Technical Support Manuel For TutorMe Android Application

Developed by: Conor Payton, Randy Hash, Michael Cantley

***Table of Contents:***

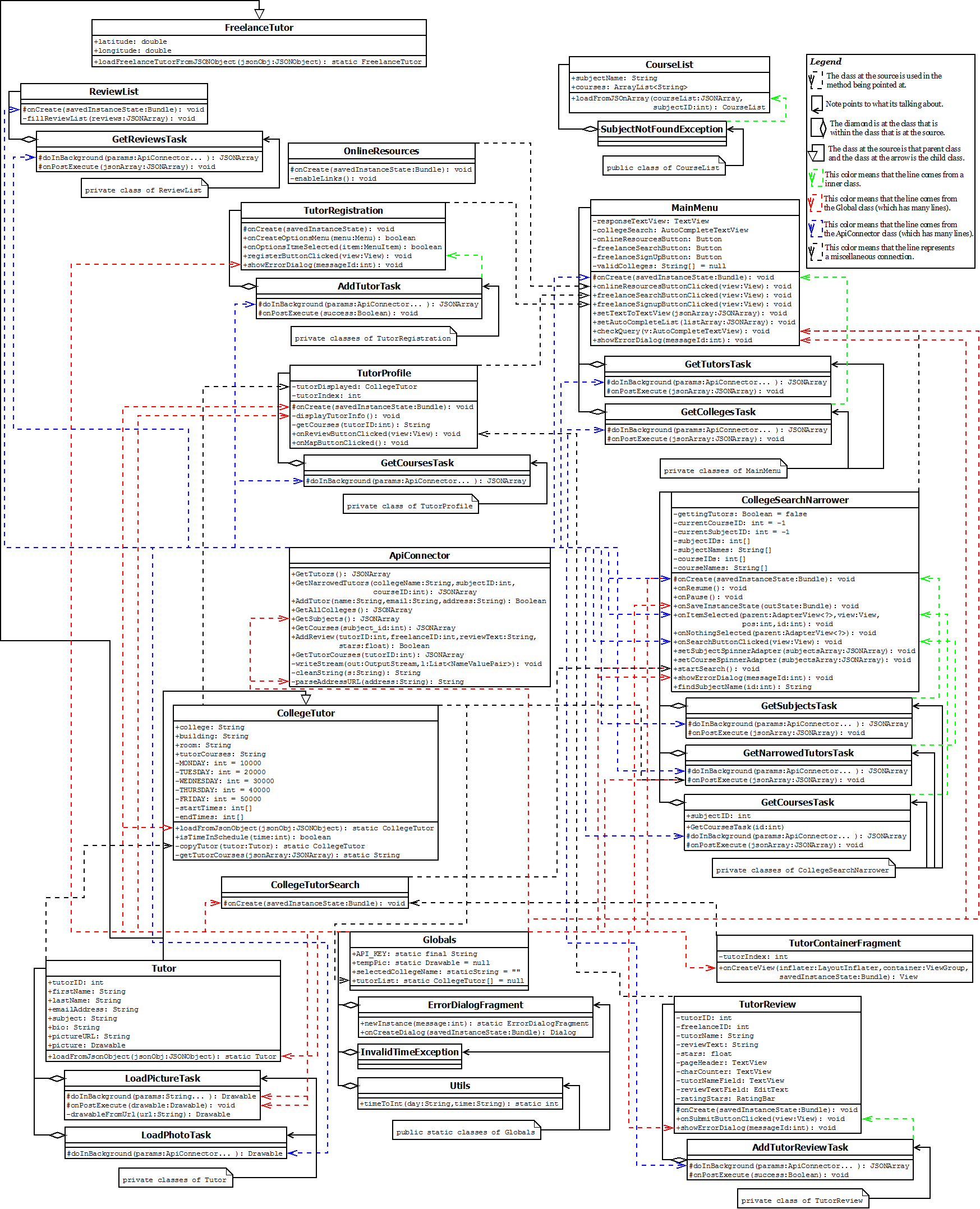
1. Overview of the Program
2. UML Diagram
3. Class and Their Methods Overviews

***1) Overview of the Program***

The purpose of this program is to provide users a convenient way of finding tutors both on a campus, or freelance tutors in their surrounding area. The application has been made with JAVA, JSON, MySQL, Google Maps API, and PHP. For development, we used the IDE Android Studio.

