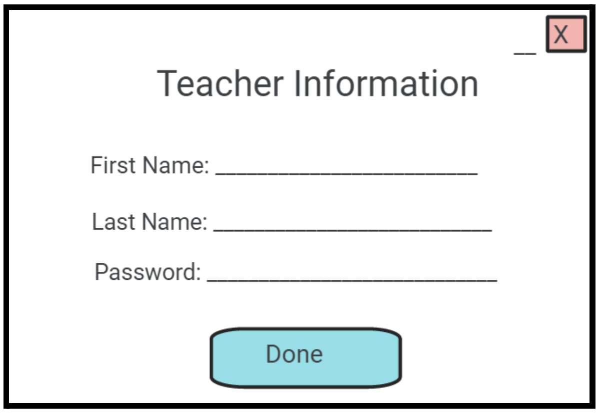
Dour done today: ____
Number of quarters done: ____
New current dour sapara: _____

Main Menu Options - Change program chosen

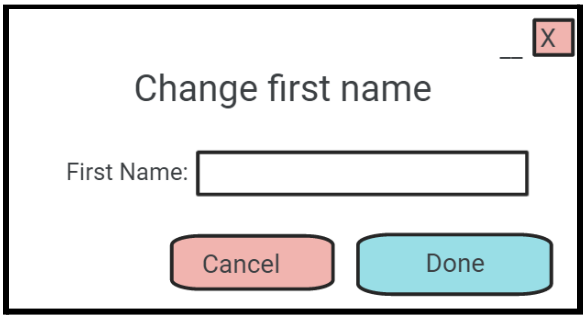
The teacher can change the program of the student they wish when this dialogue is shown. The done and cancel button navigated to the main menu but done saves the new program chosen while cancel does not.



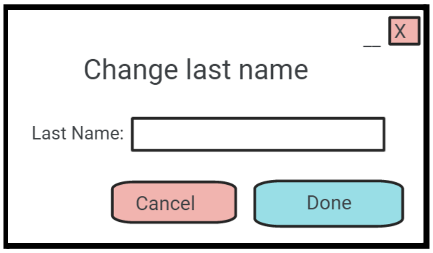
Main Menu Options - Manage personal information - view teacher information
Dialog shows up that displays teacher information. Done button navigates to the main menu.


_ X
Teacher Information
First Name: ______________________
Last Name: _______________________
Password: ________________________
Done

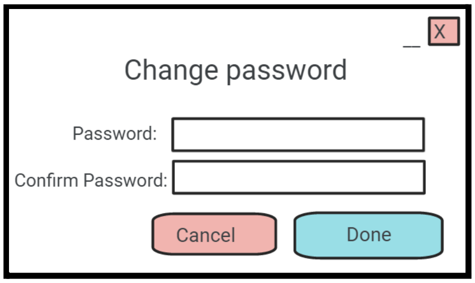
Main Menu Options - Manage personal information - change your first name
Dialog with a text field that allows the teacher to add their new first name. Done navigated to the main menu and cancel discards the changes made.



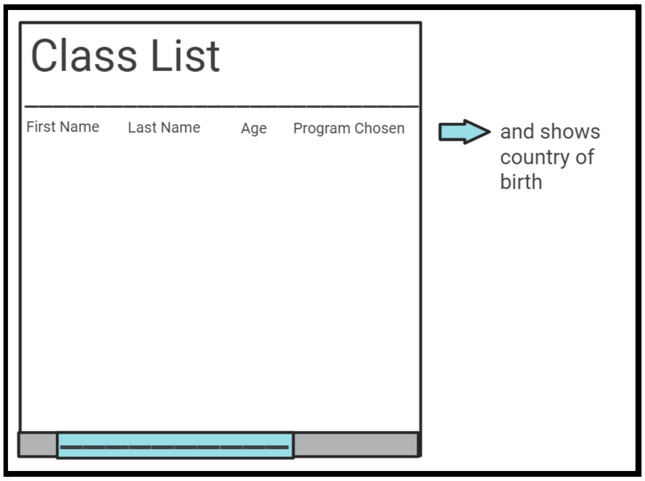
Main Menu Options - Manage personal information - change your last name
Dialog with a text field that allows the teacher to add their new last name. Done navigated to the main menu and cancel discards the changes made.



Main Menu Options - Manage personal information - change your password
Dialog with text fields that allow the teacher to add their new password used when logging in to the program. Done navigated to the main menu and cancel discards the changes made.

