|  |
| --- |
| Microsoft |
| I.T. P.A.T. |
| *System Design Document* |
| Best viewed 2 pages at a time. (Zoom level +/- 60%) |
| **Jayden Maree** |
|  |
| **Glenwood House** |

*Table of Contents*

[*User Interface Design* 4](#_Toc491810406)

[Log In Screen 4](#_Toc491810407)

[Security 5](#_Toc491810408)

[Data Access and Error-checking 5](#_Toc491810409)

[Registration Screen 6](#_Toc491810410)

[Security 7](#_Toc491810411)

[Data Access and Error-checking 7](#_Toc491810412)

[Tutor Profile Screen 8](#_Toc491810413)

[Security 9](#_Toc491810414)

[Data Access and Error-checking 9](#_Toc491810415)

[Student Profile Screen 10](#_Toc491810416)

[Security 11](#_Toc491810417)

[Data Access and Error-checking 11](#_Toc491810418)

[Admin Profile Screen 12](#_Toc491810419)

[Security 13](#_Toc491810420)

[Data Access and Error-checking 13](#_Toc491810421)

[Admin Screen 14](#_Toc491810422)

[Security 15](#_Toc491810423)

[Data Access and Error-checking 15](#_Toc491810424)

[Student Screen 16](#_Toc491810425)

[Security 17](#_Toc491810426)

[Data Access and Error-checking 17](#_Toc491810427)

[Tutor Screen 18](#_Toc491810428)

[Security 19](#_Toc491810429)

[Data Access and Error-checking 19](#_Toc491810430)

[Confirmation Dialogue Screen 20](#_Toc491810431)

[Security 21](#_Toc491810432)

[Data Access and Error-checking 21](#_Toc491810433)

[JFileChooser Screen 22](#_Toc491810434)

[*Sequencing* 23](#_Toc491810435)

[Key: 23](#_Toc491810436)

[Flow Diagram (A) 24](#_Toc491810437)

[Flow Diagram (B) 25](#_Toc491810438)

[Flow Diagram (D) 26](#_Toc491810439)

[Flow Diagram (E) 27](#_Toc491810440)

[*Class Design* 28](#_Toc491810441)

[‘Database’ Class 28](#_Toc491810442)

[Fields 28](#_Toc491810443)

[Methods 28](#_Toc491810444)

[‘Main’ Class 30](#_Toc491810445)

[Fields 30](#_Toc491810446)

[Methods 30](#_Toc491810447)

[*Persistent Storage Design* 31](#_Toc491810448)

[tblUser 31](#_Toc491810449)

[Table: 31](#_Toc491810450)

[Formatting: 31](#_Toc491810451)

[Sample Data: 31](#_Toc491810452)

[tblPeer 32](#_Toc491810453)

[Table: 32](#_Toc491810454)

[Formatting: 32](#_Toc491810455)

[Sample Data: 32](#_Toc491810456)

[tblTimetable 33](#_Toc491810457)

[Table: 33](#_Toc491810458)

[Formatting: 33](#_Toc491810459)

[Sample Data: 33](#_Toc491810460)

[tblSubjects 34](#_Toc491810461)

[Table: 34](#_Toc491810462)

[Formatting: 34](#_Toc491810463)

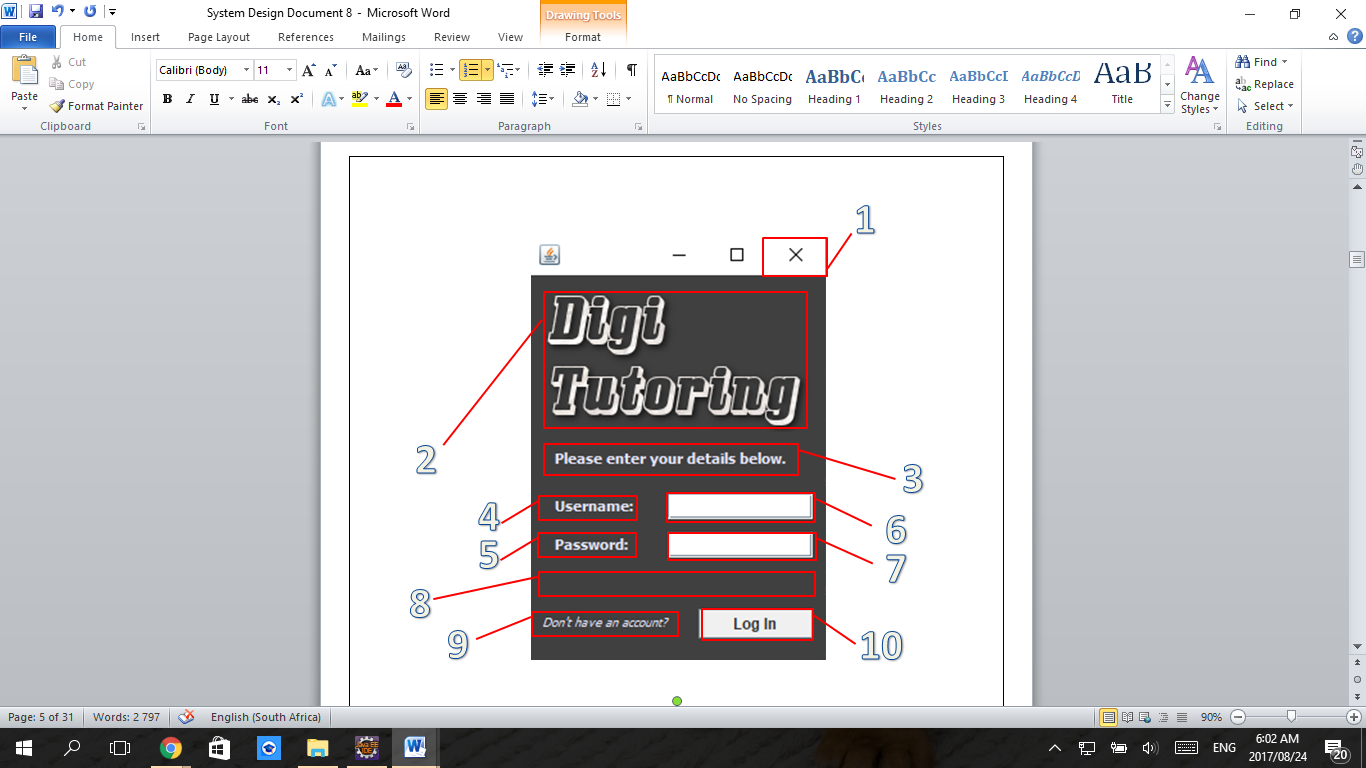
[Sample Data: 34](#_Toc491810464)

[Relationship Diagram 35](#_Toc491810465)

[*Explanation of Storage Design* 36](#_Toc491810466)

# *User Interface Design*

## Log In Screen



1. The clickable ‘Exit’ button that closes the program.
2. The application logo.
3. The label instructing users what to do.
4. The label indicating that usernames should be entered adjacently.
5. The label indicating that passwords should be entered adjacently.
6. The text field for users to enter their usernames.
7. The password field for users to enter their passwords.
8. The dynamically changing label used by the application to inform users of their input-errors.
9. The clickable label that opens the registration screen.
10. The clickable button that opens the user’s respective screen (Student, Tutor, Admin).

