Variable explanations

There are numerous variables used in the project. Sometimes there are variables used several times across different pages, or indeed in the same page where they perform different tasks and may originate from different sources. One example is when a user adds a request for another. The entries for the current user's ID and type come from the user's session variables, whereas when rejecting a request from another user these values originate from the database because the requester's details are not known beforehand. Both tasks exist in the same page, but do significantly different actions. This is why there may be the requirement to explain the variables for each case.

The use of session variables may be used to store information without forwarding it through insecure methods: GET or POST requests. Variables labelled with the 'SESSION' prefix are usually identical to the page's variable and its value. For example, UserType is often used in conjunction with SESSION['UserType']. They perform the same task, except that the session variable is used to store the information during the session as opposed to during the page being browsed at a given time. The standard variable used in the page is easier to program and sometimes must be loaded only during loading of the page. The use of StudentID in ViewMeetingPage varies depending on the current user. If a student views it, then their session variable for UserID can be used. If a mentor views it then their UserID is used to find students. The StudentID values here are fetched in the search.

LoginPage

- UserLogin

The login username of the user. This is unique for each user for a given user type. Apostrophes and quotation marks are unsuitable for us with this field, as with any other field. This is because of the format programming languages often follow within their syntax.

- UserPassword

The password of the user for use with logging into their user account. This may consist of any suitable character, with the exception of apostrophes and quotation marks.

- UserType

The type of user. This is an integer value. '1' represents a student, '2' represents a parent and '3' represents a mentor.

DatabaseConnectPage

- DatabaseHost

The host server. This is given the name 'localhost' because the files are local on the server.

Variable explanations

- DatabaseUser

The user login for accessing the server's databases. In this case it is 'sachasimon'.

- DatabasePassword

The password used for accessing the server's databases. In this case it is 'qp12zm'.

- ConnectServer

This is an action to connect to the database on the server. In the case of success, the database can then be queried and have entries added to its tables. If not, report an error message to confirm that database is inaccessible.

- DatabaseName

The name of the database. Note that it is different to the login. Here it is also named 'sachasimon'.

ProcessLoginPage

- UserLogin

The user login to use for searching the database. This value is then assigned to StudentLogin/ParentLogin/MentorLogin depending on the value for UserType.

- UserPassword

The user password to use for searching the database. As with the login fields, this field will then be assigned to its respective user type.

- UserType

The type of user. This decides which table to search. A value of '1' will result in the querying for student details from StudentTable, a value of '2' will result in the querying for parent details from ParentTable and a value of '3' will result in the querying for mentor details from MentorTable.

- logsql

Defines the query for the SQL search. Consists of criteria and table directories.

- logquery

Transforms the SQL into a query.

- logarray

Obtains the information and stores it into memory as an array.

- StudentLogin

Student's login username.

- StudentFirstName

Student's first name.

Variable explanations

- StudentSurname

Student's surname.

- StudentID

Student's ID.

- StudentForm

Student's form group.

- StudentEmail

Student's email address (optional).

- StudentPhoneNumber

Student's phone number (optional).

- ParentLogin

Parent's login username.

- ParentFirstName

Parent's first name.

- ParentSurname

Parent's surname.

- ParentID

Parent's ID.

- ParentEmail

Parent's email address (optional).

- ParentPhonenumber

Parent's phone number (optional).

- MentorLogin

Mentor's login username.

- MentorFirstName

Mentor's first name.

- MentorSurname

Mentor's surname.

- MentorID

Mentor's ID.

Variable explanations

- MentorEmail

Mentor’s email address (optional).

- MentorPhoneNumber

Mentor’s phone number (optional).

- UserTypeName

Name of the user type (student/parent/mentor).

- UserFirstName

User’s first name.

- UserSurname

User’s surname.

- UserEmail

User’s email address (optional).

- UserPhoneNumber

User’s phone number (optional).

- LoggedIn

Defines if the session is logged in or not. Used primarily to notify the user if they are logged in or not.

LoggedInPage

- UserType

The type of user. Decides which details to display.

- LoggedIn

Defines whether current user is logged in or not.

- UserLogin

The user login name for the current user.

- UserFirstName

User's first name. Used in displaying the name of the current user.

- UserSurname

User's surname. Used in displaying the surname of the current user.

- UserEmail

Variable explanations

User's email address, if existent, will be displayed to the user.

