APPENDIX C

TABLE OF CONTENTS

CLIENT INTERACTION (15/07/18)

The transcript below details a meeting I had with my client regarding the XML pages for my application.

Me: Hello Suyash, how have you been?

Client: Hi Dhruv, I have been well. Have you coded the designs for me to see?

Me: Yes I have. Here, you can have a final look and suggest any final changes you want to implement.

Client: Wow! They look excellent. OK, great. You seem to have acted on most of the feedback I gave you on your initial designs from before. However, I have one more request. Would that be ok?

Me: Of course, what would you like me to change?

Client: I hope this does not cause a lot of trouble, but I am not entirely convinced about displaying the user history in a table format.

Me: Oh, I see. How else would you like me to display the user history?

Client: Currently, I think the user would be overwhelmed by seeing so much data at once. Would it be possible for you to display the results of each survey individually as cards on a scrollable list?

Me: That might take some time, but I think it's manageable. You want the user to be able to scroll through their results, right?

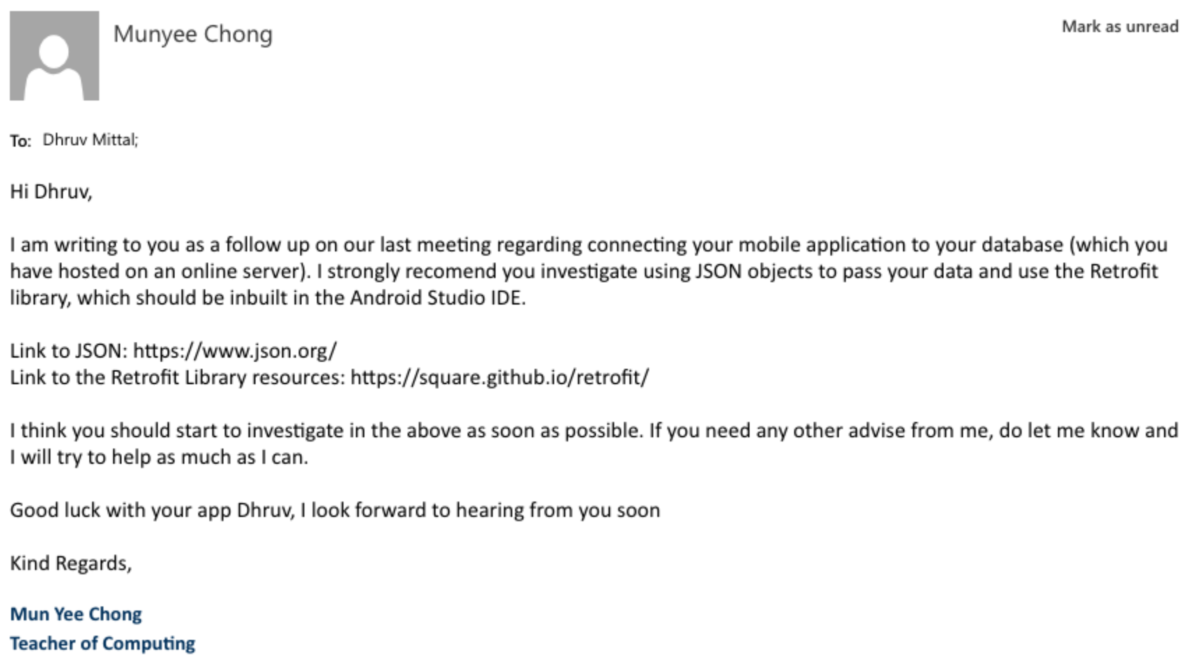
Client: Yes. If they can see their results one at a time in a scrollable view or something along those lines then that would be better. Sorry for any inconvenience caused, I hope it's not too late.

Me: Not at all, no worries. I will make those changes as soon as possilble.

Client: I look forward to seeing the whole application function and come to life.

Me: Thank you. Yes, I should start working on making this app functional now.

ADVISOR INTERACTION (24/07/18)



Munyee Chong

Mark as unread

To: Dhruv Mittal;

Hi Dhruv,

I am writing to you as a follow up on our last meeting regarding connecting your mobile application to your database (which you have hosted on an online server). I strongly recomend you investigate using JSON objects to pass your data and use the Retrofit library, which should be inbuilt in the Android Studio IDE.

Link to JSON: https://www.json.org/
Link to the Retrofit Library resources: https://square.github.io/retrofit/

I think you should start to investigate in the above as soon as possible. If you need any other advise from me, do let me know and I will try to help as much as I can.

Good luck with your app Dhruv, I look forward to hearing from you soon

Kind Regards,

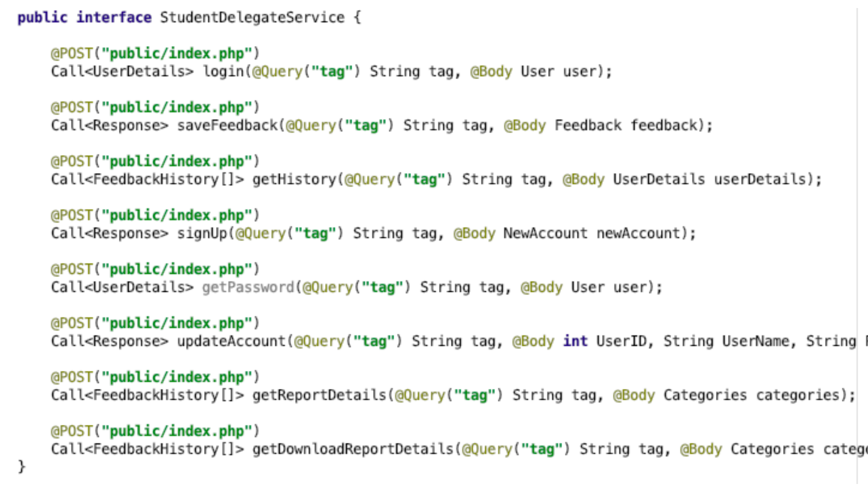
Mun Yee Chong
Teacher of Computing

The email above shows my advisor recommending me to use JSON and the Retrofit libraries.

CODE

HELPER CLASSES AND INTERFACES

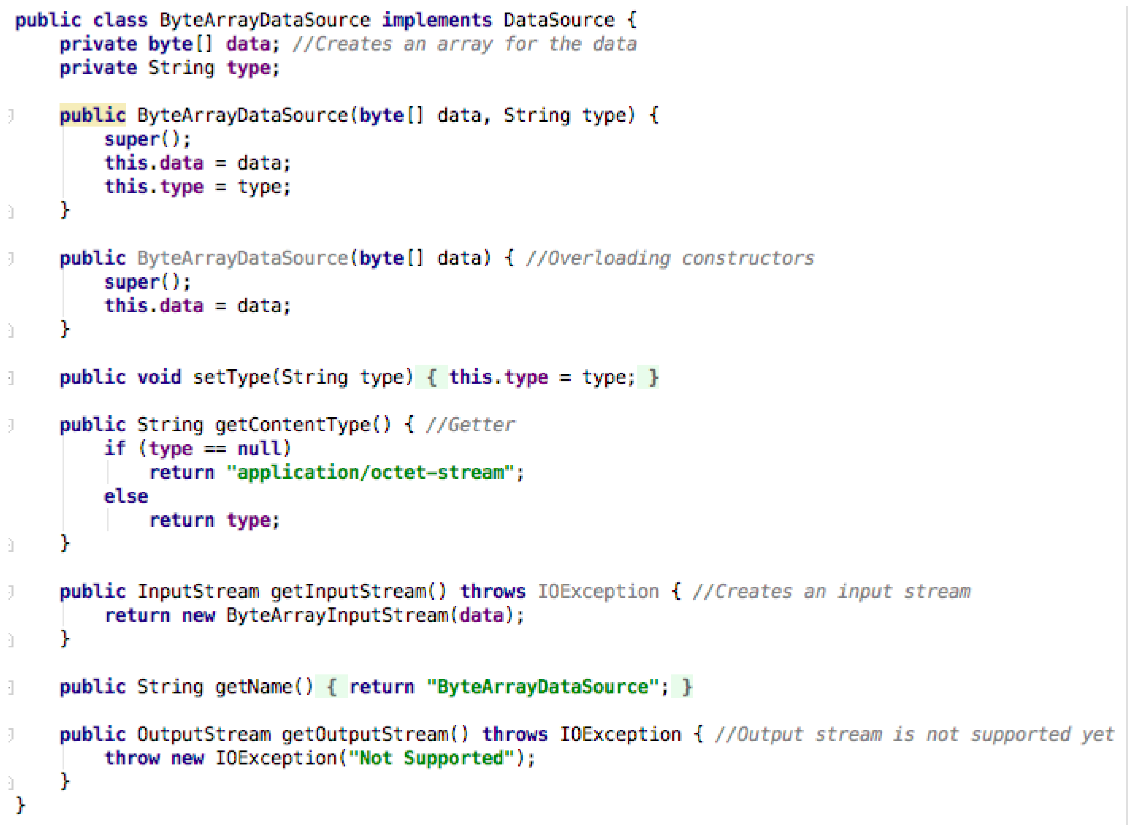
STUDENTDELEGATESERVICE INTERFACE



```java
public interface StudentDelegateService {

    @POST("public/index.php")
    Call<UserDetails> login(@Query("tag") String tag, @Body User user);

    @POST("public/index.php")
    Call<Response> saveFeedback(@Query("tag") String tag, @Body Feedback feedback);

    @POST("public/index.php")
    Call<FeedbackHistory[]> getHistory(@Query("tag") String tag, @Body UserDetails userDetails);

    @POST("public/index.php")
    Call<Response> signUp(@Query("tag") String tag, @Body NewAccount newAccount);

    @POST("public/index.php")
    Call<UserDetails> getPassword(@Query("tag") String tag, @Body User user);

    @POST("public/index.php")
    Call<Response> updateAccount(@Query("tag") String tag, @Body int UserID, String UserName, String Password, String YearGroup);

    @POST("public/index.php")
    Call<FeedbackHistory[]> getReportDetails(@Query("tag") String tag, @Body Categories categories);

    @POST("public/index.php")
    Call<FeedbackHistory[]> getDownloadReportDetails(@Query("tag") String tag, @Body Categories categories);
}
```

The above code shows my interface which consists of HTTP calls which connect to my PHP APIs online

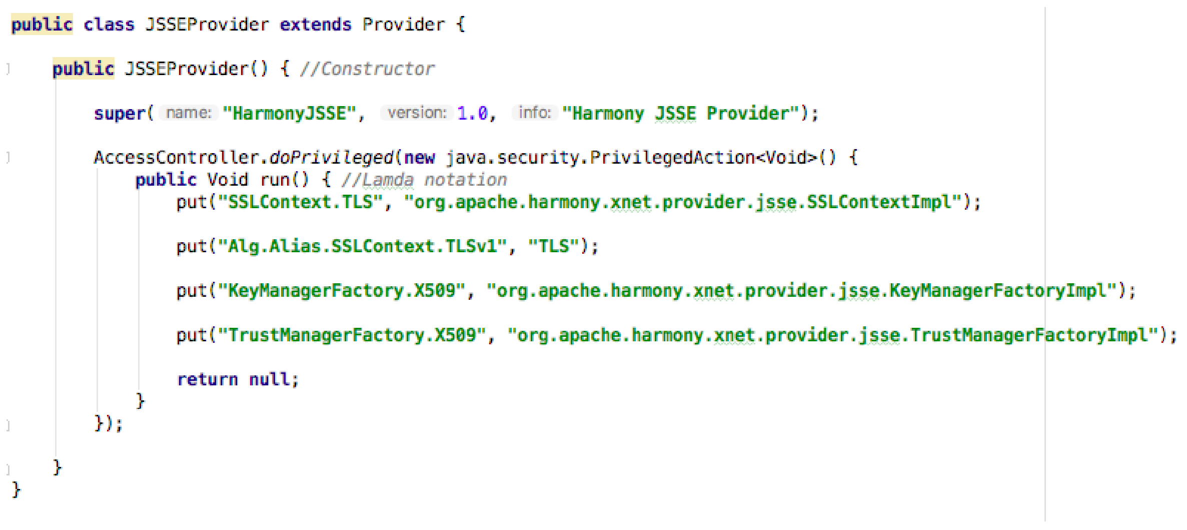
BYTEARRAYDATASOURCE



```java
public class ByteArrayDataSource implements DataSource {
    private byte[] data; //Creates an array for the data
    private String type;

    public ByteArrayDataSource(byte[] data, String type) {
        super();
        this.data = data;
        this.type = type;
    }

    public ByteArrayDataSource(byte[] data) { //Overloading constructors
        super();
        this.data = data;
    }

    public void setType(String type) { this.type = type; }

    public String getContentType() { //Getter
        if (type == null)
            return "application/octet-stream";
        else
            return type;
    }

    public InputStream getInputStream() throws IOException { //Creates an input stream
        return new ByteArrayInputStream(data);
    }

    public String getName() { return "ByteArrayDataSource"; }

    public OutputStream getOutputStream() throws IOException { //Output stream is not supported yet
        throw new IOException("Not Supported");
    }
}
```

The above code shows a helper class for sending emails. The main purpose of this file is to provide a structure to store my data.