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 2,3,4,5,8 | 6,7 | 1,9,10 |

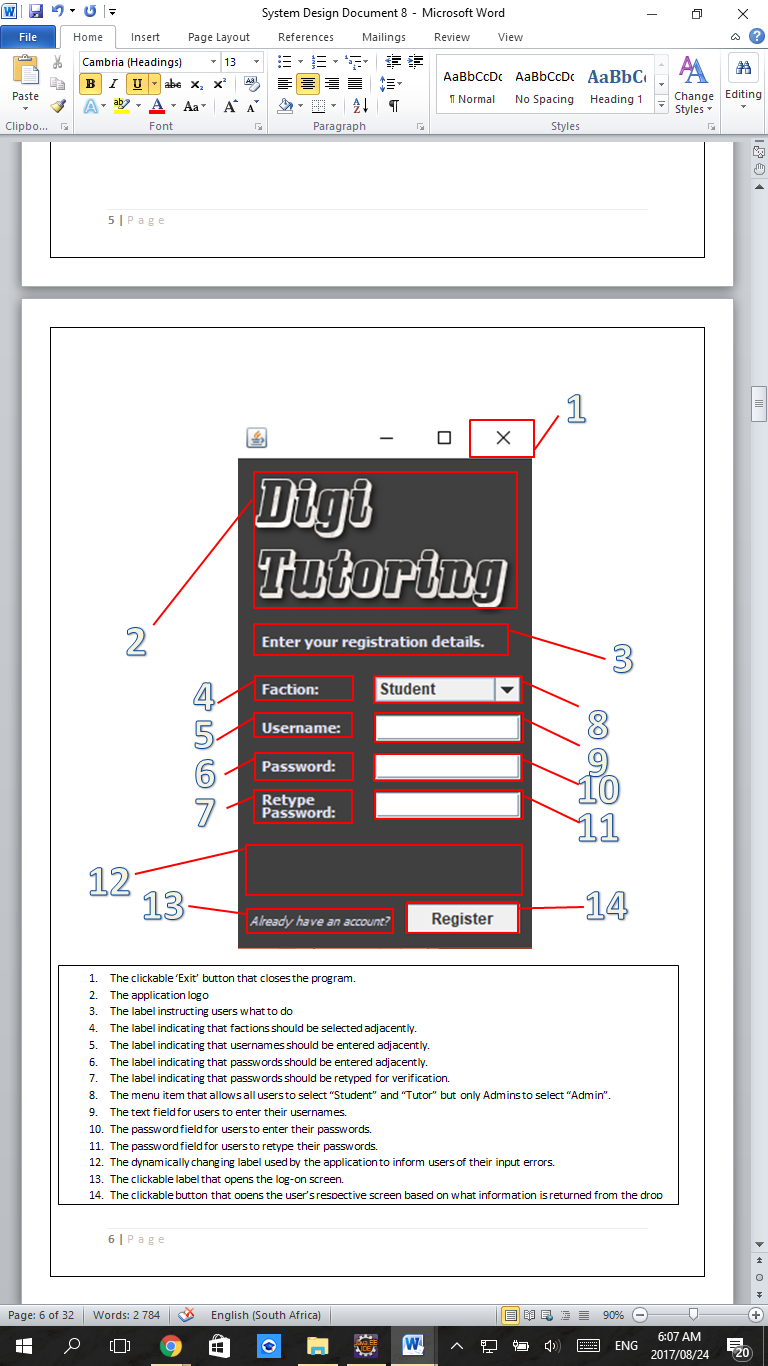
### Security

This screen is opened when the program is launched, thus all users view this screen.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. No error-checking takes place and no data is accessed or stored.
3. 6 and 7 are checked for validity – it is checked whether or not the username (6) is in the database and whether or not it has the corresponding password (7).
   1. If there is no correspondence or existence then label 8 will change to inform the user (textually), and fields 6 and 7 will change to the colour red, indicating an error.

## Registration Screen



1. The clickable ‘Exit’ button that closes the program.
2. The application logo.
3. The label instructing users what to do
4. The label indicating that factions should be selected adjacently.
5. The label indicating that usernames should be entered adjacently.
6. The label indicating that passwords should be entered adjacently.
7. The label indicating that passwords should be retyped for verification adjacently.
8. The menu field that allows all users to select “Student” or “Tutor” but only Admins to select “Admin”.
9. The text field for users to enter their usernames.
10. The password field for users to enter their passwords.
11. The password field for users to retype their passwords for verification.
12. The dynamically changing label used by the application to inform users of their input errors.
13. The clickable label that opens the log-in screen.
14. The clickable button that opens the user’s respective screen (Student, Tutor, Admin).
15. The clickable button that opens the user’s respective screen based on what information is returned from the drop down menu item (Student, Tutor or Admin).

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 2,3,4,5,6,7,12 | 8,9,10,11 | 1,13,14 |

### Security

This screen is opened when the Action Element ‘9’ is pressed in the Log In Screen, no data checking takes place when pressed. The Log In Screen is accessible by all users, thus so is this Registration Screen if users choose to access it.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. No error-checking takes place and no data is accessed or stored.
3. 8 is checked to be non-default (validation). 9 is checked for availability in the database. 10 and 11 are verified to be identical.
   1. If there is an error then label 8 will change to inform the user (textually), and fields 9, 10 and 11 will turn red to indicate errors appropriately.

## Tutor Profile Screen

1. The clickable ‘Exit’ button that closes the program.
2. The clickable label that opens the Tutor screen, where tutors view students.
3. The label indicating that the user’s name should be entered adjacently.
4. The label indicating that the user’s surname should be entered adjacently.
5. The label indicating that the user’s date of birth should be entered adjacently.
6. The label indicating that the user’s name should be selected from the menu adjacent.
7. The label indicating that the user’s subjects should be entered or removed in the fields adjacent.
8. The label indicating that the user’s cell phone number should be entered adjacently.
9. The label indicating that the user’s profile picture should be entered adjacently.
10. The label indicating that the user’s timetable should be entered adjacently.
11. The dynamically changing label used by the application to inform users of their input errors.
12. The clickable label that opens the confirmation screen, where users can delete their account.
13. The text field for users to enter their name.
14. The text field for users to enter their surname.
15. The field for users to enter their date of birth.(verified)
16. The menu field for users to enter their grade.
17. The text field for users to enter their subject to add.
18. The text field for users to select their subject to remove.
19. The text field for users to enter their cell phone number. (verified)
20. The field that displays the users profile picture.
21. The clickable button used to open the JFileChooser Screen.
22. The timetable for users to enter their tutoring times.
23. The clickable button used to upload data into the database using information entered.
24. The clickable button used to download data from the database and populate the screen
25. The clickable button used to remove selected subjects from the adjacent window.
26. The clickable button used to add entered subjects into the adjacent window.

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 3,4,5,6,7,8,9,10,11,20 | 13,14,15,16,17,18,19,22 | 1,2,12,21,23,24,25,26 |

### Security

This screen is accessible by users that have previously registered as Tutors. This screen is opened after registration or after the Log In process. If the user Logs In then their profile data is displayed.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. It is checked that the database is populated with profile data, else 11 displays an error.
3. No error-checking takes place, verification is asked via the Confirmation Dialogue Screen, data is deleted from the database if the dialogue approved.
4. No error checking and no data directly accessed but rather indirectly via JFileChooser.
5. Error Checking: are Data Entered Fields empty? If True then 11 displays error, if False then Data Access: update the entered data into corresponding database tables. Save image to file.
6. Error Checking: are Data Entered Fields empty? If True then 11 displays error, if False then Data Access: update the Profile Screen to display the user’s database data.
7. And 26. No error checking and no database data access.

## Student Profile Screen

1. The clickable ‘Exit’ button that closes the program.
2. The clickable label that opens the Student screen, where students view tutors.
3. The label indicating that the user’s name should be entered adjacently.
4. The label indicating that the user’s surname should be entered adjacently.
5. The label indicating that the user’s date of birth should be entered adjacently.
6. The label indicating that the user’s name should be selected from the menu adjacent.
7. The label indicating that the user’s subjects should be entered or removed in the fields adjacent.
8. The label indicating that the user’s cell phone number should be entered adjacently.
9. The label indicating that the user’s profile picture should be entered adjacently.
10. The dynamically changing label used by the application to inform users of their input errors.
11. The clickable label that opens the confirmation screen, where users can delete their account.
12. The text field for users to enter their name.
13. The text field for users to enter their surname.
14. The field for users to enter their date of birth.(verified)
15. The menu field for users to enter their grade.
16. The text field for users to enter their subject to add.
17. The text field for users to select their subject to remove.
18. The text field for users to enter their cell phone number. (verified)
19. The field that displays the users profile picture.
20. The clickable button used to open the JFileChooser Screen.
21. The clickable button used to remove selected subjects from the adjacent window.
22. The clickable button used to add entered subjects into the adjacent window.
23. The clickable button used to upload data into the database using information entered.
24. The clickable button used to download data from the database and populate the screen

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 3,4,5,6,7,8,9,10,19 | 12,13,14,15,16,17,18 | 1,2,11,20,21,22,23,24 |