- UserPhoneNumber

User's phone number, if existent, will be displayed to the user.

- UserID

User's ID is displayed to the user in this page.

- UserTypeName

User's type name is displayed to the user as a word, each possibility being 'student', 'parent' and 'mentor'.

- StudentForm

The form group of the student.

ViewMeetingPage

- LoggedIn

Confirms if the session still exists. If not, the user is notified of this.

- UserID

User's ID. Used in conjunction with UserType to find meetings of current user.

- UserTypeName

Name given to type of user.

- UserType

Type of user.

- UserFirstName

First name of user.

- UserSurname

Surname of user.

- studentno

In the context of students, this is automatically adjusted to one student. Parents and mentors may have several students, so this is used to store each student property as an array. Each student is displayed followed by each of their meetings.

- totalstudentno

Total number of students. For students, this is set to 1. For parents and mentors this value is the total number of students they have a connection with. This value is decremented by 1 before the meetings of each student are fetched due to a bug in the program when a parent or mentor browses this page. I have not been able to resolve this.

- StudentFirstName

Variable explanations

Student's first name is displayed for each student retrieved and for meetings of each student. This may be an array if more than one student is fetched.

- StudentID

The student's ID may already exist for student browsers. This is used solely in regard to them. Parents and mentors require their student's ID for each student they have. This enables the meetings of each student to be viewed. This is stored as an array.

- ParentIDsql

Students and mentors have a query for finding the parent information.

- ParentIDquery

SQL query for the above SQL definition.

- ParentIDarray

ID of the parent returned from the query.

- ParentID

ID of the current parent.

- Parentsql

SQL definition for searching parent properties.

- Parentquery

Query for parentsql.

- Parentarray

Array for parent results fetched.

- MentorIDsql

ID query SQL for MentorTable.

- MentorIDquery

SQL query for the above definition.

- MentorIDarray

ID of the mentor returned from the query.

- MentorID

ID of the current mentor.

- Mentorsql

SQL definition for searching mentor properties.

- Mentorquery

Query for mentorsql.

- Mentorarray

Array for mentor results fetched.

- studentsql

Variable explanations

SQL statement for students.

- studentvalidationquery

Query for verifying if student found.

- studentarray

Array for any results fetched from queries.

- studentquery

Query for fetching students from StudentTable.

- meetcheck

Verifies whether to check for meetings or not.

- StudentFirstName

First name of each student. Stored as an array.

- StudentSurname

Surname of each student. Stored as an array.

- StudentID

ID of student. Stored as an array.

- meetsql

SQL query to define the query for MeetingTable.

- meetqueryvalidate

Validation query for the validation for whether a meeting exists.

- meetarray

Fetches results using any meeting query.

- meetquery

Query for meetsql to find existing meetings.

- meetno

Meeting number in for loop.

- MeetingID

ID of current meeting. All MeetingID values are stored in an array.

- MeetingDate

Date of meeting.

- yyyy

Year in date display.

- mm

Month in date display.

Variable explanations

- dd

Day in date display.

- datetime

MeetingDate is a number which is unsuitable for use with the interface so the variable datetime is used to display the date appropriately.

AddMeetingPage

- UserType

Decides if type of user is a mentor. If not then the user is redirected back to ViewMeetingPage.

- action

Decides whether to display inputs or actually add a record to MeetingTable.

- day

Mentor input for day of date to insert.

- month

Mentor input of month of date to insert.

- StudentID

ID of student selected by mentor to add a meeting for.

- yy

Current year; mathematically recalculated to fit current academic year. Year to insert into MeetingDate.

- maxday

Maximum number of days in given month.

- dd

Day to add into MeetingDate.

- mm

Month to add into MeetingDate.

- currentmonth

Current month in calendar when viewed.

- MeetingDate

Date of meeting. Format is yyyy-mm-dd.

- UserID

Variable explanations

ID of mentor.

- confirmsql

SQL query to confirm that mentor is eligible for adding meeting into MeetingTable for selected student.

- confirmquery

Query for SQL.

- confirmarray

Array of result fetched.

- addsql

SQL statement of meeting to be added into MeetingTable with specified fields.

- addquery

Variable for adding record into MeetingTable.

- formno

Iterative variable used in for loop for fetching mentor's students.

- studentsql

SQL for fetching student information.