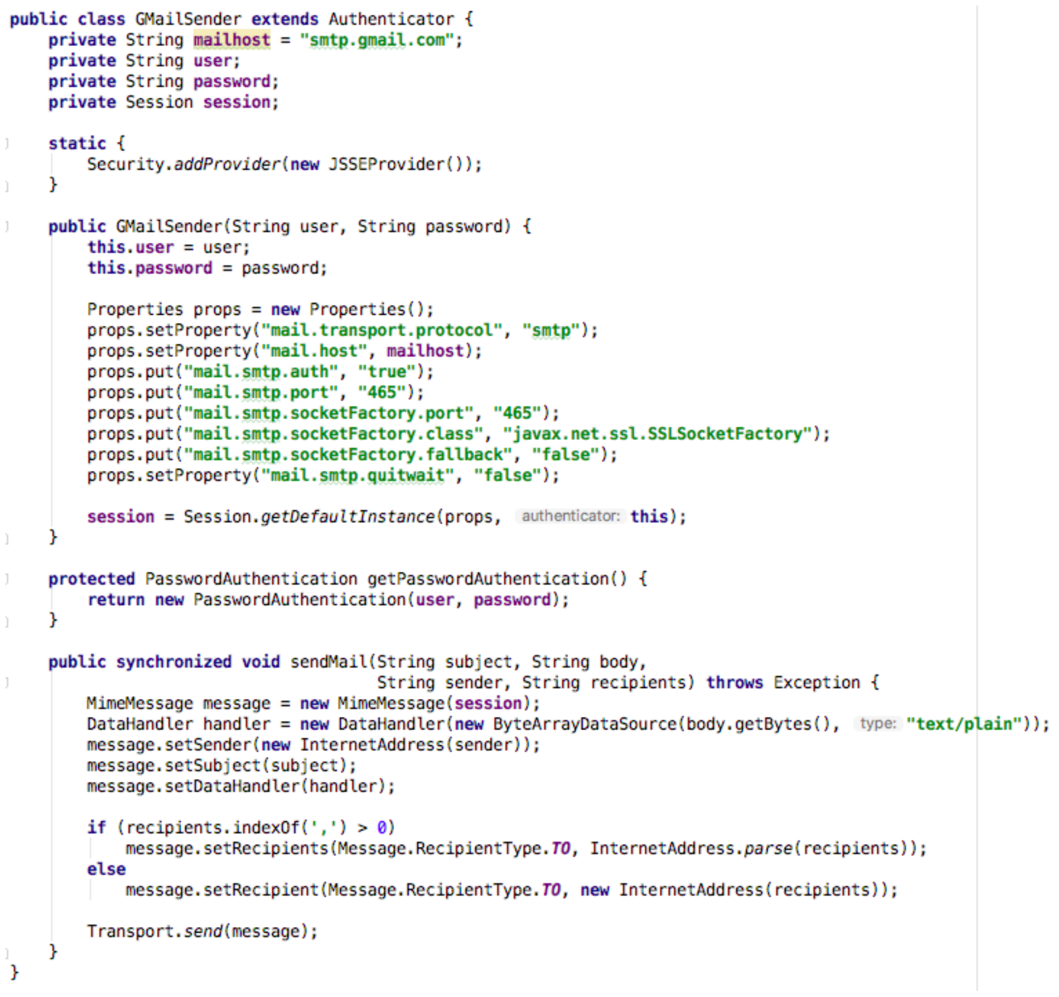
JSSEPROVIDER



```java
public class JSSEProvider extends Provider {

    public JSSEProvider() { //Constructor

        super( name: "HarmonyJSSE",  version: 1.0,  info: "Harmony JSSE Provider");

        AccessController.doPrivileged(new java.security.PrivilegedAction<Void>() {
            public Void run() { //Lamda notation
                put("SSLContext.TLS", "org.apache.harmony.xnet.provider.jsse.SSLContextImpl");

                put("Alg.Alias.SSLContext.TLSv1", "TLS");

                put("KeyManagerFactory.X509", "org.apache.harmony.xnet.provider.jsse.KeyManagerFactoryImpl");

                put("TrustManagerFactory.X509", "org.apache.harmony.xnet.provider.jsse.TrustManagerFactoryImpl");

                return null;
            }
        });

    }
}
```

This provider class is used by the main send email function

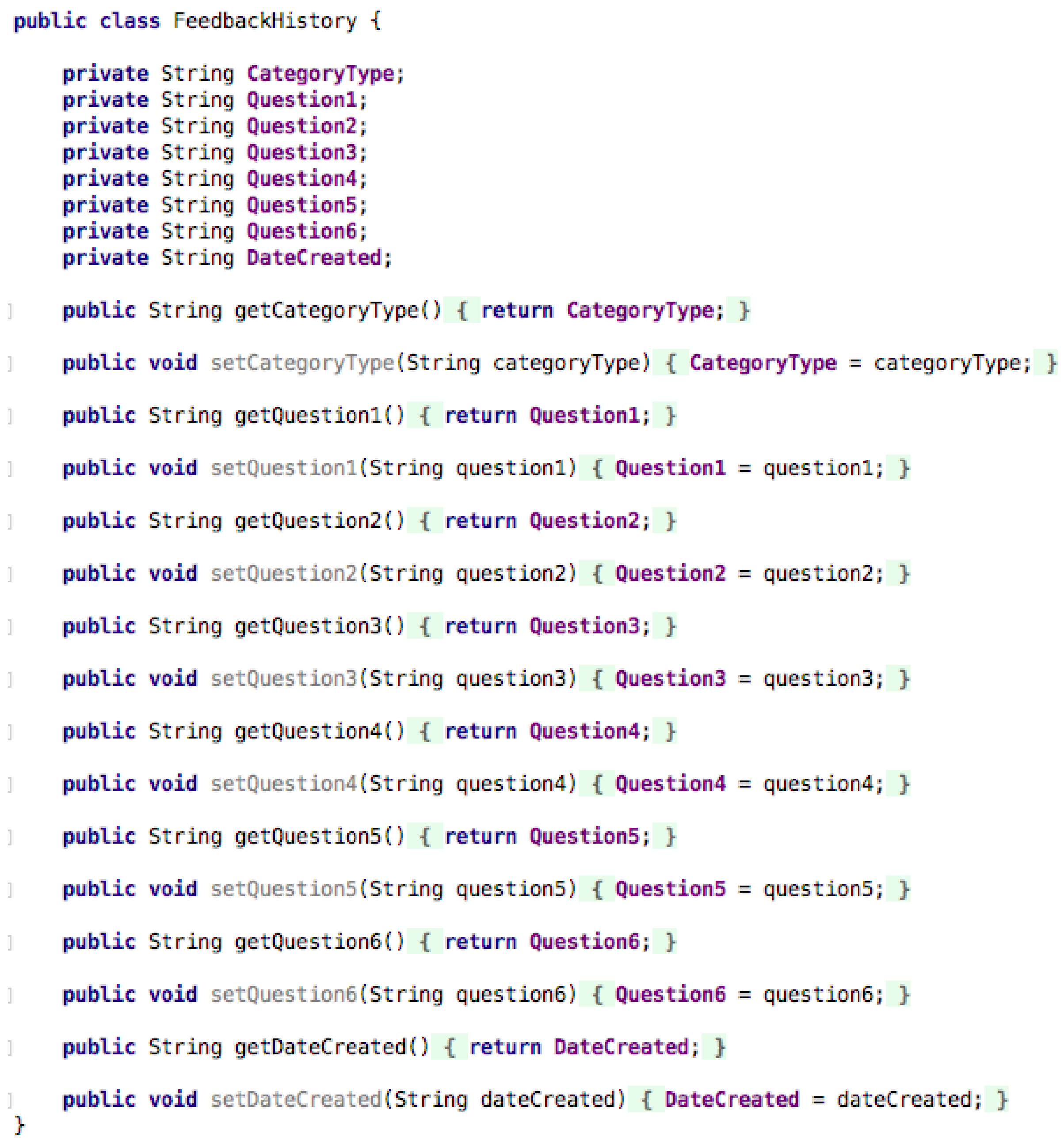
GMAILSENDER



```java
public class GMailSender extends Authenticator {
    private String mailhost = "smtp.gmail.com";
    private String user;
    private String password;
    private Session session;

    static {
        Security.addProvider(new JSSEProvider());
    }

    public GMailSender(String user, String password) {
        this.user = user;
        this.password = password;

        Properties props = new Properties();
        props.setProperty("mail.transport.protocol", "smtp");
        props.setProperty("mail.host", mailhost);
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.port", "465");
        props.put("mail.smtp.socketFactory.port", "465");
        props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        props.put("mail.smtp.socketFactory.fallback", "false");
        props.setProperty("mail.smtp.quitwait", "false");

        session = Session.getDefaultInstance(props,  authenticator: this);
    }

    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(user, password);
    }

    public synchronized void sendMail(String subject, String body,
                                      String sender, String recipients) throws Exception {
        MimeMessage message = new MimeMessage(session);
        DataHandler handler = new DataHandler(new ByteArrayDataSource(body.getBytes(),  type: "text/plain"));
        message.setSender(new InternetAddress(sender));
        message.setSubject(subject);
        message.setDataHandler(handler);

        if (recipients.indexOf(',') > 0)
            message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(recipients));
        else
            message.setRecipient(Message.RecipientType.TO, new InternetAddress(recipients));

        Transport.send(message);
    }
}
```

The GMailSender class is mainly used in the main send email function

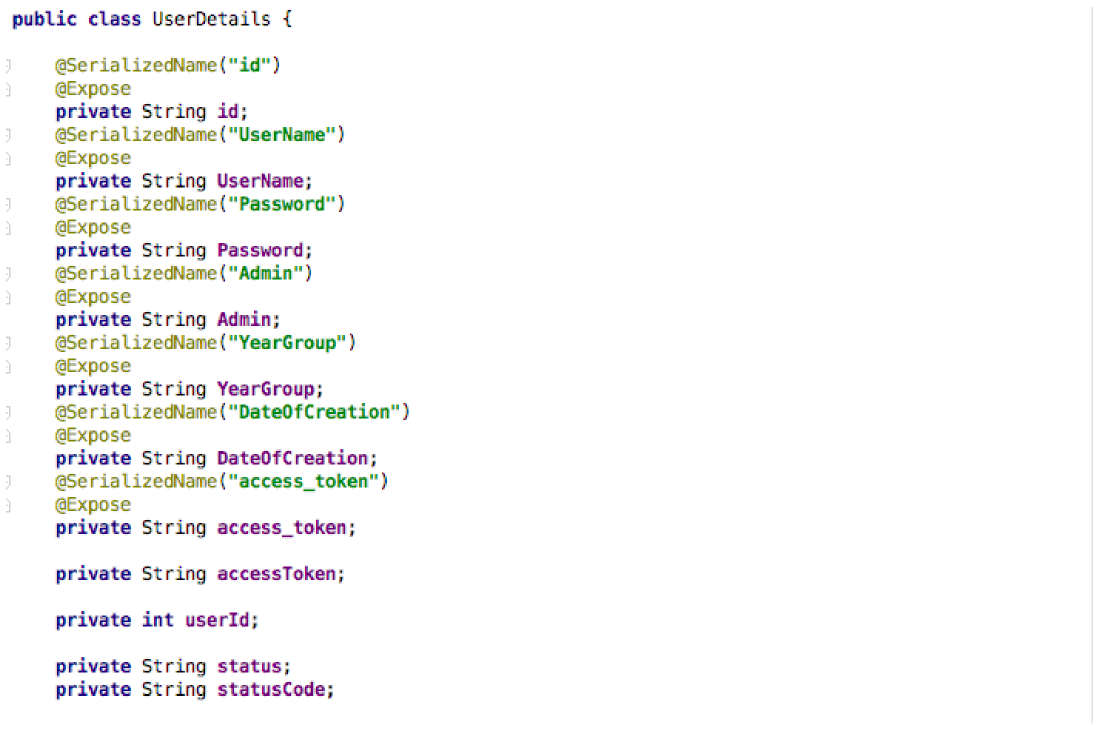
FEEDBACKHISTORY MODEL CLASS



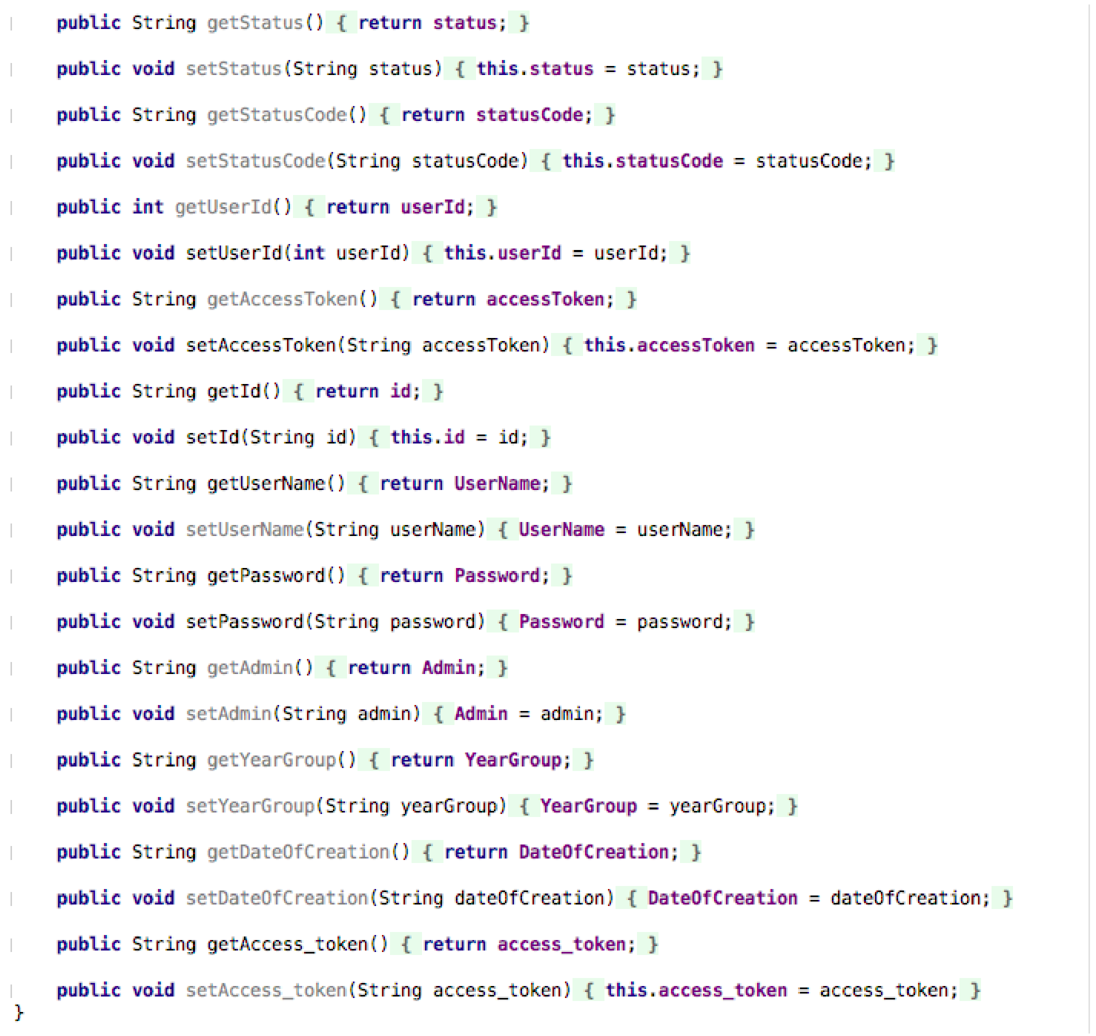
```
public class FeedbackHistory {

    private String CategoryType;
    private String Question1;
    private String Question2;
    private String Question3;
    private String Question4;
    private String Question5;
    private String Question6;
    private String DateCreated;

    public String getCategoryType() { return CategoryType; }

    public void setCategoryType(String categoryType) { CategoryType = categoryType; }

    public String getQuestion1() { return Question1; }

    public void setQuestion1(String question1) { Question1 = question1; }

    public String getQuestion2() { return Question2; }

    public void setQuestion2(String question2) { Question2 = question2; }

    public String getQuestion3() { return Question3; }

    public void setQuestion3(String question3) { Question3 = question3; }

    public String getQuestion4() { return Question4; }

    public void setQuestion4(String question4) { Question4 = question4; }

    public String getQuestion5() { return Question5; }

    public void setQuestion5(String question5) { Question5 = question5; }

    public String getQuestion6() { return Question6; }

    public void setQuestion6(String question6) { Question6 = question6; }

    public String getDateCreated() { return DateCreated; }

    public void setDateCreated(String dateCreated) { DateCreated = dateCreated; }
}
```