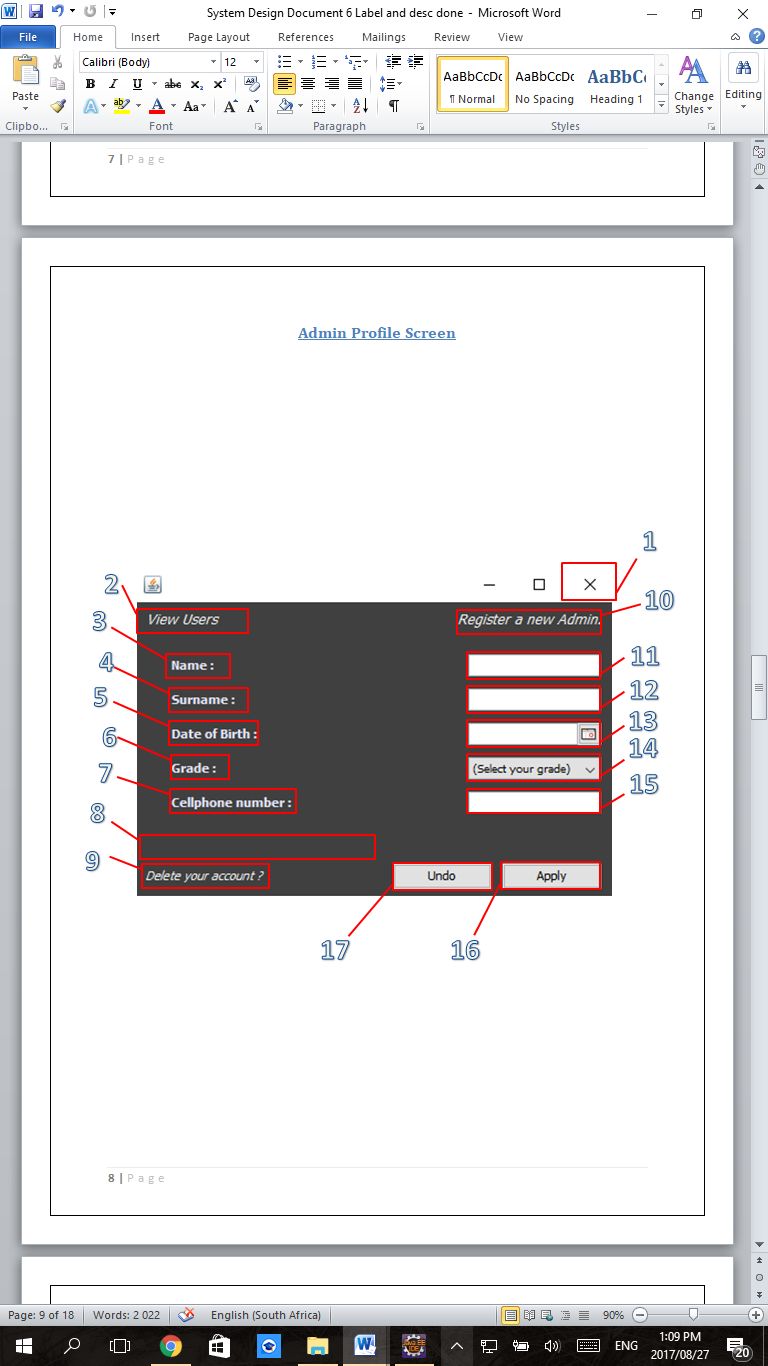
### Security

This screen is accessible by users that have previously registered as Students. This screen is opened after registration or after the Log In process. If the user Logs In then their profile data is displayed.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. It is checked that the database is populated with profile data, else 10 displays an error.
3. No error-checking takes place, verification is asked via the Confirmation Dialogue Screen, data is deleted from the database if the dialogue approved.
4. No error checking and no data directly accessed but rather indirectly via JFileChooser.
5. Error Checking: are Data Entered Fields empty? If True then 10 displays error, if False then Data Access: update the entered data into corresponding database tables. Save image to file.
6. Error Checking: are Data Entered Fields empty? If True then 10 displays error, if False then Data Access: update the Profile Screen to display the user’s database data.
7. And 24. No error checking and no database data access.

## Admin Profile Screen



1. The clickable ‘Exit’ button that closes the program.
2. The clickable label that opens the Admin screen, where admins view users.
3. The label indicating that the user’s name should be entered adjacently.
4. The label indicating that the user’s surname should be entered adjacently.
5. The label indicating that the user’s date of birth should be entered adjacently.
6. The label indicating that the user’s name should be selected from the menu adjacent.
7. The label indicating that the user’s cell phone number should be entered adjacently.
8. The dynamically changing label used by the application to inform users of their input errors.
9. The clickable label that opens the confirmation screen, where users can delete their account.
10. The clickable label that opens the registration screen with ‘Admin Privileges’.
11. The text field for users to enter their name.
12. The text field for users to enter their surname.
13. The field for users to enter their date of birth.(verified)
14. The menu field for users to enter their grade
15. The text field for users to enter their cell phone number. (verified)
16. The clickable button used to upload data into the database using information entered.
17. The clickable button used to download data from the database and populate the screen

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 3,4,5,6,7,8 | 11,12,13,14,15 | 1,2,9,10,16,17 |

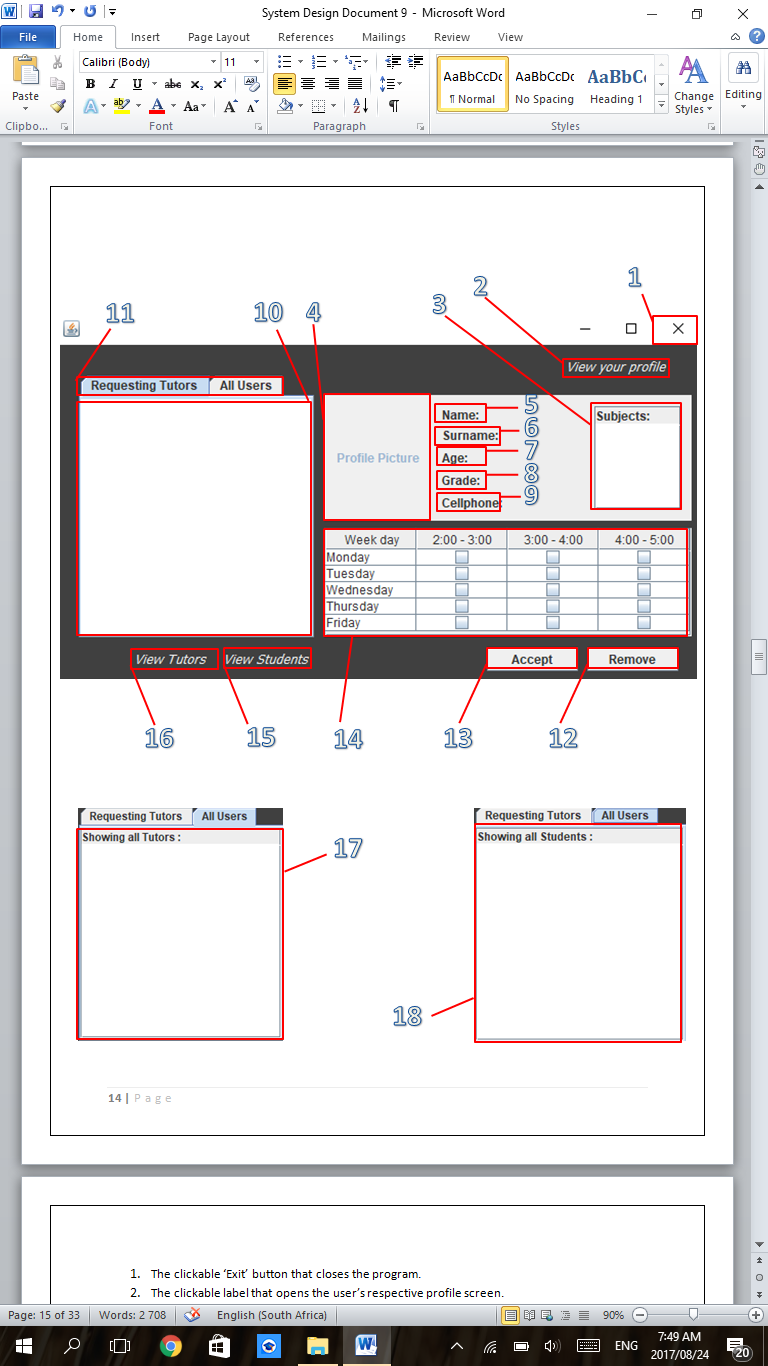
### Security

This screen is accessible by users that have previously registered as Admins. This screen is opened after registration or after the Log In process. If the user Logs In then their profile data is displayed.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. It is checked that the database is populated with profile data, else 11 displays an error.
3. No error-checking takes place, verification is asked via the Confirmation Dialogue Screen, data is deleted from the database if the dialogue approved.
4. No error-checking and no data access.
5. Error Checking: are Data Entered Fields empty? If True then 8 displays error, if False then Data Access: update the entered data into corresponding database tables.
6. Error Checking: are Data Entered Fields empty? If True then 8 displays error, if False then Data Access: update the Profile Screen to display the user’s database data.

## Admin Screen



1. The clickable ‘Exit’ button that closes the program.
2. The clickable label that opens the user’s respective profile screen.
3. The list of subjects associated with the user selected in 10, 17 or 18.
4. The field that displays the selected user’s profile picture.
5. The label indicating that the selected user’s name will be displayed adjacently.
6. The label indicating that the selected user’s surname will be displayed adjacently.
7. The label indicating that the selected user’s age will be displayed adjacently.
8. The label indicating that the selected user’s grade will be displayed adjacently.
9. The label indicating that the selected user’s cell phone number will be displayed adjacently.
10. The field that displays the list of tutors that are requesting admin approval.
11. The clickable tabs that allow the user to change between viewing all users and viewing requesting tutors.
12. The clickable button used to delete a user selected in 10, 17 or 18.
13. The clickable button used to give admin approval to a user selected in 10, after which they move to 18.
14. The timetable where tutoring times of tutors selected in 10 or 17 are displayed.
15. The clickable label that that brings the field 18 to the front when tab ‘All Users’ has been selected.
16. The clickable label that that brings the field 17 to the front when tab ‘All Users’ has been selected.
17. The field that displays the list of tutors that have been given admin approval.
18. The field that displays the list of students that have been registered into the application (are in the database).

