

Contents

Introduction.....	1
Scope	1
Limitation.....	1
Problem Statement and Objectives	2
Problem Statement.....	2
Objectives	2
Analysis & Design	3
Rough idea of the application	3
Use-case diagram.....	4
Use-case template	5
List of classes	6
Class Relationship Diagram.....	7
Sequence Diagram	8
How the data structures or method or technique works to achieve the project functionalities?	10
Driver Program (StudentSurveySystem.java).....	10
DataAnalysis class (DataAnalysis.java).....	11
FormFunction class (FormFunction.java)	16
Administrator class (Administrator.java)	18
Student class (Student.java).....	21
Inheritance class (Inheritance.java)	22
Form class (Form.java)	24
Overall	25
Screenshot of Student Survey Application	26
Testing report or set of validations you applied	31
Additional Feature – Graphical User Interface (Extra)	41
Screenshot of Student Survey Application Graphical User Interface	41
How the data structures or method or technique works to achieve the project functionalities?	44
Administrator class (Administrator.java)	44
DataAnalysis class (DataAnalysis.java).....	45
Form class (Form.java)	45

FormFunction class (FormFunction.java)	45
Inheritance class (Inheritance.java)	45
Student class (Student.java)	45
UserMenu class (UserMenu.java)	45
AdministratorMenu class (AdministratorMenu.java)	46
AnalysisForm class (AnalysisForm.java)	47
Survey class (Survey.java)	49
StudentSurveySystem class (StudentSurveySystem.java)	50
Testing report or set of validations you applied	51
Work distribution	54
Conclusion	54

Introduction

In this project, a java project that will be create is a Student Survey application. The survey application allows students to take survey and the administrator will be able to access the feedbacks as well as analyze it. The administrator can also control the status of the survey, i.e. edit the survey questions in survey and reset the survey.

Scope

The scope of this project includes java basic expressions and flow controls, java arrays, java array list feature, java file I/O, and most importantly, java classes and objects and inheritance.

In this project, basic expression will be used for function such as basic loop counter and analyzing student's feedback. The flow controls such as do...while loop can be used to loop a section many times until user enter a valid option and switch function can be used in choosing actions for user. Java arrays will be used to store the questions of the survey as there are more than one questions. Java array list feature can be used in creating array list of a survey class so that survey can be added many times. Java file I/O can be used in storing values that will be reduced such as survey questions, students' feedback and administrator password. This is to allow the values can be saved and accessed after each run.

Classes will be created so that the tasks in the Student Survey application can be stored in methods and distributed in its specific category and be used by creating an object instance of the class when needed. Inheritance will also be used to share common variables or method so that classes can share these variables and methods without creating them again in their own class. This can reduce redundancy or avoid duplication.

The Student Survey application can be used by 2 different actors which are administrator and student. In this application, students able to take the survey while administrator can control the survey application and get access the details of the feedback. Administrator can edit the survey, analyze survey feedbacks, reset survey and even edit administrator password. Student will not be able to all these control as password is required to login into administrator functions.

Limitation

The limitation of this Student Survey application is that the template of the survey is fixed and administrator is not allowed to change it in the program. Administrator is only allowed edit the questions in the survey template. The survey also only allows rating questions, which is rating of between 1 to 5. Besides, only 1 set of survey can be created and all subject must share the same set of survey. Also, if there are many administrator, the same password has to be shared among the administrator. Other than that, another limitation of this project is that the administrator can edit the survey question while the survey feedback remains the same. Moreover, administrator only allowed to change password after entering administrator functions. Lastly, students have to answered the previous survey questions before moving to the next.

Problem Statement and Objectives

Problem Statement

How to design and implement a Student Survey application using object oriented approach?

Objectives

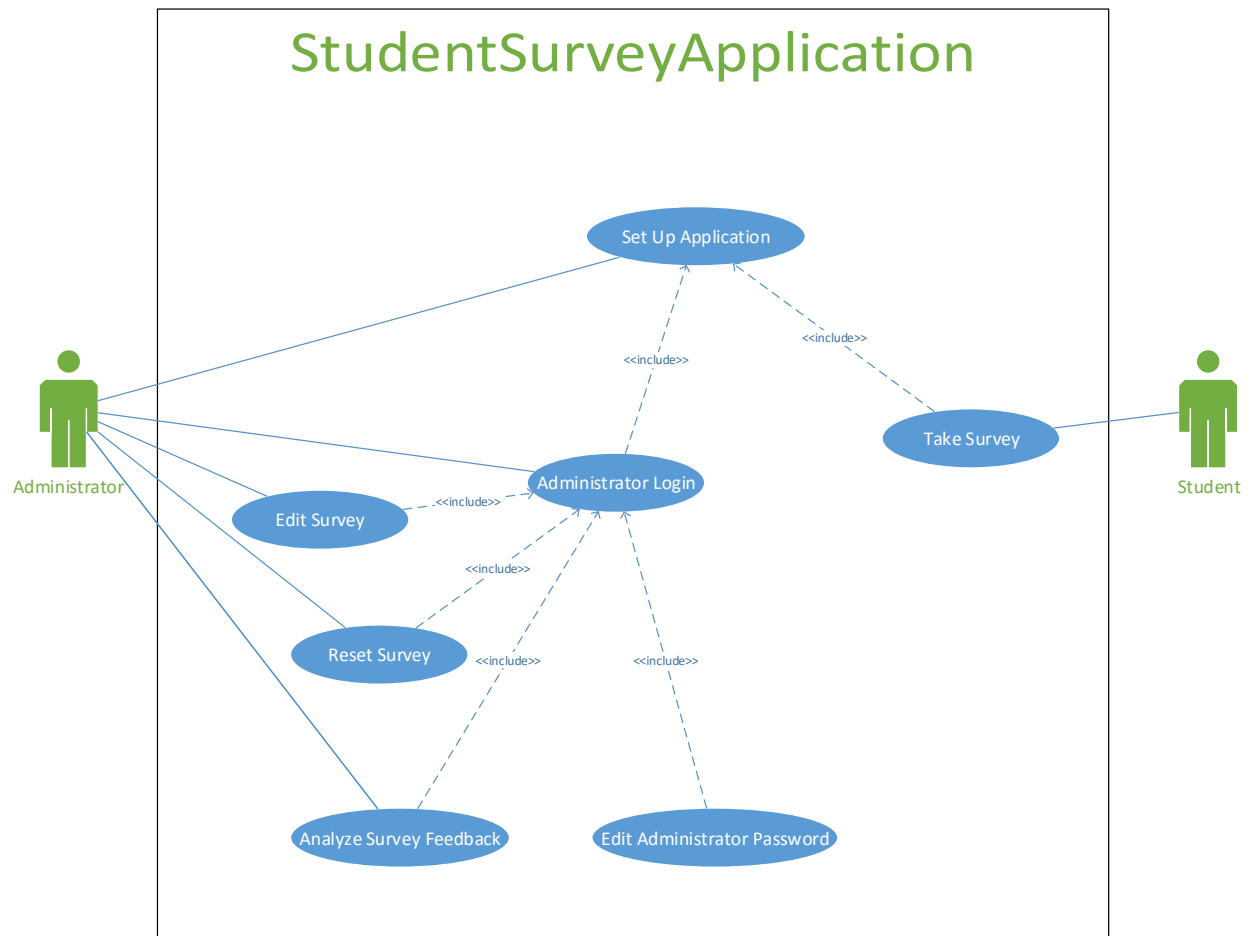
- Create a Student Survey application and implementing it as a Java application.
- Fully use all the Java knowledge that has learned and apply it in this Student Survey application.
- Allow students to take survey and administrator to access the students' feedbacks in the application.

Analysis & Design

Rough idea of the application

- This is a student survey application that allows survey to be taken by students and feedback to be received by administrator.
- In this survey application, student have 4 options, which are to take the survey for four different subject while administrator have few options such as analyze students' feedback, edit survey question, edit administrator password and reset survey.
- The survey application works in this way:
 - FIRST run
 - Program will require administrator ask to set administrator password. Administrator have to insert password two times to be able to start setting up the survey application.
 - Administrator will then direct transfer to user choosing page.
 - After FIRST run
 - Program directs user to a page with 2 options, Administrator and Student.
 - If administrator is chosen:
 - User required to login using the password that was set initially or password that has changed.
 - After password verification, if the password is correct, user will be directed to administrator page with options such as analyze students' feedback, edit survey question, reset survey, edit administrator password and exit.
 - The function of each options will be as below:
 - Analyze students' feedback: Analyze aspects that students entered in survey for each subjects.
 - Edit survey question: Edit survey question for new survey or if survey question is entered wrongly.
 - Reset survey: Remove all the survey question and reset the database that stores survey feedback.
 - Edit administrator password: Edit administrator password by entering it two times.
 - Exit: exit administrator page.
 - Exit administrator page by entering specified option.
 - If student is chosen:
 - User will be brought to survey page to take survey.
 - User will first require to insert gender and age so that this information can be reused in other subject survey.
 - After entering gender and age, user will have options for 4 different subject survey or quit the survey.
 - When a survey chosen, user have to fill up the survey to be able to exit back to home page.

Use-case diagram



Use-case template

Name: Set up application

Goal: Create password for the survey application in first run.

Actors: Administrator

Precondition: -

Name: Administrator login

Goal: Login to get more functions on the survey

Actors: Administrator

Precondition: Survey application have to be set up first

Name: Edit survey

Goal: Edit the questions to be asked in the survey

Actors: Administrator

Precondition: Login to administrator and survey application have to be set up first

Name: Analyze survey feedback

Goal: Analyze the information key in by participants

Actors: Administrator

Precondition: Login to administrator and survey application have to be set up first

Name: Edit administrator password

Goal: Edit the password that allows administrator to enter administrator function

Actors: Administrator

Precondition: Login to administrator and survey application have to be set up first

Name: Take survey

Goal: Survey form is filled up

Actors: Students

Precondition: Survey application have to be set up first

Name: Reset survey

Goal: Reset the questions in the survey and the database of surveyor.

Actors: Administrator

Precondition: Login to administrator and survey application have to be set up first

List of classes

StudentSurveySystem – Driver program which contain main method.

Administrator – Contain methods related to administrator such as administrator password, administrator login etc.

Student – Contain method that print the survey and take input so that student can take the survey

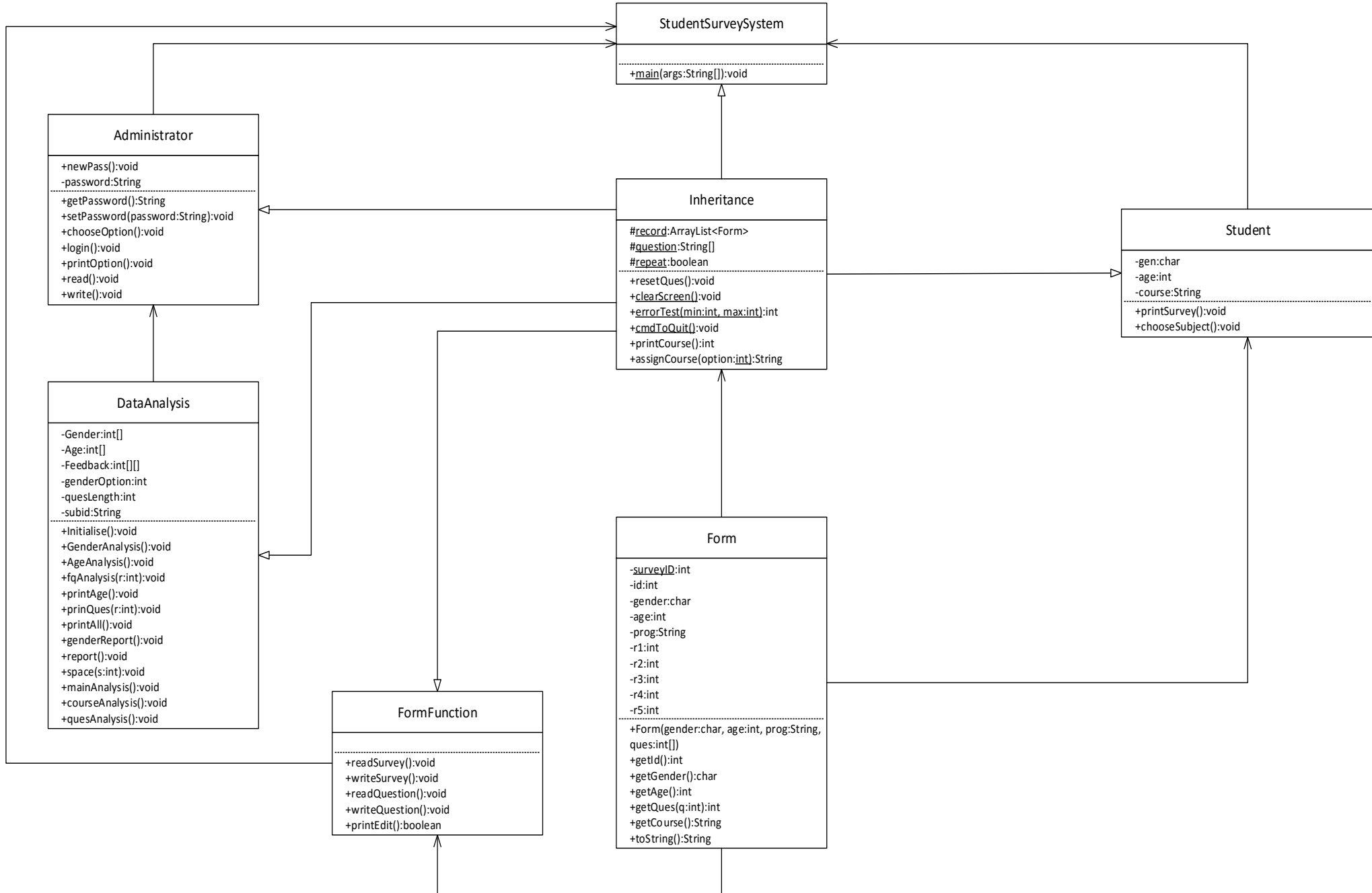
FormFunction – Contain some method for functions that deal with survey.