- studentquery

Query variable for SQL query for students.

- studentarray

Array variable for storing information on each student.

EditMeetingPage

- UserType

Variable defining the type of user. Mentors may access this page only.

- MeetingID

ID of meeting to edit.

- action

Decides which task to perform: display the menu for editing meeting or to overwrite meeting details.

- day

Variable explanations

Retrieved by fetching data from MeetingTable. Also used as a mentor input for day of MeetingDate.

- month

Retrieved by fetching data from MeetingTable. Also used as a mentor input for month of MeetingDate.

- StudentID

Retrieved by fetching data from MeetingTable. Also used as a mentor input for StudentID of selected student.

- yy

Year of MeetingDate fetched from MeetingDate/current year within academic year.

- maxday

Maximum number of days in given month.

- dd

Day to insert into MeetingDate.

- mm

Month to insert into MeetingDate.

- currentmonth

Month of current date.

- MeetingDate

Date of meeting.

- confirmsql

SQL query - confirms that student selected is the correct student.

- confirmquery

Query for confirmsql.

- confirmarray

Array for storing fetched results. Used to confirm if student connection exists between mentor and student.

- editsql

SQL statement for editing meeting.

- editquery

Query for editing meeting; overwrites existing meeting details.

Variable explanations

- fetchsql

SQL query definition for retrieving information based on a certain meeting.

- fetchquery

Query for fetchsql.

- fetcharray

Array for storing information on fetchquery.

- yyyy

Year of meeting.

- datetime

Date of meeting for appropriate display to mentor.

- studentsql1

SQL statement for finding StudentID from MeetingTable entry with MeetingID.

- studentquery1

Query of studentsql1.

- studentarray

Array for storing information from queries for students.

- studentsql2

SQL query definition using StudentID to find student information.

- studentquery2

Query for studentsql2.

- studentsql

SQL statement for searching all the mentor's students. This is used for enabling the mentor to choose their student.

- studentquery

Query for studentsql.

- studentarray

Array for storing current student's details.

DeleteMeetingPage

- UserType

Variable explanations

Type of user. Only mentors are allowed to use this page.

- action

Decides between displaying a menu and deleting the meeting from MeetingTable.

- UserID

ID of mentor.

- MeetingID

ID of meeting to be deleted.

- confirmsql

SQL statement for verifying that meeting is related to mentor.

- confirmquery

Query for confirmsql.

- confirmarray

Array for storing any results for confirmsql. In this case only 1 result is sufficient.

- deletesql

SQL statement for deleting the meeting from MeetingTable.

- deletequery

Query for deletesql; deletes the meeting from MeetingTable.

- day

Day of MeetingDate; day of meeting.

- month

Month of MeetingDate; month of meeting.

- StudentID

ID of current student.

- fetchsql

SQL statement to fetch information on meeting using MeetingID.

- fetchquery

SQL query for fetchsql.

- fetcharray

Array for storing results from fetchquery.

Variable explanations

- yyyy

Year of meeting.

- mm

Month of meeting.

- dd

Day of meeting.

- datetime

Displayable date to mentor.

- studentsql1

SQL statement to check if MeetingID and StudentID have a match.

- studentquery1

Query for studentsql1.

- studentarray

Array for storing information on queries for student information.

- studentsql2

SQL statement for searching other student information from StudentTable.

- studentquery2

Query for studentsql2.

ViewTargetPage

- UserID

ID of current user.

- UserType

Type of user.

- userlinked

Verifies if meeting is relevant to current user.

- MeetingID

ID of meeting currently being viewed.

- confirmsql

SQL statement to find meeting where MeetingID and relevant UserID match (parents require a StudentID instead of their own).

Variable explanations

- confirmquery

Query for confirmsql.

- confirmarray

Array for storing results on confirmquery search.

- StudentID

Student's ID (when browsing as a student or parent).

- studentsql

SQL query for finding a relevant student for a parent.

- studentquery

SQL query for studentsql.

- studentarray

Array for storing results from studentquery.

- MentorID

Mentor's ID (when browsing as a mentor).

- targetsql

SQL statement for searching targets from TargetTable.

- targetqueryvalidation

Query for verifying whether a target is to be found. Query based on targetsql.

- targetarray

Array for storing information from a search for targets.

- targetquery

Query for searching for targets based on targetsql.