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 3,4,5,6,7,8,9,10,14,17,18 |  | 1,2,11,12,13,15,16 |

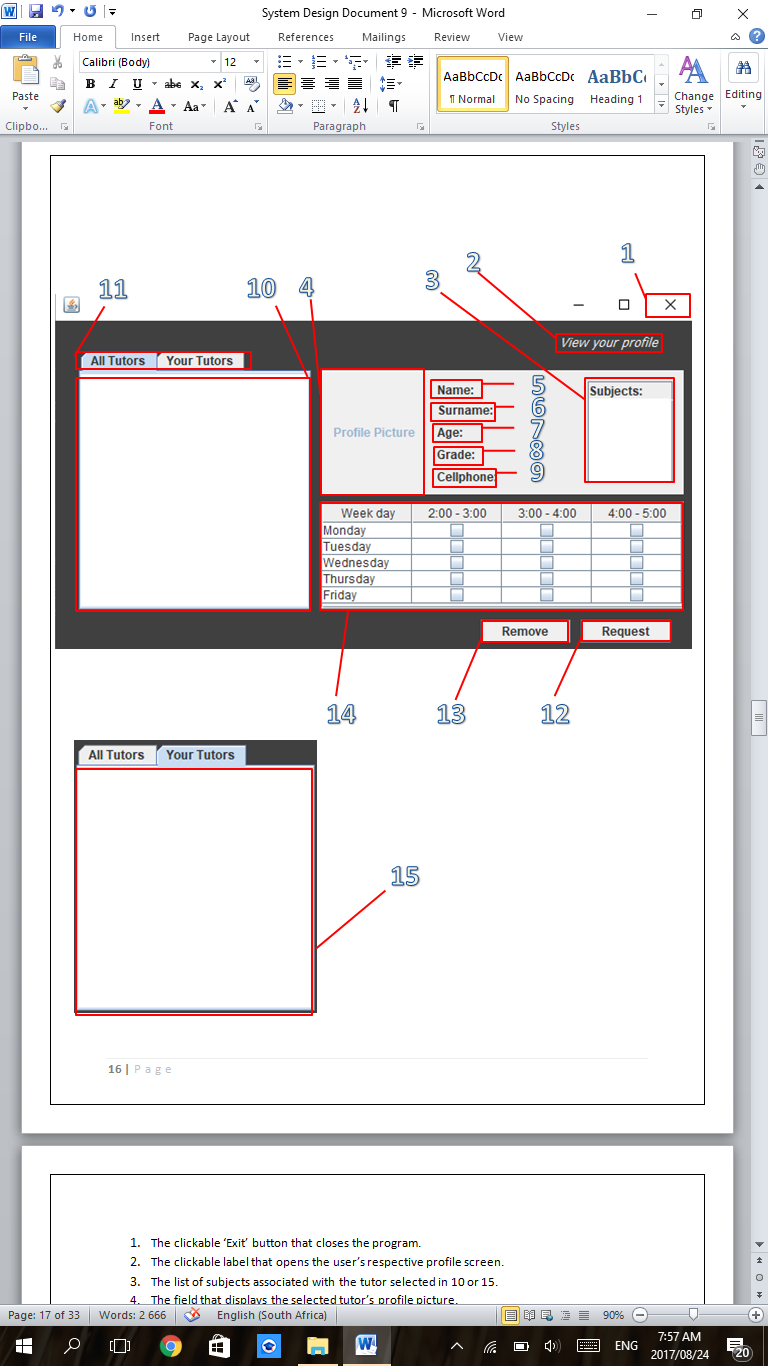
### Security

This screen is accessible by all Admins which have complete profiles. This screen is opened from the Admin Profile Screen.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. No error checking or data access.
3. No error-checking and no data access.
4. It is checked that only Requesting Tutors are selected, database data is updated
5. No error checking or data access.
6. No error checking or data access

## Student Screen



1. The clickable ‘Exit’ button that closes the program.
2. The clickable label that opens the user’s respective profile screen.
3. The list of subjects associated with the tutor selected in 10 or 15.
4. The field that displays the selected tutor’s profile picture.
5. The label indicating that the selected tutor’s name will be displayed adjacently.
6. The label indicating that the selected tutor’s surname will be displayed adjacently.
7. The label indicating that the selected tutor’s age will be displayed adjacently.
8. The label indicating that the selected tutor’s grade will be displayed adjacently.
9. The label indicating that the selected tutor’s cell phone number will be displayed adjacently.
10. The field that displays the list of tutors that have been granted admin approval. (‘All Tutors’)
11. The clickable tabs that allow the user to change between viewing all tutors and viewing tutors that have accepted them as their student.
12. The clickable button used by a student to request to be tutored by a tutor displayed in 10.
13. The clickable button used to remove a tutor field selected in 15.
14. The timetable where tutoring times of tutors selected in 10 or 15 are displayed.
15. The field that displays the list of tutors that have accepted the user as their student.

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 3,4,5,6,7,8,9,10,14,15 |  | 1,2,11,12,13 |

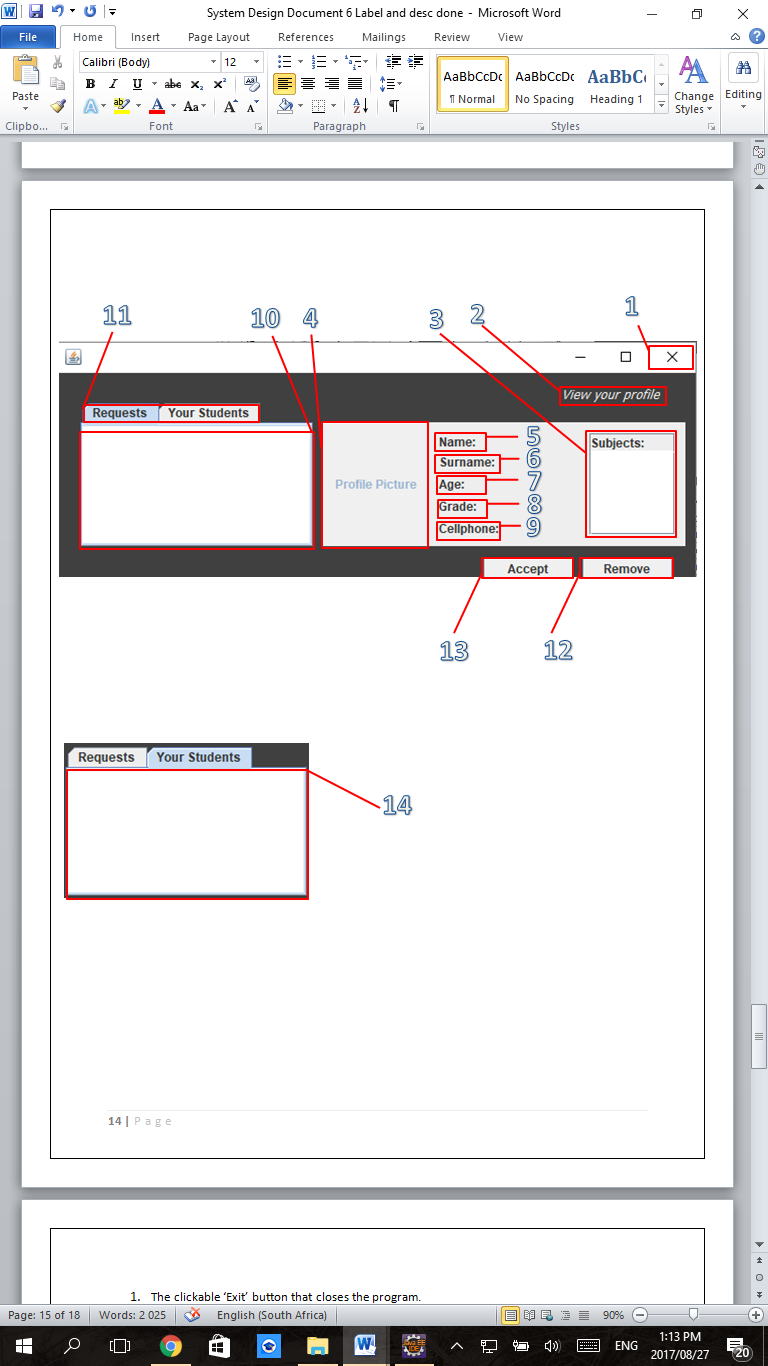
### Security

This screen is accessible by all Students which have complete profiles. This screen is opened from the Student Profile Screen.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. No error checking or data access.
3. No error-checking and no data access.
4. The action is only performed if a Tutor is selected, database data is updated.
5. The action is only performed if a Tutor is selected, database data is deleted.