Form – Contain the feedback that student enter for the survey.

DataAnalysis – Contain methods that allows the feedback of students to be analyzed.

Inheritance – Contain some variables and method that will be reused in the classes to be inherited using this class to reduce redundancy.

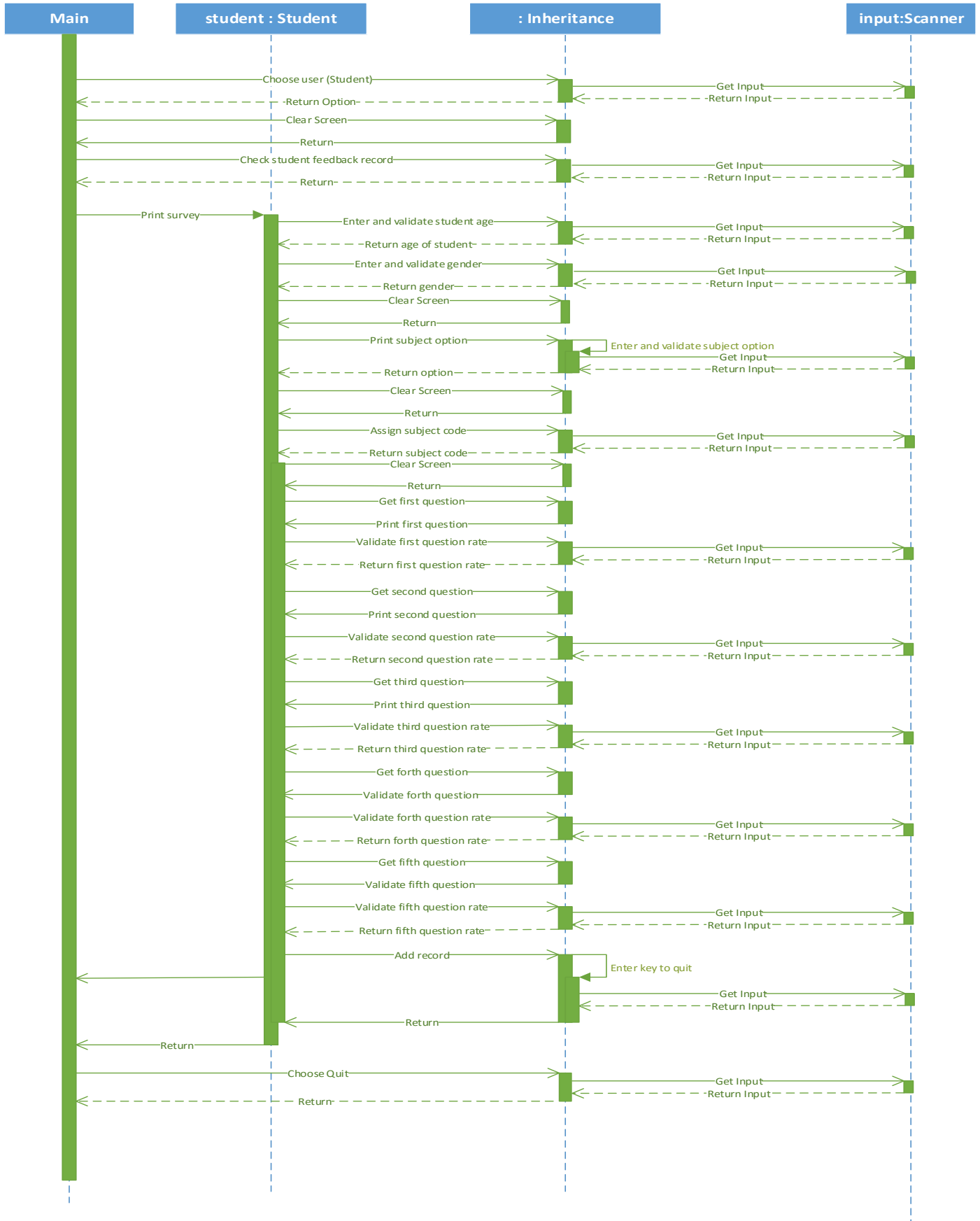
Class Relationship Diagram

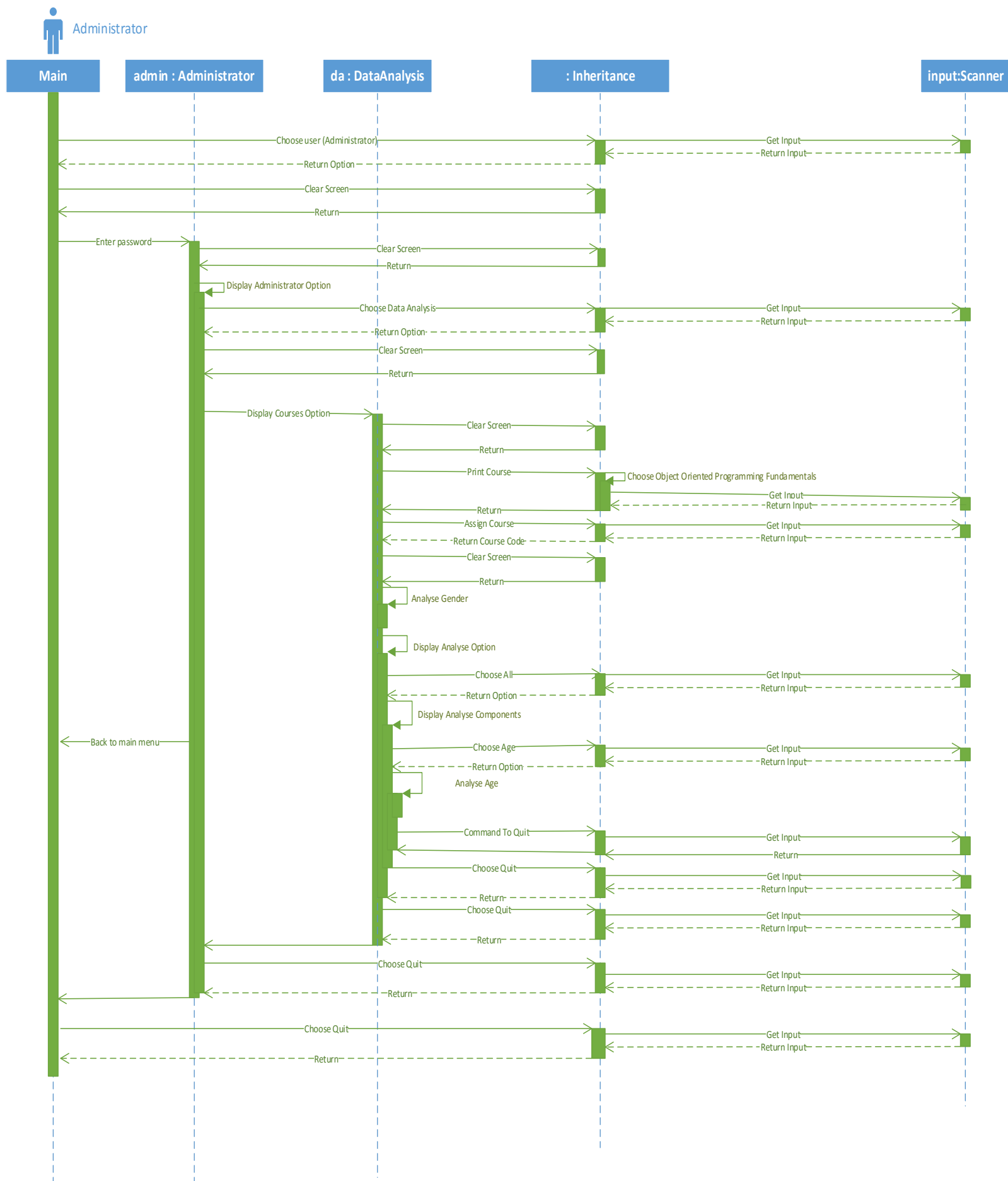


Sequence Diagram



Student





How the data structures or method or technique works to achieve the project functionalities?

Driver Program (StudentSurveySystem.java)

This is the driver program which contains the main method and extends Inheritance class. The main method contains all the instructions required for the application to work correctly. In this method, object instances for formFunction class and Administrator class are created. This is to allow the access of methods in these classes. In the beginning of the application, object instance of formFunction class, formFunction, is created. It is then used to call readQuestion() and readSurvey() methods in formFunction class to read the survey questions and feedbacks from their respective text files, question.txt and database.txt. These two text files will contain all questions edited and feedbacks given by students. Next, object instance of Administrator class, admin, is created. It then calls the read() method to read in the administrator password to Administrator class which will be used later. The read() method also makes sure that the program is activated by administrator first by asking administrator to set password in the first run.

The application then prints out option menu for users to input their status. To ask for input, errorTest(0, 2) method which is in Inheritance class is called to ensure correct input format and range. The first argument, 0, represents the minimum of the range for the option and second argument, 2, represent the maximum of the range of for the option. Then, clearScreen() method which is in the Inheritance class is called to clear the console screen. After getting the input from user, a switch statement is used to check the option entered by the user so that appropriate task and service can be performed.

In the switch statement, if the user enters 1, which represents administrator, login() method in Administrator class is called for administrator login and administrator functions.

If the user enters 2, which represents student, an object instance of Student class, student, is created. chooseSubject() method from Student class is then called for printing survey form. The last instruction of case 2 will then be break. These two instructions are looped until repeat is false.

After exiting the switch statement, repeat is set to the boolean where user option not equals to 0. By using this statement, if the user option is equals to 0, repeat will be set to false, control will quit the loop; if the user's option is not equals to 0, repeat will be set to true, the control will return to printing user option menu and start again.

The last instruction in the driver program will be calling writeSurvey() method in formFunction class. This method will write the feedbacks obtained to database.txt text file and the program ended.

DataAnalysis class (DataAnalysis.java)

This class contains all the methods required to carry out tasks for data analysis by administrator. It extends Inheritance class so that some of the common array list, array, variables and methods in Inheritance class can be used.

In this class, 2 private one-dimensional integer array (Gender[], Age[]), 1 private two-dimensional integer array(Feedback[][]), 2 private integer instance variable (genderOption, quesLength) and 1 private String instance variable (subid) is declared. Gender[] with array length 2 is used to store the number of male and female students that took the survey. Age[] with array length 11 which stores the number of students in different age range. Feedback[][] is used to store the number of students that choose specific rating (1 to 5) for each question. The genderOption instance variable stores the option that is chosen by administrator according to the menu that determines gender to analyse. quesLength instance variable stores the number of alphabet in longest question and subid instance variable used to store the subject code that is chosen by administrator.

Initialise() method

This method is used to initialize the arrays. Two for loop is used in this method to initialize the Age array and Feedback array so that it can count the number of students involved using accumulation method.

GenderAnalysis() method

This method is used to analyse the number of male and female student for a specific subject. The first two instruction initialize the Gender array to 0 so that accumulation can be used to count number of students for each gender. Then, an enhanced for loop is used to loop through the record array list which contain the survey of each student. An if statement with condition the course matches the subject code, subid is used. In the if statement, there is another if...else statement. It is used to check the gender of the students and increase the Gender array accordingly.