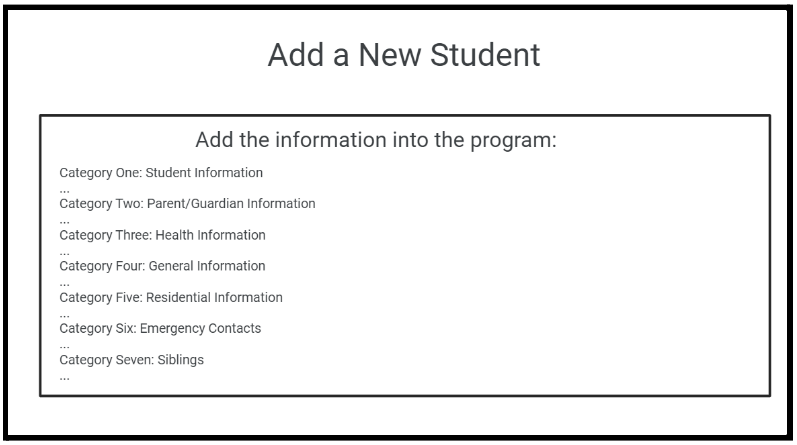


Main Menu Options - Display class list
Is a scrollable dialogue that pops up with key information for all students. Can be exited by clicking anywhere outside of the box.



and shows country of birth

Main Menu Options - Add a new student
Teacher can add all the information needed to register a new student below each category.



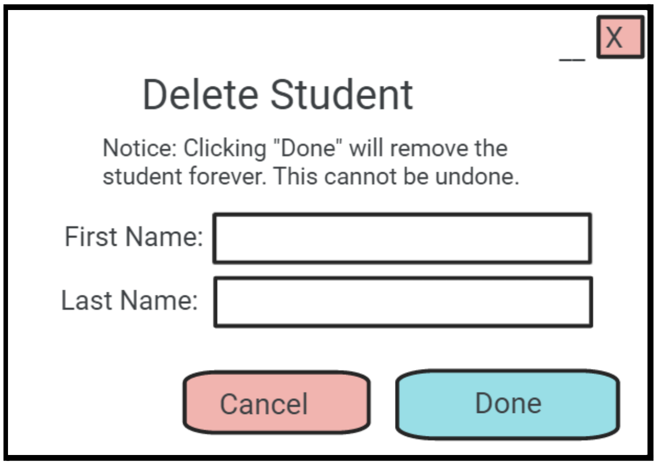
Add a New Student

Add the information into the program:

Category One: Student Information
...
Category Two: Parent/Guardian Information
...
Category Three: Health Information
...
Category Four: General Information
...
Category Five: Residential Information
...
Category Six: Emergency Contacts
...
Category Seven: Siblings
...

Main Menu Options - Delete student

Teacher can input the first and last name of the student they wish to permanently remove. Cancel and Done buttons both navigate to the main menu but done removes the student from the file.