## Tutor Screen



1. The clickable ‘Exit’ button that closes the program.
2. The clickable label that opens the user’s respective profile screen.
3. The list of subjects associated with the student selected in 10 or 14.
4. The field that displays the selected student’s profile picture.
5. The label indicating that the selected student’s name will be displayed adjacently.
6. The label indicating that the selected student’s surname will be displayed adjacently.
7. The label indicating that the selected student’s age will be displayed adjacently.
8. The label indicating that the selected student’s grade will be displayed adjacently.
9. The label indicating that the selected student’s cell phone number will be displayed adjacently.
10. The field that displays the list of students that have requested to be tutored by the user.
11. The clickable tabs that allow the user to change between viewing requesting students and viewing students that have already been accepted for tutoring.
12. The clickable button used to remove a student field selected in 10 or 14.
13. The clickable button used by a tutor to accept a student in 10 for tutoring, after which the selected field transfers to 14.
14. The field that displays the list of students that have been accepted for tutoring by the user.

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 3,4,5,6,7,8,9,10,14 |  | 1,2,11,12,13 |

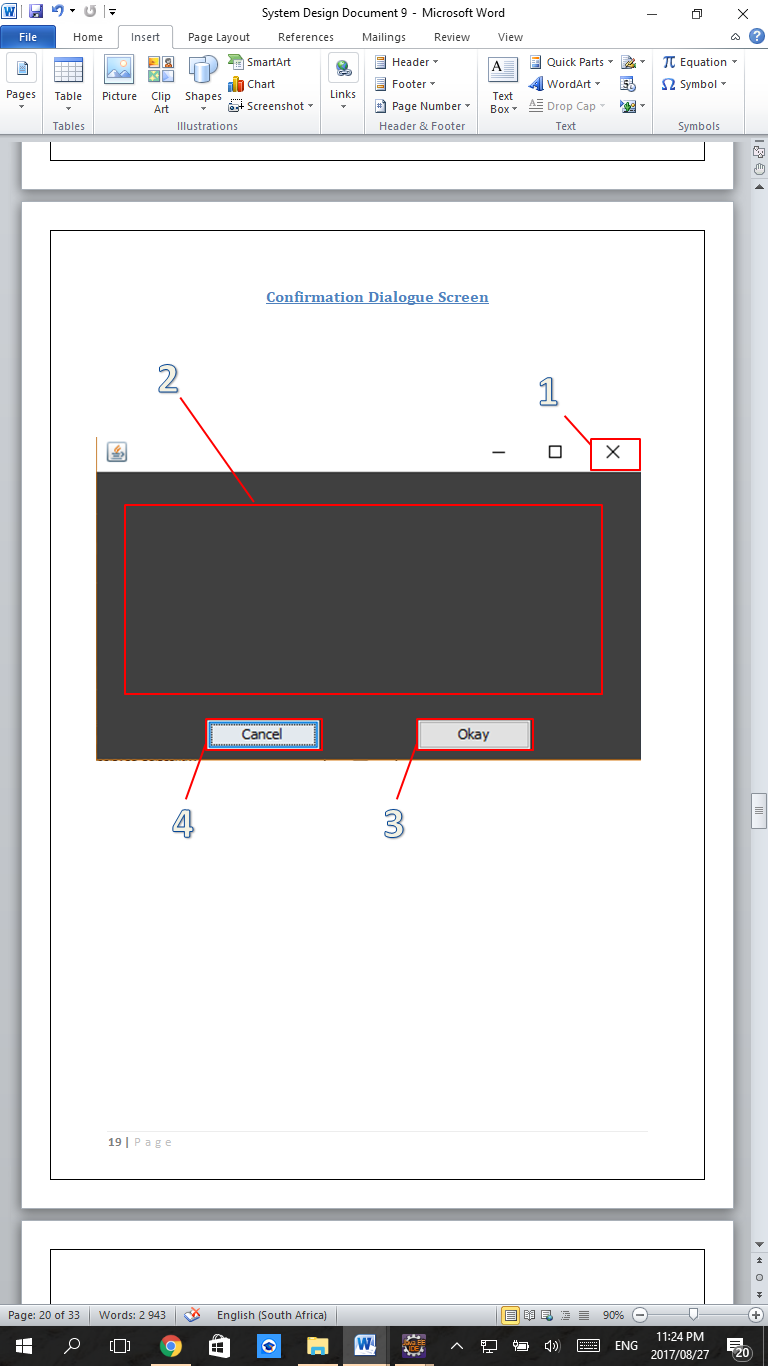
### Security

This screen is accessible by all Tutors which have complete profiles. This screen is opened from the Tutor Profile Screen.

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. No error checking or data access.
3. No error-checking and no data access.
4. The action is only performed if a Student is selected, database data is deleted.
5. The action is only performed if a Student in 10 is selected, database data is updated.

## Confirmation Dialogue Screen



1. The clickable ‘Exit’ button that closes the program.
2. The dynamically changing label used by the application to request user confirmation for the action in progress.
3. The clickable button used to confirm the action in progress and progress to the next screen (the next screen could also be the original screen).
4. The clickable button used to cancel the action in progress and return to the original screen.

|  |  |  |
| --- | --- | --- |
| Data Displayed | Data Entered | Action Elements |
| 2 |  | 1,3,4 |

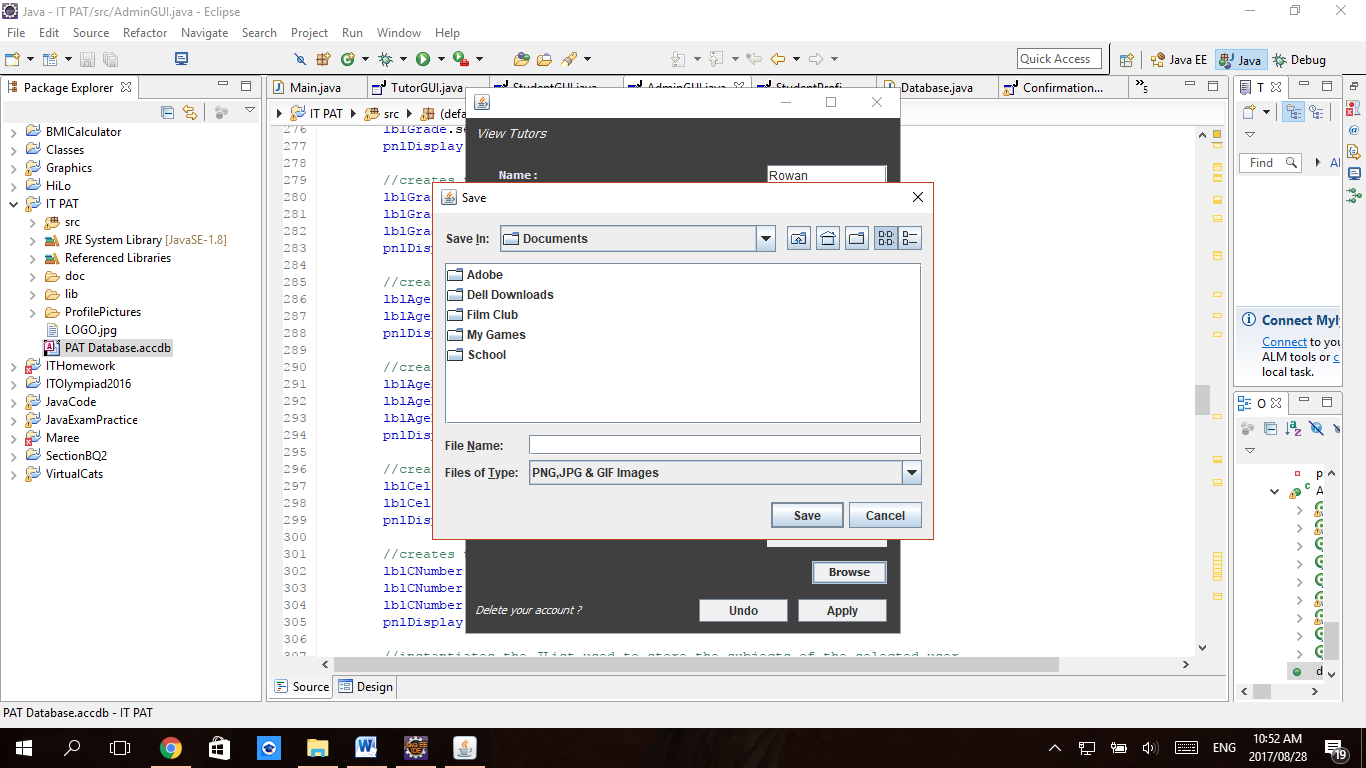
### Security

This screen is accessible by all Tutors which have complete profiles. This screen is opened from the Tutor Profile Screen,

### Data Access and Error-checking

1. No error-checking takes place and no data is accessed or stored, the database connection is closed.
2. No error checking takes place but data access could be deletion of user accounts.
3. No error checking or data access takes place.

## JFileChooser Screen



This screen has not been designed by me but I have used it as a javax.swing import.

This screen allows the user to select an image from their computer, after which when the profile ‘apply’ button is pressed this image will be copied into the project folder ‘ProfilePictures’ file and the new image path stored in the database with the rest of the user’s data.

# *Sequencing*

## Key:

Data Flow

Parent –Child Relationship

Database Queries used.