AgeAnalysis() method

This method is used to analyse the age range of students. A switch statement is used depend on the genderOption variable. If the genderOption is 1, meaning that administrator chose to analyse the age for male students. Therefore, in case 1, there is an enhanced for loop. In the enhanced for loop, there is an if statement. When looping through each survey form from record array list, if the character that getGender() method returns is equals to "M" and the course that getCourse() method returns is equals to the subject code administrator chosen, subid, the Age array with index ((int) ((x.getAge()) / 10) will be pre-incremented. The last instruction for case 1 will be break the switch.

((int) ((x.getAge()) / 10) is actually a formula to increase performance of the accumulation. It first get the age of the student using getAge() method, then it divides the age by 10. The result will be a float value. By using explicit casting, the float value will then be converted to integer which will represents which age range the student will be in. For example, if the getAge() method returns 67, by dividing it, it will becomes 6.7. After explicit casting, it then becomes 6, which means that the index is 6 and the age range will be 60 to 69.

For case 2 in the switch statement, the enhanced for loop will be the same except the `getGender()` method returned character is compared with character "F", meaning that only female students will be analysed.

For case 3 in switch statement, the enhanced for loop same as case 1 and 2 except no more gender comparison as all will be analysed according to the course of student.

fqAnalysis(int r) method

Due to the function to analyse different gender separately, this method also uses switch statement which will perform its task based on different gender or all students. In this method, a parameter `r` is set to receive the passed in value. This parameter `r` will receive question no to be analysed. As stated earlier, there are only five questions, so if `r` is 0, this means that it will analyse the first question according to the gender or all.

The switch statement depends on `genderOption`. In case 1, male students' feedbacks will be analysed. So an enhanced for loop is used to loop through all the forms in record array list. An if statement is placed in this enhanced for loop. If the return character of `getGender()` method from Form class is 'M', which means is male, and the course that `getCourse()` method returns is equals to the subject code administrator chosen, `subid`, pre-increment the Feedback array with row index as `r` and column index as integer is obtained by calling `getQues(r)` method from Form class which will return the rating feedback of question `r` for that male student. The last instruction in the case will then be `break` the switch. Even though this way is a bit confusing, it can reduce redundancy by preventing creation of many analysis methods which is almost the same.

In case 2, the instructions are similar to case 1 but the return value of `getGender()` method is compared with 'F', which indicates only female will be considered in analysis.

In case 3, all the students are considered. The instructions are similar to case 1 and 2 but no gender comparison will be done. It will consider all students without limiting the gender.

printAge() method

This method is used to print out the results of analysed age range. It first call `AgeAnalysis()` method to analyse the age range then it prints out the age range and its respective number of students involved in the age range. 'void' is used in the first line of the method to indicate that this method will not return anything.

printQues(int r) method

This method is used to print out the analysed feedback for a specific question. It receives the question in parameter `r`. First, it calls `fqAnalysis(r)` method and passed in the question to be analysed, the method will analyse and store the results in Feedback two-dimensional array. The Feedback array with row index as `r` and column index `i` will be displayed through for loop.

printAll() method

This method prints all the survey feedback information according to gender or all. First, the header is printed. Then a switch statement is used depending on `genderOption`. In case 1, male

students' feedbacks will be analysed. So an enhanced for loop is used to loop through all the forms in record array list. An if statement is placed in this enhanced for loop. If the return character of `getGender()` method from Form class is 'M', which means is male, and the course that `getCourse()` method returns is equals to the subject code administrator chosen, `subid`, print the returned integer from `getId()` method and concatenate with `x`, the form. The reason of using `getId()` method when printing is because the Form class survey id information is not included in the `toString()` method. The last instruction for case 1 will be break statement.

In case 2, the instructions are similar to case 1. The only difference is that the returned character of `getGender()` method is compared with 'F' to consider only female students.

In case 3, instructions are similar to case 1 and 2 except if statement only contain course comparison and not gender comparison as it involves all the students.

generateReport() method

This method is used to generate report which will display all information. This will allow administrator to print the report to paper easily. In `generateReport()` method, for male, female and all students, the number of students are displayed. A for loop is then used to find for longest question and store the number alphabet of longest question into `quesLength`. The `genderOption` variable is set to 1, 2 and 3 at different lines when analyzing the information. After setting the `genderOption`, `record()` method is called so that the information will be displayed.

report() method

This method is used to analyse and print result of analysis. It first set all elements in Age array and Feedback array to 0 by calling `Initialise()` method. Then, `printAge()` method is called to print the age analysis. `space(quesLength)` method is called to print spaces for alignment. Next, a for loop is used to analyse the feedback of each question by calling `fqAnalysis(i)` method in the for loop. Lastly, a table that shows the Feedback array stored values is created using nested for loop.

space(int s)

This method is used to print spaces for content alignment in table. The argument passed in is the number of spaces required to print. A for loop is used to print the spaces according to the argument.

mainAnalysis() method

This method is used to display the menu of data analysis for a specific subject and input the administrator's option. The instructions are placed in a `do...while` loop until the repeat variable is set as false. The screen is first cleared by calling `clearScreen()` method in Inheritance class. The menu for data analysis is then displayed on screen. Administrator input is carried out by calling `errorTest(0, 4)` method in Inheritance class. The first argument, 0, represents the minimum option range and the second argument, 4, represents the maximum option range. The returned value of `errorTest(0, 4)` method is placed into `genderOption` variable. A test is carried out using `if...else` statement to check whether to repeat from beginning of method. If the `genderOption` which contain administrator input is not equals to 0, meaning that it will be between 1 to 4, `quesAnalysis()` method. The repeat variable is then set to

true. This is to allow administrator to go back to this menu again after entering into deeper menu; else, the genderOption should be equals to 0 and repeat will be set to false, meaning the loop will be quitted.

courseAnalysis() method

This method is used to ask administrator which course to analyse. The instructions are placed in a do...while statement with repeat as condition so that it can return to previous menu when repeat is false. cleanScreen() is first called to clear the screen. printCourse() method is called and the value is returned to courseOption integer variable. This is method is used to print courses options for administrator to choose. An if...else statement is used with condition courseOption not equals to 0. This is to check whether courseOption is 0. If courseOption is not 0, assignCourse(courseOption) method is called and value is returned to subid instance variable. This is to assign subject code to subid. GenderAnalysis() and mainAnalysis() methods are then called to analyse gender and carry out analysis for that specific subject. repeat variable is then set to true. If courseOption is 0, repeat is set to false so that the loop will be exited.

quesAnalysis() method

This method is used to display different analysis options after choosing gender to be analysed. All the instruction is placed in a do...while loop to ensure administrator only can quit when enter 0. The first statement calls clearScreen() method from Inheritance class. Then, Initialise() method is called to set all elements in Age and Feedback array to 0.

Next, if...else statements are used to determine set of instructions to be used. If genderOption is between 1 and 3 inclusive, it will first print the menu for components to analysed. Switch statements are used in printing menu to change words such as male, female and all according to the genderOption. The errorTest(0, 7) method is called and minimum option range, 0, and maximum option range, 7, is passed into this method. Administrator input value will then return to variable option2. clearScreen() method is called again to clear the screen.

A switch statement depending on option2 is used. In case 1, printAge() will be called to analyse the age range and display it. cmdToQuit() method is then called to allow administrator to enter any key and press enter to return to previous menu.

In case 2, the administrator request for Question 1 analysis. printQues(0) will be called to analyse Question 1's feedbacks. cmdToQuit() method is then called to allow administrator to enter any key and press enter to return to previous menu.

In case 3, the administrator request for Question 2 analysis. printQues(1) will be called to analyse Question 2's feedbacks. cmdToQuit() method is then called to allow administrator to enter any key and press enter to return to previous menu.

In case 4, the administrator request for Question 3 analysis. printQues(2) will be called to analyse Question 3's feedbacks. cmdToQuit() method is then called to allow administrator to enter any key and press enter to return to previous menu.

In case 5, the administrator request for Question 4 analysis. `printQues(3)` will be called to analyse Question 4's feedbacks. `cmdToQuit()` method is then called to allow administrator to enter any key and press enter to return to previous menu.

In case 6, the administrator request for Question 5 analysis. `printQues(4)` will be called to analyse Question 5's feedbacks. `cmdToQuit()` method is then called to allow administrator to enter any key and press enter to return to previous menu.

In case 7, the administrator request to print all the information in survey for student involved. `printAll()` will be called to print those survey information. `cmdToQuit()` method is then called to allow administrator to enter any key and press enter to return to previous menu.

By default, break statement will be executed.

After exiting the switch statement, `clearScreen()` is called to clear the screen. Variable `repeat` is set to `option2` not equals to 0 so that `repeat` will be set to true when `option2` is number other than 2 and it will be set as false when `option2` is 0 which administrator request to quit.

Back to the if...else statements, if `option1` is not between 1 and 3, the control will go to the next else if to check whether `option1` is equals to 4. If `option1` equals to 4, `generateReport()` method is called to generate a report of all analysis. `cmdToQuit()` method is then called to allow administrator to enter any key and press enter to return to previous menu.

Back to the if...else statements again, if `option1` is not between 1 and 4, then `option1` should be equals to 0. Therefore, break statement is called to break the loop and go back to the gender analysis choosing menu.

FormFunction class (FormFunction.java)

This class is used contains all the functions that deal with survey management, such as reading survey feedback from database.txt text file, writing survey feedback to database.txt text file etc. This class extends Inheritance class, meaning the methods and protected variables in Inheritance class can be used in this class.

readSurvey() method

This method is used to read survey from database.txt text file. File object is first created to link to database.txt text file. An if statement is used to test whether the text file exist before continue. If the text file does not exist, new file with name 'database' text file will be created and the method will be exited. If the file exist after testing, a new Scanner object is created to allow reading in database.txt. a while loop with hasNext() method in scanner object created is used to ensure that content exist in database.txt after each read. The gender, age and course of student is first read, then the 5 questions' feedbacks are read into an array with size of 5 using a while loop with hasNextInt() method as condition. These information is then added to the array list to create a new Form object in record array list. After all contents in database.txt are read into record array list, the Scanner object the call close() method to close the function of Scanner which input information from text file.

writeSurvey() method

This method is used to write the survey form information into database.txt. This method will be called at the end of the program.

A Print Writer object is created and connected with database.txt text file. An enhanced for loop is then used to loop through the record array list and print the toString() method string in each Form into database.txt using Print Writer's println() method. After completing the enhanced for loop, the Print Writer is closed by calling the close() method in Print Writer object created.

readQuestion() method

Since the five questions in the student survey application can be edited, these five questions should have a text file that store these questions. This method is used to read in the five questions from question.txt file.

A File object is created to link to question.txt file. Then, an if statements is used to test whether this file exists. If this file does not exists, the file is created and resetQues() in Inheritance class is called to set the five question to default strings (Question 1, Question 2, Question 3, Question 4, Question 5). writeQues() method is then called to write these questions into question.txt. If the file exists, a Scanner object will be created and link with the File object created preciously. Then, a counter, i, is declared. A while loop is created to read in each line of question into question array in Inheritance class. While next line exists, it will be read into question array with index set in counter i. The counter i is post-incremented after each loop. After reading in all the five questions into question array, close() method in the Scanner object is called.

writeQues() method

This method is used to write questions into question.txt file. A Print Writer object is created and link to question.txt file. A for loop is used to loop through question array and write the five questions into question.txt using println() method in Print Writer object. After output all the questions into question.txt file, close() method in Print Writer object is called.

printEdit() method

This method is used to edit the five questions for the student survey. It is declared as boolean and will return a boolean. Menu will be printed out to ask which question to be changed. When printing the menu, a for loop is used to print out the questions in question array so that administrator will know which question to edit and make appropriate changes.

errorTest(0, question.length) method in Inheritance class is called with passing first argument, 0, as the minimum of option range and second argument, length of question array as maximum of option range. The value return from the method is then store into option variable. A Scanner object is created to read input from user later. Then, an if statement is used to test whether the option is equals to 0. If option is not equals to 0, it will ask user to enter the new question and the Scanner object created previously will be used to read in the question and store it into its respective question array element position. Then, the method will return true. If the is equals to 0, the if statement above will be ignored and this method will straightaway return false.

Administrator class (Administrator.java)

This is a class that stores instance variables and methods related to administrator. It extends Inheritance class so that the question array, record array list and some other methods can be used in this class. It contains one private String instance variable, password, which stores the password of administrator.

getPassword() method

This is a getter method which return the password so that it can be used for functions in other classes.

setPassword(String password) method

This is a setter method sets the password in Administrator class to the password passed in from arguments.

chooseOption() method

This method prints the menu for administrator's operations and allow administrator to choose functions that they want. printOption() method is first called to print the administrator menu. errorTest(0, 4) method is then called to allow administrator input option. First argument, 0, which represents the minimum option range and second argument, 4, represents the maximum option range are passed in to ensure that the allowed range for option will only be between 0 and 4. The option input by administrator is then returned and store in adminOption variable.