The PHP files are used to interact with a web-hosted database. At the moment, we are using 000webhost.com, which provides 1500Mb of free memory. 000webhost provides access to phpMyAdmin, which is the tool used to construct the database, and all of the PHP files are used to interact with it (the application goes through only 1 though which detects the command the application is attempting to do). The application will get the information from the hosting site by giving a query to the PHP access file, which will output a JSONArray that contains all of the data that we queried the database for in NameValue pairs.

***2) UML Diagram***



***3) Class and Their Methods Overviews***

* ApiConnector is a public class used to interact with the web-hosted database. It has the methods:
* ApiConnector has one staic variable inside of it. String url = "http://tutorapp.net76.net/application\_api.php". This url is the location of the api for the website, where the application will make requests for information from the database.
* public JSONArray GetTutors() throws IOException

GetTutors is a public method that: has no parameters ; returns a JSONArray ; throws IOException. GetTutors will query the hosted database for a complete list of all freelance tutors in the database.

* public JSONArray GetNarrowedTutors(String collegeName, int subjectID, int courseID) throws IOException

GetNarrowedTutors is a public method that: has parameters of String collegeName, int subjectID, and int courseID ; returns a JSONArray ; throws IOException. GetNarrowedTutors will query the hosted database for a list of college tutors based on the chosen college, subject, and course. Subject and course do not need to be chosen for it to work, they will be passed in as -1 if their value was not specified. The returned list of tutors will be sorted by their schedules.

* public JSONArray GetNarrowedFreelanceTutors(int subjectID) throws IOException

GetNarrowedFreelanceTutors is a public method that: has the parameter of int subjectID ; returns JSONArray ; throws IOException. This method works similar to the GetNarrowedTutors method, in that it queries the database for a list of freelance tutors that fall into the catagories provided through the parameters. There are plans to eventually add a second parameter, which would be distance, but it has not been implemented yet.

* public Boolean AddTutor(String name, String email, String address) throws IOException

AddTutor is a public method that: has parameters of String name, String email, and String address ; returns a Boolean ; throws IOException. AddTutor will be used to register a freelance tutor into the webhosted database through the application. It uses the Google Maps geocoding to get the longitude and latitude of the address given, so it won’t store the actual address into the database, just the coordinates. The Boolean returned will say whether or not the attempt succeeded or failed.

* public JSONArray GetAllColleges() throws IOException

GetAllColleges is a public method that: has no parameters ; returns a JSONArray ; throws IOException. GetAllColleges will be used to return a list of every college within the database. It returns a list of names of every college in the database.

* public JSONArray GetSubjects() throws IOException

GetSubjects is a public method that: has no parameters ; returns a JSONArray ; throws IOException. GetSubjects will be used to return a list of all subjects of a particular college. The college name will be obtained from a global variable “selectedCollegeName”.

* public JSONArray GetFreelanceSubjects() throws IOException

GetFreelanceSubjects is a public method that: has no parameters ; returns a JSONArray ; throws IOException. GetFreelanceSubjects will be used to return a list of all subjects that freelancers in the database offer.

* public JSONArray GetReviews() throws IOException

GetReviews is a public method that: has no parameters ; returns a JSONArray ; throws IOException. GetReviews will be used to return a list of all the reviews for a particular tutor (whose ID is obtained from a global array “tutorList”.

* public JSONArray GetCourses(int subject\_id) throws IOException

GetCourses is a public method that: has the parameter of int subject\_id ; returns a JSONArray ; throws IOException. GetCourses will be used to return all of the courses associated with a particular subject at a college.

* public Boolean AddReview(int tutorID, int freelanceID, String reviewText, float stars) throws MalformedURLException, IOException

AddReview is a public method that: has the parameters of int tutorID, int freelanceID, String reviewText, float stars ; returns a Boolean ; throws a MalformedURLException, IOException. Add Review will be used to allow the user to add a review for either the tutor selected or the freelancer selected through their IDs.

* public JSONArray getTutorCourses(int tutorID) throws IOException

getTutorCourses is a public method that: has the parameter of tutorID ; returns a JSONArray ; throws IOException. getTutorCourses has the simple functionality of returning all of the courses that a particular tutor has listed within the database.