Reversible flow between screens

Joint arrow indicating one-way funnelling, joining two screens to one.

Reversible flow between the Registration screen and the Admin Profile screen is only available if the Registration screen is opened from within the Admin Profile screen.

## Flow Diagram (A)

Log In Screen

Registration Screen

Admin Profile Screen

Tutor Profile Screen

Student Profile Screen

Admin Privileges

## Flow Diagram (B)

Tutor Profile Screen

Tutor Application   
Screen

Student Profile Screen

Admin Profile Screen

Student Application Screen

Admin Application   
Screen

NOTE: ALL SCREENS IN DIAGRAM B READ AND WRITE DATA.

NON-IMAGE DATA IS STORED IN A DATABASE, IMAGES ARE STORED IN A FOLDER

## Flow Diagram (D)

tblPeer

tblUser

Database:

‘PAT Database’

tblTimetable

tblSubjects

Admin Application Screen.

Student Application Screen.

Tutor Application Screen.

Admin Application Screen.

Student Application Screen.

Tutor Application Screen

Tutor Profile Screen.

Student Profile Screen

All Screens

Admin Application Screen.

Student Application Screen.

Tutor Profile Screen.

## Flow Diagram (E)

Student Screen

Tutor Screen

Admin Screen

Student Profile Screen

Tutor Profile Screen

Image Folder

‘ProfilePictures’

# *Class Design*

## ‘Database’ Class

### Fields

|  |  |  |
| --- | --- | --- |
| **Modifier and Type** | **Field** | **Description** |
| private Connection | connection | Stores the connection received via the DriverManager class |
| private String | dbURL | Stores the URL of the database in terms |
| private String | driver | Stores the name of the driver used |
| private String | msAccDB | Stores the name of the database |
| private ResultSet | resultSet | Stores the result set of select queries |
| private Statement | statement | Stores the jdbc statement |
| private int | subjectsIDAutoNumber | Used to store and generate the primary key of tblSubjects |
| private int | timetableIDAutoNumber | Used to store and generate the primary key of tblTimetable |

### Methods

|  |  |  |
| --- | --- | --- |
| Type | Methods and Parameters | Description |
|  | [Database](file:///C:\Users\jmare\Documents\School\GR%2011%20IT\Java%20Projects\IT%20PAT\doc\Database.html#Database--)() | Constructor |
| + void | approveTutor(String username) | Used by admins to accept prospective tutors. |
| + void | closeConnection() | This closes the database connection, is runned in Main's main ShutDownHook |
| + int | countUsername(String usrName) | Used in Main to see if the username is unique |
| + void | deletePeerRelation(String tutorUsername, String studentUsername) | used if a user is deleted from the database |
| + void | deleteUser(String usrName) | Deletes any user and all their associated tables and data. |
| + void | deleteUserSubjectList(String username) | Deletes the subject records for the user but not their ID. |
| + String | getAge(String usrName) | Calculates and returns the User's age based on the year. |
| + String | getAllStudents() | Used by Admins to see all the students in the database. |
| + String | getAllTutors() | Used by both admins and students to see all the tutors that have admin approval in the database. |
| + String | getBDay(String usrName) | Returns the user’s date of birth. |
| + String | getCellphone(String usrName) | Returns the user’s cell phone number. |
| + String | getFaction(String usrName) | Returns the user’s faction. |
| + String | getGrade(String usrName) | Returns the user’s grade. |
| + String | getName(String usrName) | Returns the user’s name |
| + String | getPassword(String usrName) | Returns the user’s password |
| + String | getPicture(String usrName) | Returns the picture path stored in the database. |
| + String | getProspectiveTutors() | Used by Admins to see all the tutors that are new in the database needing admin approval. |
| + String | getRequestingStudents(String username) | Used by tutors to see which students are requesting tutoring. |
| + int | getSubjectID(String usrName) | Returns the user's subject ID |
| + String | getSubjects(String usrName) | Returns the user’s subjects as a comma delimited string. |
| + String | getSurname(String usrName) | Returns the user’s surname. |
| +boolean | getTableData(String usrName, int row, int column) | Returns a specific user timetable value. |
| + int | getTimetableID(String usrName) | Returns the user's timetable ID |
| + String | getUserStudents(String username) | Used by tutors to see which students they are already tutoring. |
| + String | getUserTutors(String username) | Used by students to see which tutors are currently tutoring them |
| + void | insertPeerRelation(String tutorUsername, String studentUsername, boolean newValue) | used by students to request tutoring |
| + void | insertUser(String uName, String pWord, String fction) | Used in registration only to insert a user into the database |
| + void | insertUserSubjectList(String username, String[] subjectArray) | Inserts a tutor or student subject list and gives them an appropriate ID. |
| + void | insertUserTable(String username, JTable timetable) | Inserts an empty array. |
| +boolean | isPaired(String tutorUsername, String studentUsername) | Used to see if there is already a Peer relationship between the student and user in the database |
| +boolean | isPopulated(String usrName) | Used to see if the user has already completed their user profile. |
| + void | updateAdmin(String username, String name, String surname, String bDate, String grade, String cphone) | Updates tblUser with the Admin User’s values. |
| + void | updatePRRequest(String tutorUsername, String studentUsername, boolean newValue) | used by tutors to decline requesting students or students to re-request tutors |
| + void | updateStudent(String username, String name, String surname, String bDate, String grade, String[] subjectArray, String cphone, String filePath) | Updates tblUser with the Student User’s values. |
| + void | updateTutor(String username, String name, String surname, String bDate, String grade, String[] subjectArray, String cphone, String filePath, JTable timetable, int timetableID) | Updates tblUser with the Tutor User’s values. |
| + void | updateUserTable(String username, JTable timetable) | Changes any already inserted timetables. |

## ‘Main’ Class

### Fields

|  |  |  |
| --- | --- | --- |
| **Modifier and Type** | **Field** | **Description** |
| static JFrame | currentFrame | Stores the frame currently displayed |
| static boolean | isEmpty | True if profile parameters are empty |
| static MasterFrame | master | The parent frame that instantiates the GUI Screens. |
| static JFrame | nextFrame | Stores the frame to be displayed after confirmation dialogue is shows |
| static Database | storage | A database object to run queries from |
| static boolean | tooManySubjects | Checks if the user has entered too many subjects in their profile |
| static String | username | Stores the username of the active user. |

### Methods

|  |  |  |
| --- | --- | --- |
| **Type** | **Methods and Parameters** | **Description** |
| static void | btnAdmnApplyPressed(String name, String surname, JDateChooser dateChooser, String grade, String cellphone) | Admin info received as parameters , is validated and verified and updated into database. |
| static void | btnLogInPressed(String uNameText, char[] pWord) | Checks the LogInGUI's fields for verification and changes the user's screen. |
| static void | btnRegisterPressed(String uNameText, char[] pWord\_1, char[] pWord\_2, String fction) | Checks the RegisterGUI's fields for verification, changes the user's screen, calls method to insert user details into database. |
| static void | btnStudApplyPressed(String name, String surname, JDateChooser dateChooser, String grade, String[] subjects, String cellphone, String filePath) | Student info received as parameters , is validated and verified and updated into database. |
| static void | btnTutApplyPressed(String name, String surname, JDateChooser dateChooser, String grade, String[] subjects, String cellphone, String filePath, JTable timetable) | Tutor info received as parameters , is validated and verified and updated into database. info into database. |
| static void | main(String[] args) | The method which starts the program |

# *Persistent Storage Design*

## tblUser

### Table:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Names | Username | Password | Faction | Name | Surname | DateofBirth | Cellphone | PicturePath | NeedsAdminApproval | SubjectID | TimetableID |
| Type: | Text | Text | Text | Text | Text | Date | Text | Text | Boolean | Integer | Integer |
| Keys: | Primary |  |  |  |  |  |  |  |  | Foreign | Foreign |
|  |  |  |  |  |  |  |  |  |  |  |  |