A switch statement is used to determine the instructions to carry out according to administrator's option. In case 1, the administrator chose to carry out data analysis. A new DataAnalysis object instance, da, is created. This instance of object is then used to call chooseAnalysis() method in DataAnalysis class. Last instruction for case 1 will be break statement.

In case 2, administrator chose to edit the survey question. An instance of object for FormFunction class is created. A do...while loop is used for question edit. clearScreen() method from Inheritance class is called to clear the screen after each loop. printEdit() method from FormFunction class is called to allow question manipulation in the condition of do...while loop. The method will then return a boolean to determine whether the loop should be break. After exiting the loop, this means that questions are edited. writeQuestion() method from FormFunction class is called to write the edited question to question.txt for future use. Last instruction for case 2 will be break statement.

In case 3, administrator chose to reset the survey. clear() method from record array list in Inheritance class is called to clear all the Forms in record array list. Last instruction for case 3 will be break statement.

In case 4, administrator chose to change administrator password. clearScreen() method is called to clear the screen. newPass() method is called to allow setting new password. Last instruction for case 4 will be break statement.

After the switch statement, adminOption variable is compared with 0. If adminOption variable is not 0, repeat will be set as true. If adminOption variable is 0, repeat variable will be set as false.

login() method

This method is used to let administrator login and allow administrator functions to be accessible. All the instructions are placed in a do...while loop with repeat as condition. An instance object of Scanner is first created to be used for password input later. `nextLine()` method from the Scanner object is called to allow administrator to input password and store it into a String variable. An if...else statement is then used to return the status. If the password typed by administrator is same as the private password instance variable in this class, a do...while loop is used to loop `chooseOption()` method until repeat is false. This is for the administrator functions; else, if the typed password is "quit", repeat is set to false. If the password is uncorrect and also not equals to "quit", repeat is set to true. This is to allow login request to loop many times as administrator might typed the wrong password accidentally.

printOption() method

This method is used to print out the functions option for administrator. `clearScreen()` method is called to clear the screen. The options for administrator are then printed using print statements.

read() method

This method is used to read the administrator password from `adminPass.txt` file and check whether the student survey application is first run. A File object is first created and linked to `adminPass.txt` file. An instance object of Scanner is created to allow input of password later. An if statement is used to check whether `adminPass.txt` file exists. If file does not exists or file is empty, meaning this is first run of student survey application. `createNewFile()` method from the File object created previously is called to create `adminPass.txt` file. A do...while loop is used to ask administrator for administrator password setting. The password is saved in a String variable. The condition of the loop is checking whether the password entered by user matches "quit". If it matches "quit", it will loop again as "quit" is the sentinel value which allows user to quit from logging in. `setPassword(pass)` method is called to set private variable password as the password passed in the argument. `write()` method is called to write the password into `adminPass.txt`. If the file exists, a new Scanner object instance is created and link to the File created previously. The first line in `adminPass.txt` will then be read into the private password variable which will be used to allow password checking.

write() method

This method is used to write password into `adminPass.txt` file for future use. A Print Writer object instance is created and link to `adminPass.txt` file. The private password variable is then printed into `adminPass.txt` by calling `println(this.password)` method and passing in password variable as argument. `close()` method of Print Writer object is then called in the last statement.

newPass() method

This method is used to allow changing of administrator password. A Scanner object instance is created to input password. 2 variables which store new password and repeated password is declared. A do...while loop is used with condition new password not equals to repeated password so that when repeated password different from new password, loop will continue. In the do..while loop, administrator

is asked to enter new password. `nextLine()` method in the Scanner object created is called to read in new password. Administrator is then asked to repeat the new password. `nextLine()` method in the Scanner object created is called to repeat new password. Loop will continue while new password not equals to old password or new password equals to "quit". This is because if new password not equals to repeated password the loop should repeat. This is the same if new password equals to "quit". After exiting the loop, `setPassword(newPass)` method is called with new password as argument to set the new password into the password private variable. `write()` method is called to write the new password into `adminPass.txt` file.

Student class (Student.java)

This class contains method that allow students to take survey. It extends Inheritance class as it needs to use the repeat variable for looping control. There are three private instance variable in this class. These instance variables are gen(private character instance variable), age(private integer instance variable) and course(private String instance variable).

printSurvey() method

This method is use to print survey and get feedbacks from students. The subject code and survey id is first printed. A new integer array with length of 5 is declared to be used to store student's questions feedback. A for loop which loops five times with nested do...while loop is then used to print the question from the question array in Inheritance class and ask for input from student using errorTest(1, 5) method. The questions require student to rate between 1 to 5. Therefore, 1 and 5 is passed into the method as minimum rate and maximum rate for feedback.

A new Form object is then created to add the Form into record array list which store student's feedback. cmdToQuit() method in Inheritance class is called.

chooseSubject() method

This method is used to get gender, age and choose subject for survey. An instance of object of Scanner, input, is created to get input from student later.

A do...while loop is used to make sure student enter the correct character, which is either 'M'(Male) or 'F'(Female) only. Due to the limitation of Java in getting a single character only, when student enter a line, only the first character of the line will be taken and store into char variable declared earlier. This variable is the compared with both 'M' and 'F' character using OR operator in an if...else statement to ensure that the character is valid. If the user entered character 'M' or 'F', repeat will be set to false as the gender is valid. If is not 'M' or 'F', repeat will be set to true, so that student will be asked to input valid gender character again.

After exiting the do...while loop for gender, another do...while loopis entered for age. In the do...while loop, the application will ask student for his/her age and errorTest(1, 100) method from Inheritance class is called to allow student input their age. First argument, 1, represent the minimum valid age for the student, and second argument, 100, represent the maximum valid age for student are passed into the method. The purpose of do...while loop for student age input is to allow "How old are you?" to be printed if invalid input occurs.

A do...while loop is entered for subject choosing. clearScreen() method is first called to clear the console screen. Subjects menu is then printed and student input is obtained by calling printCourse() method in Inheritance class which will return the subject option to stuSub integer variable.. An if statement is used with condition stuSub is not equals to 0. If stuSub is not 0, instance variable course gets the return value of assignCourse(stuSub). printSurvey() method is then called to print survey for students and get feedbacks. The repeat variable is set to true. If stuSub is 0, repeat is set to false to exit the loop.

Inheritance class (Inheritance.java)

This class is used to store common variable, array, array list, and methods that are used commonly in more than one class. It is the super class and Administrator, DataAnalysis, FormFunction, Student and StudentSurveySystem are subclasses of it. These subclasses can access its protected variables and methods without needing to create object instance.

A protected static Form class array list is created to store feedback of students. A protected static String array with length of 5 is created to store the questions of survey. A protected static boolean variable is created to determine store true or false to determine whether the loop should continue or exit.

resetQues() method

This method is used to reset the questions of survey to default string. A for loop is used which allow 5 times looping to loop through the question array and insert the default question string.

clearScreen() method

This method is used to clear the screen of the console. It actually uses key on keyboard to clear the screen. A robot object is created initially. It is then used to call keypress() method in it two times to press CTRL and L key and then keyRelease() method to release CTRL and L key. sleep(50) method is called from Thread to allow 50 milliseconds pause for the program to prevent it from accidentally clear next instruction that will be printed on the screen. This method is applicable only to NetBean IDE.

errorTest(int min, int max) method

This method is used in most of the subclass to allow users enter option or number input. It can test the validity of input and carry out appropriate actions. Two arguments are passed into this method when calling it. The first argument is the minimum range option and the second argument is the maximum range option.

A Scanner object instance is created to get the input integer from user later. An integer variable is declared for storing user input later. A do...while loop is used to loop multiple times until the user input a valid option or number. In the do...while loop, an if statement is first carried out to test whether the minimum option is equals to 0. If it is equals to 0, "Enter Option : " will be printed. This is because this method is used in student survey form as well. These words are not suitable to appear in student survey form. try...catch statement is then used to allow certain instruction statements to be carried out for error handling. In the try block, user is asked to input an integer by calling nextInt() method from the Scanner object created previously. nextLine() method from Scanner object is called to flush the input (absorb ENTER) to prevent errors. An if...else statement is the used to check whether user input range is in the valid range. If the option entered by user is between minimum range and maximum range, repeat is set to false; else, the repeat will be set to true and an if...else statement is carried out. If the minimum option is not equals to 0, break the loop; else, print error message. This is because the minimum option range of 0 is only applied to administrator function. The catch block will be the instructions that will carry out if any type of errors is thrown. It will do the same instructions as the previous else block except that the nextLine() method from Scanner object is called to flush the input (absorb ENTER) to prevent errors. The condition for while will be the variable repeat. It will continue looping if repeat is true.

After exiting the do...while loop, the user input option is confirmed a valid option. It is the returned to the subclass and store into an option variable.

cmdToQuit() method

This method is called to allow use to enter any key to go back. A Scanner object instance is created to receive any value. The application will ask user to insert any key. The user then press ENTER and the control will go back.

printCourse() method

This method printthe subject menu and get input from user. After printing the subject menu, errorTest(0, 4) is called for user input and user's option is returned to option integer variable. First argument, 0, represent the minimum option range and second argument, 4, represent maximum option range. The option variable is then returned to the subclass method that call this method.

assignCourse(int option)

This method is used to return the subject code to the method that call it. option parameter stores the option of user for subject. A switch statement is used to return the subject code. In case 1, Object Oriented Programming Fundamentals is chosen and "PRG1203" will be returned. In case 2, Computer Organisation is chosen and " CSC1202" will be returned. In case 3, Database Fundamentals is chosen and " SEG1201" will be returned. In case 4, Web Fundamentals is chosen and " WEB1201" will be returned.

Form class (Form.java)

This class acts like a student survey form. Each student survey feedback information will be save into one Form object. This class is also used to create an array list which can create many Form object for undefined student number.

In this class there are 1 private static variable, 4 private final variables and 1 private final array. The private static integer variable, surveyID, is used to assign survey form id. A private final integer variable, id, is declared to store the id of survey form. A private final character, gender, variable is declared to store the gender of student. A private final integer variable, age, is declared to store the age of student. A private final String variabl, course is declared to store the course of the student. A private final integer array, ques, is declared to the feedback ratings of 5 questions by student.

Form(char gender, int age, String prog, int[] ques) constructor

This is an overloaded constructor to assign the student survey information into the form. The id of the form is set to pre-incrementing the static surveyID. The gender instance variable is set to the value of passed in gender. The age instance variable is set to the value of passed in age. The course instance variable is set to the value of passed in prog. The ques instance array variable is set to the array values of passed in ques.

getId() method

This is a getter method for student survey form id which returns the form id.

getGender() method

This is a getter method for student gender which returns the gender of student.

getAge() method

This is a getter method for student age which returns the age of student.

getQues(int q) method

This is a getter method for student question feedback ratings by returning the feedback for array index passed in.

toString() method

This is a toSting method which allows the information of the Form to be returned as a String.

Overall

In this application, some of the most commonly used are do...while loop with repeat boolean variable as condition and also `errorTest(int min, int max)` method to check for validity of input values. The do...while loop with repeat boolean variable is useful as it allows control to be in a loop and repeat the instructions many times until user asked to exit. This makes the program to work in many layers. The `errorTest(int min, int max)` method contains try...catch statement that will prevent user from entering invalid input. This method is used by many subclasses of Inheritance class as it can prevent errors and infinite loop if user enter an invalid value.