The above model class was used as a template for storing and sending the survey data

USERDETAILS MODEL CLASS

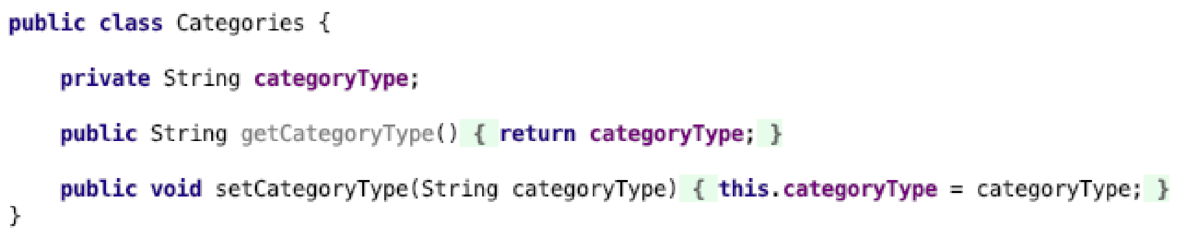


```java
public class UserDetails {

    @SerializedName("id")
    @Expose
    private String id;
    @SerializedName("UserName")
    @Expose
    private String UserName;
    @SerializedName("Password")
    @Expose
    private String Password;
    @SerializedName("Admin")
    @Expose
    private String Admin;
    @SerializedName("YearGroup")
    @Expose
    private String YearGroup;
    @SerializedName("DateOfCreation")
    @Expose
    private String DateOfCreation;
    @SerializedName("access_token")
    @Expose
    private String access_token;

    private String accessToken;

    private int userId;

    private String status;
    private String statusCode;
```



```java
    public String getStatus() { return status; }

    public void setStatus(String status) { this.status = status; }

    public String getStatusCode() { return statusCode; }

    public void setStatusCode(String statusCode) { this.statusCode = statusCode; }

    public int getUserId() { return userId; }

    public void setUserId(int userId) { this.userId = userId; }

    public String getAccessToken() { return accessToken; }

    public void setAccessToken(String accessToken) { this.accessToken = accessToken; }

    public String getId() { return id; }

    public void setId(String id) { this.id = id; }

    public String getUserName() { return UserName; }

    public void setUserName(String userName) { UserName = userName; }

    public String getPassword() { return Password; }

    public void setPassword(String password) { Password = password; }

    public String getAdmin() { return Admin; }

    public void setAdmin(String admin) { Admin = admin; }

    public String getYearGroup() { return YearGroup; }

    public void setYearGroup(String yearGroup) { YearGroup = yearGroup; }

    public String getDateOfCreation() { return DateOfCreation; }

    public void setDateOfCreation(String dateOfCreation) { DateOfCreation = dateOfCreation; }

    public String getAccess_token() { return access_token; }

    public void setAccess_token(String access_token) { this.access_token = access_token; }
}
```

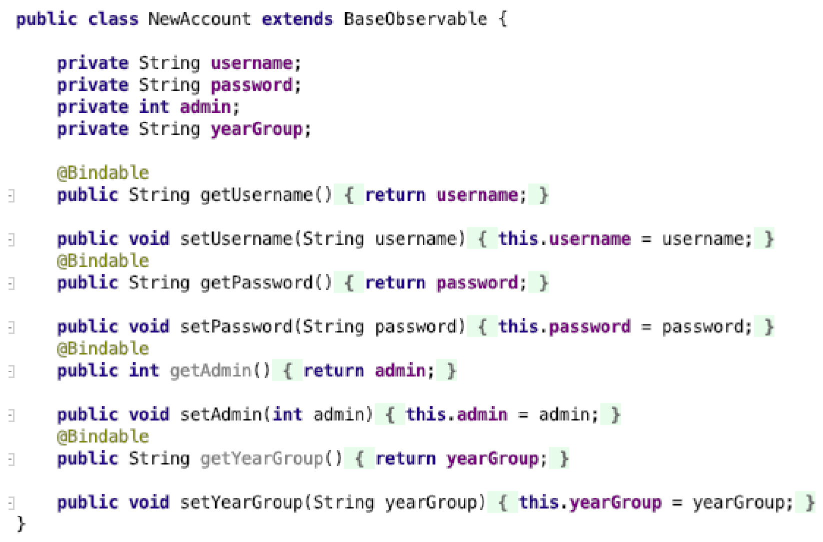
CATEGORIES MODEL CLASS



```
public class Categories {

    private String categoryType;

    public String getCategoryType() { return categoryType; }

    public void setCategoryType(String categoryType) { this.categoryType = categoryType; }
}
```

This class is used the create piecharts and reports

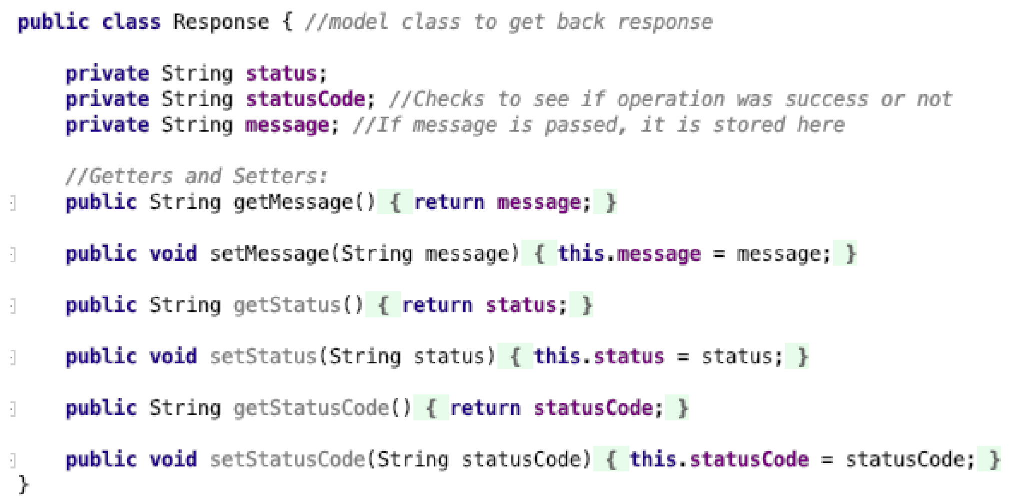
NEW ACCOUNT MODEL CLASS



```
public class NewAccount extends BaseObservable {

    private String username;
    private String password;
    private int admin;
    private String yearGroup;

    @Bindable
    public String getUsername() { return username; }

    public void setUsername(String username) { this.username = username; }
    @Bindable
    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }
    @Bindable
    public int getAdmin() { return admin; }

    public void setAdmin(int admin) { this.admin = admin; }
    @Bindable
    public String getYearGroup() { return yearGroup; }

    public void setYearGroup(String yearGroup) { this.yearGroup = yearGroup; }
}
```

This model class is used to create a new account

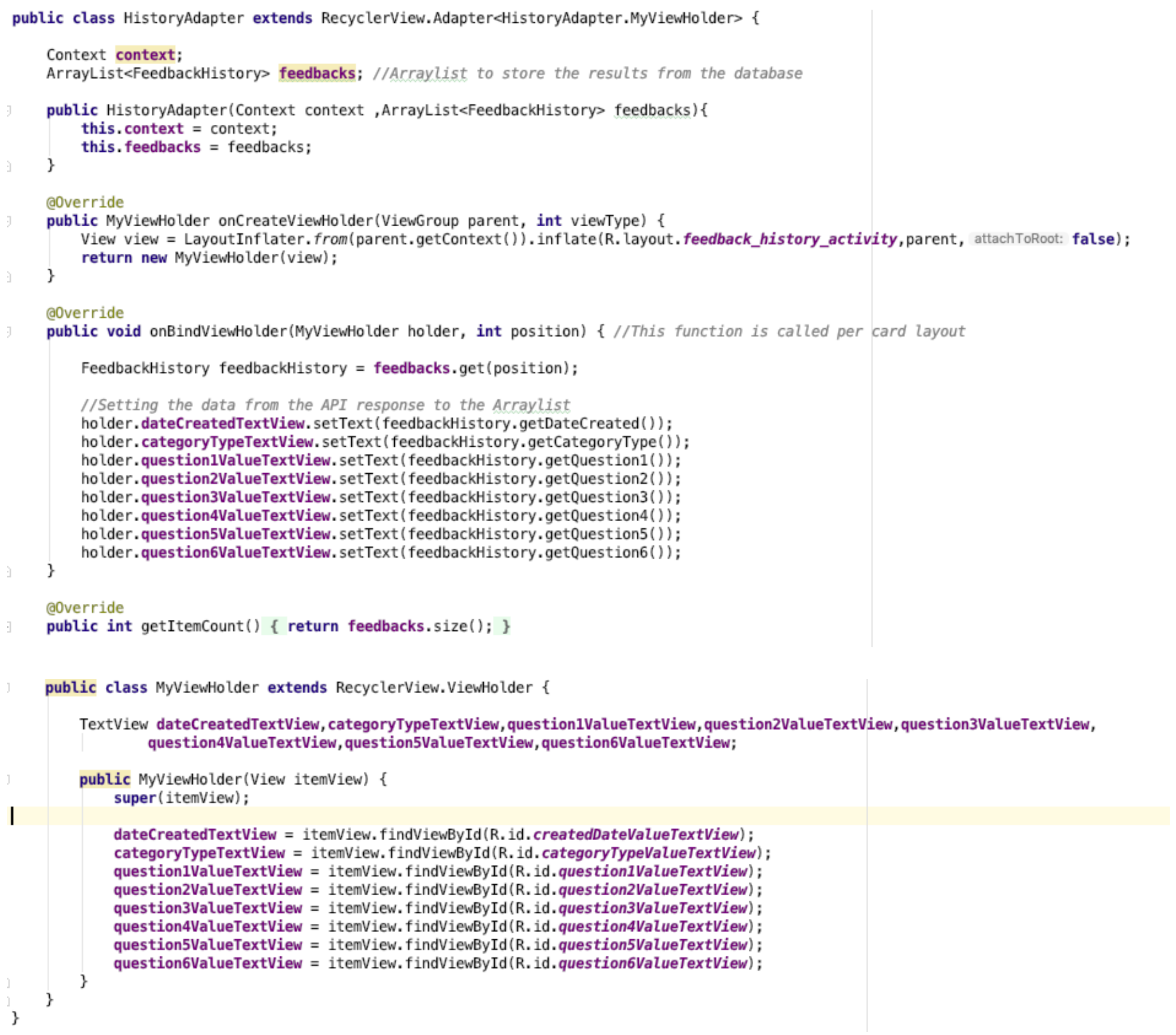
RESPONSE MODEL CLASS



```
public class Response { //model class to get back response

    private String status;
    private String statusCode; //Checks to see if operation was success or not
    private String message; //If message is passed, it is stored here

    //Getters and Setters:
    public String getMessage() { return message; }

    public void setMessage(String message) { this.message = message; }

    public String getStatus() { return status; }

    public void setStatus(String status) { this.status = status; }

    public String getStatusCode() { return statusCode; }

    public void setStatusCode(String statusCode) { this.statusCode = statusCode; }
}
```

Model class which is used to store the reponse from database calls, the statusCode variable is used to check if the operation was valid or not

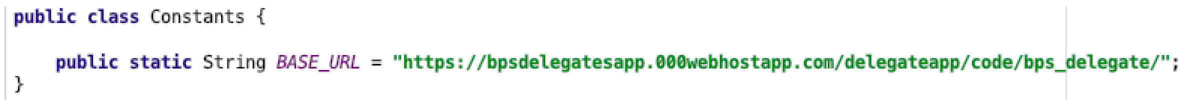
HISTORY ADAPTER CLASS



```java
public class HistoryAdapter extends RecyclerView.Adapter<HistoryAdapter.MyViewHolder> {

    Context context;
    ArrayList<FeedbackHistory> feedbacks; //Arraylist to store the results from the database

    public HistoryAdapter(Context context ,ArrayList<FeedbackHistory> feedbacks){
        this.context = context;
        this.feedbacks = feedbacks;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.feedback_history_activity,parent, attachToRoot: false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(MyViewHolder holder, int position) { //This function is called per card layout

        FeedbackHistory feedbackHistory = feedbacks.get(position);

        //Setting the data from the API response to the Arraylist
        holder.dateCreatedTextView.setText(feedbackHistory.getDateCreated());
        holder.categoryTypeTextView.setText(feedbackHistory.getCategoryType());
        holder.question1ValueTextView.setText(feedbackHistory.getQuestion1());
        holder.question2ValueTextView.setText(feedbackHistory.getQuestion2());
        holder.question3ValueTextView.setText(feedbackHistory.getQuestion3());
        holder.question4ValueTextView.setText(feedbackHistory.getQuestion4());
        holder.question5ValueTextView.setText(feedbackHistory.getQuestion5());
        holder.question6ValueTextView.setText(feedbackHistory.getQuestion6());
    }

    @Override
    public int getItemCount() { return feedbacks.size(); }


    public class MyViewHolder extends RecyclerView.ViewHolder {

        TextView dateCreatedTextView,categoryTypeTextView,question1ValueTextView,question2ValueTextView,question3ValueTextView,
                question4ValueTextView,question5ValueTextView,question6ValueTextView;

        public MyViewHolder(View itemView) {
            super(itemView);

            dateCreatedTextView = itemView.findViewById(R.id.createdDateValueTextView);
            categoryTypeTextView = itemView.findViewById(R.id.categoryTypeValueTextView);
            question1ValueTextView = itemView.findViewById(R.id.question1ValueTextView);
            question2ValueTextView = itemView.findViewById(R.id.question2ValueTextView);
            question3ValueTextView = itemView.findViewById(R.id.question3ValueTextView);
            question4ValueTextView = itemView.findViewById(R.id.question4ValueTextView);
            question5ValueTextView = itemView.findViewById(R.id.question5ValueTextView);
            question6ValueTextView = itemView.findViewById(R.id.question6ValueTextView);
        }
    }
}
```