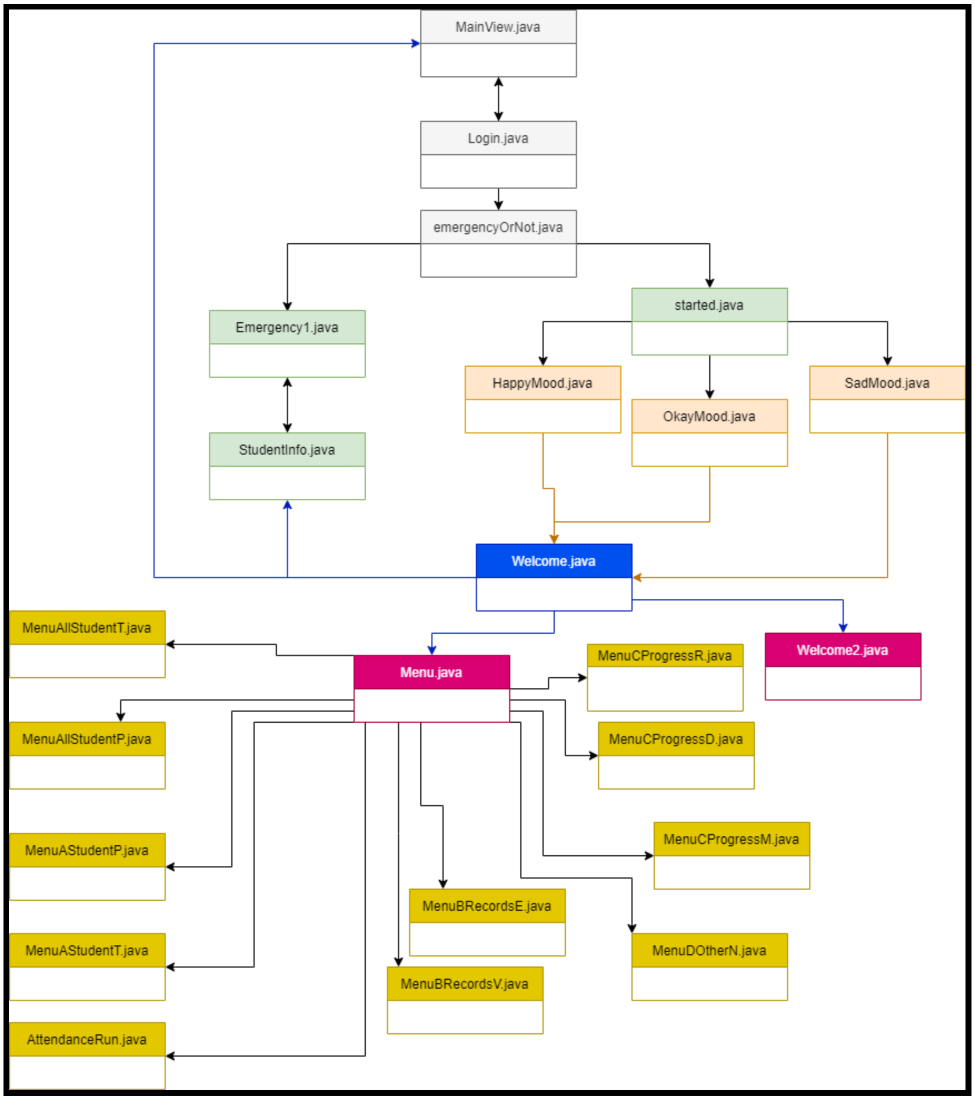


Delete Student

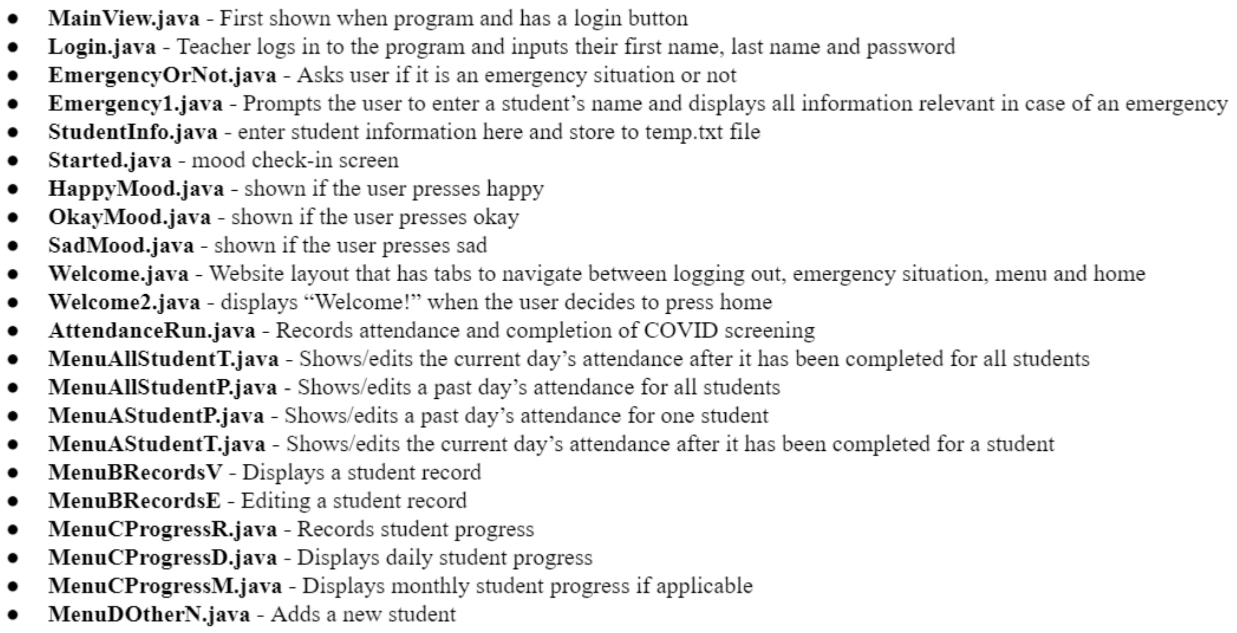
Notice: Clicking "Done" will remove the student forever. This cannot be undone.

Relationship Between GUI Classes

Planning how the classes are connected to each other. The user can navigate between the classes in the given ways when using the program.