Screenshot of Student Survey Application

```
Set your administrator password:
```

Figure 1 : Set Administrator for Survey Set Up (Administrator)

```
Who are you?
1 Administrator
2 Student
0 Exit system
Enter Option :
```

Figure 2 : User Main Menu (Administrator and Student)

```
Please enter administrator password. (Enter 'quit' to return)
|
```

Figure 3 : Administrator Login (Administrator)

```
Welcome administrator. What do you want to do?
1 Data Analysis
2 Edit Survey Questions
3 Reset Survey
4 Change Password
0 Quit
Enter Option :
```

Figure 4 : Administrator Menu (Administrator)

```
Course Analysis (Total = 20 student feedback(s))
1 PRG1203 - Object Oriented Programming Fundamentals
2 CSC1202 - Computer Organisation
3 SEG1201 - Database Fundamentals
4 WEB1201 - Web Fundamentals
0 Exit
Enter Option : |
```

Figure 5 : Administrator Course Choosing for Data Analysis (Administrator)

```

Data Analysis for PRG1203
1 Male
2 Female
3 All
4 Generate Report
0 Exit
Enter Option : |

```

Figure 6 : Type of Data Analysis for Chosen Course (Administrator)

```

All Male students' Feedback(s) Analysis (Total Male student = 5)
1 Age
2 Question 1 : The course objectives were clear.
3 Question 2 : The course procedures and assignments support course objectives.
4 Question 3 : The amount of reading you were asked to do was appropriate.
5 Question 4 : The amount of writing or other class work you were asked to do was enough.
6 Question 5 : What overall rating would you give the course?
7 List All Males' Feedback(s)
0 Exit
Enter Option : |

```

Figure 7 : Data Analysis of Male Students for Chosen Course (Administrator)

```

All Female students' Feedback(s) Analysis (Total Female student = 2)
1 Age
2 Question 1 : The course objectives were clear.
3 Question 2 : The course procedures and assignments support course objectives.
4 Question 3 : The amount of reading you were asked to do was appropriate.
5 Question 4 : The amount of writing or other class work you were asked to do was enough.
6 Question 5 : What overall rating would you give the course?
7 List All Females' Feedback(s)
0 Exit
Enter Option : |

```

Figure 8 : Data Analysis of Female Students for Chosen Course (Administrator)

```

All Feedback(s) Analysis (Total = 7)
1 Age
2 Question 1 : The course objectives were clear.
3 Question 2 : The course procedures and assignments support course objectives.
4 Question 3 : The amount of reading you were asked to do was appropriate.
5 Question 4 : The amount of writing or other class work you were asked to do was enough.
6 Question 5 : What overall rating would you give the course?
7 List All Feedback(s)
0 Exit
Enter Option : |

```

Figure 9 : Data Analysis of All Students for Chosen Course (Administrator)

Overall Analysis for PRG1203

MALE

Number of male student : 5

Feedback from male student

Age 1-9 : 0

Age 10-19 : 1

Age 20-29 : 1

Age 30-39 : 1

Age 40-49 : 0

Age 50-59 : 0

Age 60-69 : 1

Age 70-79 : 1

Age 80-89 : 0

Age 90-99 : 0

Age 100 : 0

	R1	R2	R3	R4	R5
The course objectives were clear.	0	0	0	1	4
The course procedures and assignments support course objectives.	0	0	0	2	3
The amount of reading you were asked to do was appropriate.	0	0	0	2	3
The amount of writing or other class work you were asked to do was enough.	0	0	0	2	3
What overall rating would you give the course?	0	0	0	3	2

FEMALE

Number of female student : 2

Feedback from female student

Age 1-9 : 0

Age 10-19 : 0

Age 20-29 : 0

Age 30-39 : 0

Age 40-49 : 1

Age 50-59 : 1

Age 60-69 : 0

Age 70-79 : 0

Age 80-89 : 0

Age 90-99 : 0

Age 100 : 0

	R1	R2	R3	R4	R5
The course objectives were clear.	0	0	0	2	0
The course procedures and assignments support course objectives.	0	0	0	0	2
The amount of reading you were asked to do was appropriate.	0	0	0	1	1
The amount of writing or other class work you were asked to do was enough.	0	0	0	0	2
What overall rating would you give the course?	0	0	0	1	1

ALL

Number of students : 7

Feedback from all student

Age 1-9 : 0

Age 10-19 : 1

Age 20-29 : 1

Age 30-39 : 1

Age 40-49 : 1

Age 50-59 : 1

Age 60-69 : 1

Age 70-79 : 1

Age 80-89 : 0

Age 90-99 : 0

Age 100 : 0

	R1	R2	R3	R4	R5
The course objectives were clear.	0	0	0	3	4
The course procedures and assignments support course objectives.	0	0	0	2	5
The amount of reading you were asked to do was appropriate.	0	0	0	3	4
The amount of writing or other class work you were asked to do was enough.	0	0	0	2	5
What overall rating would you give the course?	0	0	0	4	3

Press any key and ENTER to quit: |

Figure 10 : Report Generated for Chosen Course (Administrator)

```
Which question do you want to edit?
1 The course objectives were clear.
2 The course procedures and assignments support course objectives.
3 The amount of reading you were asked to do was appropriate.
4 The amount of writing or other class work you were asked to do was enough.
5 What overall rating would you give the course?
0 Quit
Enter Option : |
```

Figure 11 : Choose Survey Question to Edit (Administrator)

```
Which question do you want to edit?
1 The course objectives were clear.
2 The course procedures and assignments support course objectives.
3 The amount of reading you were asked to do was appropriate.
4 The amount of writing or other class work you were asked to do was enough.
5 What overall rating would you give the course?
0 Quit
Enter Option : 1
Enter your question:
```

Figure 12 : Edit Chosen Survey Question (Administrator)

```
Enter your new password
1234
Repeat your new password
1234|
```

Figure 13 : Change Administrator Password (Administrator)

```
What is your gender? (M/F)
M

How old are you?
20
```

Figure 14 : Enter Student's Gender and Age (Student)

```
Subject
1 PRG1203 - Object Oriented Programming Fundamentals
2 CSC1202 - Computer Organisation
3 SEG1201 - Database Fundamentals
4 WEB1201 - Web Fundamentals
0 Exit
Enter Option :
```

Figure 15 : Choose Subject for Survey (Student)

Survey Subject : PRG1203

Your survey id is 1.

The course objectives were clear. (1 - 5 ONLY) : 1

The course procedures and assignments support course objectives. (1 - 5 ONLY) : 2

The amount of reading you were asked to do was appropriate. (1 - 5 ONLY) : 3

The amount of writing or other class work you were asked to do was enough. (1 - 5 ONLY) : 4

What overall rating would you give the course? (1 - 5 ONLY) : 5

Thank you :)

Press any key and ENTER to quit: |

Figure 16 : Student Survey Form (Student)

Testing report or set of validations you applied

After successfully creating this program, it is tested with a set of values that are typed into the database.txt. These values are used to test whether the reading in survey, writing survey, numbering survey id and data analysis working correctly. The values are as below:

M	18	PRG1203	4	5	5	5	4
F	30	SEG1201	3	2	1	5	4
M	27	PRG1203	5	5	4	5	5
F	67	CSC1202	5	2	5	4	5
F	41	SEG1201	3	3	2	5	5
M	39	PRG1203	5	4	5	4	4
M	51	CSC1202	3	4	5	2	4
M	32	SEG1201	3	2	3	4	3
F	45	PRG1203	4	5	4	5	5
F	62	CSC1202	5	5	5	4	5
M	77	PRG1203	5	4	5	4	4
F	15	CSC1202	1	2	3	4	5
M	66	PRG1203	5	5	4	5	5
M	43	SEG1201	3	4	4	5	5
F	57	PRG1203	4	5	5	5	4
F	46	WEB1201	4	5	3	4	5
M	45	WEB1201	3	3	3	3	3
M	85	CSC1202	1	2	3	4	5
F	34	WEB1201	3	3	4	5	3
M	56	PRG1201	3	5	4	5	5

*first column : Gender

*second column : Age

* Third column : Subject Code

*Forth, fifth, sixth, seventh and eighth column : First, second, third, forth and fifth question's feedback

However, due to the number of subjects, analysis will only be done on Object Oriented Programming Fundamentals, PRG1203. To make sure that the analysis is carrying out correctly in the code, this set of survey feedback values are analysed using Microsoft Excel 2016 as below.



To prove that the data analysis works correctly, the screenshot for data analysis for PRG 1203 is attached. (Comparison can be made between Excel Analysis and Student Survey Application Analysis)

```
Course Analysis (Total = 20 student feedback(s))
1 PRG1203 - Object Oriented Programming Fundamentals
2 CSC1202 - Computer Organisation
3 SEG1201 - Database Fundamentals
4 WEB1201 - Web Fundamentals
0 Exit
Enter Option : 1|
```

Figure 17 : Choose PRG1203 as subject/course to analyse

```
Data Analysis for PRG1203
1 Male
2 Female
3 All
4 Generate Report
0 Exit
Enter Option : 3|
```

Figure 18 : Choose to analyse all PRG1203 students

```
All Feedback(s) Analysis (Total = 8)
1 Age
2 Question 1 : The course objectives were clear.
3 Question 2 : The course procedures and assignments support course objectives.
4 Question 3 : The amount of reading you were asked to do was appropriate.
5 Question 4 : The amount of writing or other class work you were asked to do was enough.
6 Question 5 : What overall rating would you give the course?
7 List All Feedback(s)
0 Exit
Enter Option : |
```

Figure 19 : Choose elements to analyse

```
Age 1-9      : 0
Age 10-19   : 1
Age 20-29   : 1
Age 30-39   : 1
Age 40-49   : 1
Age 50-59   : 2
Age 60-69   : 1
Age 70-79   : 1
Age 80-89   : 0
Age 90-99   : 0
Age 100     : 0
Press any key and ENTER to quit: |
```

Figure 20 : Age Analysis for all PRG1203 students

```
Data Analysis for Question 1
Rate 1: 0
Rate 2: 0
Rate 3: 1
Rate 4: 3
Rate 5: 4
Press any key and ENTER to quit: |
```

Figure 21 : Question 1 Analysis for all PRG1203 students

```
Data Analysis for Question 2
Rate 1: 0
Rate 2: 0
Rate 3: 0
Rate 4: 2
Rate 5: 6
Press any key and ENTER to quit: |
```

Figure 22 : Question 2 Analysis for all PRG1203 students

```
Data Analysis for Question 3
Rate 1: 0
Rate 2: 0
Rate 3: 0
Rate 4: 4
Rate 5: 4
Press any key and ENTER to quit: |
```

Figure 23 : Question 3 Analysis for all PRG1203 students

```
Data Analysis for Question 4
Rate 1: 0
Rate 2: 0
Rate 3: 0
Rate 4: 2
Rate 5: 6
Press any key and ENTER to quit: |
```

Figure 24 : Question 4 Analysis for all PRG1203 students

```
Data Analysis for Question 5
Rate 1: 0
Rate 2: 0
Rate 3: 0
Rate 4: 4
Rate 5: 4
Press any key and ENTER to quit: |
```

Figure 25 : Question 5 Analysis for all PRG1203 students

ID	Gender	Age	Course	Q1	Q2	Q3	Q4	Q5
1	M	18	PRG1203	4	5	5	5	4
3	M	27	PRG1203	5	5	4	5	5
6	M	39	PRG1203	5	4	5	4	4
9	F	45	PRG1203	4	5	4	5	5
11	M	77	PRG1203	5	4	5	4	4
13	M	66	PRG1203	5	5	4	5	5
15	F	57	PRG1203	4	5	5	5	4
20	M	56	PRG1203	3	5	4	5	5

Press any key and ENTER to quit: |

Figure 26 : All PRG1203 students' feedback information

After making sure the student survey application data analysis works correctly, validation of data is tested. This is to make sure that the application will loop the instructions until the user enter correct input format. It must make sure that student survey application will not crash and require force stop due to wrong format entered.