This history adapter class is called to create to populate each history card with the relevant data

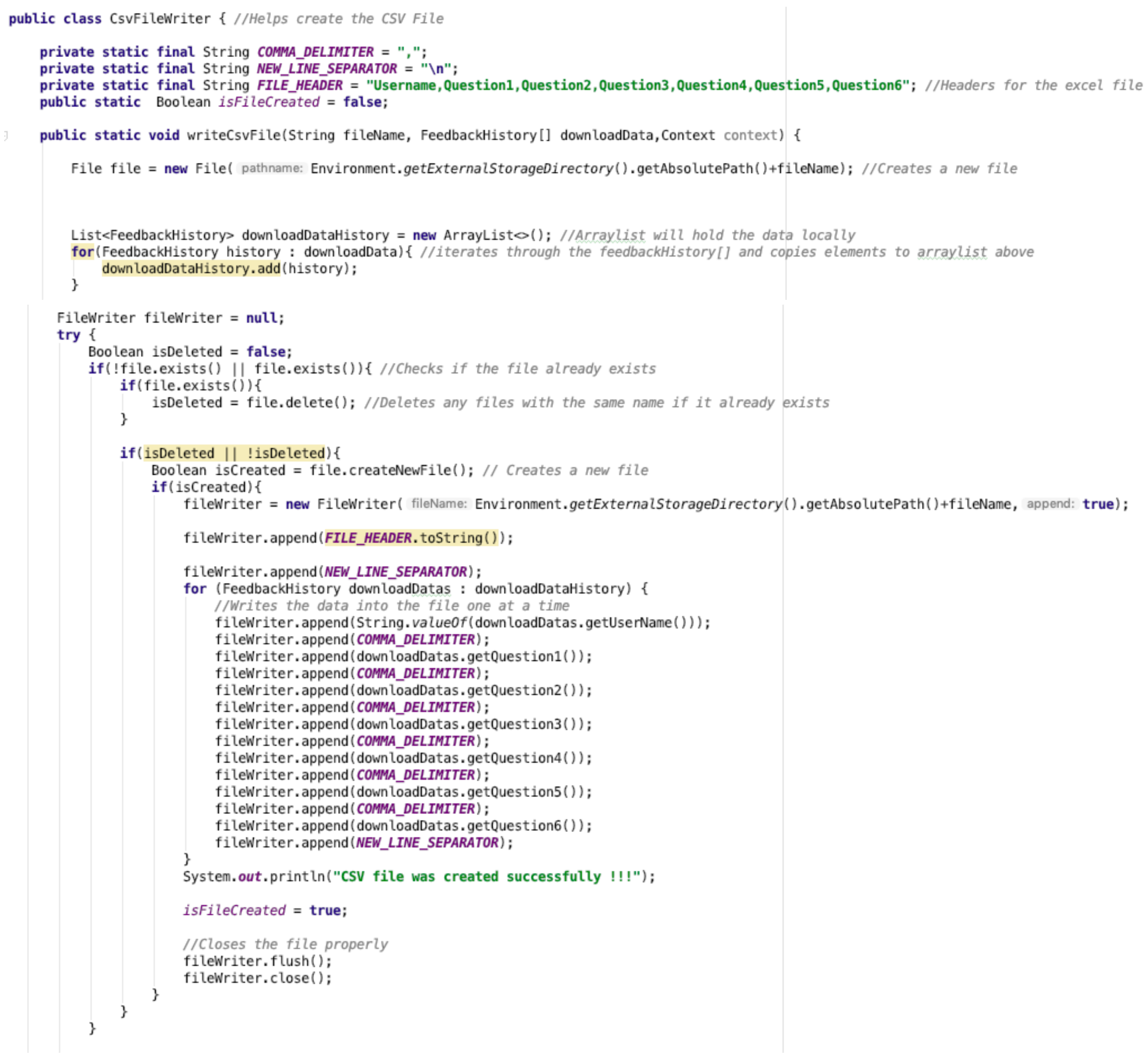
CONSTANTS



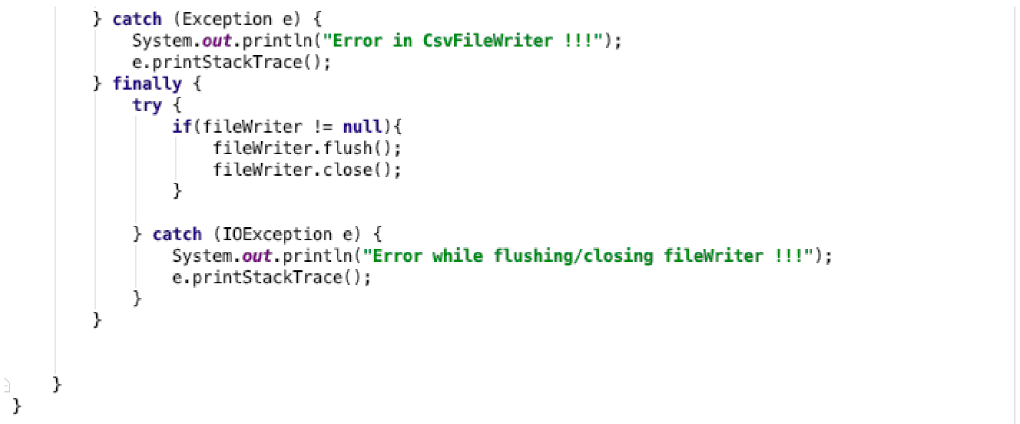
```java
public class Constants {

    public static String BASE_URL = "https://bpsdelegatesapp.000webhostapp.com/delegateapp/code/bps_delegate/";
}
```

This class stores the URL to my online database

CSVFILEWRITER



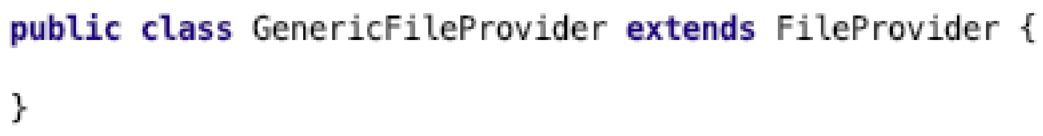
```
public class CsvFileWriter { //Helps create the CSV File

    private static final String COMMA_DELIMITER = ",";
    private static final String NEW_LINE_SEPARATOR = "\n";
    private static final String FILE_HEADER = "Username,Question1,Question2,Question3,Question4,Question5,Question6"; //Headers for the excel file
    public static  Boolean isFileCreated = false;

    public static void writeCsvFile(String fileName, FeedbackHistory[] downloadData,Context context) {

        File file = new File( pathname: Environment.getExternalStorageDirectory().getAbsolutePath()+fileName); //Creates a new file


        List<FeedbackHistory> downloadDataHistory = new ArrayList<>(); //Arraylist will hold the data locally
        for(FeedbackHistory history : downloadData){ //iterates through the feedbackHistory[] and copies elements to arraylist above
            downloadDataHistory.add(history);
        }

      FileWriter fileWriter = null;
      try {
          Boolean isDeleted = false;
          if(!file.exists() || file.exists()){ //Checks if the file already exists
              if(file.exists()){
                  isDeleted = file.delete(); //Deletes any files with the same name if it already exists
              }

              if(isDeleted || !isDeleted){
                  Boolean isCreated = file.createNewFile(); // Creates a new file
                  if(isCreated){
                      fileWriter = new FileWriter( fileName: Environment.getExternalStorageDirectory().getAbsolutePath()+fileName, append: true);

                      fileWriter.append(FILE_HEADER.toString());

                      fileWriter.append(NEW_LINE_SEPARATOR);
                      for (FeedbackHistory downloadDatas : downloadDataHistory) {
                          //Writes the data into the file one at a time
                          fileWriter.append(String.valueOf(downloadDatas.getUserName()));
                          fileWriter.append(COMMA_DELIMITER);
                          fileWriter.append(downloadDatas.getQuestion1());
                          fileWriter.append(COMMA_DELIMITER);
                          fileWriter.append(downloadDatas.getQuestion2());
                          fileWriter.append(COMMA_DELIMITER);
                          fileWriter.append(downloadDatas.getQuestion3());
                          fileWriter.append(COMMA_DELIMITER);
                          fileWriter.append(downloadDatas.getQuestion4());
                          fileWriter.append(COMMA_DELIMITER);
                          fileWriter.append(downloadDatas.getQuestion5());
                          fileWriter.append(COMMA_DELIMITER);
                          fileWriter.append(downloadDatas.getQuestion6());
                          fileWriter.append(NEW_LINE_SEPARATOR);
                      }
                      System.out.println("CSV file was created successfully !!!");

                      isFileCreated = true;

                      //Closes the file properly
                      fileWriter.flush();
                      fileWriter.close();
                  }
              }
          }
```



```
        } catch (Exception e) {
            System.out.println("Error in CsvFileWriter !!!");
            e.printStackTrace();
        } finally {
            try {
                if(fileWriter != null){
                    fileWriter.flush();
                    fileWriter.close();
                }

            } catch (IOException e) {
                System.out.println("Error while flushing/closing fileWriter !!!");
                e.printStackTrace();
            }
        }


    }
}
```

The class above is used to write the excel file

GENERICFILEPROVIDER



```
public class GenericFileProvider extends FileProvider {

}
```

This file provider class is used in conjunction with the CSV writer class

ACTIVITY CONTROLLER CLASSES

LOGIN ACTIVITY

```java
public class LogInActivity extends AppCompatActivity {

    private LogInActivityBinding activityBinding;
    private SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.log_in_activity);

        sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);

//Activity binder saves lines of code by allowing direct access through XML files
        activityBinding = DataBindingUtil.setContentView(this, R.layout.log_in_activity);
        activityBinding.setUser(new User());
        activityBinding.setActivity(this);
    }

    public void onLoginButtonClick(User user){
        doLogin(user);
    }

    public void openSignUpPage(){
        Intent myIntent = new Intent(this, SignUpActivity.class);
        startActivity(myIntent);
    }



    public void doLogin(User user){

//Retrofit + Service facilitates the interaction between the app and the database
        Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
        StudentDelegateService service = retrofit.create(StudentDelegateService.class);
        String tag = "login";
        service.login(tag,user).enqueue(new Callback<UserDetails>() {

            //Handles the response from the API
            @Override
            public void onResponse(@NonNull Call<UserDetails> call, @NonNull Response<UserDetails> response) {
                String userId = response.body().getId();

                if(userId != null ){

                    UserDetails userDetails = response.body();
                    //Stores the userdetails on the local storage of the device
                  sharedPreferences.edit().putString("UserId",userDetails.getId() != null ? userDetails.getId() : "0").apply();
                    sharedPreferences.edit().putString("AccessToken",userDetails.getAccess_token()).apply();
                    sharedPreferences.edit().putString("UserName",userDetails.getUserName()).apply();

                    //Start the menu activity from the log in page
                    Intent intent = new Intent(LogInActivity.this,MenuActivity.class);
                    startActivity(intent);

                }else{
                    Toast.makeText(getApplicationContext(),"Something went wrong. Please check your credentials and try again",
                            Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(@NonNull Call<UserDetails> call, @NonNull Throwable t) {
                Log.d("Failure====","Failure===="+t.getMessage());
                Toast.makeText(getApplicationContext(),"Something went wrong. Please check your credentials and try again",
                        Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

SIGN UP ACTIVITY

```
public class SignUpActivity extends AppCompatActivity {

    private Button createAccountButton, viewAllData, logInButton;
    private EditText userNamePassword;
    private EditText userNameSignUp;
    private EditText emailIDSignUp;
    private Spinner yearGroupSpinner;
    private SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sign_up_activity);

        sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);

        userNameSignUp = findViewById(R.id.userNameSignUp);
        userNamePassword = findViewById(R.id.passwordSignUp);
        createAccountButton = findViewById(R.id.btnCreateAccount);
        yearGroupSpinner = findViewById(R.id.SignUpYearGroupSpinner);
        logInButton = findViewById(R.id.btnLogIn);
        emailIDSignUp = findViewById(R.id.emailIDSignUP);
        String[] items = new String[]{"Year 6", "Year 7", "Year 8", "Year 9", "Year 10", "Year 11", "Year 12", "Year 13"};
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, items);
        yearGroupSpinner.setAdapter(adapter);


        createAccountButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                NewAccount newAccount = new NewAccount();

                //Get user inputs from the screen and store into newAccount object for future use
                newAccount.setUsername(userNameSignUp.getText().toString());
                newAccount.setPassword(userNamePassword.getText().toString());
                newAccount.setYearGroup(yearGroupSpinner.getSelectedItem().toString());
                newAccount.setAdmin(0);

                if(validateSignUp()) { //ensure that userName is of the correct format
                        doSignUp(newAccount); //if the userName is of a valid format, then sign up
                }

            }
        });

        logInButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openLoginPage();
            }
        });
    }


    private void doSignUp(final NewAccount newAccount){

        //Retrofit facilitates the connection of app to API files
        Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
        StudentDelegateService service = retrofit.create(StudentDelegateService.class);
        String tag = "register"; //Used to determine the signUp process within the API

        service.signUp(tag,newAccount).enqueue(new
Callback<com.example.dhruvmittal.studentdelegatesapp.model.Response>() {
```