- targetno

Target number used in for loop for searching each target.

- DateDue

Date target is due.

- yyyy

Year of target due date.

- mm

Variable explanations

Month of target due date.

- dd

Day of target due date.

- datetime

DueDate in a format to display to the user.

- TargetUserName

Specifies whether the target is set for the student, parent or mentor related to the meeting.

AddTargetPage

- UserType

Type of user. Mentors may access this page but students and parents may not.

- MeetingID

ID of meeting the target will be added to.

- UserID

ID of mentor.

- validatemeetingsql

SQL statement to verify whether mentor and target match.

- validatemeetingquery

Query for validatemeetingsql.

- validatemeetingarray

Array for storing information for validatemeetingquery.

- action

Decides whether to display the menu or add a target to TargetTable.

- TargetUserType

Mentor input. Defines which user involved in the meeting the target is set for.

- UserTarget

Mentor input. This is the target that the user must complete.

- day

Mentor input. This is the day of the date the target is due (for DateDue).

Variable explanations

- month

Mentor input. This is the month of the date the target is due (for DateDue).

- MentorComment

Mentor input. Optional but additional information in regard to the target may be added here.

- yyyy

Year of the date the target is due (for DateDue).

- maxday

Maximum number of days in given month.

- dd

Day for input into DateDue.

- mm

Month for input into DateDue.

- currentmonth

Current month of which the target is added (month target is added in).

- currentday

Current day of which target is added (day target is added in).

- DateDue

Date target is due in for. It must be no earlier than current day in the academic year and may be no later than the limit (30th June).

- addsql

SQL statement for adding target with set fields.

- addquery

Query for addsql - adds target to TargetTable.

- StudentComment

Student's comment for target. Automatically set to blank.

- ParentComment

Parent's comment for target. Automatically set to blank.

- TargetMetYet

Defines whether a target is set. When adding a target this is set to '0'. This defines the target as being incomplete.

Variable explanations

- TargetID

ID of target to be added.

- displayparent

If a parent exists, offer the option to set a target for the student's parent.

- studentidsql

SQL statement for finding the ID of the student in given meeting.

- studentidquery

Query for studentidsql.

- studentidarray

Array for storing any result from studentidquery.

- studentsql

SQL query for finding student's other details including name and ID.

- studentquery

Query for studentsql.

- studentarray

Array for storing information retrieved from studentquery.

- parentidsql

SQL query for finding ID of parent related to meeting using StudentID.

- parentidquery

Query for parentidsql.

- parentidarray

Array for storing retrieved data from parentidquery.

- parentsql

SQL statement for finding other parent details if a ParentID entry found in StudentTable for current student.

- parentquery

Query for parentsql.

- parentarray

Array for storing information retrieved from parentquery.

Variable explanations

EditTargetPage

- UserType

Type of user. Only mentors may access this page.

- MeetingID

ID of meeting current target belongs to.

- UserID

Mentor's ID.

- validatemeetingsql

SQL statement to verify whether mentor and target match.

- validatemeetingquery

Query for validatemeetingsql.

- validatemeetingarray

Array for storing information for validatemeetingquery.

- TargetID

ID of current target being edited.

- targetsql

SQL statement to fetch information on current target.

- targetquery

Query for targetsql.

- targetarray

Array for storing data retrieved from targetquery.

- action

Decides whether to display the editing menu or overwrite existing meeting.

- TargetUserType

Mentor input. Defines which user involved in the meeting the target is set for.

- UserTarget

Mentor input. This is the target that the user must complete.

- DateDue

Date target is due.

Variable explanations

- StudentComment

Student's comment on current target.

- ParentComment

Parent's comment on current target.

- MentorComment

Mentor input. Mentor's comment on current target.

- day

Mentor input. This is the day of the date the target is due.

- month

Mentor input. This is the month of the date the target is due.

- TargetMetYet

Mentor input. Defines whether a target is complete or not.

- yyyy

Year of target due date. Loaded into memory upon loading of target details. Also input into DueDate for current academic year.

- maxday

Maximum number of days in given month.

- dd

Day for input into DateDue.

- mm

Month for input into DateDue.

- currentmonth

Current month of which the target is added (month target is added in).

- currentday

Current day of which target is added (day target is added in).

- editsql

SQL statement for updating current target with new details.