For input validation, do...while loop and try...catch statement is used. try...catch statement is placed inside do...while loop and repeat variable from Inheritance class will be used as condition for do...while loop.

```

Who are you?
1 Administrator
2 Student
0 Exit system
Enter Option : abc
Please enter option between 0 and 2 only
Enter Option : abc def ghi jkl mno
Please enter option between 0 and 2 only
Enter Option : -1
Please enter option between 0 and 2 only
Enter Option : 3
Please enter option between 0 and 2 only
Enter Option : (^&&* (^*% ^& *% ^(*^ *( *( ^ (
Please enter option between 0 and 2 only
Enter Option : 12 33 45 67 8 7
Please enter option between 0 and 2 only
Enter Option : |

```

Figure 27 : Input validation for user menu

Figure 27 shows input validation for user menu. In this menu, the valid format is integer with range of 0 to 2. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 0, number more than 2, symbols and multiple numbers. From Figure 27, it can be seen that the program will continue to ask for valid input unless valid input is given. (Valid number cannot be demonstrated here as clearScreen() method will be called)

```

Please enter administrator password. (Enter 'quit' to return)
12345
Please enter administrator password. (Enter 'quit' to return)
capeinv ieonrivo ern
Please enter administrator password. (Enter 'quit' to return)
^&*% &^^(*)
Please enter administrator password. (Enter 'quit' to return)
|

```

Figure 28 : Password checking for administrator

Figure 28 shows the password checking for administrator. The password set previously is 1234. From Figure 28, it can be seen that values other than 1234 will not work. To quit from entering password, the password must either be 1234 and direct to administrator menu or insert “quit” to return to user menu. (1234 and “quit” cannot be demonstrated here as clearScreen() method will be called)

```

Welcome administrator. What do you want to do?
1 Data Analysis
2 Edit Survey Questions
3 Reset Survey
4 Change Password
0 Quit
Enter Option : abc
Please enter option between 0 and 4 only
Enter Option : abc def ghi jkl mno
Please enter option between 0 and 4 only
Enter Option : -10
Please enter option between 0 and 4 only
Enter Option : 12
Please enter option between 0 and 4 only
Enter Option : &*(*&)( &(^#$%^ %$^&*& #@@%
Please enter option between 0 and 4 only
Enter Option : 12334 3 28 79847 59
Please enter option between 0 and 4 only
Enter Option : |

```

Figure 29 : Input validation for administrator menu

Figure 29 shows input validation for administrator menu. In this menu, the valid format is integer with range of 0 to 4. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 0, number more than 4, symbols and multiple numbers. From Figure 29, it can be seen that the program will continue to ask for valid input unless valid input is given. (Valid number cannot be demonstrated here as clearScreen() method will be called)

```

Course Analysis (Total = 20 student feedback(s))
1 PRG1203 - Object Oriented Programming Fundamentals
2 CSC1202 - Computer Organisation
3 SEG1201 - Database Fundamentals
4 WEB1201 - Web Fundamentals
0 Exit
Enter Option : abc
Please enter option between 0 and 4 only
Enter Option : abc def ghi jkl mno
Please enter option between 0 and 4 only
Enter Option : -34
Please enter option between 0 and 4 only
Enter Option : 34
Please enter option between 0 and 4 only
Enter Option : ^(*&) &(*^ ^ & ^ *^^ ^^ % & ^* ^* ^&
Please enter option between 0 and 4 only
Enter Option : 132 54 6575 76 8 98 7897
Please enter option between 0 and 4 only
Enter Option : |

```

Figure 30 : Input validation for subject/course menu

Figure 30 shows input validation for subject/course menu. In this menu, the valid format is integer with range of 0 to 4. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 0, number more than 4, symbols and multiple numbers. From Figure 30, it can be seen that the program will continue to ask for valid input unless valid input is given. (Valid number cannot be demonstrated here as clearScreen() method will be called)

```

Data Analysis for PRG1203
1 Male
2 Female
3 All
4 Generate Report
0 Exit
Enter Option : abc
Please enter option between 0 and 4 only
Enter Option : abc def ghi jkl mno
Please enter option between 0 and 4 only
Enter Option : -23
Please enter option between 0 and 4 only
Enter Option : 23
Please enter option between 0 and 4 only
Enter Option : *&^*%& ^ %$ &^ &*^ *( & *^ %^%&
Please enter option between 0 and 4 only
Enter Option : 8743534 45 739 753 79372 347
Please enter option between 0 and 4 only
Enter Option : |

```

Figure 31 : Input validation for data analysis menu

Figure 31 shows input validation for data analysis menu. In this menu, the valid format is integer with range of 0 to 4. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 0, number more than 4, symbols and multiple numbers. From Figure 30, it can be seen that the program will continue to ask for valid input unless valid input is given. (Valid number cannot be demonstrated here as clearScreen() method will be called)

```

All Feedback(s) Analysis (Total = 8)
1 Age
2 Question 1 : The course objectives were clear.
3 Question 2 : The course procedures and assignments support course objectives.
4 Question 3 : The amount of reading you were asked to do was appropriate.
5 Question 4 : The amount of writing or other class work you were asked to do was enough.
6 Question 5 : What overall rating would you give the course?
7 List All Feedback(s)
0 Exit
Enter Option : abc
Please enter option between 0 and 7 only
Enter Option : abc def ghi jkl mno
Please enter option between 0 and 7 only
Enter Option : 77
Please enter option between 0 and 7 only
Enter Option : -77
Please enter option between 0 and 7 only
Enter Option : (*&&^^^ %^&    ^% *&^ &^^ ^*& ^ &^( ^ ( ^ (*
Please enter option between 0 and 7 only
Enter Option : 124 7534 6456 464 645    6 5467847 64 4 4
Please enter option between 0 and 7 only
Enter Option : |

```

Figure 32 : Input validation for element analysis menu

Figure 32 shows input validation for element analysis menu. In this menu, the valid format is integer with range of 0 to 7. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 0, number more than 7, symbols and multiple numbers. From Figure 32, it can be seen that the program will continue to ask for valid input unless valid input is given. (Valid number cannot be demonstrated here as clearScreen() method will be called)

```

Which question do you want to edit?
1 The course objectives were clear.
2 The course procedures and assignments support course objectives.
3 The amount of reading you were asked to do was appropriate.
4 The amount of writing or other class work you were asked to do was enough.
5 What overall rating would you give the course?
0 Quit
Enter Option : abc
Please enter option between 0 and 5 only
Enter Option : abc def ghi jkl mno
Please enter option between 0 and 5 only
Enter Option : -10
Please enter option between 0 and 5 only
Enter Option : 23
Please enter option between 0 and 5 only
Enter Option : *(&(& ( &*(&*(&(*& *&*&)
Please enter option between 0 and 5 only
Enter Option : 123 456 756 7 8 889
Please enter option between 0 and 5 only
Enter Option : |

```

Figure 33 : Input validation for edit question menu

Figure 33 shows input validation for edit question menu. In this menu, the valid format is integer with range of 0 to 5. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 0, number more than 5, symbols and multiple numbers. From Figure 33, it can be seen that the program will continue to ask for valid input unless valid input is given. (Valid number cannot be demonstrated here as clearScreen() method will be called)


```

Enter your new password
quit
Repeat your new password
quit
Enter your new password
1234
Repeat your new password
12345
Enter your new password
abc abcd
Repeat your new password
abcabcd
Enter your new password
124 32534
Repeat your new password
12432534
Enter your new password
|

```

Figure 34 : Administor change password

Figure 34 shows password changing. From Figure 34, it can be seen that the sentinel value, “quit” cannot be set as password because it is the keyword to quit from entering password. The new password and repeated password must also be equals or else it will loop again until new password same as repeated password.

```

What is your gender? (M/F)
abc def ghi jkl mno
What is your gender? (M/F)
12
What is your gender? (M/F)
-80
What is your gender? (M/F)
123 432 4535 645
What is your gender? (M/F)
*^&^(^  &*&&* ^*  &*&
What is your gender? (M/F)
1243 4536547 56 58 678 67
What is your gender? (M/F)
M

How old are you?
abc
How old are you?
abc def ghi jkl mno
How old are you?
-50
How old are you?
102
How old are you?
123 432 4535 645
How old are you?
*) (*&(*  *&(&&*^( (*  *(&*&&&*
How old are you?
|

```

Figure 35 : Input validation for student gender and age

Figure 35 shows input validation for student gender. The valid format for gender is character ‘M’ or ‘F’. To make sure it works correctly, it is tested with string with spaces, numbers, symbols and multiple numbers. From Figure 35, it can be seen that the program will continue to ask for valid input unless valid input is given.

For age, the range of age allowed is between 1 to 100. String, numbers not between 1 to 100, multiple numbers and symbols are tested. (Valid number cannot be demonstrated here as clearScreen() method will be called)

```

Subject
1 PRG1203 - Object Oriented Programming Fundamentals
2 CSC1202 - Computer Organisation
3 SEG1201 - Database Fundamentals
4 WEB1201 - Web Fundamentals
0 Exit
Enter Option : abc
Please enter option between 0 and 4 only
Enter Option : abc def ghi jkl mno
Please enter option between 0 and 4 only
Enter Option : -23'
Please enter option between 0 and 4 only
Enter Option : 23
Please enter option between 0 and 4 only
Enter Option : *%(*% & ^% % &^% %* & ^(*^ (
Please enter option between 0 and 4 only
Enter Option : 123 456 457 56756 8 5 78
Please enter option between 0 and 4 only
Enter Option : |

```

Figure 36 : Input validation for student survey for specific subject

Figure 36 shows input validation for student survey for specific subject. In this menu, the valid format is integer with range of 0 to 4. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 0, number more than 4, symbols and multiple numbers. From Figure 36, it can be seen that the program will continue to ask for valid input unless valid input is given. (Valid number cannot be demonstrated here as clearScreen() method will be called)

```

Survey Subject : PRG1203

Your survey id is 21.

The course objectives were clear. (1 - 5 ONLY) : abc

The course objectives were clear. (1 - 5 ONLY) : abc def ghi jkl mno

The course objectives were clear. (1 - 5 ONLY) : 67

The course objectives were clear. (1 - 5 ONLY) : -78

The course objectives were clear. (1 - 5 ONLY) : *%(%^% (%^ %* ^*% ^% % %*

The course objectives were clear. (1 - 5 ONLY) : 123 435 7587 869 78 9

The course objectives were clear. (1 - 5 ONLY) : 1

The course procedures and assignments support course objectives. (1 - 5 ONLY) : |

```

Figure 37 : Input validation for student survey questions

Figure 37 shows input validation for student survey question. In this menu, the valid format is integer with range of 1 to 5. To make sure it works correctly, it is tested with string without spaces, string with spaces, number lesser than 1, number more than 2, symbols and multiple numbers. From Figure 37, it can be seen that the program will continue to ask for valid input unless valid input is given.

Additional Feature – Graphical User Interface (Extra)

After coding the Student Survey Application using console (without interface), some of the classes now contains the “brain” methods of the functions. To further enhance the user friendliness, graphical user interface will have to be added. These Graphical User Interface (GUI) is additional feature for this Student Survey Application. They are created using NetBean Jform.