* public Drawable getTutorPhoto(String photoURL)

getTutorPhoto is a public method that: has the parameter of String photoURL ; returns a Drawable object. getTutorPhoto will return a Drawable object, which is obtained by creating a HTTP connection to the photoURL provided as a parameter. The photoURL is obtained from tutors in the database.

* private void writeStream(OutputStream out, List<NameValuePair> l) throws UnsupportedEncodingException, IOException

writeStream is a private method that: has the parameters of OutputStream out, List<NameValuePair> l ; it is a void function ; throws UnsupportedEncodingException, IOException. writeStream is used to write a request URL to the server connection's output stream.

* private String cleanString(String s)

cleanString is a private method that: has the parameter String s ; returns a String. cleanString is used to clean a JSON response from the server so it can be used as a JSONArray.

* private String parseAddressURL(String address)

parseAddressURL is a private method that has the parameter String address ; returns a String. parseAddressURL is used to turns an address provided into a URL request for Google's geocoding API.

* CollegeSearchNarrower is a public class that extends Activity and implements AdapterView.OnItemSelectedListener. It is used to help the user narrow their search for college-based tutors by subject, course, and times. The variables and methods are:
* CollegeSearchNarrower has the variables private boolean gettingTutors = false (it sees if a task is currently running to fetch tutors). int currentCourseID = -1 and private int currentSubjectID = -1 (these are used to keep track of currently selected information). private int[] subjectIDs and private String[] subjectNames (these are the lists of subject names/ids returned from the database). private int[] courseIDs and private String[] courseNames (these are the lists of course names/ids returned from the database). All of these variables are accessed class-wide, as they are all private.
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. This function is only meant to set the activity to its “default” setting.

* public void onResume()

public void onPause()

onResume and onPause both are public methods that: have no parameters ; are void functions. These two are basically two parts of the same thing. When onPause() is called, it will call super.onPause(), and when onResume() is called, it will call super.onResume().

* public void onSaveInstanceState(Bundle outState)

onSaveInstanceState is a public method that: has the parameter of Bundle outstate ; is a void function. It will I dunno. **EDIT THIS ONE LATER.**

* public void onItemSelected(AdapterView<?> parent, View view, int pos, long id)

onItemSelected is a public method that: has the parameters of AdapterView<?> parent, View view, int pos, long id ; is a void function. onItemSelected is called when an item in one of the spinners is selected.

* public void onNothingSelected(AdapterView<?> parent)

onNothingSelected is a public method that: has the parameter of AdapterView<?> parent ; is a void function. It does nothing!

* public void onSearchButtonClicked(View view)

onSearchButtonClicked is a public method that: has the parameter of View view ; is a void function. It determines what happens when the search button is clicked.

* public void setSubjectSpinnerAdapter(JSONArray subjectsArray)

setSubjectSpinnerAdapter is a public method that: has the parameter of JSONArray subjectsArray ; it is a void function. It builds and attaches an adapter to the subject spinner.

* public void setCourseSpinnerAdapter(JSONArray subjectsArray)

setCourseSpinnerAdapter is a public method that: has the parameter of JSONArray subjectsArray ; it is a void function. It builds and attaches an adapter to the course spinner.

* public void startSearch()

startSearch is a public method that: has no parameters ; is a void function. startSearch will start the activity for the tutorSearchIntent if the global array tutorList has stuff to sort through.

* void showErrorDialog(int messageId)

showErrorDialog is a void method that: has the parameter int messageId. It will show a fragment that has the error message it needs to show.

* public String findSubjectName(int id)

findSubjectName is a public method that: has the parameter int id ; returns a String. The purpose is to get the name of a subject, based on it’s id.

* There is also 3 internal private classes: GetSubjectsTask, GetNarrowedTutorsTask, and GetCoursesTask. They all extend AsyncTask<ApiConnector,Long,JSONArray>. All of these tasks have the same two methods within (except for GetCoursesTask, it has a third function).
* protected JSONArray doInBackground(ApiConnector... params).

This method, in all 3 private classes, will return a JSONArray from the appropriate ApiConnector method (i.e. GetSubjectsTask will call ApiConnector’s GetSubjects method.)