```
            //Get response from the API and store it in the "Response" model class
            @RequiresApi(api = Build.VERSION_CODES.KITKAT)
            @Override
            public void onResponse(@NonNull Call<com.example.dhruvmittal.studentdelegatesapp.model.Response> call,
@NonNull Response<com.example.dhruvmittal.studentdelegatesapp.model.Response> response) {
                if(response.code() == 200){
                    Toast.makeText(getApplicationContext(),"User registered successfully",Toast.LENGTH_LONG).show();
                    sendEmail(newAccount);
                    Intent intent = new Intent(SignUpActivity.this, MenuActivity.class);
                    startActivity(intent);

                }else{
                    Toast.makeText(getApplicationContext(),"Something went wrong. Please check your credentials and try
again",Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(@NonNull Call<com.example.dhruvmittal.studentdelegatesapp.model.Response> call, @NonNull
Throwable t) {
                Log.d("Failure====","Failure===="+t.getMessage());
                Toast.makeText(getApplicationContext(),"Something went wrong. Please check your credentials and try
again",Toast.LENGTH_LONG).show();
            }
        });

    }

    @RequiresApi(api = Build.VERSION_CODES.KITKAT)
    private void sendEmail(NewAccount newAccount){

        final String senderEmailID = "mitta52639@gmail.com";
        final String senderEmailPassword = "Dhruv2911";
        final String subject = "Your new Student Delegates Feedback App account";
        final String recieverEmailID = emailIDSignUp.getText().toString();
        String userName = newAccount.getUsername();
        String password = newAccount.getPassword();
        final String body = "Thank you for creating an account on the Student Delegates Feedback App"+ System.lineSeparator()+
                "Your log in details are sent below, please store these for future use:"
                + System.lineSeparator()+ System.lineSeparator()+
                "Username: "+ userName + System.lineSeparator()+ "Password: "+ password;



        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    GMailSender sender = new GMailSender(senderEmailID,
                            senderEmailPassword);
                    sender.sendMail(subject, body, senderEmailID, recieverEmailID);
                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(),"Email could not be sent successfully",Toast.LENGTH_LONG).show();
                }
            }

        }).start();

        Toast.makeText(getApplicationContext(),"Email sent successfully",Toast.LENGTH_LONG).show();
    }

    private boolean validateSignUp(){ //Checks if the entered username if of the correct format (Eg: abcd12)
        boolean isvalid = false;

        if (userNameSignUp.getText().toString().equalsIgnoreCase("")){
            userNameSignUp.setError("Please enter user name");
            userNameSignUp.requestFocus();
            isvalid = false;
        }else if (userNameSignUp.getText().toString().length() < 6){
            userNameSignUp.setError("Please Enter valid username that is 6 digits long");
            userNameSignUp.requestFocus();
            isvalid = false;
        }else {
            String user = userNameSignUp.getText().toString();
            String d = user.substring(user.length()-2,user.length()); // 12
            String c = user.substring(0,user.length()-2); // char

            try{
                Integer.parseInt(d);
                isvalid = true;

            }catch (NumberFormatException e){
                isvalid = false;
                userNameSignUp.setError("Please Enter valid username");
                userNameSignUp.requestFocus();
            }

            if (c.matches(".*\\d.*")){
                isvalid = false;
                userNameSignUp.setError("Please Enter valid username");
            }
        }

        if(userNamePassword.getText().toString().equalsIgnoreCase("")){
            userNamePassword.setError("Please enter a password");
            userNamePassword.requestFocus();
            isvalid = false;
        } else if(userNamePassword.getText().toString().length() > 15){
            userNamePassword.setError("Please ensure your password is less than 15 characters");
            userNamePassword.requestFocus();
            isvalid = false;
        }
        return isvalid;
    }

    private void openLoginPage(){
        Intent myIntent = new Intent(this, LogInActivity.class);
        startActivity(myIntent);
    }
}
```

MENU ACTIVIY

```
public class MenuActivity extends AppCompatActivity {

    private MenuActivitityBinding activityBinding;
    private TextView displayUserName;
private SharedPreferences sharedPreferences;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.menu_activitity);

    sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);

    displayUserName = findViewById(R.id.DisplayUsername);
    String userName = sharedPreferences.getString("UserName", null);
    displayUserName.setText("Username: "+ userName); //Displays the current username on the menu screen



    activityBinding = DataBindingUtil.setContentView(this, R.layout.menu_activitity);
    activityBinding.setActivity(this);



}
public void openCanteenPage(){
    Intent openCanteenOptionPage = new Intent(this, CanteenOptionPageActivity1.class);
    startActivity(openCanteenOptionPage);
}

public void openSubjectPage(){
    Intent openSubjectOptionPage = new Intent(this, SubjectOptionPageActivity1.class);
    startActivity(openSubjectOptionPage);
}

public void openECAPage(){
    Intent openECAOptionPage = new Intent(this, ECAOptionPageActivity1.class);
    startActivity(openECAOptionPage);
}

public void openFreePeriodPage(){
    Intent openFreePeriodPage = new Intent(this, StudyPeriodOptionPageActivity1.class);
    startActivity(openFreePeriodPage);
}

public void openTeacherPage(){
    Intent openTeachersPage = new Intent(this, TeacherOptionPageActivity1.class);
    startActivity(openTeachersPage);
}

public void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }

    private void openReportPage(){
        Intent openReportPage = new Intent(this, ViewReportActivity.class);
        startActivity(openReportPage);
    }

    public void openHistoryPage(){
        Intent openHistoryPage = new Intent(this, ViewHistoryActivity.class);
        startActivity(openHistoryPage);
    }

    public void openUpdateAccountPage(){
        Intent openUpdateAccountPage = new Intent(this, UpdateAccountActivity.class);
        startActivity(openUpdateAccountPage);
    }
}
```

CANTEEN OPTION PAGE 1 ACTIVITY

```
public class CanteenOptionPageActivity1 extends AppCompatActivity {

    private Button menu, logOut, nextPage;
    private char Q1, Q2;
    private boolean Q3;
    private CheckBox q1a, q1b, q1c, q1d;
    private CheckBox q2a, q2b, q2c, q2d;
    private CheckBox q3a, q3b;

    private CanteenOptionPageActivity1Binding activity1Binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.canteen_option_page_activity_1);

        activity1Binding = DataBindingUtil.setContentView(this, R.layout.canteen_option_page_activity_1);
        activity1Binding.setActivity(this);
        menu = findViewById(R.id.CanteenMenuButton1);
        logOut = findViewById(R.id.CanteenLogOutButton1);
        nextPage = findViewById(R.id.CanteenNextPageButton);

        q1a = findViewById(R.id.CanteenAnswer1A);
        q1b = findViewById(R.id.CanteenAnswer1B);
        q1c = findViewById(R.id.CanteenAnswer1C);
        q1d = findViewById(R.id.CanteenAnswer1D);

        q2a = findViewById(R.id.CanteenAnswer2A);
        q2b = findViewById(R.id.CanteenAnswer2B);
        q2c = findViewById(R.id.CanteenAnswer2C);
        q2d = findViewById(R.id.CanteenAnswer2D);



        q3a = findViewById(R.id.CanteenAnswer3A);
        q3b = findViewById(R.id.CanteenAnswer3B);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openMenu();
            }
        });

        logOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openLogInPage();
            }
        });

        nextPage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //When opening the next page, all the data from the current page must be passed into the next page so
the data is not lost
                //Here I have taken the user inputs and then passed them to the next page in a "bundle" below:

                Q1 = getAnswer1();
                Q2 = getAnswer2();
                Q3 = getAnswer3();

                Intent openNextCanteenPage = new Intent(CanteenOptionPageActivity1.this,
        CanteenOptionPageActivity2.class);
                Bundle canteenOptionPage1DataBundle = new Bundle();
                canteenOptionPage1DataBundle.putChar("q1", Q1);
                canteenOptionPage1DataBundle.putChar("q2", Q2);
                canteenOptionPage1DataBundle.putBoolean("q3", Q3);

                openNextCanteenPage.putExtras(canteenOptionPage1DataBundle);
                startActivity(openNextCanteenPage);

            }
        });
    }

    public void onCheckboxQ1Checked(){
        q1a.setEnabled(false);
        q1b.setEnabled(false);
        q1c.setEnabled(false);
        q1d.setEnabled(false);
    }

    public void onCheckboxQ2Checked(){
        q2a.setEnabled(false);
        q2b.setEnabled(false);
        q2c.setEnabled(false);
        q2d.setEnabled(false);
    }

    public void onCheckboxQ3Checked(){
        q3a.setEnabled(false);
        q3b.setEnabled(false);
    }

    private char getAnswer1(){
        char answer = 'f'; //f is a default value corresponding to error

        if(q1a.isChecked() && !q1b.isChecked()&& !q1c.isChecked()&& !q1d.isChecked()){
```

```
        System.out.println("Q1A selected");
        answer = 'a';
    } else if(!q1a.isChecked() && q1b.isChecked()&& !q1c.isChecked()&& !q1d.isChecked()){
        System.out.println("Q1B Selected");
        answer = 'b';
    } else if(!q1a.isChecked() && !q1b.isChecked()&& q1c.isChecked()&& !q1d.isChecked()){
        System.out.println("Q1C Selected");
        answer = 'c';
    }
    else if(!q1a.isChecked() && !q1b.isChecked()&& !q1c.isChecked()&& q1d.isChecked()){
        System.out.println("Q1D Selected");
        answer = 'd';
    } else{
        System.out.println("Please check only 1 box.");
    }
    return answer;
}

private char getAnswer2(){
    char answer = 'f'; //f is a default value corresponding to error

    if(q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
        System.out.println("Q2A selected");
        answer = 'a';
    } else if(!q2a.isChecked() && q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
        System.out.println("Q2B Selected");
        answer = 'b';
    } else if(!q2a.isChecked() && !q2b.isChecked()&& q2c.isChecked()&& !q2d.isChecked()){
        System.out.println("Q2C Selected");
        answer = 'c';
    }
    else if(!q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& q2d.isChecked()){
        System.out.println("Q2D Selected");
        answer = 'd';
    } else{
        System.out.println("Please check only 2 box.");
    }
    return answer;
}
```

```
    private boolean getAnswer3(){
        boolean answer = false; //f is a default value corresponding to error

        if(q3a.isChecked() && !q3b.isChecked()){
            System.out.println("Q3A selected");
            answer = true;
        } else if(!q3a.isChecked() && q3b.isChecked()){
            System.out.println("Q3B Selected");
            answer = false;
        } else{
            System.out.println("Please check only 3 box.");
        }
        return answer;
    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }
}
```

CANTEEN OPTION PAGE 2 ACTIVITY

```
public class CanteenOptionPageActivity2 extends AppCompatActivity {

    private Button menu, logOut, submit;
    private char Q1, Q2;
    private String Q3;
    private String Q4, Q5, Q6;
    private EditText Q4EditText,Q5EditText,Q6EditText;

    SharedPreferences sharedPreferences;

    int userId = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.canteen_option_page_activity_2);

        sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);

        userId = Integer.parseInt(sharedPreferences.getString("UserId",null));

        menu = findViewById(R.id.CanteenMenuButton2);
        logOut = findViewById(R.id.CanteenLogOutButton2);
        submit = findViewById(R.id.CanteenSubmitButton);
        Q4EditText = findViewById(R.id.CanteenQuestion4EditText);
        Q5EditText = findViewById(R.id.CanteenQuestion5EditText);
        Q6EditText = findViewById(R.id.CanteenOtherCommentsEditText);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openMenu();
            }
        });
```

```java
        logOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openLogInPage();
            }
        });

        submit.setOnClickListener(new View.OnClickListener() { //Get the required data and store in database from here.
            @Override
            public void onClick(View view) {

                //Get data from page 1:
                Intent intentExtras = getIntent();
                Bundle bundle = intentExtras.getExtras();

                if(bundle != null){
                    Q1 = bundle.getChar("q1", 'f');
                    Q2 = bundle.getChar("q2",'f');
                    if(String.valueOf(bundle.getBoolean("q3")).equals("true")){
                        Q3 = "true";
                    } else if (String.valueOf(bundle.getBoolean("q3")).equals("false")){
                        Q3 = "false";
                    } else{
                        Q3 = "error";
                        System.out.println("Error in getting Q3 from CanteenOptionPage1");
                    }
                }



                //Get data from the current page:
                Q4 = Q4EditText.getText().toString();
                Q5 = Q5EditText.getText().toString();
                Q6 = Q6EditText.getText().toString();

                //Add data to table in database
                addEntryToTblOption(userId, String.valueOf(Q1),String.valueOf(Q2),Q3,Q4,Q5,Q6);

            }
        });

    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }

    private void addEntryToTblOption(int userId, String Q1, String Q2, String Q3, String Q4, String Q5, String Q6){

        String categoryType = "Canteen"; //This will store the category type in the table
        String tag = "saveFeedBack"; //This tag will determine which API function to call

        //Access token is used to verify if the user has authority to add feedback
        String accessToken = sharedPreferences.getString("AccessToken",null);

        //Creates an object from the model class "feedback"
        Feedback feedback = new Feedback();
        feedback.setAccessToken(accessToken);
        feedback.setUserId(userId);
        feedback.setQuestion1(Q1);
        feedback.setQuestion2(Q2);
        feedback.setQuestion3(Q3);
        feedback.setQuestion4(Q4);
        feedback.setQuestion5(Q5);
        feedback.setQuestion6(Q6);
        feedback.setCategoryType(categoryType);



        //Retrofit + service interface facilitate connection to API
        Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
        StudentDelegateService service = retrofit.create(StudentDelegateService.class);

        service.saveFeedback(tag,feedback).enqueue(new Callback<Response>() {
            @Override
            public void onResponse(Call<Response> call, retrofit2.Response<Response> response) { //Executes when API
responds
                if(response.code() == 200){ //200 is the success code for my APIs
                    Toast.makeText(getApplicationContext(),"Feedback for canteen saved
successfully",Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(getApplicationContext(),"Something went wrong.Please try
again",Toast.LENGTH_LONG).show();
                }
            }
            @Override
            public void onFailure(Call<Response> call, Throwable t) { //Executes if something went wrong in the API
                Log.d("Error==","Error=="+t.getMessage());
                Toast.makeText(getApplicationContext(),"Something went wrong.Please try again",Toast.LENGTH_LONG).show();
            }
        });