Screenshot of Student Survey Application Graphical User Interface

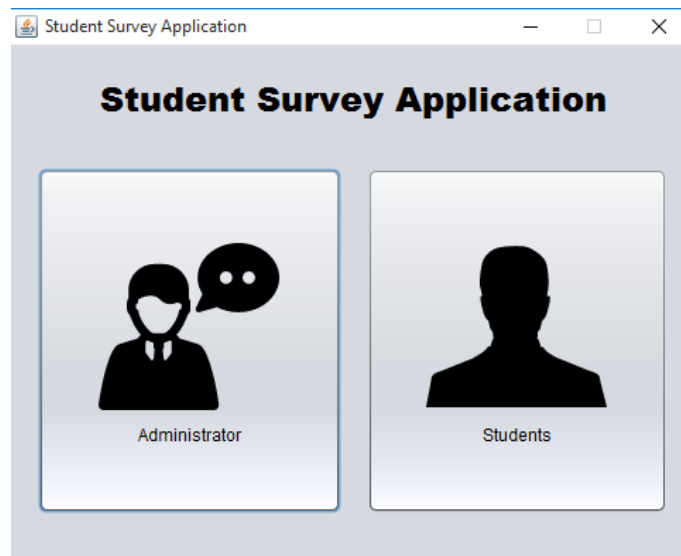


Figure 38 : GUI of User Menu

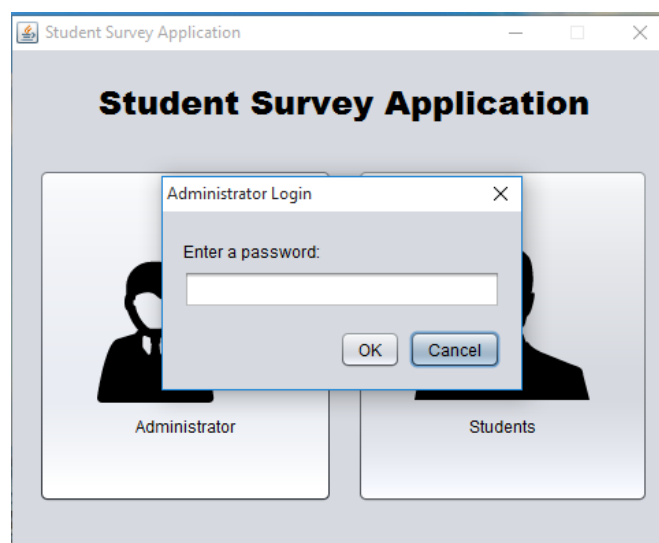


Figure 39 : GUI of Administrator Login

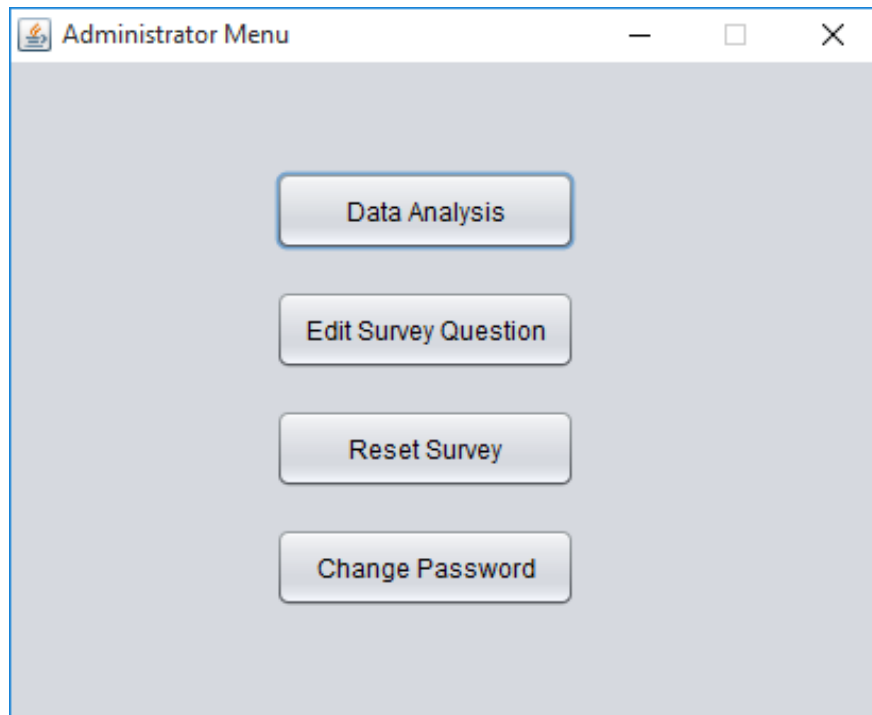


Figure 40 : GUI of Administrator Menu

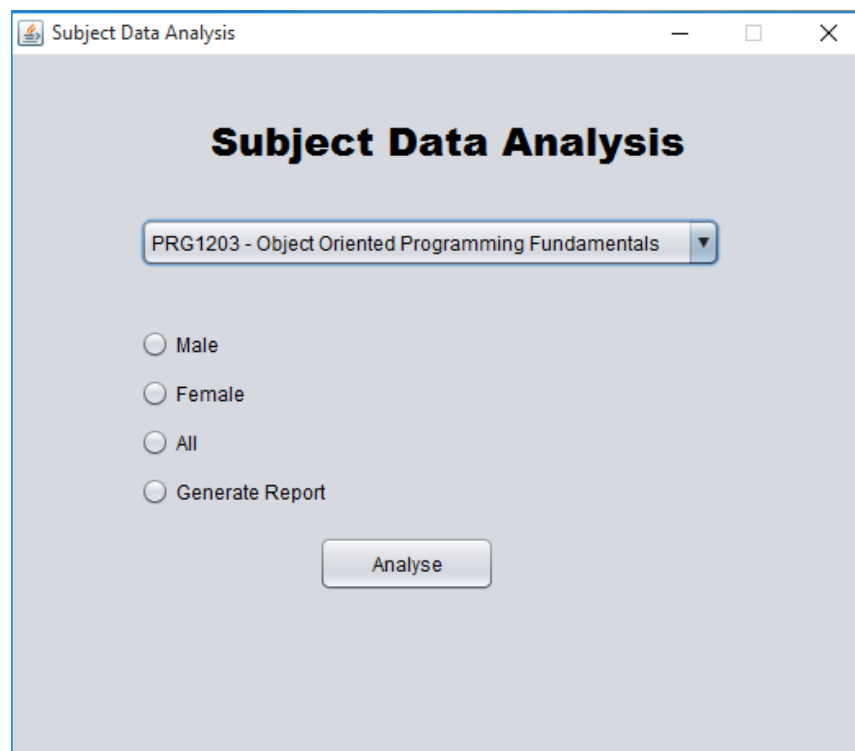


Figure 41 : GUI of Subject Data Analysis Menu

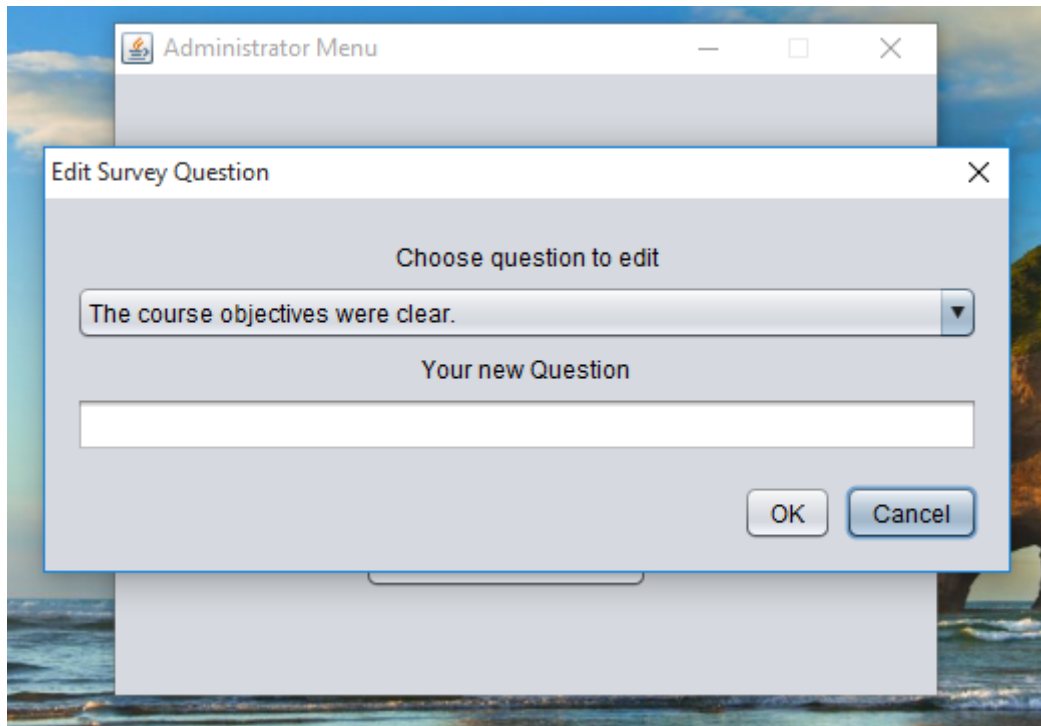


Figure 42 : GUI of Question Editing

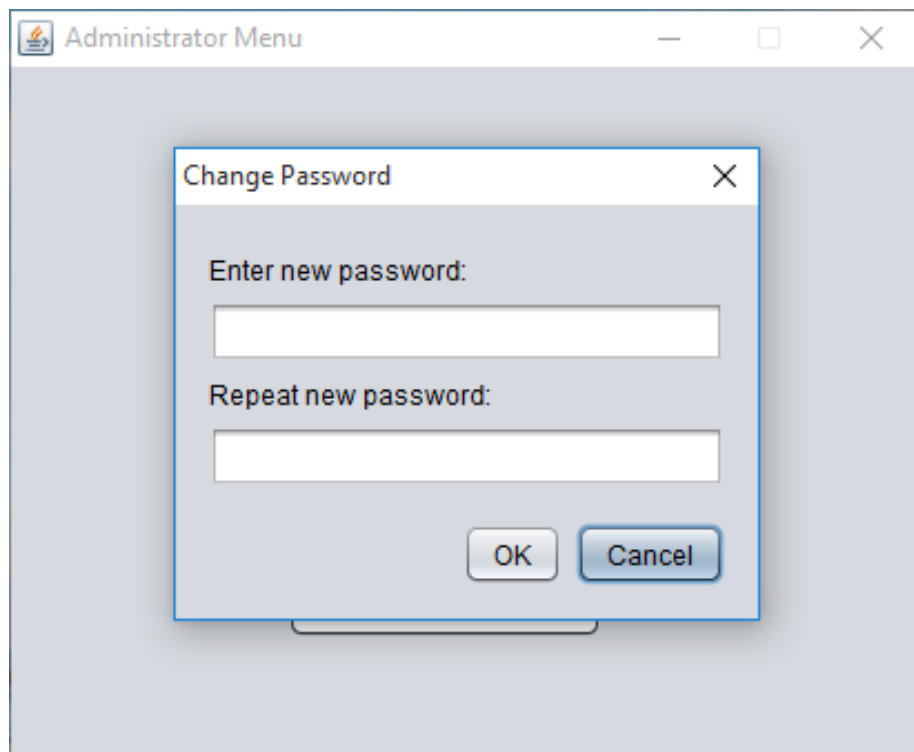


Figure 43 : GUI of Password Changing

Figure 44 : Student Survey

After converting the application to GUI, we realized that more classes can be created for user interface and the methods for some of the classes for console interface can be removed.

How the data structures or method or technique works to achieve the project functionalities?

Administrator class (Administrator.java)

In administrator class, some of the methods such as chooseOption() method, login() method, printOption method and newPassmrthod are removed. These methods are used for console interface and some of the codes in these methods are moved to new AdministratorMenu class which is designed for Administrator Menu and new User Menu class which is designed for User Menu. However, the private String type instance variable, password, is maintained to store password. getPassword() method, setPassword(String password method) and write() method are remain the same. However, there is a minor changes to read() method to make it more suitable to the class designed for GUI.

DataAnalysis class (DataAnalysis.java)

In this class, mainAnalysis() method, courseAnalysis() method and quesAnalysis method are removed and other methods and instance variables remain unchanged. This is because these three methods are methods that are called to create console interface for administrator's subject data analysis menu.

Form class (Form.java)

This class is the same as the project without graphical user interface. It is used to create objects to store information for students survey feedback.

FormFunction class (FormFunction.java)

The printEdit method which is used to print interface for question editing is removed. Some of the codes will be replaced into AdministratorMenu class under a button used for creating JoptionPane for question editing. Other methods are maintained as they will be needed for reading and writing survey questions and survey feedbacks.

Inheritance class (Inheritance.java)

In this class, repeat instance variable is removed as it is no longer required for looping control since GUI will be implemented. Methods such as clearScreen(), errorTest(int min, int max), cmdToQuit(), printCourse() and assignCourse(int option) will be removed as implementing GUI will not require these methods. These methods are mainly used to support error detection and print console interface for user. Even though these methods are removed, some of the codes or statements are used in classes designed for GUI.

Student class (Student.java)

This class is totally removed as GUI components will be used for student feedback options.

UserMenu class (UserMenu.java)

UserMenu() Constructor

In this constructor, initComponents() method is called to initialize all components and place them at the appropriate places. A FormFunction object instance, ff, is created for reading and writing data. A Window Event Listener is also placed in this constructor to make sure that writeSurvey() method is called when window is closing so that all survey feedback can be stored into database.txt. An Administrator object instance is created and read() method is called to read in password from

adminPass.txt. If password exist, readSurvey() and readQuestion() methods will be called; else, it will ask to enter new password.

initComponents() method

This is a generated method which contain codes for the components in GUI. It will be called to placed compoonents in appropriate places.

AdministratorActionPerformed(java.awt.event.ActionEvent evt) method

This is a method that acts as action listener. It carries codes to execute inside the block when Administrator button is pressed. This method allows user to enter password for password validation. If password correct, an AdministratorMenu object instance is created and show method is called; else, it will display error using JOptionPane dialog box.

studentActionPerformed(java.awt.event.ActionEvent evt) method

This is a method that acts as action listener. It carries code to execute inside the block when student button is pressed. A SurveyApplication object instance is created and show method is called.

main(String args[]) method

This method will be invoked by create and display the form when a new object instance of UserMenu class is called to run.

AdministratorMenu class (AdministratorMenu.java)

AdministratorMenu() Constructor

In this constructor, initComponents() method is called to initialize all components and placed them at the appropriate places. A Window Event Listener is also placed in this constructor to make sure that an object instance of UserMenu class will be created and show() method is called when the window is closing.

initComponents() method

This is a generated method which contain codes for the components in GUI. It will be called to placed compoonents in appropriate places.

dataAnalysisbtnActionPerformed(java.awt.event.ActionEvent evt) method

This is a method that will craete an object instance of AnalysisForm class and show() method is called when dataAnalysisbtn button is pressed.

resetActionPerformed(java.awt.event.ActionEvent evt) method

This is a method that will call clear() method of record array list in Inheritance class to remove all Form object in array list.

editQuesActionPerformed(java.awt.event.ActionEvent evt) method

This is a method that contains statements or codes to be executed when editQues Button is pressed. When editQues Button is pressed, a panel with combo box and textfield will appear. The Combo Box will contain all the five questions that are from Inheritance class question array. Administrator can choose question to edit and enter new question in the text field. After pressing OK, the text in the text field will be obtained by calling the getText() method and the String will replace the question chosen by using assignment operator. If text field is empty, JOptionPane window error will be displayed. FormFunction object instance is then created and writeQuestion() method is called to write questions to question.txt.

chgPassActionPerformed(java.awt.event.ActionEvent evt) method

This is a method that will be invoked when chgPass button is pressed. It will require administrator to enter new password two times. If password is different, error message will be displayed and user can continue to change password.

callSetPass() method

This is a method that will be called when this application is first run. Administrator will have to enter administrator password to set up this application.

getPassword() method

This is a getter method that will return value of instance String variable password.

main(String args[]) method

This method will be invoked by create and display the form when a new object instance of AdministratorMenu class is called to run.