* protected void onPostExecute(JSONArray jsonArray)

This method will be executed after the doInBackground is finished, and take doInBackground’s returned variable as a parameter.

* GetCoursesTask: public GetCoursesTask(int id)

This sets the subjectID being used for the task.

* CollegeTutor is a public class that extends Tutor. This class is used to contain all of the information on a specific college-based tutor and their drop-in schedules it has the following variable/methods:
* CollegeTutor has 5 public variables (for easy access when manipulating them) and 7 private variables, which are used to maintain schedules. The 5 public variables are: public String college, public String building, public String room, public String tutorCourses, public String tutorSchedule; all of these variables are attributes of the tutor. The 5 private variables that are basically static numbers are: private int MONDAY = 10000, private int TUESDAY = 20000, private int WEDNESDAY = 30000, private int THURSDAY = 40000, private int FRIDAY = 50000; these are numbers used to perform the algorithm on finding a schedule time. There are also 2 more private variables, which are arrays: private int[] startTimes, private int[] endTimes; these arrays are used to store the start and end times of the tutor’s schedule.
* public static CollegeTutor loadFromJsonObject(JSONObject jsonObj)

loadFromJsonObject is a public method that: has the parameter of JSONObject jsonObj ; returns a CollegeTutor. This method will take the JSONObject that was obtained from the ApiConnector class and then return the CollegeTutor object that now has all of the data parsed out.

* public boolean isTimeInSchedule(int time)

isTimeInschedule is a public method that: has the parameter int time ; returns a Boolean. The purpose of this method is to simply see if the time passed in is between any start and end times for the tutor’s schedule.

* private static String makeSchedule(CollegeTutor tutor)

makeSchedule is a private static method that: has the parameter CollegeTutor tutor ; returns a String. This method will build the schedule of the tutor into a single string, and then return it.

* private static int[] sortSchedule(int[] scheduleTimes)

sortSchedule is a private static method that: has the parameter int[] scheduleTimes ; returns a int[]. The purpose of this method is to sort the times in the integer array passed in.

* private static CollegeTutor copyTutor(Tutor tutor)

copyTutor is a private static method that: has the parameter Tutor tutor ; returns a CollegeTutor. This method will copy all of the information from the passed in tutor into a CollegeTutor and then return the CollegeTutor.

* CollegeTutorSearch is a public class that extends Activity. This class is used to set up the collegeTutorSearch activity page. It has a single method:
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The onCreate method will setup the CollegeTutorSearch activity on its creation.

* FreelanceContainerFragment is a public class that extends Fragment. This class is used to help display Freelancer information on the FreelanceSearchNarrower activity.
* There are 2 class variables in the class. private int tutorIndex, which contains the index of the tutor that is containes in the tutorList global array. There is also ImageView iv, which is used to hold the display of the image (the tutor’s profile picture).
* public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

onCreateView is a public method that has: the parameters LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState ; returns a View. This method is meant to construct the fragment on its creation.

* There is also one private class within FreelanceContainerFragment. LoadPictureTask is a private class that extends AsyncTask<ApiConnector, Void, Drawable>. It is meant to load pictures from URL to Bitmap for Freelancers. It has the methods:
* protected Drawable doInBackground(ApiConnector... params)

doInBackground is a protected method that: has the parameter of ApiConnector… params ; returns a Drawable. This method allows the task, when called, to run the GetTutorPhoto method from the ApiConnector class.

* protected void onPostExecute(Drawable drawable)

onPostExecute is a protected method that: has the parameter of Drawable drawable ; is a void function. This method, when called, will add the picture from the database to the correct tutor in the global tutorList array and set the ImageView of the fragment.

* FreelanceProfile is a public class that extends Activity. This class is used to set up the FreelaceProfile activity page. It has 2 class-wide variables and some methods:
* FreelanceProfile has 2 variables: private FreelanceTutor tutorDisplayed, and private int tutorIndex.
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The purpose of this method is to set up the activity FreelanceProfile on creation.

* private void displayTutorInfo()

displayTutorInfo is a private method that: has no parameters ; is a void function. displayTutorInfo will populate the fields on the screen with the correct information that is contained in the tutorDisplayed object.