    }
}
```

ECA OPTION PAGE 1 ACTIVITY

```
public class ECAOptionPageActivity1 extends AppCompatActivity {

    private Button menu, logOut, nextPage;
    private char Q1, Q2, Q3;
    private CheckBox q1a, q1b, q1c, q1d;
    private CheckBox q2a, q2b, q2c, q2d;
    private CheckBox q3a, q3b, q3c, q3d;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.eca_option_page_activity_1);

        menu = findViewById(R.id.ECAMenuButton1);
        logOut = findViewById(R.id.ECALogOutButton1);
        nextPage = findViewById(R.id.ECANextPageButton);



        q1a = findViewById(R.id.ECAAnswer1A);
        q1b = findViewById(R.id.ECAAnswer1B);
        q1c = findViewById(R.id.ECAAnswer1C);
        q1d = findViewById(R.id.ECAAnswer1D);

        q2a = findViewById(R.id.ECAAnswer2A);
        q2b = findViewById(R.id.ECAAnswer2B);
        q2c = findViewById(R.id.ECAAnswer2C);
```

```
q2d = findViewById(R.id.ECAAnswer2D);


q3a = findViewById(R.id.ECAAnswer3A);
q3b = findViewById(R.id.ECAAnswer3B);
q3c = findViewById(R.id.ECAAnswer3C);
q3d = findViewById(R.id.ECAAnswer3D);

menu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        openMenu();
    }
});

logOut.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        openLogInPage();
    }
});

nextPage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Q1 = getAnswer1();
        Q2 = getAnswer2();
        Q3 = getAnswer3();

        Intent openNextECAPage = new Intent(ECAOptionPageActivity1.this, ECAOptionPageActivity2.class);
        Bundle ECAOptionPage1DataBundle = new Bundle();
        ECAOptionPage1DataBundle.putChar("q1", Q1);
        ECAOptionPage1DataBundle.putChar("q2", Q2);
        ECAOptionPage1DataBundle.putChar("q3", Q3);

        openNextECAPage.putExtras(ECAOptionPage1DataBundle);
        startActivity(openNextECAPage);

        //openNextECAPage();
    }
```

```
        });
    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }
/*
    private void openNextECAPage(){
        Intent openNextECAPage = new Intent(this, ECAOptionPageActivity2.class);
        startActivity(openNextECAPage);
    }
*/

    private char getAnswer1(){
        char answer = 'f'; //f is a default value corresponging to error

        if(q1a.isChecked() && !q1b.isChecked()&& !q1c.isChecked()&& !q1d.isChecked()){
            System.out.println("Q1A selected");
            answer = 'a';
        } else if(!q1a.isChecked() && q1b.isChecked()&& !q1c.isChecked()&& !q1d.isChecked()){
            System.out.println("Q1B Selected");
            answer = 'b';
        } else if(!q1a.isChecked() && !q1b.isChecked()&& q1c.isChecked()&& !q1d.isChecked()){
            System.out.println("Q1C Selected");
            answer = 'c';
        }
        else if(!q1a.isChecked() && !q1b.isChecked()&& !q1c.isChecked()&& q1d.isChecked()){
            System.out.println("Q1D Selected");
            answer = 'd';
        } else{
            System.out.println("Please check only 1 box.");
        }
        return answer;
    }
```

```
private char getAnswer2(){
    char answer = 'f'; //f is a default value corresponging to error

    if(q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
        System.out.println("Q2A selected");
        answer = 'a';
    } else if(!q2a.isChecked() && q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
        System.out.println("Q2B Selected");
        answer = 'b';
    } else if(!q2a.isChecked() && !q2b.isChecked()&& q2c.isChecked()&& !q2d.isChecked()){
        System.out.println("Q2C Selected");
        answer = 'c';
    }
    else if(!q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& q2d.isChecked()){
        System.out.println("Q2D Selected");
        answer = 'd';
    } else{
        System.out.println("Please check only 1 box.");
    }
    return answer;
}

private char getAnswer3(){
    char answer = 'f'; //f is a default value corresponging to error

    if(q3a.isChecked() && !q3b.isChecked()&& !q3c.isChecked()&& !q3d.isChecked()){
        System.out.println("Q3A selected");
        answer = 'a';
    } else if(!q3a.isChecked() && q3b.isChecked()&& !q3c.isChecked()&& !q3d.isChecked()){
        System.out.println("Q3B Selected");
        answer = 'b';
    } else if(!q3a.isChecked() && !q3b.isChecked()&& q3c.isChecked()&& !q3d.isChecked()){
        System.out.println("Q3C Selected");
        answer = 'c';
    }
    else if(!q3a.isChecked() && !q3b.isChecked()&& !q3c.isChecked()&& q3d.isChecked()){
        System.out.println("Q3D Selected");
        answer = 'd';
    } else{
            System.out.println("Please check only 1 box.");
        }
        return answer;
    }

}
```

ECA OPTION PAGE 2 ACTIVITY

```
public class ECAOptionPageActivity2 extends AppCompatActivity {

    private Button menu, logOut, submit;
    private char Q1, Q2, Q3;
    private String Q4, Q5, Q6;
    private EditText Q4EditText,Q5EditText,Q6EditText;

    SharedPreferences sharedPreferences;

    int userId = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.eca_option_page_activity_2);
```

```java
sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);
userId = Integer.parseInt(sharedPreferences.getString("UserId",null));

menu = findViewById(R.id.ECAMenuButton2);
logOut = findViewById(R.id.ECALogOutButton2);
submit = findViewById(R.id.ECASubmitButton);

Q4EditText = findViewById(R.id.ECAQuestion4EditText);
Q5EditText = findViewById(R.id.ECAQuestion5EditText);
Q6EditText = findViewById(R.id.ECAOtherCommentsEditText);

menu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        openMenu();
    }
});

logOut.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        openLogInPage();
    }
});

submit.setOnClickListener(new View.OnClickListener() { //Get the required data and store in database from here.
    @Override
    public void onClick(View view) {

        //Get data from page 1:
        Intent intentExtras = getIntent();
        Bundle bundle = intentExtras.getExtras();

        if(bundle != null){
            Q1 = bundle.getChar("q1", 'f');
            Q2 = bundle.getChar("q2",'f');
            Q3 = bundle.getChar("q3", 'f');
        }

        //Get data from the current page:
        Q4 = Q4EditText.getText().toString();
        Q5 = Q5EditText.getText().toString();
        Q6 = Q6EditText.getText().toString();
```

```
                addEntryToTblOption(userId, String.valueOf(Q1),String.valueOf(Q2),String.valueOf(Q3),Q4,Q5,Q6);

                //System.out.println(Q1 +"    "+ Q2+"    "+ Q3+ "    "+ Q4 + "    "+ Q5+ "    "+ Q6);

                //Update userHistory after submitting
                //updateUserHistory();
            }
        });

    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }

    //Here, we must CREATE a new record with the UserName,
    private void addEntryToTblOption(int userId, String Q1, String Q2, String Q3, String Q4, String Q5, String Q6){
        String categoryType = "ECA";
        //System.out.print("UserId==="+userId+" Q1=="+Q1+" Q2="+Q2+" Q3="+Q3+" Q4="+Q4+" Q5="+Q5+" Q6="+Q6);
        String tag = "saveFeedBack";
        String accessToken = sharedPreferences.getString("AccessToken",null);
        Feedback feedback = new Feedback();
        feedback.setAccessToken(accessToken);
        feedback.setUserId(userId);
        feedback.setQuestion1(Q1);
        feedback.setQuestion2(Q2);
        feedback.setQuestion3(Q3);
        feedback.setQuestion4(Q4);
        feedback.setQuestion5(Q5);
        feedback.setQuestion6(Q6);
        feedback.setCategoryType(categoryType);



        Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
        StudentDelegateService service = retrofit.create(StudentDelegateService.class);

        service.saveFeedback(tag,feedback).enqueue(new Callback<Response>() {
            @Override
            public void onResponse(Call<Response> call, retrofit2.Response<Response> response) {
                if(response.code() == 200){
                    Toast.makeText(getApplicationContext(),"Feedback for ECA saved successfully",Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(getApplicationContext(),"Something went wrong.Please try
again",Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(Call<Response> call, Throwable t) {
                Log.d("Error==","Error=="+t.getMessage());
                Toast.makeText(getApplicationContext(),"Something went wrong.Please try again",Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

STUDY PERIOD OPTION PAGE 1 ACTIVITY

```
public class StudyPeriodOptionPageActivity1 extends AppCompatActivity {

    private Button menu, logOut, nextPage;
    private char Q1, Q2;
    private boolean Q3;
    private CheckBox q1a, q1b, q1c, q1d;
    private CheckBox q2a, q2b, q2c, q2d;
    private CheckBox q3a, q3b;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.study_period_option_page_activity_1);
        menu = findViewById(R.id.StudyPeriodMenuButton1);
        logOut = findViewById(R.id.StudyPeriodLogOutButton1);
        nextPage = findViewById(R.id.StudyPeriodNextPageButton);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openMenu();
            }
        });

        logOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openLogInPage();
            }
        });

        nextPage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Q1 = getAnswer1();
                Q2 = getAnswer2();
                Q3 = getAnswer3();

                Intent openNextStudyPeriodPage = new Intent(StudyPeriodOptionPageActivity1.this,
StudyPeriodOptionPageActivity2.class);
                Bundle StudyPeriodOptionPage1DataBundle = new Bundle();
                StudyPeriodOptionPage1DataBundle.putChar("q1", Q1);
                StudyPeriodOptionPage1DataBundle.putChar("q2", Q2);
                StudyPeriodOptionPage1DataBundle.putBoolean("q3", Q3);

                openNextStudyPeriodPage.putExtras(StudyPeriodOptionPage1DataBundle);
                startActivity(openNextStudyPeriodPage);

                //openNextCanteenPage();
            }
        });
        q1a = findViewById(R.id.StudyPeriodAnswer1A);
        q1b = findViewById(R.id.StudyPeriodAnswer1B);
        q1c = findViewById(R.id.StudyPeriodAnswer1C);
        q1d = findViewById(R.id.StudyPeriodAnswer1D);

        q2a = findViewById(R.id.StudyPeriodAnswer2A);
        q2b = findViewById(R.id.StudyPeriodAnswer2B);
        q2c = findViewById(R.id.StudyPeriodAnswer2C);
        q2d = findViewById(R.id.StudyPeriodAnswer2D);



        q3a = findViewById(R.id.StudyPeriodAnswer3A);
        q3b = findViewById(R.id.StudyPeriodAnswer3B);

    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }

    private void openNextStudyPeriodPage(){
        Intent openNextStudyPeriodPage = new Intent(this, StudyPeriodOptionPageActivity2.class);
        startActivity(openNextStudyPeriodPage);
    }



    private char getAnswer1(){
        char answer = 'f'; //f is a default value corresponging to error

        if(q1a.isChecked() && !q1b.isChecked()&& !q1c.isChecked()&& !q1d.isChecked()){
            System.out.println("Q1A selected");
            answer = 'a';
        } else if(!q1a.isChecked() && q1b.isChecked()&& !q1c.isChecked()&& !q1d.isChecked()){
            System.out.println("Q1B Selected");
            answer = 'b';
        } else if(!q1a.isChecked() && !q1b.isChecked()&& q1c.isChecked()&& !q1d.isChecked()){
            System.out.println("Q1C Selected");
            answer = 'c';
        }
        else if(!q1a.isChecked() && !q1b.isChecked()&& !q1c.isChecked()&& q1d.isChecked()){
            System.out.println("Q1D Selected");
            answer = 'd';
        } else{
            System.out.println("Please check only 1 box.");
        }
        return answer;
    }

    private char getAnswer2(){
        char answer = 'f'; //f is a default value corresponging to error

        if(q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
            System.out.println("Q2A selected");
            answer = 'a';
        } else if(!q2a.isChecked() && q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
            System.out.println("Q2B Selected");
            answer = 'b';
        } else if(!q2a.isChecked() && !q2b.isChecked()&& q2c.isChecked()&& !q2d.isChecked()){
            System.out.println("Q2C Selected");
            answer = 'c';
        }
        else if(!q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& q2d.isChecked()){
            System.out.println("Q2D Selected");
            answer = 'd';
        } else{
            System.out.println("Please check only 2 box.");
        }
        return answer;
    }

    private boolean getAnswer3(){
        boolean answer = false;

        if(q3a.isChecked() && !q3b.isChecked()){
            System.out.println("Q3A selected");
            answer = true;
        } else if(!q3a.isChecked() && q3b.isChecked()){
            System.out.println("Q3B Selected");
            answer = false;
        } else{
            System.out.println("Please check only 3 box.");
        }
        return answer;
    }
}
```