- editquery

Query for editsql.

- displayparent

Variable explanations

If a parent exists, offer the option to set a target for the student's parent.

- studentidsql

SQL statement for finding the ID of the student in given meeting.

- studentidquery

Query for studentidsql.

- studentidarray

Array for storing any result from studentidquery.

- studentsql

SQL query for finding student's other details including name and ID.

- studentquery

Query for studentsql.

- studentarray

Array for storing information retrieved from studentquery.

- parentidsql

SQL query for finding ID of parent related to meeting using StudentID.

- parentidquery

Query for parentidsql.

- parentidarray

Array for storing retrieved data from parentidquery.

- parentsql

SQL statement for finding other parent details if a ParentID entry found in StudentTable for current student.

- parentquery

Query for parentsql.

- parentarray

Array for storing information retrieved from parentquery.

DeleteTargetPage

- UserType

This page is for access to mentors only. This variable is fed in what type of user has requested this page. Students and parents are redirected back to ViewTargetPage.

Variable explanations

- MeetingID

ID of meeting target belongs to.

- UserID

ID of user browsing page.

- validatemeetingsql

SQL statement to verify whether mentor and target match.

- validatemeetingquery

Query for validatemeetingsql.

- validatemeetingarray

Array for storing information for validatemeetingquery.

- TargetID

ID of target to be deleted.

- targetsql

SQL statement to retrieve target information.

- targetquery

Query for targetsql.

- targetarray

Array for storing data retrieved from targetquery.

- action

Decides which action to take; ‘0’ represents the action to display deletion prompt and ‘1’ decides to delete the target from TargetTable.

- deletesql

SQL statement to define the query to delete target from TargetTable using TargetID.

- deletequery

Query for deletesql – actually deletes the target.

- currentday

Unused variable.

- currentmonth

Unused variable.

Variable explanations

- day

Unused variable.

- month

Unused variable.

- yyyy

First reset, this variable is then loaded into memory as part of the year the date the target is due (Derived from DateDue).

- mm

This variable is reset upon loading, then loaded into memory from the month part of the date the target is due (derived from DateDue).

- dd

Also reset upon loading this variable represents the day in DateDue.

- displayparent

Unused variable.

- TargetUserType

Used to determine who the target was set for.

- DateDue

Date the target is due.

- StudentComment

This is the student's comment. Regardless of contents (including no value inserted) this is displayed to the mentor.

- ParentComment

This is the parent's comment. Regardless of contents (including no value inserted) this is displayed to the mentor.

- MentorComment

This is the mentor's comment. Regardless of contents (including no value inserted) this is displayed to the mentor.

- TargetMetYet

Determines whether the target is completed or if it is incomplete.

- datetime

Variable explanations

Uses yyyy, mm and dd to represent DateDue in a user-friendly manner.

CommentPage

- UserType

Type of user browsing this page. Each type of user has a respective comment input.

- UserID

ID of the user within their user type.

- TargetID

ID of the target the user is commenting on.

- MeetingID

The meeting the target is based on.

- targetlinked

Determines whether or not the meeting is relevant to the user. If not then the user is redirected to ViewTargetPage.

- confirmsql1

SQL statement containing the query definition for finding the meeting in MeetingTable with the target's MeetingID.

- confirmsql2

This varies depending on the type of user. Students and parents will use their value for StudentID. Mentors will have their MentorID in the search.

- confirmsql

Concatenates confirmsql1 and confirmsql2 to form a statement to search MeetingTable for the correct ID of the meeting and the student/mentor.

- confirmquery

Query for confirmsql.

- confirmarray

Array for retrieving data from confirmquery.

- studentsql

Parents only, uses ParentID in searching for any relevant students from StudentTable.

- studentquery

Query for studentsql.

- studentarray

Array for storing retrieved student IDs for meeting.

- studentno

Iterative variable. Used in a for loop. Each of the parent's students is returned from

Variable explanations

the search.

- StudentID

StudentID is used in verification whether the student/parent is eligible to comment on this target.

- MentorID

ID of mentor browsing the page. This is used to verify that the mentor commenting on the target is correct.

- findsql

Retrieves the comments from TargetTable using TargetID.

- findquery

Query for findsql.

- findarray

Array for storing results from query.

- action

decides whether to add, edit or delete the contents of the comment, or to display the menu.