* public void onReviewButtonClicked(View view)

onReviewButtonClicked is a public method that: has the parameter of View view ; is a void function. onReviewButtonClicked will, when the button on the screen is clicked, start the activity for the ReviewList class.

* public void onMapButtonClicked()

**NOT YET IMPLEMENTED EDIT LATER!**

* FreelanceSearchNarrower is a public class that extends Activity and implements AdapterView.OnItemSelectedListener. This class is used to set up and maintain the FreelaceSearchNarrower activity page. FreelanceSearchNarrower is meant to search for freelancers by their subjects. There are plans in the future to implement searching by distance away from user, but as of right now that is not implemented.
* FreelanceSearchNarrower has the variables private boolean gettingTutors = false (it sees if a task is currently running to fetch tutors). private int currentSubjectID = -1 (this is used to keep track of currently selected information). private int[] subjectIDs and private String[] subjectNames (these are the lists of subject names/ids returned from the database).
* protected void onCreate(Bundle savedInstanceState)
* onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The purpose of this method is to set up the activity FreelanceSearchNarrower on creation. It will run the GetFreelanceSubjectsTask.
* public void onItemSelected(AdapterView<?> parent, View view, int pos, long id)
* onItemSelected is a public method that: has the parameters AdapterView<?> parent, View view, int pos, long id ; is a void function. This method will see if the item the user selected in the spinners were valid.
* public void onNothingSelected(AdapterView<?> parent)

**NOTHING EDIT LATER!**

* public void onSearchButtonClicked(View view)

onSearchButtonClicked is a public method that: has the parameter View view ; is a void function. This method will run the GetNarrowedFreelanceTutorsTask when the user clicks the search button.

* public void setSubjectSpinnerAdapter(JSONArray subjectsArray)

setSubjectSpinnerAdapter is a public method that: has the parameter of JSONArray subjectsArray ; it is a void function. It builds and attaches an adapter to the subject spinner.

* public String findSubjectName(int id)

findSubjectName is a public method that: has the parameter int id ; returns a String. This method is meant to find the match from subjectIDs that goes with the id passed in. It will return the subject name if it finds a match, and return an empty string if not.

* public void startSearch()

startSearch is a public method that: has no parameters ; is a void function. startSearch will start the activity for the tutorSearchIntent if the global array freelanceList has stuff to sort through.

* void showErrorDialog(int messageId)

showErrorDialog is a void method that: has the parameter int messageId. It will show a fragment that has the error message it needs to show.

* There is a private class GetFreelanceSubjectsTask that extends AsyncTask<ApiConnector,Long,JSONArray>. This class is meant to get subjects from the database.
* protected JSONArray doInBackground(ApiConnector... params).

doInBackground is a protected method that: has the parameter of ApiConnector... params ; returns a JSONArray. This method will call ApiConnector’s GetFreelanceSubjects method and return it’s result if it can.

* protected void onPostExecute(JSONArray jsonArray)

This method will be executed after the doInBackground is finished, and take doInBackground’s returned variable as a parameter. It will setup the subject spinner’s adapter with the contents of the JSONArray.

* There is a private class GetNarrowedFreelanceTutorsTask that extends AsyncTask<ApiConnector,Long,JSONArray>. This class is meant to get subjects from the database.
* protected JSONArray doInBackground(ApiConnector... params).

doInBackground is a protected method that: has the parameter of ApiConnector... params ; returns a JSONArray. This method will call ApiConnector’s GetNarrowedFreelanceTutors method and return it’s result if it can.

* protected void onPostExecute(JSONArray jsonArray)

This method will be executed after the doInBackground is finished, and take doInBackground’s returned variable as a parameter. It will set the gettingTutors flag to false (since it is done) and then go about inserting them into the freelanceList global variable.

* FreelanceTutor is a public class that extends Tutor. This class is used like CollegeTutor, as it is a child class of Tutor and is meant to store the contents of a Freelance Tutor.
* There are 2 public variables in this class: public double latitude, and public double longitude. These variable are used to help the user find freelance tutors in their area.
* public static FreelanceTutor loadFromJsonObject(JSONObject jsonObj)

loadFromJsonObject is a public method that: has the parameter of JSONObject jsonObj ; returns a FreelanceTutor. This method will take the JSONObject that was obtained from the ApiConnector class and then return the FreelanceTutor object that now has all of the data parsed out.