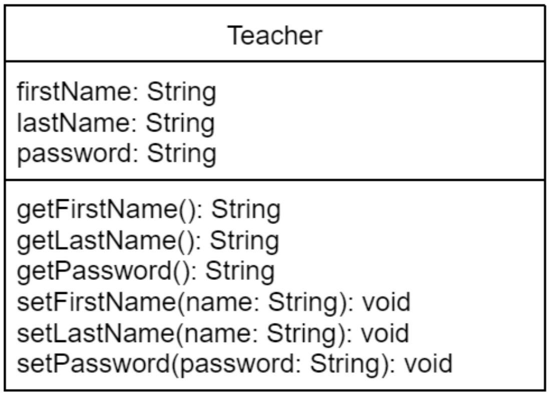
The functionality of each class



- **MainView.java** - First shown when program and has a login button
- **Login.java** - Teacher logs in to the program and inputs their first name, last name and password
- **EmergencyOrNot.java** - Asks user if it is an emergency situation or not
- **Emergency1.java** - Prompts the user to enter a student's name and displays all information relevant in case of an emergency
- **StudentInfo.java** - enter student information here and store to temp.txt file
- **Started.java** - mood check-in screen
- **HappyMood.java** - shown if the user presses happy
- **OkayMood.java** - shown if the user presses okay
- **SadMood.java** - shown if the user presses sad
- **Welcome.java** - Website layout that has tabs to navigate between logging out, emergency situation, menu and home
- **Welcome2.java** - displays "Welcome!" when the user decides to press home
- **AttendanceRun.java** - Records attendance and completion of COVID screening
- **MenuAllStudentT.java** - Shows/edits the current day's attendance after it has been completed for all students
- **MenuAllStudentP.java** - Shows/edits a past day's attendance for all students
- **MenuAStudentP.java** - Shows/edits a past day's attendance for one student
- **MenuAStudentT.java** - Shows/edits the current day's attendance after it has been completed for a student
- **MenuBRecordsV** - Displays a student record
- **MenuBRecordsE** - Editing a student record
- **MenuCProgressR.java** - Records student progress
- **MenuCProgressD.java** - Displays daily student progress
- **MenuCProgressM.java** - Displays monthly student progress if applicable
- **MenuDOtherN.java** - Adds a new student

<u>UML Diagrams of Classes</u>

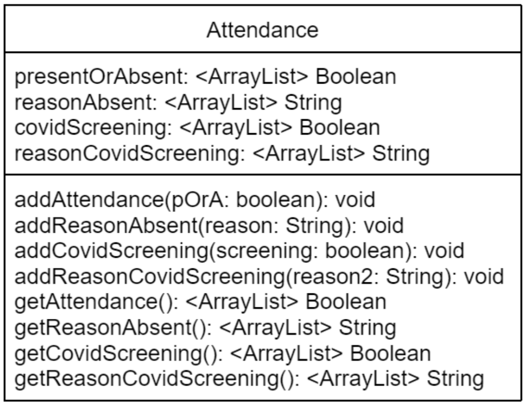
There are four object classes that are used in the program. A UML diagram for each of the classes is below.



Teacher

firstName: String
lastName: String
password: String

getFirstName(): String
getLastName(): String
getPassword(): String
setFirstName(name: String): void
setLastName(name: String): void
setPassword(password: String): void

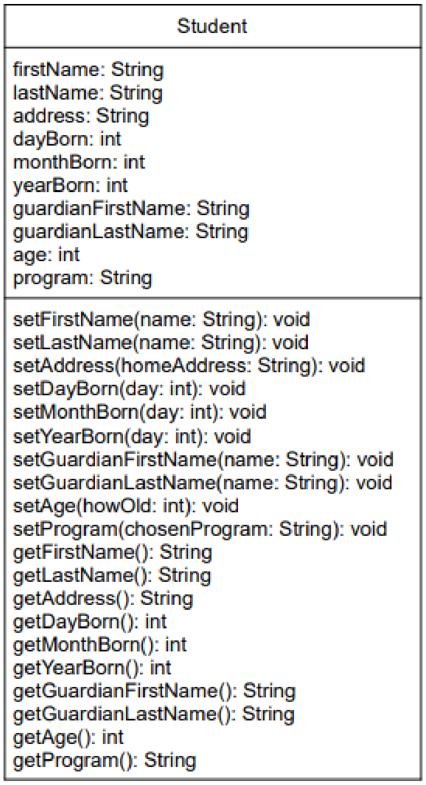


Attendance

presentOrAbsent: <ArrayList> Boolean
reasonAbsent: <ArrayList> String
covidScreening: <ArrayList> Boolean
reasonCovidScreening: <ArrayList> String

addAttendance(pOrA: boolean): void
addReasonAbsent(reason: String): void
addCovidScreening(screening: boolean): void
addReasonCovidScreening(reason2: String): void
getAttendance(): <ArrayList> Boolean
getReasonAbsent(): <ArrayList> String
getCovidScreening(): <ArrayList> Boolean
getReasonCovidScreening(): <ArrayList> String

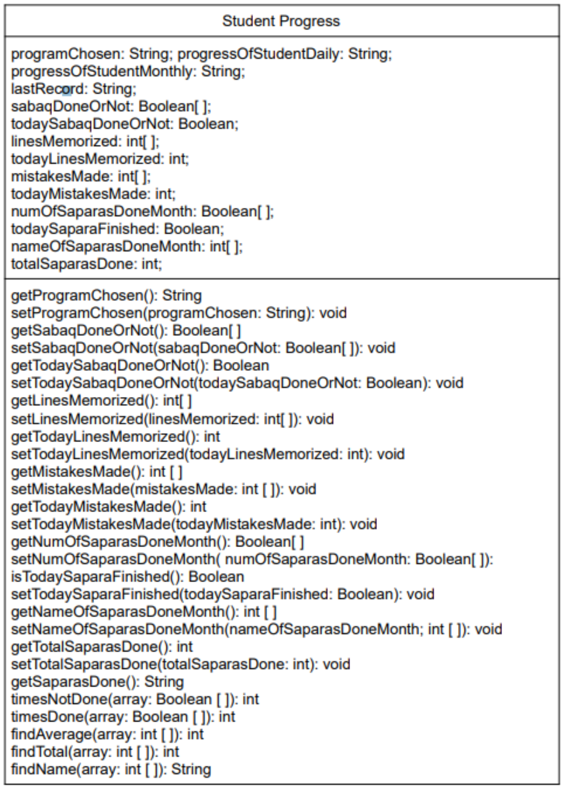
UML diagrams for the first version of the last two classes are below. Client's requests from the program required more variables than could fit in a suitable way in this report in version two. Check appendix IV for the full UML diagrams.