STUDY PERIOD OPTION PAGE 2 ACTIVITY

```
public class StudyPeriodOptionPageActivity2 extends AppCompatActivity {
```

```java
private Button menu, logOut, submit;
private char Q1, Q2;
private String Q3;
private String Q4, Q5, Q6;
private EditText Q4EditText,Q5EditText,Q6EditText;

SharedPreferences sharedPreferences;

int userId = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.study_period_option_page_activity_2);

    menu = findViewById(R.id.StudyPeriodMenuButton2);
    logOut = findViewById(R.id.StudyPeriodLogOutButton2);
    submit = findViewById(R.id.StudyPeriodSubmitButton);

    sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);

    userId = Integer.parseInt(sharedPreferences.getString("UserId",null));

    Q4EditText = findViewById(R.id.StudyPeriodQuestion4EditText);
    Q5EditText = findViewById(R.id.StudyPeriodQuestion5EditText);
    Q6EditText = findViewById(R.id.StudyPeriodOtherCommentsEditText);



    menu.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            openMenu();
        }
    });

    logOut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            openLogInPage();
        }
    });

    submit.setOnClickListener(new View.OnClickListener() { //Get the required data and store in database from here.
        @Override
        public void onClick(View view) {
            //Get data from page 1:
            Intent intentExtras = getIntent();
            Bundle bundle = intentExtras.getExtras();

            if(bundle != null){
                Q1 = bundle.getChar("q1", 'f');
                Q2 = bundle.getChar("q2",'f');
                if(String.valueOf(bundle.getBoolean("q3")).equals("true")){
                    Q3 = "true";
                } else if (String.valueOf(bundle.getBoolean("q3")).equals("false")){
                    Q3 = "false";
                } else{
                    Q3 = "error";
                    System.out.println("Error in getting Q3 from StudyPeriodOptionPage1");
                }
            }

            //Get data from the current page:
            Q4 = Q4EditText.getText().toString();
            Q5 = Q5EditText.getText().toString();
            Q6 = Q6EditText.getText().toString();

            addEntryToTblOption(userId, String.valueOf(Q1),String.valueOf(Q2),String.valueOf(Q3),Q4,Q5,Q6);

            //System.out.println(Q1 +"    "+ Q2+"    "+ Q3+ "   "+ Q4 + "    "+ Q5+ "   "+ Q6);

            //Update userHistory after submitting
            //updateUserHistory();
        }
    });

}

private void openMenu(){
    Intent openMenuPage = new Intent(this, MenuActivity.class);
    startActivity(openMenuPage);
}

private void openLogInPage(){
    Intent openLogInPage = new Intent(this, LogInActivity.class);
    startActivity(openLogInPage);
}

//Here, we must CREATE a new record with the UserName,
    private void addEntryToTblOption(int userId, String Q1, String Q2, String Q3, String Q4, String Q5, String Q6){
        String categoryType = "StudyPeriod";
        //System.out.print("UserId==="+userId+" Q1=="+Q1+" Q2="+Q2+" Q3="+Q3+" Q4="+Q4+" Q5="+Q5+" Q6="+Q6);
        String tag = "saveFeedBack";
        String accessToken = sharedPreferences.getString("AccessToken",null);
        Feedback feedback = new Feedback();
        feedback.setAccessToken(accessToken);
        feedback.setUserId(userId);
        feedback.setQuestion1(Q1);
        feedback.setQuestion2(Q2);
        feedback.setQuestion3(Q3);
        feedback.setQuestion4(Q4);
        feedback.setQuestion5(Q5);
        feedback.setQuestion6(Q6);
        feedback.setCategoryType(categoryType);



        Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
        StudentDelegateService service = retrofit.create(StudentDelegateService.class);

        service.saveFeedback(tag,feedback).enqueue(new Callback<Response>() {
            @Override
            public void onResponse(Call<Response> call, retrofit2.Response<Response> response) {
                if(response.code() == 200){
                    Toast.makeText(getApplicationContext(),"Feedback for Study Periods saved
successfully",Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(getApplicationContext(),"Something went wrong.Please try
again",Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(Call<Response> call, Throwable t) {
                Log.d("Error==","Error=="+t.getMessage());
                Toast.makeText(getApplicationContext(),"Something went wrong.Please try again",Toast.LENGTH_LONG).show();
            }
        });

    }
}
```

SUBJECT OPTION PAGE 1 ACTIVITY

```java
public class SubjectOptionPageActivity1 extends AppCompatActivity {

    private Button menu, logOut, nextPage;
    private String Q1;
    private int Q2;
    private boolean Q3;
    private CheckBox q1a, q1b, q1c, q1d;
    private CheckBox q2a, q2b, q2c, q2d;
    private CheckBox q3a, q3b;
    private Spinner subjectSpinner, homeworkTimeSpinner;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.subject_option_page_activity_1);

        menu = findViewById(R.id.SubjectMenuButton1);
        logOut = findViewById(R.id.SubjectLogOutButton1);
        nextPage = findViewById(R.id.SubjectNextPageButton);

        q3a = findViewById(R.id.SubjectAnswer3A);
        q3b = findViewById(R.id.SubjectAnswer3B);

        subjectSpinner = findViewById(R.id.SubjectSpinner);
        homeworkTimeSpinner = findViewById(R.id.HomeworkTimeSpinner);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openMenu();
            }
        });

        logOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openLogInPage();
            }
        });

        nextPage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Q1 = getAnswer1();
                Q2 = getAnswer2();
                Q3 = getAnswer3();
                Intent openNextSubjectPage = new Intent(SubjectOptionPageActivity1.this, SubjectOptionPageActivity2.class);
                Bundle SubjectOptionPage1DataBundle = new Bundle();
                SubjectOptionPage1DataBundle.putString("q1", Q1);
                SubjectOptionPage1DataBundle.putInt("q2", Q2);
                SubjectOptionPage1DataBundle.putBoolean("q3", Q3);

                openNextSubjectPage.putExtras(SubjectOptionPage1DataBundle);
                startActivity(openNextSubjectPage);

                //openNextSubjectPage();
            }
        });

        String[] items = new String[]{"Mathematics", "Biology", "Physics", "Chemistry", "Geography", "History", "Physical Education (P.E)",
"Business Studies", "Economics", "Art"
                , "DT", "ITGS", "Computer Science", "Theory of Knowledge", "CAS", "Chinese", "Spanish", "German", "French"
                , "Thai", "Other Language", "Drama", "Music", "ESS"};
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, items);
        subjectSpinner.setAdapter(adapter);

        String[] homeworkHours = new String[]{"0 Hours", "1 Hour", "2 Hours", "3 Hours", "4 Hours", "5 Hours", "6 Hours", "6+ Hours"};
        ArrayAdapter<String> adapter1 = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, homeworkHours);
        homeworkTimeSpinner.setAdapter(adapter1);

    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }

/*
    private void openNextSubjectOptionPage(){
        Intent openNextSubjectOptionPage = new Intent(this, SubjectOptionPageActivity2.class);
        startActivity(openNextSubjectOptionPage);
    }
*/

    private String getAnswer1(){
        String answer = subjectSpinner.getSelectedItem().toString();
        return answer;
    }
```

```java
private int getAnswer2(){
    int answer;
    String text = homeworkTimeSpinner.getSelectedItem().toString();

    switch (text){
        case "0 Hours":
            answer = 0;
            break;
        case "1 Hour":
            answer = 1;
            break;
        case "2 Hours":
            answer = 2;
            break;
        case "3 Hours":
            answer = 3;
            break;
        case "4 Hours":
            answer = 4;
            break;
        case "5 Hours":
            answer = 5;
            break;
        case "6 Hours":
            answer = 6;
            break;
        default:
            answer = 7;
            break;
    }
    return answer;
}

private boolean getAnswer3(){
    boolean answer = false; //f is a default value corresponging to error

    if(q3a.isChecked() && !q3b.isChecked()){
        System.out.println("Q3A selected");
        answer = true;
    } else if(!q3a.isChecked() && q3b.isChecked()){
        System.out.println("Q3B Selected");
        answer = false;
    } else{
        System.out.println("Please check only 3 box.");
    }
    return answer;
}
```

```
}
```

SUBJECT OPTION PAGE 2 ACTIVITY

```
public class SubjectOptionPageActivity2 extends AppCompatActivity {

    private Button menu, logOut, submit;
    private String Q1;
    private int Q2;
    private String Q3;
    private String Q4, Q5, Q6;
    private EditText Q4EditText,Q5EditText,Q6EditText;

    SharedPreferences sharedPreferences;
    int userId = 0;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.subject_option_page_activity_2);

        sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);
        userId = Integer.parseInt(sharedPreferences.getString("UserId",null));

        menu = findViewById(R.id.SubjectMenuButton2);
        logOut = findViewById(R.id.SubjectLogOutButton2);
        submit = findViewById(R.id.SubjectSubmitButton);

        Q4EditText = findViewById(R.id.SubjectQuestion4EditText);
        Q5EditText = findViewById(R.id.SubjectQuestion5EditText);
        Q6EditText = findViewById(R.id.SubjectOtherCommentsEditText);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openMenu();
            }
        });

        logOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openLogInPage();
            }
        });

        submit.setOnClickListener(new View.OnClickListener() { //Get the required data and store in database from here.
            @Override
            public void onClick(View view) {

                //Get data from page 1:
                Intent intentExtras = getIntent();
                Bundle bundle = intentExtras.getExtras();

                if(bundle != null){
                    Q1 = bundle.getString("q1", "Error in retrieving answer 1 in Subject Page 1 ");
                    Q2 = bundle.getInt("q2",0);
                    if(String.valueOf(bundle.getBoolean("q3")).equals("true")){
                        Q3 = "true";
                    } else if (String.valueOf(bundle.getBoolean("q3")).equals("false")){
                        Q3 = "false";
                    } else{
                        Q3 = "error";
                        System.out.println("Error in getting Q3 from CanteenOptionPage1");
                    }
                }

                //Get data from the current page:
                Q4 = Q4EditText.getText().toString();
                Q5 = Q5EditText.getText().toString();
                Q6 = Q6EditText.getText().toString();

                addEntryToTblOption(userId, String.valueOf(Q1),Integer.toString(Q2),Q3,Q4,Q5,Q6);

                //System.out.println(Q1 +"    "+ Q2+"    "+ Q3+ "    "+ Q4 + "    "+ Q5+ "    "+ Q6);

                //Update userHistory after submitting
                //updateUserHistory();
            }
        });
    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }

    //Here, we must CREATE a new record with the UserName,
    private void addEntryToTblOption(int userId, String Q1, String Q2, String Q3, String Q4, String Q5, String Q6){
        String categoryType = "Subject";
        //System.out.print("UserId==="+userId+" Q1=="+Q1+" Q2="+Q2+" Q3="+Q3+" Q4="+Q4+" Q5="+Q5+" Q6="+Q6);
        String tag = "saveFeedBack";
        String accessToken = sharedPreferences.getString("AccessToken",null);
        Feedback feedback = new Feedback();
        feedback.setAccessToken(accessToken);
        feedback.setUserId(userId);
        feedback.setQuestion1(Q1);
        feedback.setQuestion2(Q2);
        feedback.setQuestion3(Q3);
        feedback.setQuestion4(Q4);
        feedback.setQuestion5(Q5);
        feedback.setQuestion6(Q6);
        feedback.setCategoryType(categoryType);

        Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
        StudentDelegateService service = retrofit.create(StudentDelegateService.class);

        service.saveFeedback(tag,feedback).enqueue(new Callback<Response>() {
            @Override
            public void onResponse(Call<Response> call, retrofit2.Response<Response> response) {
                if(response.code() == 200){
                    Toast.makeText(getApplicationContext(),"Feedback for subjects saved
  successfully",Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(getApplicationContext(),"Something went wrong.Please try
  again",Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(Call<Response> call, Throwable t) {
                Log.d("Error==","Error=="+t.getMessage());
                Toast.makeText(getApplicationContext(),"Something went wrong.Please try again",Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

TEACHER OPTION PAGE 1 ACTIVITY

```
public class TeacherOptionPageActivity1 extends AppCompatActivity {

    private Button menu, logOut, nextPage;
    private String Q1;
    private char Q2;
    private boolean Q3;
    private Spinner subjectSpinnerTeachersPage;
    private CheckBox q1a, q1b, q1c, q1d;
    private CheckBox q2a, q2b, q2c, q2d;
    private CheckBox q3a, q3b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.teachers_option_page_activity_1);

        menu = findViewById(R.id.TeachersMenuButton1);
        logOut = findViewById(R.id.TeachersLogOutButton1);
        nextPage = findViewById(R.id.TeachersNextPageButton);
        subjectSpinnerTeachersPage = findViewById(R.id.SubjectSpinnerTeachersPage);

        q2a = findViewById(R.id.TeachersAnswer2A);
        q2b = findViewById(R.id.TeachersAnswer2B);
        q2c = findViewById(R.id.TeachersAnswer2C);
        q2d = findViewById(R.id.TeachersAnswer2D);



        q3a = findViewById(R.id.TeachersAnswer3A);
        q3b = findViewById(R.id.TeachersAnswer3B);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openMenu();
            }
        });

        logOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                openLogInPage();
            }
        });

        nextPage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Q1 = getAnswer1();
                Q2 = getAnswer2();
                Q3 = getAnswer3();

                Intent openNextTeachersPage = new Intent(TeacherOptionPageActivity1.this, TeacherOptionPageActivity2.class);
                Bundle TeachersOptionPage1DataBundle = new Bundle();
                TeachersOptionPage1DataBundle.putString("q1", Q1);
                TeachersOptionPage1DataBundle.putChar("q2", Q2);
                TeachersOptionPage1DataBundle.putBoolean("q3", Q3);

                openNextTeachersPage.putExtras(TeachersOptionPage1DataBundle);
                startActivity(openNextTeachersPage);

            }
        });

        String[] items = new String[]{"Mathematics", "Biology", "Physics", "Chemistry", "Geography", "History", "Physical Education
(P.E)", "Business Studies", "Economics", "Art"
                , "DT", "ITGS", "Computer Science", "Theory of Knowledge", "CAS", "Chinese", "Spanish", "German", "French"
                , "Thai", "Other Language", "Drama", "Music", "ESS"};
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, items);
        subjectSpinnerTeachersPage.setAdapter(adapter);
    }

    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }
    private String getAnswer1(){
        String answer = subjectSpinnerTeachersPage.getSelectedItem().toString();
        return answer;
    }

    private char getAnswer2(){
        char answer = 'f'; //f is a default value corresponging to error

        if(q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
            System.out.println("Q2A selected");
            answer = 'a';
        } else if(!q2a.isChecked() && q2b.isChecked()&& !q2c.isChecked()&& !q2d.isChecked()){
            System.out.println("Q2B Selected");
            answer = 'b';
        } else if(!q2a.isChecked() && !q2b.isChecked()&& q2c.isChecked()&& !q2d.isChecked()){
            System.out.println("Q2C Selected");
            answer = 'c';
        }
        else if(!q2a.isChecked() && !q2b.isChecked()&& !q2c.isChecked()&& q2d.isChecked()){
            System.out.println("Q2D Selected");
            answer = 'd';
        } else{
            System.out.println("Please check only 2 box.");
        }
        return answer;
    }

    private boolean getAnswer3(){
        boolean answer = false; //f is a default value corresponging to error

        if(q3a.isChecked() && !q3b.isChecked()){
            System.out.println("Q3A selected");
            answer = true;
        } else if(!q3a.isChecked() && q3b.isChecked()){
            System.out.println("Q3B Selected");
            answer = false;
        } else{
            System.out.println("Please check only 3 box.");
        }
        return answer;
    }