* private static FreelanceTutor copyTutor(Tutor tutor)

copyTutor is a private static method that: has the parameter Tutor tutor ; returns a FreelanceTutor. This method will copy all of the information from the passed in tutor into a FreelanceTutor object and then return the FreelanceTutor.

* Globals is a public class. This class is used as a container for all global variables within the program. It has many variables and a few classes as well that are meant to be accessed anywhere in the program when needed.
* In Globals, there are currently 5 public static variabls. These are: public static final String API\_KEY = “stuff” **EXPLAIN LATER!** , public static Drawable noPhoto = null, public static String selectedCollegeName = “”, public static CollegeTutor[] tutorList = null, public static FreelanceTutor[] freelanceList = null.
* There are also a few public static classes within the Globals class.
* ErrorDialogFragment is a public static class that extends DialogFragment.
* public static ErrorDialogFragment newInstance(int message)

newInstance is a public static method that: takes a parameter of int message ; returns an ErrorDialogFragment. This method will take a message into it and then return the ErrorDialogFragment with the message inside of it.

* public Dialog onCreateDialog(Bundle savedInstanceState)

onCreateDiaglog is a public method that: has the parameter Bundle savedInstanceState ; returns a Dialog. This method will set up the layout for the ErrorDialogFragment.

* InvalidTimeException is a public static class that extends Exception. It will be thrown whenever a time is invalid.
* Utils is a public static class. It is used to provide some functions/methods all across the program.
* public static int timeToInt(String day, String time) throws InvalidTimeException

timeToInt is a public static method that: has the parameters String day, String time ; returns an int ; throws InvalidTimeException. This method is used to construct the number that is going to be used to do time manipulation.

* MainMenu is a public class that extends Activity. This class is used as the initial hub. From here the user can do things such as: search for the college of their choosing, search for freelancers in their area, signup as a freelance tutor, or go to the online resources page.
* There is one private variable within this class: private String[] validColleges = null. This array will hold the list of all valid colleges returned from the database.
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter of Bundle savedInstanceState ; is a void function. This method will set up the activity for the main menu on the start of the activity.

* public void onlineResourcesButtonClicked(View view)

onlineResourcesButtonClicked is a public method that: has the parameter View view ; is a void function. This method states what happens when the user clicks the Online Resources button (spoiler, it starts the activity for online resources page).

* public void freelanceSearchButtonClicked(View view)

freelanceSearchButtonClicked is a public method that: has the parameter View view ; is a void function. This method states what happens when the user clicks the Freelance Search button.

* public void freelanceSignupButtonClicked(View view)

freelanceSignupButtonClicked is a public method that: has the parameter View view ; is a void function. This method states what happens when the user clicks the Freelance Signup button.

* public void setAutoCompleteList(JSONArray listArray)

setAutoCompleteList is a public method that: has the parameter JSONArray listArray ; is a void function. The purpose of this method is to allow the search bar on the MainMenu activity to have autocomplete that is provided from the validColleges array.

* public void checkQuery(AutoCompleteTextView v)

checkQuery is a public method that: has the parameter of AutoCompleteTextView v ; is a void function. This method makes sure that the college entered into the search field is valid from the validColleges array. If it is it will start the activity for a CollegeSearchNarrower.

* void showErrorDialog(int messageId)

showErrorDialog is a constructor that: has the parameter of int messageId. **EDIT LATER DUN KNOW WTF IS GOING ON!**

* There is also a private class inside of the MainMenu class. This class is: private class GetCollegesTask that extends AsyncTask<ApiConnector,Long,JSONArray>.
* protected JSONArray doInBackground(ApiConnector... params)

doInBackground is a protected method that: has the parameter of ApiConnector… params ; returns a JSONArray. This method allows the task, when called, to run the GetAllColleges method from the ApiConnector class.

* protected void onPostExecute(JSONArray jsonArray)

onPostExecute is a protected method that: has the parameter of JSONArray jsonArray ; is a void function. This method, when called, will set up the autocomplete list for the activity.

* OnlineResources is a public class that extends Activity. This Activity is used to act as the Online Resources page that is to be used by users. This Activity has two methods:
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The purpose of this method is to set up the activity for the Online Resources.