Student

firstName: String
lastName: String
address: String
dayBorn: int
monthBorn: int
yearBorn: int
guardianFirstName: String
guardianLastName: String
age: int
program: String

setFirstName(name: String): void
setLastName(name: String): void
setAddress(homeAddress: String): void
setDayBorn(day: int): void
setMonthBorn(day: int): void
setYearBorn(day: int): void
setGuardianFirstName(name: String): void
setGuardianLastName(name: String): void
setAge(howOld: int): void
setProgram(chosenProgram: String): void
getFirstName(): String
getLastName(): String
getAddress(): String
getDayBorn(): int
getMonthBorn(): int
getYearBorn(): int
getGuardianFirstName(): String
getGuardianLastName(): String
getAge(): int
getProgram(): String



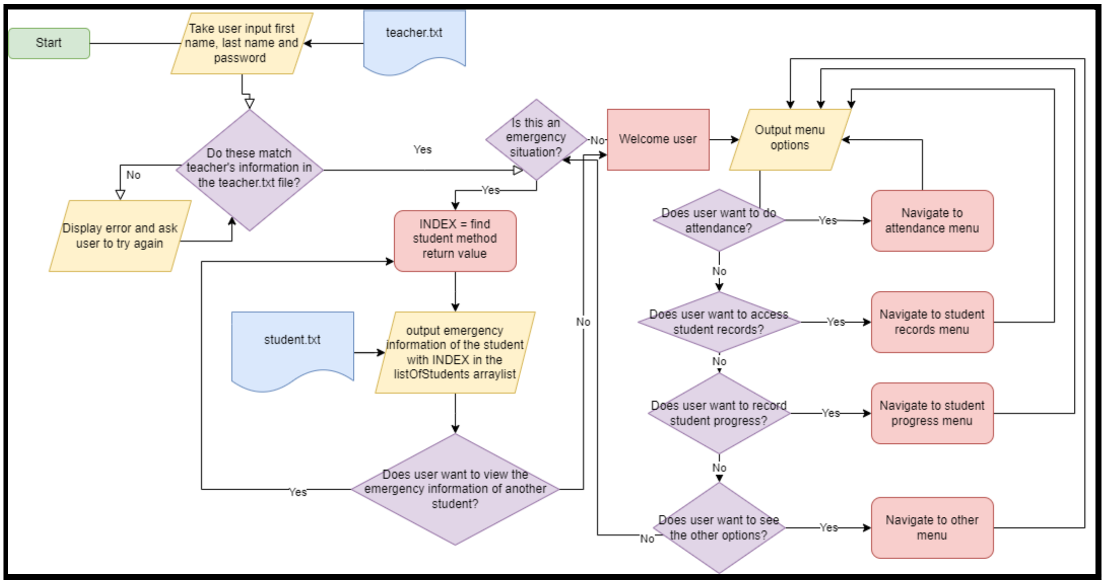
Student Progress

programChosen: String; progressOfStudentDaily: String;
progressOfStudentMonthly: String;
lastRecord: String;
sabaqDoneOrNot: Boolean[ ];
todaySabaqDoneOrNot: Boolean;
linesMemorized: int[ ];
todayLinesMemorized: int;
mistakesMade: int[ ];
todayMistakesMade: int;
numOfSaparasDoneMonth: Boolean[ ];
todaySaparaFinished: Boolean;
nameOfSaparasDoneMonth: int[ ];
totalSaparasDone: int;