- UserComment

This is the user's input comment. Any existing comment will be loaded for editing.

- commentsql1

Part of the SQL statement to update the comment.

- commentsql2

Defines which user type's comment is to be updated.

- commentsql3

Sets the comment to the input comment value for the current target.

- commentsql

SQL statement of commentsql1, commentsql2 and commentsql3 concatenated together. This essentially updates the existing contents of the user's comment for the current target.

- commentquery

Query for commentsql which updates the comment.

ViewContactPage

- UserType

Type of user. This enables the system to display an appropriate interface for each different user type.

- UserID

Variable explanations

ID of user within their user type.

- requesterdetaildisplay

Determines whether to display requests from other users to the current user.

- requesteddetaildisplay

Determines whether to display requests for other users to the current user.

- StudentID

Students: this is their ID. Parents and mentors: this is the current student being searched for in the student queries.

- parentsql

SQL statement for searching any parent of the student using the student's ID.

- parentquery

Query for parentsql.

- parentarray

Array for storing results from parentquery.

- ParentFirstName

Current parent's first name.

- ParentSurname

Current parent's surname.

- ParentID

Current parent's ID.

- ParentEmail

Current parent's email address.

- ParentPhoneNumber

Current parent's phone number.

- mentorsql

SQL statement for searching StudentTable using student's StudentID value for a MentorID.

- mentorquery

Query for mentorsql.

- mentorarray

Variable explanations

Array for storing results from mentorquery.

- MentorFirstName

Current mentor's first name.

- MentorSurname

Current mentor's surname.

- MentorID

Current mentor's ID.

- MentorEmail

Current mentor's email address.

- MentorPhoneNumber

Current mentor's phone number.

- studentsql

SQL statement for searching the parent's/mentor's student in StudentTable using their value for ParentID/MentorID.

- studentvalidationquery

Query for studentsql. Verifies whether the parent/mentor has a student.

- studentquery

Query for studentsql. This is a new query to avoid omission of queried results.

- studentarray

Array for storing data for query searches for studentvalidationquery and studentquery.

- studentno

Iterative variable. Also used as an array pointer for student properties.

- StudentFirstName

Array variable. Stores student first names.

- StudentSurname

Array variable. Stores student surnames.

- StudentID

Array variable. Stores student ID values.

- StudentForm

Variable explanations

Array variable. Stores student form groups.

- StudentEmail

Array variable. Stores student email addresses.

- StudentPhoneNumber

Array variable. Stores student phone numbers.

- requestedsql

SQL statement for searching for requests from other users.

- requestedquery

Query for requestedsql.

- requestno

Iterative variable. Used in a for loop where requests between the current user and other users are queried by a search in RequestTable.

- requestedarray

Array for storing results from requestedquery.

- RequesterUserType

Type of user who requested the current user.

- RequesterUserID

ID of the user who requested the current user.

- RequesterFirstName

First name of requester.

- RequesterSurname

Surname of requester.

- StudentForm

Form group of any student who requests current user.

- requestersql

SQL statement for finding requests the current user has made for other users. RequestTable is to be queried.

- requesterquery

Query for requestersql.

- requesterarray

Variable explanations

Array for storing results from requesterquery.

- RequestedUserType

Type of user the requested user is

- RequestedUserID

ID of the requested user.

- RequestedUserFirstName

First name of the requested user.

- RequestedUserSurname

Surname of the requested user.

- deleteuserid

ID of the parent/mentor to delete from StudentTable for a given StudentID.

- deleteusertype

Type of user to delete connection from in StudentTable for a given StudentID.

- confirmsql1

SQL statement fragment for checking StudentTable for the given StudentID value.

- confirmsql2

SQL statement fragment for the search where the ParentID/MentorID value is equal to deleteuserid.

- confirmsql

SQL statement concatenated from confirmsql1 and confirmsql2.

- confirmquery

Query for confirmsql.

- confirmarray

Array for storing results from confirmquery.

- deletesql1

SQL statement fragment for setting a field.

- deletesql2

SQL statement fragment for stating whether ParentID or MentorID is to be set.

- deletesql3

Variable explanations

SQL statement fragment for setting the parent/mentor's ID value to null for a given StudentID.

- deletesql

SQL statement concatenated from deletesql1, deletesql2 and deletesql3. This defines the statement to set the ParentID/MentorID value to NULL for a given StudentID.