* private void enableLinks()

enableLinks is a private method that: has no parameters ; is a void function. This method is meant to activate the links that are on the page for the Online Resources page.

* ReviewList is a public class that extends Activity. This activity is used to show the user a list of all reviews on a tutor when they choose to view them.
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The purpose of this method is to set up the activity for the Review List, which runs the GetReviewsTask to do so.

* private void fillReviewList(JSONArray reviews) throws JSONException

fillReviewList is a private method that: has the parameter of JSONArray reviews ; is a void function ; throws JSONException. The purpose of this method is to fill the list view on the page with reviews from the JSONArray.

* There is also one private class within ReviewList. GetReviewsTask is a private class that extends AsyncTask<ApiConnector,Long,JSONArray> It has the methods:
* protected JSONArray doInBackground(ApiConnector... params)

doInBackground is a protected method that: has the parameter of ApiConnector… params ; returns a JSONArray. This method allows the task, when called, to run the GetReviews method from the ApiConnector class.

* protected void onPostExecute(JSONArray jsonArray)

onPostExecute is a protected method that: has the parameter of JSONArray jsonArray ; is a void function. This method, when called, will set up the review list for the activity using the results obtained from the database.

* CollegeTutor is a public class that extends Tutor. This class is used to contain all of the information on a specific college-based tutor and their drop-in schedules it has the following variable/methods:
* Tutor has 8 public class-wide variables. All of these variables are from the database. First there is the public int tutorID, which is stored for fast look-up of other tutor relevant data. Then there is public String firstName, publicString lastName, public String emailAddress, public String subject, public String bio, and public String pictureURL; all of these variables are string data of the tutor obtained from the database. The final variable is public Drawable picture, which contains the drawable object for the tutor (their profile picture). All other variables seem pretty self-explanatory (i.e. firstName = the tutor’s first name, subject = the subject they tutor, bio = the tutor’s 255 character description of themselves).
* public static Tutor loadFromJsonObject(JSONObject jsonObj)

loadFromJsonObject is a public method that: has the parameter of JSONObject jsonObj ; returns a Tutor. This method will take the JSONObject that was obtained from the ApiConnector class and then return the CollegeTutor object that now has all of the data parsed out.

* TutorContainerFragment is a public class that extends Fragment. This class is used to help display tutor information on the TutorSearchNarrower page. **I THINK! ASK LATER EDIT LATER!**
* There are 2 class variables in the class. private int tutorIndex, which contains the index of the tutor that is containes in the tutorList global array. There is also ImageView iv, which is used to hold the display of the image (the tutor’s profile picture).
* public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

onCreateView is a public method that has: the parameters LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState ; returns a View. This method is meant to construct the fragment on its creation.

* There is also one private class within TutorContainerFragment. LoadPictureTask is a private class that extends AsyncTask<ApiConnector, Void, Drawable>. It is meant to load pictures from URL to Bitmap for Tutors. It has the methods:
* protected Drawable doInBackground(ApiConnector... params)

doInBackground is a protected method that: has the parameter of ApiConnector… params ; returns a Drawable. This method allows the task, when called, to run the GetTutorPhoto method from the ApiConnector class.

* protected void onPostExecute(Drawable drawable)

onPostExecute is a protected method that: has the parameter of Drawable drawable ; is a void function. This method, when called, will add the picture from the database to the correct tutor in the global tutorList array and set the ImageView of the fragment.

* TutorProfile is a public class that extends Activity. This activity is used to display all relevant information on a tutor to a user when they look at a particular tutor’s profile.
* There are two private class-wide variables within TutorProfile. The first is private CollegeTutor tutorDisplayed, which is simply the object for the tutor that the profile is for. The second variable is private int tutorIndex, which is the tutor’s index in the tutorList global array.
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The purpose of this method is to set up the activity for the Tutor Profile, which runs the GetCoursesTask to help do so.

* private void displayTutorInfo()

displayTutorInfo is a private method that: has no parameters ; is a void function. This method will populate all of the fields on screen with the relative information needed.

* public void onReviewButtonClicked(View view)

onReviewButtonClicked is a public method that: has the parameter View view ; is a void function. This method is supposed to start the ReviewList activity.