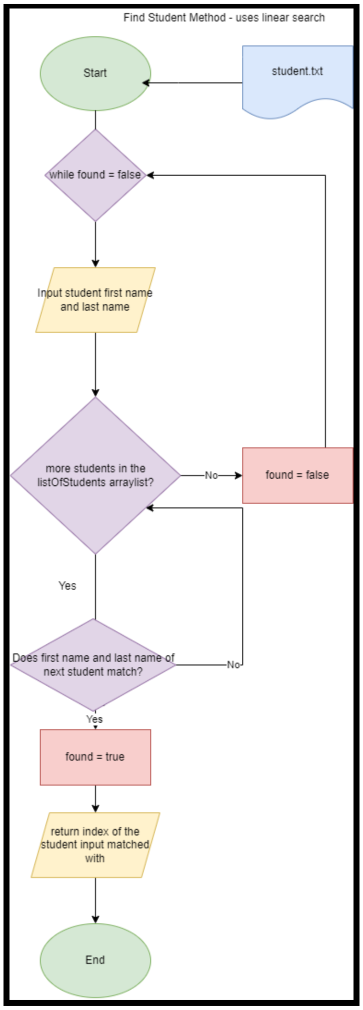
getProgramChosen(): String
setProgramChosen(programChosen: String): void
getSabaqDoneOrNot(): Boolean[ ]
setSabaqDoneOrNot(sabaqDoneOrNot: Boolean[ ]): void
getTodaySabaqDoneOrNot(): Boolean
setTodaySabaqDoneOrNot(todaySabaqDoneOrNot: Boolean): void
getLinesMemorized(): int[ ]
setLinesMemorized(linesMemorized: int[ ]): void
getTodayLinesMemorized(): int
setTodayLinesMemorized(todayLinesMemorized: int): void
getMistakesMade(): int [ ]
setMistakesMade(mistakesMade: int [ ]): void
getTodayMistakesMade(): int
setTodayMistakesMade(todayMistakesMade: int): void
getNumOfSaparasDoneMonth(): Boolean[ ]
setNumOfSaparasDoneMonth(numOfSaparasDoneMonth: Boolean[ ]):
isTodaySaparaFinished(): Boolean
setTodaySaparaFinished(todaySaparaFinished: Boolean): void
getNameOfSaparasDoneMonth(): int [ ]
setNameOfSaparasDoneMonth(nameOfSaparasDoneMonth; int [ ]): void
getTotalSaparasDone(): int
setTotalSaparasDone(totalSaparasDone: int): void
getSaparasDone(): String
timesNotDone(array: Boolean [ ]): int
timesDone(array: Boolean [ ]): int
findAverage(array: int [ ]): int
findTotal(array: int [ ]): int
findName(array: int [ ]): String

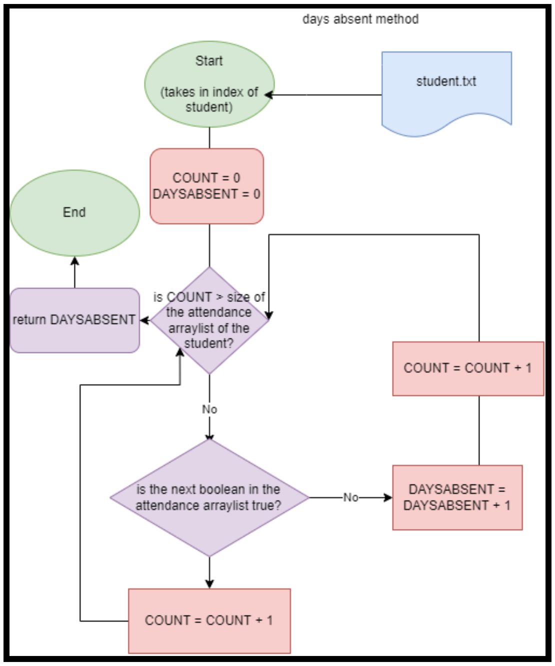
Progression of The Program and Important Algorithms

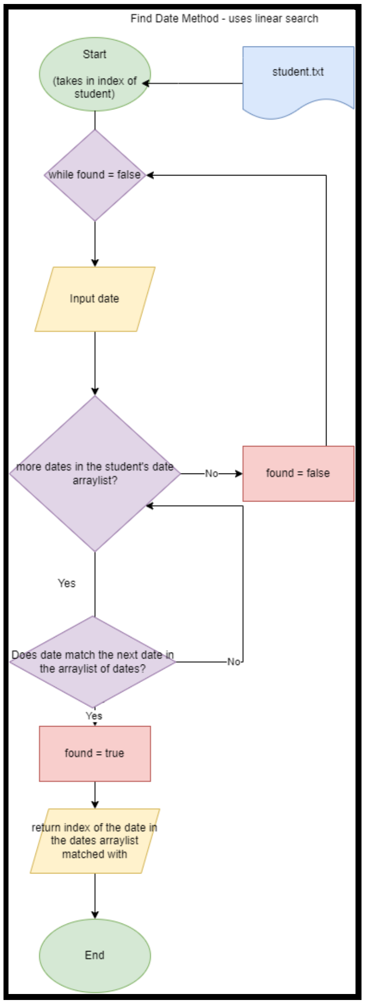
Planning for the functioning of the program and important algorithms/methods are shown below. The algorithms for each menu, however, are in appendix VI.