- deletequery

Query for deletesql. Deletes ParentID/MentorID entry from StudentTable for a given student.

SearchContactPage

- UserType

Type of user the current user is. This enables the system to display an appropriate user interface to each user of their type.

- UserID

ID of current user.

- RequestedUserFirstName

First name of user to search. Ignored if left blank.

- RequestedUserSurname

Surname of user to search. Ignored if left blank.

- RequestedUserID

ID of user to search. Ignored if left blank.

- RequestedUserType

Type of user to search. Only applicable to students without either a parent or a mentor. If this is the case a value must be selected in order for the search to be possible. This value is not set if a student has both a parent and mentor. For other cases this is automatically set.

- StudentForm

Form group of student to be searched. Applicable to parents and mentors.

- firstnamefind

Method to search the first name of the user to request. Can be used for the name to begin with, end with or contain the text entered into RequestedUserFirstName.

- surnamefind

Variable explanations

Method to search the surname of the user to request. Can be used for the name to begin with, end with or contain the text entered into RequestedUserSurname.

- idfind

Method to search the ID of the user to request. Can be used for the name to match with or contain the integer entered into RequestedUserID.

- dosearch

This is to verify that a student may search for a user type they are missing a connection with. This triggers whether a student may use this page or not.

- otherusersql

SQL statement for finding ParentID and MentorID values in StudentTable for the StudentID value of the student.

- otheruserquery

Query for otherusersql.

- otheruserarray

Array for storing results from otheruserquery.

- parentexists

For a student, this determines whether or not the student already has a parent connection with them.

- mentorexists

For a student, this determines whether or not the student already has a mentor connection with them.

- studentformsql

Used for the parent and mentor interface. SQL statement for finding all possible student form groups to search.

- studentformquery

Query for studentformsql.

- studentformarray

Array for storing results from studentformquery.

- studentsql1

SQL statement fragment for searching StudentTable for the student details.

- studentsql2

SQL statement fragment – defines how the first name is to be searched.

Variable explanations

- studentsql3

SQL statement fragment – defines that the surname is to be found.

- studentsql4

SQL statement fragment – defines which method to search the surname.

- studentsql5

SQL statement fragment - defines that the StudentID is to be searched.

- studentsql6

SQL statement fragment - defines how to search StudentID.

- studentsql7

SQL statement fragment - determines whether to search for a particular form group.

- studentsql8

SQL statement fragment - defines which user type ID to search for in StudentTable (parent/mentor).

- studentsql

SQL statement which is concatenated from studentsql1, studentsql2, studentsql3, studentsql4, studentsql5, studentsql6, studentsql7 and studentsql8. This defines the search for any student in StudentTable matching any of the criteria for the input search and those who do not have a connection with the same user type as the current user.

- studentqueryvalidation

Verifies whether any result is obtained. If so, then continue the search. If not then report an error/notification of no results. Query for studentsql.

- studentquery

When studentqueryvalidation is successful, this query is defined to avoid omission of results in returning them.

- studentarray

Array for storing data from the queries studentvalidationquery and studentquery.

- studentno

Iterative variable. Used for looping the process of searching StudentTable and returning information for each student.

- parentsql1

Variable explanations

SQL statement fragment - defines the search for parent information from ParentTable and also defines that the parent's first name is to be searched.

- parentsql2

SQL statement fragment - defines how to search the first name of the parent.

- parentsql3

SQL statement fragment - defines that the surname of the parent is to be searched.

- parentsql4

SQL statement fragment - defines how to search the parent's surname.

- parentsql5

SQL statement fragment - defines that the parent's user ID is to be searched.

- parentsql6

SQL statement fragment - defines how to search the parent's user ID.

- parentsql

SQL statement which is concatenated from parentsql1, parentsql2, parentsql3, parentsql4, parentsql5 and parentsql6. This defines the search for parent details from ParentTable with the specified input criteria.

- parentqueryvalidation

Query for parentsql. This checks to see if any results can be fetched from the search. If results are obtained, parentquery is then used to find all results. If parentqueryvalidation is unsuccessful then the search will not continue and instead report an error or a notification of an unsuccessful search.

- parentquery

Query for parentsql.

- parentarray

Array for storing data returned from parentquery.