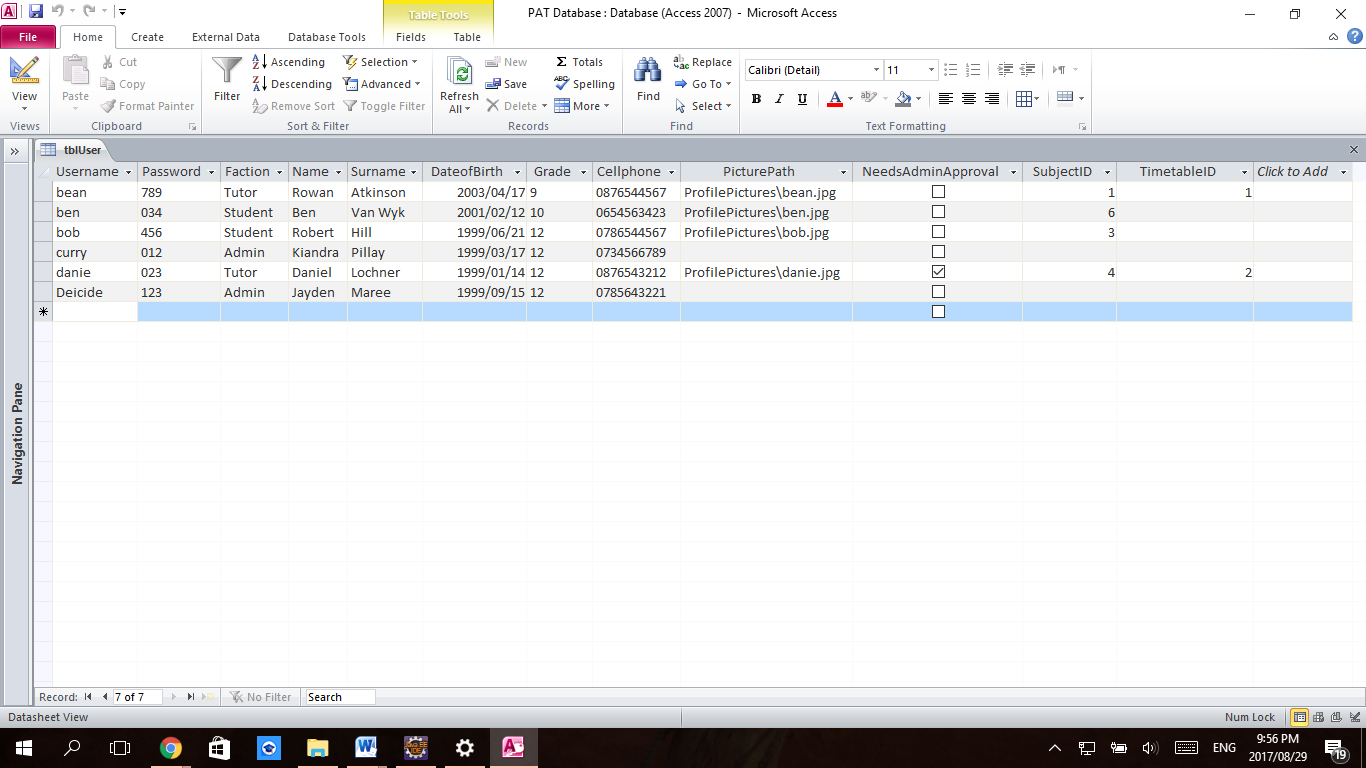
### Formatting:

DateofBirth 🡪 yyyy/MM/dd

Cellphone 🡪 digits only, no spaces.

PicturePath 🡪 Folder\Username.jpg

### Sample Data:



## tblPeer

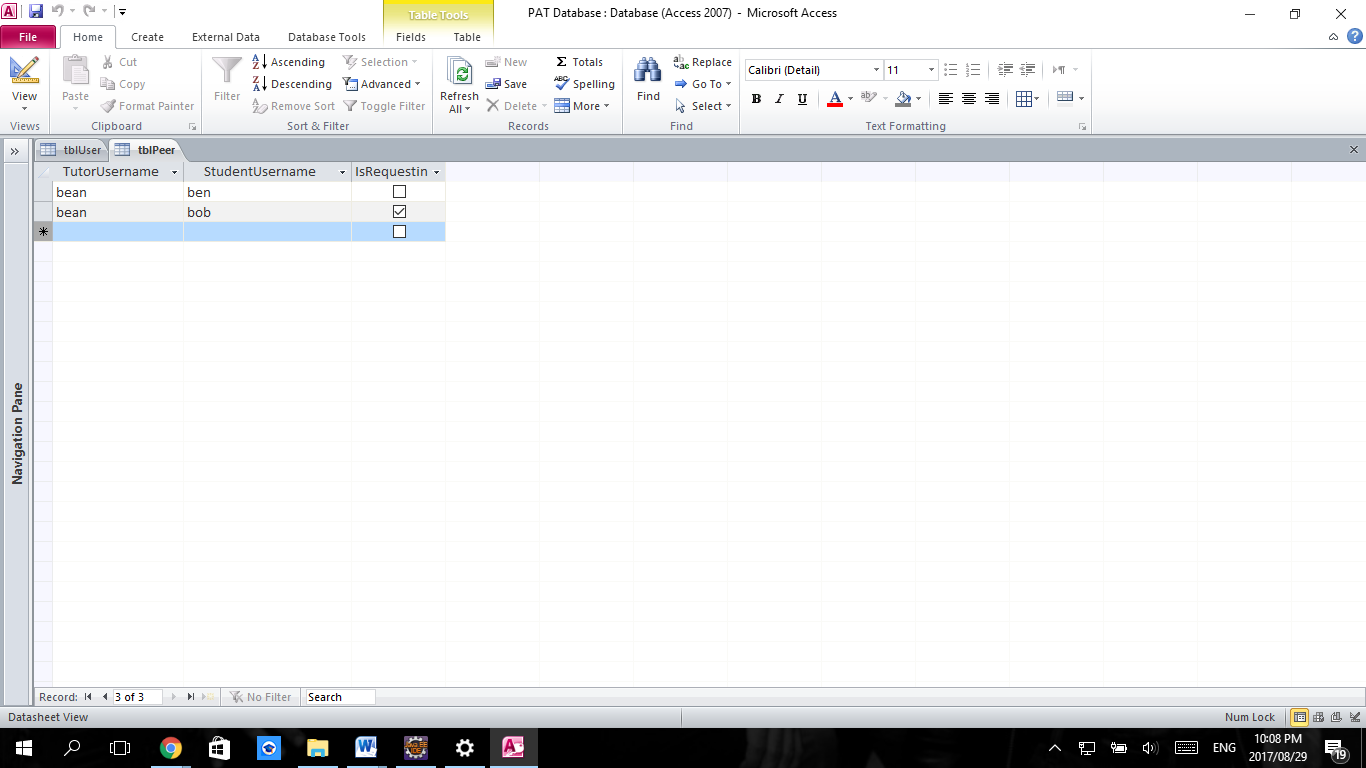
### Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Field Names | TutorUsername | StudentUsername | isRequestingTutoring |
| Type: | Text | Text | Boolean |
| Keys: | Foreign | Foreign |  |
|  | Composite Key | |  |

### Formatting:

none

### Sample Data:



## tblTimetable

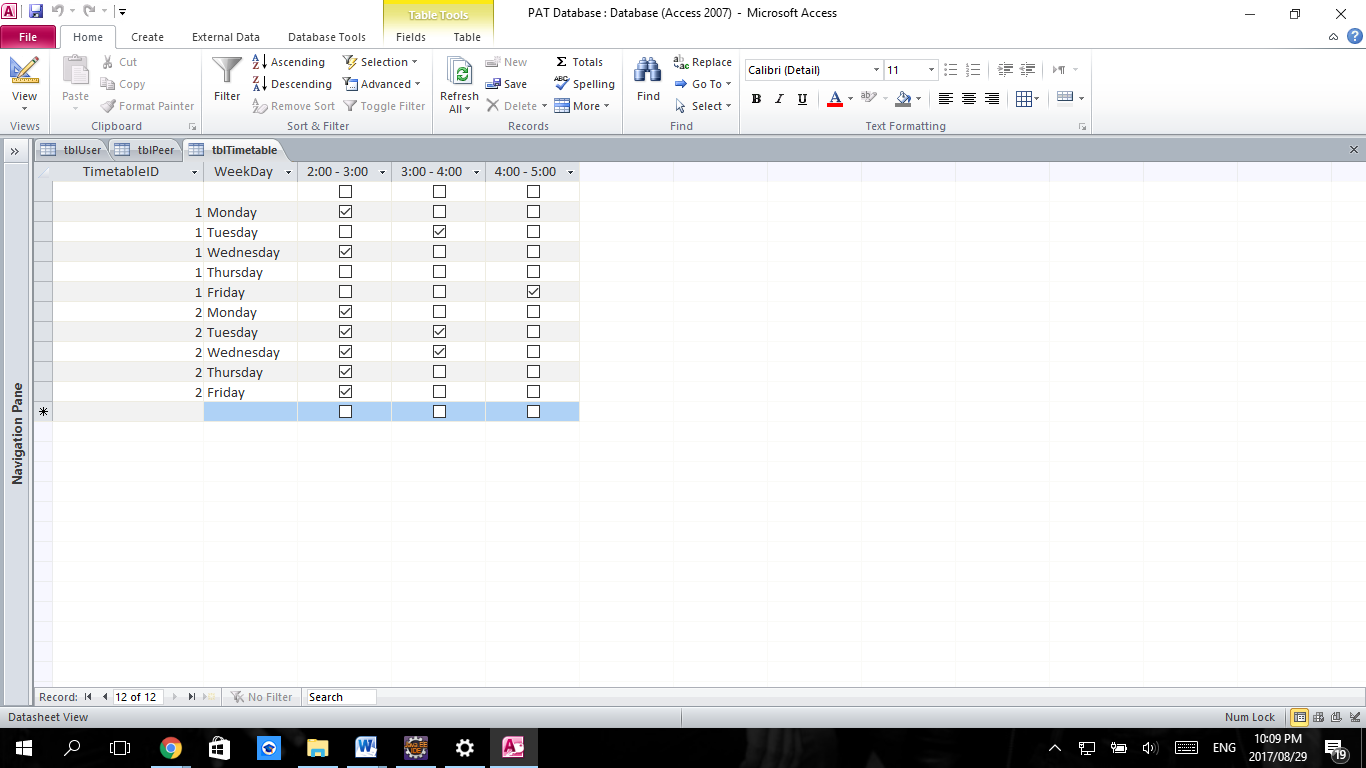
### Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Names | TimetableID | WeekDay | 2:00 – 3:00 | 3:00 – 4:00 | 4:00 – 5:00 |
| Type: | Integer | Text | Boolean | Boolean | Boolean |
| Keys: | Primary |  |  |  |  |