}
```

TEACHER OPTION PAGE 2 ACTIVITY

```
public class TeacherOptionPageActivity2 extends AppCompatActivity {

    private Button menu, logOut, submit;
    private String Q1;
    private char Q2;
    private String Q3;
    private String Q4, Q5, Q6;
    private EditText Q4EditText,Q5EditText,Q6EditText;

    SharedPreferences sharedPreferences;
    int userId = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.teachers_option_page_activity_2);

        menu = findViewById(R.id.TeachersMenuButton2);
        logOut = findViewById(R.id.TeachersLogOutButton2);
        submit = findViewById(R.id.TeachersSubmitButton);

        sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);
        userId = Integer.parseInt(sharedPreferences.getString("UserId",null));

        Q4EditText = findViewById(R.id.TeachersQuestion4EditText);
        Q5EditText = findViewById(R.id.TeachersQuestion5EditText);
        Q6EditText = findViewById(R.id.TeachersOtherCommentsEditText);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```
            openMenu();
        }
    });

    logOut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            openLogInPage();
        }
    });

    submit.setOnClickListener(new View.OnClickListener() { //Get the required data and store in database from here.
        @Override
        public void onClick(View view) {

            //Get data from page 1:
            Intent intentExtras = getIntent();
            Bundle bundle = intentExtras.getExtras();

            if(bundle != null){
                Q1 = bundle.getString("q1", "Error in retrieving answer 1 from Teacher Page 1");
                Q2 = bundle.getChar("q2",'f');
                if(String.valueOf(bundle.getBoolean("q3")).equals("true")){
                    Q3 = "true";
                } else if (String.valueOf(bundle.getBoolean("q3")).equals("false")){
                    Q3 = "false";
                } else{
                    Q3 = "error";
                    System.out.println("Error in getting Q3 from CanteenOptionPage1");
                }
            }

            //Get data from the current page:
            Q4 = Q4EditText.getText().toString();
            Q5 = Q5EditText.getText().toString();
            Q6 = Q6EditText.getText().toString();

            addEntryToTblOption(userId, String.valueOf(Q1),String.valueOf(Q2),String.valueOf(Q3),Q4,Q5,Q6);



            //Update userHistory after submitting
            //updateUserHistory();
        }
    });
}
```

```
    private void openMenu(){
        Intent openMenuPage = new Intent(this, MenuActivity.class);
        startActivity(openMenuPage);
    }

    private void openLogInPage(){
        Intent openLogInPage = new Intent(this, LogInActivity.class);
        startActivity(openLogInPage);
    }

    //Here, we must CREATE a new record with the UserName,
    private void addEntryToTblOption(int userId, String Q1, String Q2, String Q3, String Q4, String Q5, String Q6){
        String categoryType = "Teacher";
        //System.out.print("UserId==="+userId+" Q1=="+Q1+" Q2="+Q2+" Q3="+Q3+" Q4="+Q4+" Q5="+Q5+" Q6="+Q6);
        String tag = "saveFeedBack";
        String accessToken = sharedPreferences.getString("AccessToken",null);
        Feedback feedback = new Feedback();
        feedback.setAccessToken(accessToken);
        feedback.setUserId(userId);
        feedback.setQuestion1(Q1);
        feedback.setQuestion2(Q2);
        feedback.setQuestion3(Q3);
        feedback.setQuestion4(Q4);
        feedback.setQuestion5(Q5);
        feedback.setQuestion6(Q6);
        feedback.setCategoryType(categoryType);

        Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
        StudentDelegateService service = retrofit.create(StudentDelegateService.class);

        service.saveFeedback(tag,feedback).enqueue(new Callback<Response>() {
            @Override
            public void onResponse(Call<Response> call, retrofit2.Response<Response> response) {
                if(response.code() == 200){
                    Toast.makeText(getApplicationContext(),"Feedback for teachers saved
  successfully",Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(getApplicationContext(),"Something went wrong.Please try
  again",Toast.LENGTH_LONG).show();
                }
            }

            @Override
            public void onFailure(Call<Response> call, Throwable t) {
```

```
            Log.d("Error==","Error=="+t.getMessage());
            Toast.makeText(getApplicationContext(),"Something went wrong.Please try again",Toast.LENGTH_LONG).show();
        }
    });

  }

}
```

VIEW HISTORY ACITIVTY

```
public class ViewHistoryActivity extends AppCompatActivity {

  private SharedPreferences sharedPreferences;
  String userId = null;
  String token;
  RecyclerView historyRecyclerView;
  HistoryAdapter adapter;
  FeedbackHistory[] feedbacks;
  ArrayList<FeedbackHistory> feedbackHistoryArrayList;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.view_history_activity);

    sharedPreferences = getSharedPreferences("StudentDelegate", MODE_PRIVATE);
    historyRecyclerView = findViewById(R.id.feedBackHistoryRecyclerView);

    RecyclerView.LayoutManager manager = new LinearLayoutManager(ViewHistoryActivity.this);
    historyRecyclerView.setLayoutManager(manager);

    userId = sharedPreferences.getString("UserId", null);
    token = sharedPreferences.getString("AccessToken",null);
    getUserHistory(userId,token);

  }

  public void getUserHistory(String userId,String token){
    //Retrofit + service connect with API on webhost
    Retrofit retrofit = RetrofitClient.getClient(Constants.BASE_URL);
    StudentDelegateService service = retrofit.create(StudentDelegateService.class);
    String tag = "gethistory"; //Tag to ensure gethistory functions are called in the API

    UserDetails userDetails = new UserDetails();
    userDetails.setUserId(Integer.parseInt(userId));
    userDetails.setAccessToken(token); //AccessToken is required to ensure user confidentiality


    service.getHistory(tag,userDetails).enqueue(new Callback<FeedbackHistory[]>() {

        //Response from the API comes in the format of an array of feedback Class objects
        @Override
        public void onResponse(@NonNull Call<FeedbackHistory[]> call, @NonNull Response<FeedbackHistory[]> response) {

            feedbacks = response.body(); //Gets response and stores in array of objects

            if(response.code() == 200 && feedbacks.length >0){
                ArrayList<FeedbackHistory> feedbackHistoryArrayList = new ArrayList<>();

                for(FeedbackHistory history : feedbacks){ //Adds objects to the arrayList
                    feedbackHistoryArrayList.add(history);
                }

                adapter = new HistoryAdapter(ViewHistoryActivity.this, feedbackHistoryArrayList); //Passes query details to
HistoryAdapter class
                historyRecyclerView.setAdapter(adapter); //Displays the history adapter
            }
        }

        @Override
        public void onFailure(@NonNull Call<FeedbackHistory[]> call, @NonNull Throwable t) {
            Log.d("Failure","Failure");
        }
    });


  }
}
```

PHP APIS

INDEX.PHP

```
<?php
//if($_SERVER['REQUEST_METHOD'] === 'POST'){
        if (isset($_GET['tag']) && $_GET['tag'] != '') {
        // Get tag
                $tag = $_GET['tag'];
                require_once(__DIR__.'/DB_Function.php');
                $db = new DB_Function();
                $response = array("tag" => $tag, "success" => 0, "error" => 0);
        //Register START
                if ($tag == 'register') {
                        $json = json_decode(file_get_contents("php://input"));
                        $user=$db->storeUser($json);
                        //var_dump("Usersresult===".$user);
                        if ($user) {
                                $response["status"] = "Success";
                                $response["statusCode"] = "200";
                                echo json_encode($response);
                        }else{
                                $response["status"] = "Fail";
                                $response["statusCode"] = "401";
                                $response["message"] = "User Name Already Exist";
                                echo json_encode($response);
                        }
                }//Register END
        //LOGIN START
                if($tag == 'login'){
                        $json = json_decode(file_get_contents("php://input"));
                        //print_r($json);
                        $user = $db->getUser($json);
                                 if($user){
                                        echo json_encode($user[0]);
                                }else {
                                        $response["status"] = "Fail";
                                        $response["statusCode"] = "401";
                                        $response["message"] = "Incorrect username or Password";
                                        echo json_encode($response);
                                } //end of elseif
                }//LOGIN END
                if($tag=='gethistory'){
                        $json = json_decode(file_get_contents("php://input"));
                        $data = $db->getHistory($json);
                        if($data){
                                echo json_encode($data);
                        }else{
                                $response["statusCode"] = "401";
                                $response["status"] = "Something went wrong, please try again";
                                echo json_encode($response);
                        }
                }
                if($tag=='saveFeedBack'){
                   $json = json_decode(file_get_contents("php://input"));
                        $data = $db->saveFeedBack($json);
```

```
                    if($data){
                        $response["status"] = "Feedback Saved Successfully";
                        $response["statusCode"] = "200";
                        echo json_encode($response);
                    }else{
                        $response["statusCode"] = "401";
                        $response["status"] = "Something went wrong, please try again";
                        echo json_encode($response);
                    }
            }
    }
//} ?>
```

DBFUNCTION.PHP

```
<?php
class DB_Function {
 function __construct() {
      //$response = array("tag" => $tag, "success" => 0, "error" => 0);
   }
   // destructor
   function __destruct() {
   }
            public function storeUser($obj) {
                        require_once('config.php');
                        if($this->openConnection()){
                                    $username = $obj->username;
                                    $password = $obj->password;
                                    $admin = $obj->admin;
                                    $yearGroup = $obj->yearGroup;
                                    $dateOfCreation = date("Y-m-d H:i:s"); // now();
                                    //$userId = rand(10,100);
                                    $sql = "SELECT * FROM Accounts WHERE UserName='$username'";
                                    $check = $this->select($sql);
                                    $countArr = count($check);
                                    if(isset($check) && $countArr > 0){
                                                //return false;
                                    }else{
                                                $query = "INSERT into
Accounts(Username,Password,Admin,YearGroup,DateOfCreation)
values('$username','$password',$admin,'$yearGroup','$dateOfCreation')";
                                                $result = $this->insert($query);
                                                //var_dump("Insert Reult===".$result);
                                                            if(isset($result)){
                                                $data = "SELECT * FROM Accounts WHERE UserName='$username'";
                                                                        $dataResult = $this->select($data);
                                                                        return $dataResult;
                                                            }else{
                                                                        return false;
                                                            }
                                    }
                        }
            }

/*** Getting all users */
   public function getAllUsers() {
      require_once('config.php');
      $sql = "select * FROM Accounts";
      $result = mysqli_query($con,$sql);
      return $result;
   }
 public function getUser($json) {
            require_once('config.php');
            if($this->openConnection()){
                        $username = $json->username;
                        $password = $json->password;
                        $randomNumber = rand(100000, 999999);
                        //echo("RandomNumber===".$randomNumber);
            $updateSql = "UPDATE Accounts set access_token = '$randomNumber' where
UserName='$username' and Password='$password' ";
```

```
				$result = $this->edit($updateSql);
				//echo("UpdateSql====".$updateSql);
				$sql = "SELECT * FROM Accounts WHERE UserName='$username' and
Password='$password'";
				$result = $this->select($sql);
				//echo("LoginSql====".$sql);
				//var_dump($result);
				return $result;
		}
 }//End of function
		public function getHistory($obj){
				require_once('config.php');
				if($this->openConnection()){
						$usersId = $obj->userId;
						$accessToken = $obj->accessToken;
						$verifySql = "SELECT * from Accounts where access_token = '$accessToken'";
						$verifyResult = $this->select($verifySql);
						$countRes = count($verifyResult);
						if($countRes > 0){
						      $feedBackDetailSql = "SELECT * from TeacherOptionTable where UserId =
$usersId";
								$result =  $this->select($feedBackDetailSql);
								if($result){
										return $result;
								}
						}
				}
		}
		public function saveFeedBack($obj){ //Start of Save feedback function
				require_once('config.php');
				if($this->openConnection()){
				   $accessToken = $obj->accessToken;
				$categoryType = $obj->categoryType;
				$question1 = $obj->question1;
				$question2 = $obj->question2;
				$question3 = $obj->question3;
				$question4 = $obj->question4;
				$question5 = $obj->question5;
				$question6 = $obj->question6;
				$surveyId = $obj->surveyId;
				$userId = $obj->userId;
				$createdDate = date("Y-m-d H:i:s");

				$query = "SELECT * from Accounts where access_token = '$accessToken'";
				$verifyResult = $this->select($query);
				$verifyReultCount = count($verifyResult);
				if($verifyResult && $verifyReultCount > 0){
						$sql = "insert into TeacherOptionTable
(CategoryType,UserId,Question1,Question2,Question3,Question4,Question5,Question6,SurveyId,DateCreated)
values('$categoryType',$userId,'$question1','$question2','$question3','$question4','$question5','$question6','
$surveyId','$createdDate')";
						$result = $this->insert($sql);
						if($result){
```

```
                                        return $result;
                                }
                        }
                        }
                }
                public $connection = null;
                protected function openConnection() { //The open connection which connects API to the Database
                        $servername = "localhost";
                        $username = "id6328384_dhruvmittal";
                        $password = "dhruv";
                        $flag = false;
                        try {
                                $this->connection = new PDO (
"mysql:host=$servername;dbname=id6328384_bpsdelegate", $username, $password );
                                // set the PDO error mode to exception
                                $this->connection->setAttribute ( PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION );
                                //echo "Connected successfully";
                                $flag = true;
                        } catch ( PDOException $e ) {
                                //echo "Connection failed: " . $e->getMessage ();
                        }
                        return $flag;
                }
                protected function closeConnection() { //Closes the connection from database with API
                        //$this->connection=null;
                        if(isset($this->connection)) {
                                $this->connection->close();
                                unset($this->connection);
                        }

                }
                public function select($sql) {
                        //echo $sql;
                        $stmt = $this->connection->prepare($sql);
                        $stmt->execute ();

                        // echo a message to say the UPDATE succeeded
                        //echo $stmt->rowCount() . " records found";

                        // set the resulting array to associative
                        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);

                        return $result;
                }




                //insert statement
                public function insert($sql)
                {
                        $stmt = $this->connection->prepare($sql);
```

```
			$result = $stmt->execute();
			return $result;
		}
		//update statement
		public function edit($sql) {
			$stmt = $this->connection->prepare ( $sql );
			$result = $stmt->execute();
			return $result;
		}

		//delete statement
		public function delete($sql) {
			$stmt = $this->connection->prepare ( $sql );
			$result = $stmt->execute();
			return $result;
		}
	}
	?>
```