- parentno

Iterative variable. Used in a for loop. Each parent fulfilling the input criteria is returned with their information during the search.

- mentorsql1

SQL statement fragment - defines the search for mentor information from MentorTable and that mentor first names are to be searched.

- mentorsql2

Variable explanations

SQL statement fragment - defines how to search for the surname of the mentors.

- mentorsql3

SQL statement fragment - defines that the mentor's surname is to be searched.

- mentorsql4

SQL statement fragment - defines how to search the mentor's surname.

- mentorsql5

SQL statement fragment - defines that the mentor's ID is to be searched.

- mentorsql6

SQL statement fragment - defines how to search the mentor's ID.

- mentorsql

SQL statement concatenated from mentorsql1, mentorsql2, mentorsql3, mentorsql4, mentorsql5 and mentorsql6.

- mentorqueryvalidation

SQL query for mentorsql. This query determines whether or not a result is found in the search. If an unsuccessful search happens then notify the user with a message that the search did not return any results. If the search did return a result, use mentorquery to search each mentor which meets the criteria.

- mentorquery

Query for mentorsql.

- mentorarray

Array for storing data retrieved from the search mentorquery.

- mentorno

Iterative variable. Used in a for loop to enable the system to search and return each mentor relevant to the search.

AddRequestPage

- action

Decides whether to add a request for another user, cancel a request for another user or reject a request from another user.

- RequesterUserType

The type of user the requester is. If the requester is the current user this value is assigned to that of the session variable UserType. If the requester is a different user then this information is forwarded from another page via a GET request.

Variable explanations

- RequesterUserID

The ID of the requester. If the current user made the request then this variable's value is assigned to the same value as with the session variable UserID. If the requester is not the current user then this information is forwarded from another page.

- RequestedUserType

The type of user the requested user is. If the requested user is the current user this value is assigned to that of the session variable UserType. If the requested user is a different user then this information is forwarded from another page via a GET request.

- RequestedUserID

The ID of the requested user. If the current user was requested then this variable's value is assigned to the same value as with the session variable UserID. If the requested user is not the current user then this information is forwarded from another page.

- confirmsql

SQL statement which is set to verify that the current user is requesting another user they are not yet connected with.

- confirmquery

SQL query for confirmsql.

- confirmarray

Array for storing the data retrieved from confirmquery.

- checkexistingsql

SQL statement designed to check if this exact request exists before adding a new request.

- checkexistingquery

Query for checkexistingsql.

- checkexistingarray

Array for storing information from checkexistingquery.

- addsql

SQL statement which defines the query to insert a request into RequestTable with the information about who made the request and who the request is made for (if adding a request).

Variable explanations

- addquery

Query for addsql. Adds the request to RequestTable.

- removerequestsql

SQL statement for removing the request from RequestTable (if cancelling or rejecting a request).

- removequery

Query for removerequestsql. Removes the request from RequestTable.

AddContactPage

- RequestedUserType

This is the type of user who is requested. This is assigned the value from the session variable UserType.

- RequestedUserID

This is the ID of the user who is requested. This is assigned the value from the session variable UserID.

- RequesterUserType

This is the type of user who made the request. This value is forwarded from ViewContactPage.

- RequesterUserID

This is the ID of the user who made the request. This value is forwarded from ViewContactPage.

- requestsql

SQL statement for checking if the request exists in RequestTable.

- requestquery

Query for requestsql.

- requestarray

Array for storing data retrieved from requestquery.

- addsql

SQL statement for adding the parent/mentor ID to StudentTable for a given student. This is concatenated from addsql1, addsql2 and addsql3 for students.

- addquery

SQL query for addsql.

Variable explanations

- addsql1

SQL statement fragment for defining that StudentTable is to be updated.

- addsql2

SQL statement fragment for defining which field is to be altered in the query.

- addsql3

SQL statement fragment for defining that ParentID/MentorID is set to RequesterUserID for a given StudentID, which is the RequestedUserID.

- removerequestsql1

SQL statement for removing student-parent/student-mentor requests for/by the student from RequestTable.

- removequery1

SQL query for removerequestsql1.

- removerequestsql2

SQL statement for removing student-parent/student-mentor requests for/by the student from RequestTable.

- removequery2

SQL query for removerequestsql2.

LogoutPage

- LoggedIn

This variable is unset so that the session is identified as logged out and the session is eliminated.