Start
Take user input first name, last name and password
teacher.txt
Do these match teacher's information in the teacher.txt file?
No
Display error and ask user to try again
Yes
Is this an emergency situation?
No
Welcome user
Output menu options
Yes
INDEX = find student method return value
student.txt
output emergency information of the student with INDEX in the listOfStudents arraylist
Does user want to view the emergency information of another student?
Yes
No
Does user want to do attendance?
Yes
Navigate to attendance menu
No
Does user want to access student records?
Yes
Navigate to student records menu
No
Does user want to record student progress?
Yes
Navigate to student progress menu
No
Does user want to see the other options?
Yes
Navigate to other menu
No

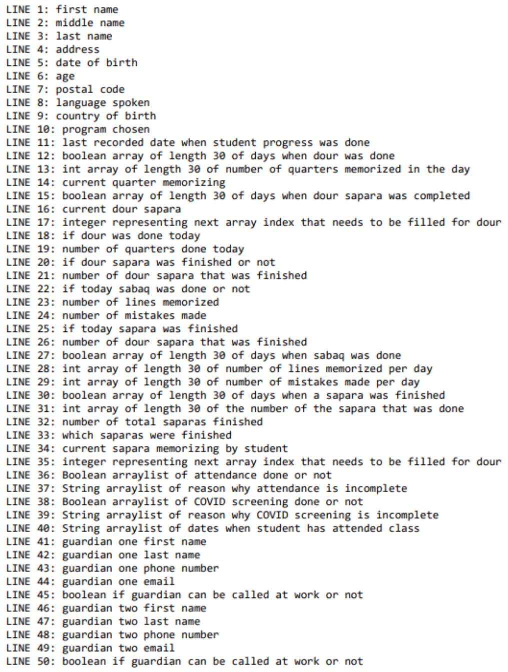

Find Student Method - uses linear search
Start
student.txt
while found = false
Input student first name and last name
more students in the listOfStudents arraylist?
No
found = false
Yes
Does first name and last name of next student match?
No
Yes
found = true
return index of the student input matched with
End


days absent method
Start
(takes in index of student)
student.txt
End
COUNT = 0
DAYSABSENT = 0
return DAYSABSENT
is COUNT > size of the attendance arraylist of the student?
COUNT = COUNT + 1
No
is the next boolean in the attendance arraylist true?
No
DAYSABSENT = DAYSABSENT + 1
COUNT = COUNT + 1

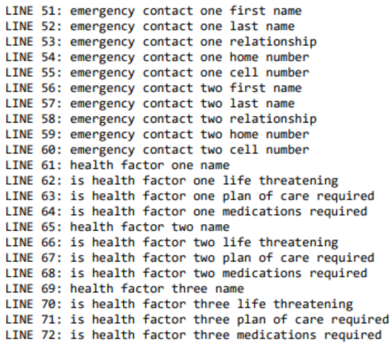

Find Date Method - uses linear search
Start
(takes in index of student)
student.txt
while found = false
Input date
more dates in the student's date arraylist?
No
found = false
Yes
Does date match the next date in the arraylist of dates?
No
Yes
found = true
return index of the date in the dates arraylist matched with
End

File Formats

student.txt - for any one student. Data for subsequent student(s) will be below the previous one's.

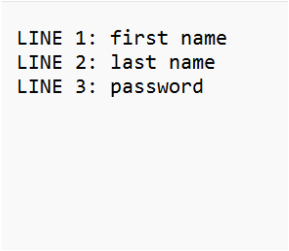


```
LINE 1: first name
LINE 2: middle name
LINE 3: last name
LINE 4: address
LINE 5: date of birth
LINE 6: age
LINE 7: postal code
LINE 8: language spoken
LINE 9: country of birth
LINE 10: program chosen
LINE 11: last recorded date when student progress was done
LINE 12: boolean array of length 30 of days when dour was done
LINE 13: int array of length 30 of number of quarters memorized in the day
LINE 14: current quarter memorizing
LINE 15: boolean array of length 30 of days when dour sapara was completed
LINE 16: current dour sapara
LINE 17: integer representing next array index that needs to be filled for dour
LINE 18: if dour was done today
LINE 19: number of quarters done today
LINE 20: if dour sapara was finished or not
LINE 21: number of dour sapara that was finished
LINE 22: if today sabaq was done or not
LINE 23: number of lines memorized
LINE 24: number of mistakes made
LINE 25: if today sapara was finished
LINE 26: number of dour sapara that was finished
LINE 27: boolean array of length 30 of days when sabaq was done
LINE 28: int array of length 30 of number of lines memorized per day
LINE 29: int array of length 30 of number of mistakes made per day
LINE 30: boolean array of length 30 of days when a sapara was finished
LINE 31: int array of length 30 of the number of the sapara that was done
LINE 32: number of total saparas finished
LINE 33: which saparas were finished
LINE 34: current sapara memorizing by student
LINE 35: integer representing next array index that needs to be filled for dour
LINE 36: Boolean arraylist of attendance done or not
LINE 37: String arraylist of reason why attendance is incomplete
LINE 38: Boolean arraylist of COVID screening done or not
LINE 39: String arraylist of reason why COVID screening is incomplete
LINE 40: String arraylist of dates when student has attended class
LINE 41: guardian one first name
LINE 42: guardian one last name
LINE 43: guardian one phone number
LINE 44: guardian one email
LINE 45: boolean if guardian can be called at work or not
LINE 46: guardian two first name
LINE 47: guardian two last name
LINE 48: guardian two phone number
LINE 49: guardian two email
LINE 50: boolean if guardian can be called at work or not
```



```
LINE 51: emergency contact one first name
LINE 52: emergency contact one last name
LINE 53: emergency contact one relationship
LINE 54: emergency contact one home number
LINE 55: emergency contact one cell number
LINE 56: emergency contact two first name
LINE 57: emergency contact two last name
LINE 58: emergency contact two relationship
LINE 59: emergency contact two home number
LINE 60: emergency contact two cell number
LINE 61: health factor one name
LINE 62: is health factor one life threatening
LINE 63: is health factor one plan of care required
LINE 64: is health factor one medications required
LINE 65: health factor two name
LINE 66: is health factor two life threatening
LINE 67: is health factor two plan of care required
LINE 68: is health factor two medications required
LINE 69: health factor three name
LINE 70: is health factor three life threatening
LINE 71: is health factor three plan of care required
LINE 72: is health factor three medications required
```