* public void onMapButtonClicked()

**NOT CURRENTLY IMPLEMENTED!**

* public String getCourses(JSONArray jsonArray) throws JSONException

getCourses is a public method that: has the parameter JSONArray jsonArray ; returns a String ; throws JSONException. This method is meant to obtain the courses from a JSONArray that is passed into the method.

* There is also a private class inside of the TutorProfile class. This class is: private class getCoursesTask that extends AsyncTask<ApiConnector,Long,String>
* Protected String doInBackground(ApiConnector... params)

doInBackground is a protected method that: has the parameter of ApiConnector… params ; returns a String. This method allows the task, when called, to run the getTutorCourses method in ApiConnector, and then pass that result to the getCourses method within TutorProfile. It will return the result of getCourses.

* protected void onPostExecute(String result)

onPostExecute is a protected method that: has the parameter of String result ; is a void function. This method, when called, will add the courses attribute to the tutor that is contained in the tutorDisplayed variable. It will also then set up the courses display for the profile.

* TutorRegistration is a public class that extends Activity. This activity is used to provide freelance tutors registration through the application.
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The purpose of this method is to set up the activity for the registration page.

* public boolean onCreateOptionsMenu(Menu menu)

onCreateOptionsMenu is a public method that: has the parameter Menu menu ; returns a Boolean. The purpose of this method is to inflate the menu passed into it.

* public boolean onOptionsItemSelected(MenuItem item)

onOptionsItemSelected is a public method that: has the parameter MenuItem item ; returns a Boolean. **DON’T UNDERSTAND THIS METHOD! EDIT LATER!**

* public void registerButtonClicked(View view)

registerButtonClicked is a public method that: has the parameter View view ; is a void function. This method dictates what happens when the register button is clicked (mainly error checking before adding the freelancer to the database through the AddTutorTask).

* void showErrorDialog(int messageId)

showErrorDialog is a I dunno **EDIT LATER!**

* There is also a private class inside of the TutorRegistration class. This class is a private class AddTutorTask that extends AsyncTask<ApiConnector, Long, boolean> and it has the two following methods:
* Protected Boolean doInBackground(ApiConnector... params)

doInBackground is a protected method that: has the parameter of ApiConnector… params ; returns a boolean. This method allows the task, when called, to run the AddTutor method in ApiConnector, and then return that result.

* protected void onPostExecute(Boolean success)

onPostExecute is a protected method that: has the parameter of Boolean success; is a void function. This method, when called, will tell the user whether or not they registered successfully (if it got past the initial error checking it will return false if the address was not valid).

* TutorReview is a public class that extends Activity. This activity is used to give the user all of the information to fill out on a review.
* There is a few private variables within TutorReview. private int tutorID, which, if the review is for a college-based tutor, this would be used over freelanceID. private int freelanceID, same as tutorID but opposite. private String reviewText, which holds the user’s written review of the tutor. private float stars, which is the user’s rating of the tutor. private TextView charCounter, which will help let the user know how many characters they have left (max of 255). private EditText reviewTextField is the field that the user enters the review into. private RatingBar ratingStars is the rating bar that the user uses to rate the tutor on a scale of 1-5.
* protected void onCreate(Bundle savedInstanceState)

onCreate is a protected method that: has the parameter Bundle savedInstanceState ; is a void function. The purpose of this method is to set up the activity for the review page.

* public void onSubmitButtonClicked(View view)

onSubmitButtonClicked is a public method that: has the parameter viewView ; is a void function. The purpose of this method is to determine what is done when the submit review button is clicked. It will run the AddTutorReviewTask if there was no errors.

* void showErrorDialog(int messageId)

**I DON’T UNDERSTAND STUFF! EDIT LATER!**

* There is also a private class inside of the TutorReview class. This class is a private class AddTutorReviewTask that extends AsyncTask<ApiConnector, Long, boolean> and it has the two following methods:
* Protected Boolean doInBackground(ApiConnector... params)

doInBackground is a protected method that: has the parameter of ApiConnector… params ; returns a boolean. This method allows the task, when called, to run the AddReview method in ApiConnector, and then return that result.

* protected void onPostExecute(Boolean success)

onPostExecute is a protected method that: has the parameter of Boolean success; is a void function. This method, when called, will tell the user whether or not they submitted the review successfully.