AnalysisForm class (AnalysisForm.java)

AnalysisForm() Constructor

This is a constructor method that will call initComponents() method which contain elements and components to be placed on the frame. It the call group2Visibility(false) method to make sure that options on the right will be hidden in the beginning.

Group2Visibility(boolean visibility) method

This method is used to set the visibility of right section components, which are the sub components that depend on the left components.

initComponents() method

This method is called to initialize the position and properties of components in the frame.

analyseActionPerformed(java.awt.event.ActionEvent evt) method

This method is invoked when analyse Button is pressed. It allows administrator to choose all the aspects that wants to be analysed. After choosing these aspects, this method contains statements to check radio buttons that are selected. It will then call methods from DataAnalysis class to analyse these aspects and a window is then displayed with the analysis results.

maleActionPerformed(java.awt.event.ActionEvent evt) method

This method is invoked when male radio button is pressed. group2Visibility(true) method is called to display radio buttons on the right for further selection. setGenderOption(1) method in DataAnalysis class is then called to set genderOption variable as 1 indicating male is chosen.

femaleActionPerformed(java.awt.event.ActionEvent evt) method

This method is invoked when female radio button is pressed. group2Visibility(true) method is called to display radio buttons on the right for further selection. setGenderOption(2) method in DataAnalysis class is then called to set genderOption variable as 2 indicating female is chosen.

allActionPerformed(java.awt.event.ActionEvent evt) method

This method is invoked when all radio button is pressed. group2Visibility(true) method is called to display radio buttons on the right for further selection. setGenderOption(3) method in DataAnalysis class is then called to set genderOption variable as 3 indicating all gender will be analysed.

reportActionPerformed (java.awt.event.ActionEvent evt)

This method is invoked when report radio button is pressed. group2Visibility(false) is called to hide radio buttons on the right as they are not required. setGenderOption(4) method in DataAnalysis class is then called to set genderOption variable as 4 indicating report will be generated.

main(String args[]) method

This method will be invoked by create and display the form when a new object instance of AnalysisForm class is called to run.

Survey class (Survey.java)

Survey() Constructor

In this constructor, initComponents() method is called to initialize all components and placed them at the appropriate places. A Window Event Listener is also placed in this constructor to make sure that an object instance of Survey class will be created and show() method is called when the window is closing.

initComponents() method

This is a generated method which contain codes for the components in GUI. It will be called to placed components in appropriate places.

maleActionPerformed(java.awt.event.ActionEvent evt) method

This method is invoked when male radio button is pressed. genderF instance variable will be set to 'M' indicating that male is chosen.

femaleActionPerformed(java.awt.event.ActionEvent evt) method

This method is invoked when female radio button is pressed. genderF instance variable will be set to 'F' indicating that female is chosen.

submitActionPerformed(java.awt.event.ActionEvent evt) method

This method contains statements that will check whether all the questions are ticked or answered. When submit Button is pressed, if all questions are answered, a new Form object will be created and added to record array list in Inheritance class. A window will be displayed indicating that survey is submitted successfully. If any one of the question is not answered, error message will be displayed.

clearActionPerformed(java.awt.event.ActionEvent evt) method

This method will be invoked when clear Button is pressed. When clear button is pressed, all the selected radio buttons will be deselect, age will be set back to default value, which is 1, and the selected index for combo box will be set back to 0, the first item.

q11 - q15ActionPerformed(java.awt.event.ActionEvent evt) method

These methods will set the ratings of students in the survey feedback to ques[0] which indicate the first question rate.

q21 – q25ActionPerformed(java.awt.event.ActionEvent evt) method

These methods will set the ratings of students in the survey feedback to ques[1] which indicate the second question rate.

q31 – q35ActionPerformed(java.awt.event.ActionEvent evt) method

These methods will set the ratings of students in the survey feedback to ques[2] which indicate the third question rate.

q41 – q45ActionPerformed(java.awt.event.ActionEvent evt) method

These methods will set the ratings of students in the survey feedback to ques[3] which indicate the forth question rate.

q51 – q55ActionPerformed(java.awt.event.ActionEvent evt) method

These methods will set the ratings of students in the survey feedback to ques[4] which indicate the fifth question rate.

main(String args[]) method

This method will be invoked by create and display the form when a new object instance of Survey class is called to run.

StudentSurveySystem class (StudentSurveySystem.java)

This class which was used as the driver program when creating project without GUI can be removed and userMenu class will be the driver program.

Testing report or set of validations you applied

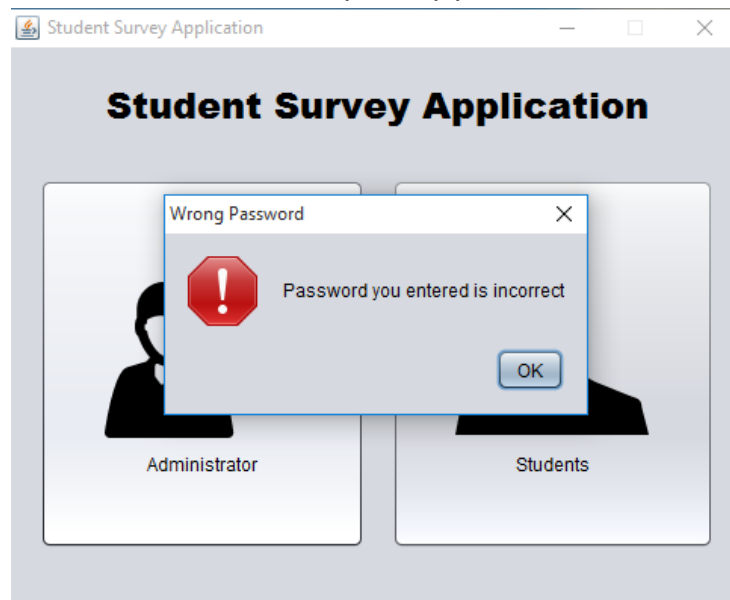


Figure 45 : Password validation for Administrator

Figure 45 will appear when user entered wrong password. User will not be able to enter administrator menu.

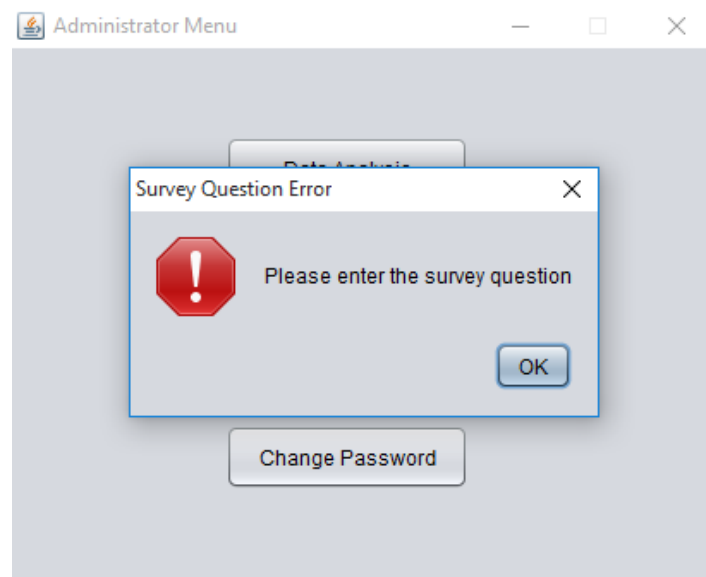


Figure 46 : Error displayed when the text field of survey question is empty

Figure 46 will appear when administrator chosen a specific question but new question is not entered during survey question editing.

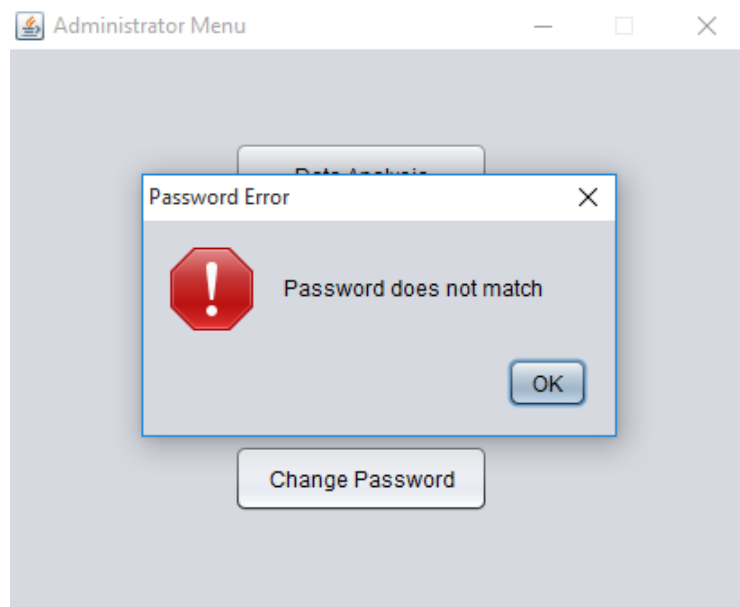


Figure 47 : Error displayed when password changing

Figure 47 will appear when administrator chose to change password and the new password entered is not equals to the repeated new password.

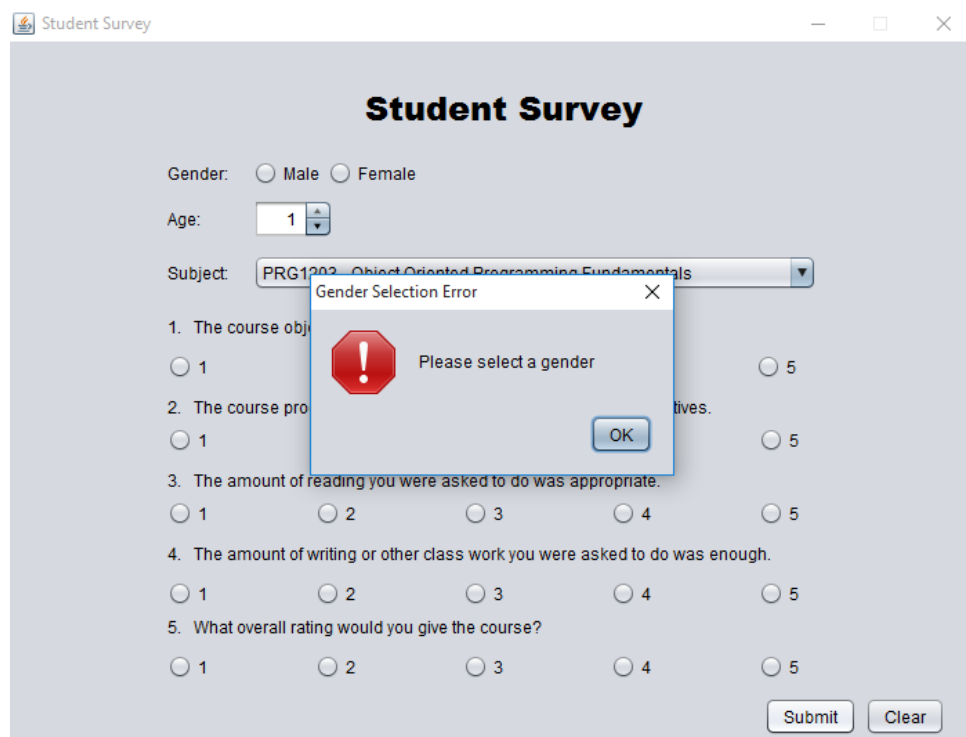


Figure 48 : Error displayed if no gender selected

Figure 48 appear when no gender is selected.

Student Survey

Student Survey

Gender: ☒ Male ☐ Female

Age:

Subject: PRG1202 - Object Oriented Programming Fundamentals

Question 1

1. The course objectives are clear and understandable.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. The course provides a good balance of theory and practical examples.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

3. The amount of reading you were asked to do was appropriate.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

4. The amount of writing or other class work you were asked to do was enough.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

5. What overall rating would you give the course?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Submit Clear

Question 1 not selected

OK

Figure 49 : Error displayed if question rating not selected

Figure 49 appear when question rating is not selected.