teacher.txt



```
LINE 1: first name
LINE 2: last name
LINE 3: password
```

Test Plan

Success Criteria	Test Performed
1. For access to the website, the program will require the teacher to log in and enter their first and last name and password.	1) Click “Login” in the main view. 2) Enter “teacher” for first name, “teacherl” for last name and “password123” for the password field. Unless teacher information has explicitly been changed in main menu, this is the default.
2. The program will allow the teacher to add the current day’s attendances and completion of COVID screenings for all students.	1) In main menu, select “Attendance”, then “Today’s Attendance” 2) Deselect checkboxes and add a reason in the column adjacent to the unchecked boxes. When finished, press “Done”. 3) Hover over “Attendance”, select “Today’s Attendance” from the drop-down menu again and confirm that attendance is done. 4) Hover over “Attendance” then “View all student attendances” then select “Today”. 5) The previously submitted attendance should reflect what is displayed. 6) For retesting purposes: edit the information displayed and press “Done”. Navigate to this menu again and recent edits should be reflected in what is displayed.
3. The program will allow the teacher to view and edit their personal information, including first name, last name and log-in password.	1) In main menu, select “Other”, “Manage passwords and personal information”, then “View teacher information”. 2) Take note of the information displayed then press “Done”. 3) Hover over, “Other”, then “Manage passwords and personal information”, then select “Change your first name”. 4) Enter a new first name and press “Change”. 5) Go back to view teacher information and confirm that the first name has changed. 6) Repeat this process with last name and password.
4. The program will allow the teacher to edit and view student records, in categories of student information, parent/guardian information, emergency contacts and health information.	1) In main menu select “Other” then “Display class list”. 2) Note the first name and last name of any particular student. 3) In the main menu, select “Student Records” then “View a student record”. 4) Input the first name and last name noted. The student record for the student should now be shown. When finished press “Done”. 5) Select “Edit a student record”, input the first name and last name of the student noted then make changes. 6) Navigate back to “View a student record” where these changes should be reflected.
5. The program will allow the teacher to record	1) In main menu, select “Other” then“Display class list”.

daily student progress depending on the student's needs in their respective programs	2) Note the first name and last name of a student with a program chosen "hafiz" and a student with another program. 3) Navigate to the main menu and hover the mouse over "Student Progress" then click on "Record Today's Progress". 4) Enter information for the student and proceed to complete the prompts till Dour and/or Sabaq are recorded. 5) Repeat this process with the other student and if the views vary, the program is successful.
6. In case of data entry errors, the program automatically gives warning messages to the teacher	1) Go to the teacher login page 2) Enter anything in the fields besides "teacher" for first name, "teacherl" for last name and "password123" for the password field. 3) A notification should be displayed that incorrect information is entered.
7. The program will generate a monthly progress report for each student	1) In main menu, select "Other" then "Display class list". 2) Note the first and last name of any student that has "Yes" displayed under the column titled "Applicable for Monthly Report?". 3) In the main menu under "Student Progress" click "View progress", then "Monthly progress". Enter in the student information of the student noted before. 4) Monthly progress should be displayed.
8. The teacher will be able to add and delete students	1) In main menu, select "Other" then "Add a new student". 2) Fill out all fields and press "Done". 3) Then click on "Other" again and "Display class list", where the new student should be. 4) To delete this student, click "Delete student" and enter the student's first and last name. 5) Press "Done" and view class list again.

Extensibility of Product

To make the program user-friendly, scalable and accessible for further development, these techniques were implemented:

- file input/output rather than hard-coding data.
- using ArrayLists instead of arrays.
- good programming style - comments, camel cases for variables and methods and understandable names of classes.
- allowing the user to add or delete their own data.

The final code listing can be seen in appendix VII.