### Formatting:

none

### Sample Data:



## tblSubjects

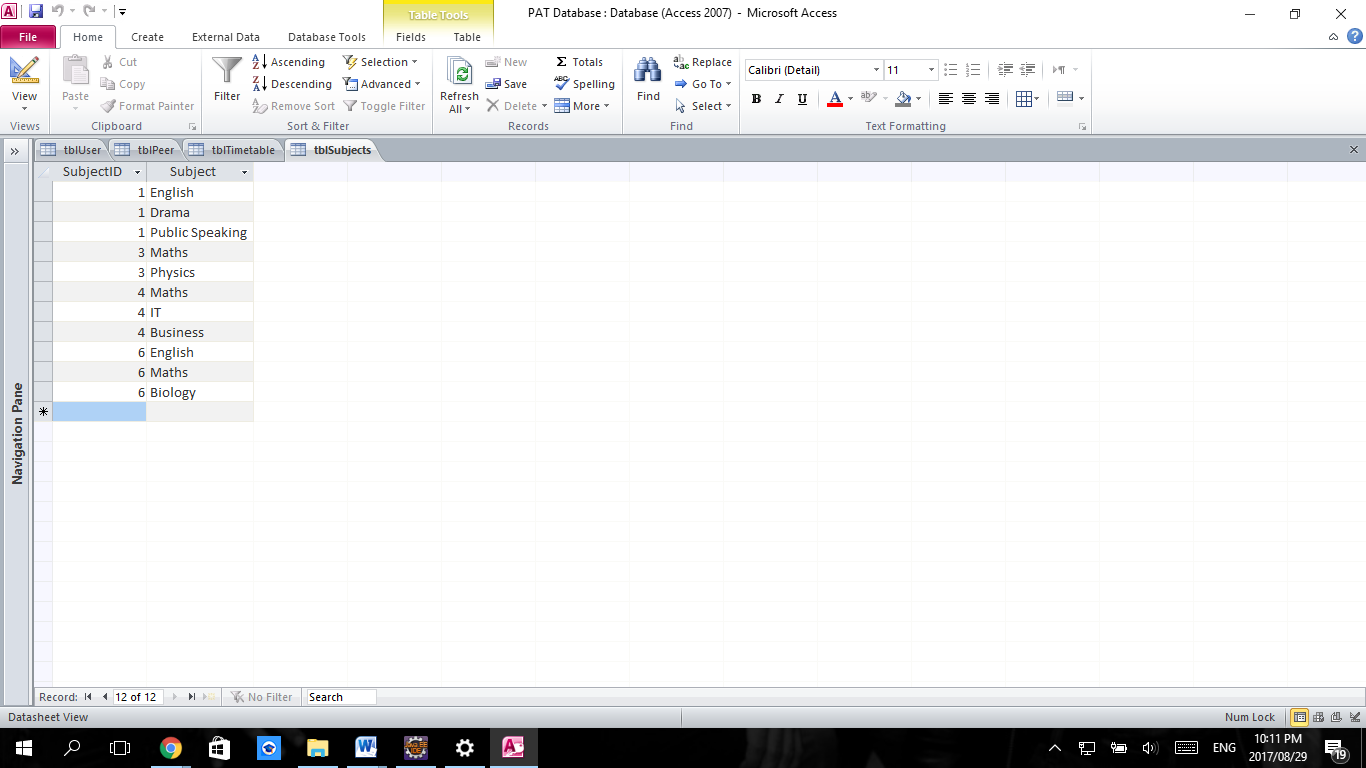
### Table:

|  |  |  |
| --- | --- | --- |
| Field Names | SubjectID | Subject |
| Type: | Integer | Text |
| Keys: | Foreign |  |

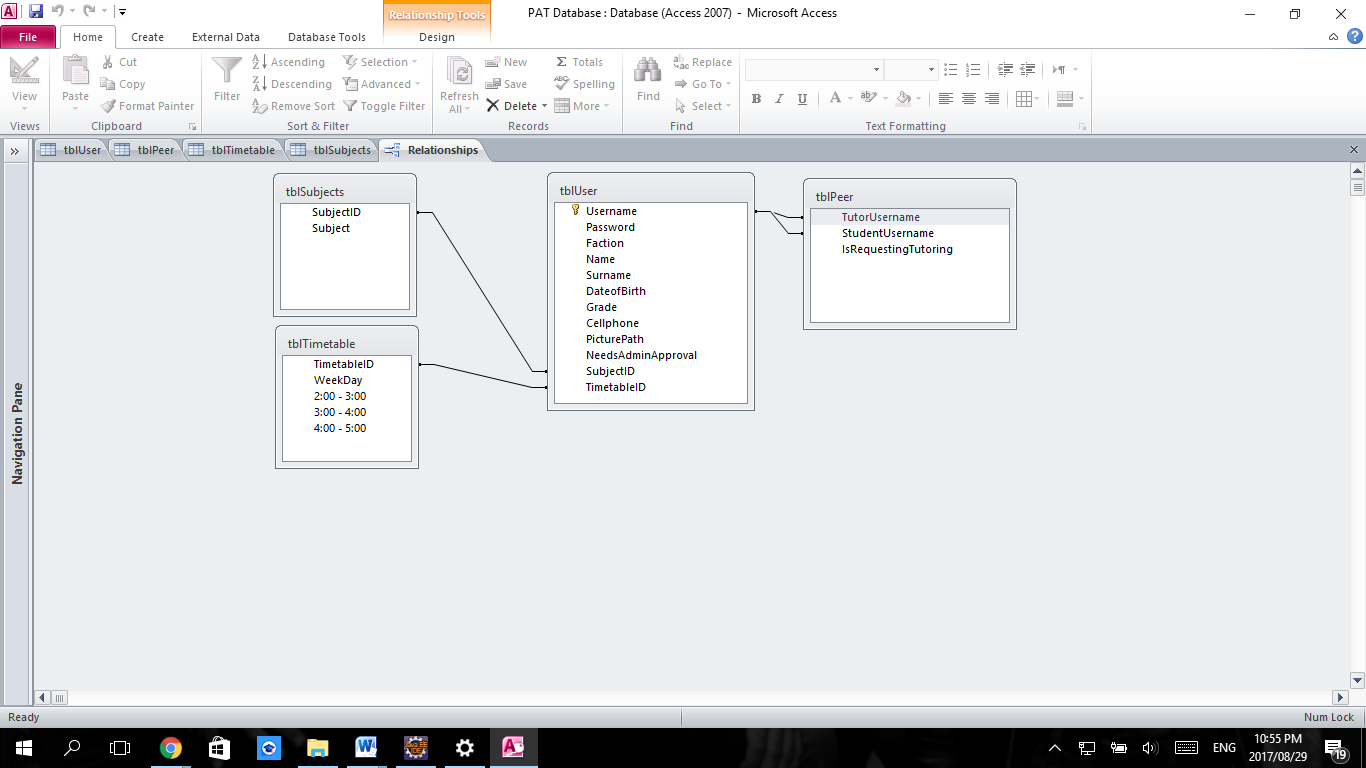
### Formatting:

none

### Sample Data:



# Relationship Diagram



# *Explanation of Storage Design*

My program does not make use of any text files for data storage. It makes use of a database to store all user data except the user’s profile picture. All data stored is stored in the project folder to allow the program to be run remotely from any terminal. Pictures uploaded by the user get stored in the project ‘ProfilePictures’ folder, all other user data is stored in the database ‘PAT Database’. All data is entered through the program and no user needs to access the database or understand how it works to store their data.

The database was used because it allowed the use of SQL statements and functions which offered the easiest way perform verification and validation checks such as if a username has already been entered into a database. The database data is also retrieved based on whichever user logs in, this is where SQL queries allow the easiest access to any unknown individual’s data because of the use of the SELECT statements and WHERE clauses.

Databases seem to be the ideal solution for running the program on a larger scale, since the program could grow to maintain a large number of users and the database would allow for organized, scalable and long term storage of the data. A text file could provide many of the same functions but text files have less security than a MS Access database file and would require additional coding in java to include features that are already included in the database’s properties and in the SQL language.