

Survey Taker

By: Dave Nndlall, Anand Timaul, Markenson Civil

Overview

- Website used HTML, CSS, Javascript, Jquery, PHP
- For the app we used web services to connect to the database.
- Data received by the app was in JSON format
- The database is queried using multiple PHP scripts which receive the data and send it back to the app.
- Used an asynchronous task class in order to save the questions to the database.

Platforms

- Used MySQL database
- Hosting : ByetHost
- Testing : MAMP
- Only created 10 questions - 4 MC, 4 AD, 2 SA
- Android App : Java, PHP, JSON
- Website : HTML, CSS, Javascript, PHP, and JQuery
- UML : Lucid Chart
- IDE & Editors: Android Studio, Sublime, TextWrangler
- Pattern: Bridge

Difficulties

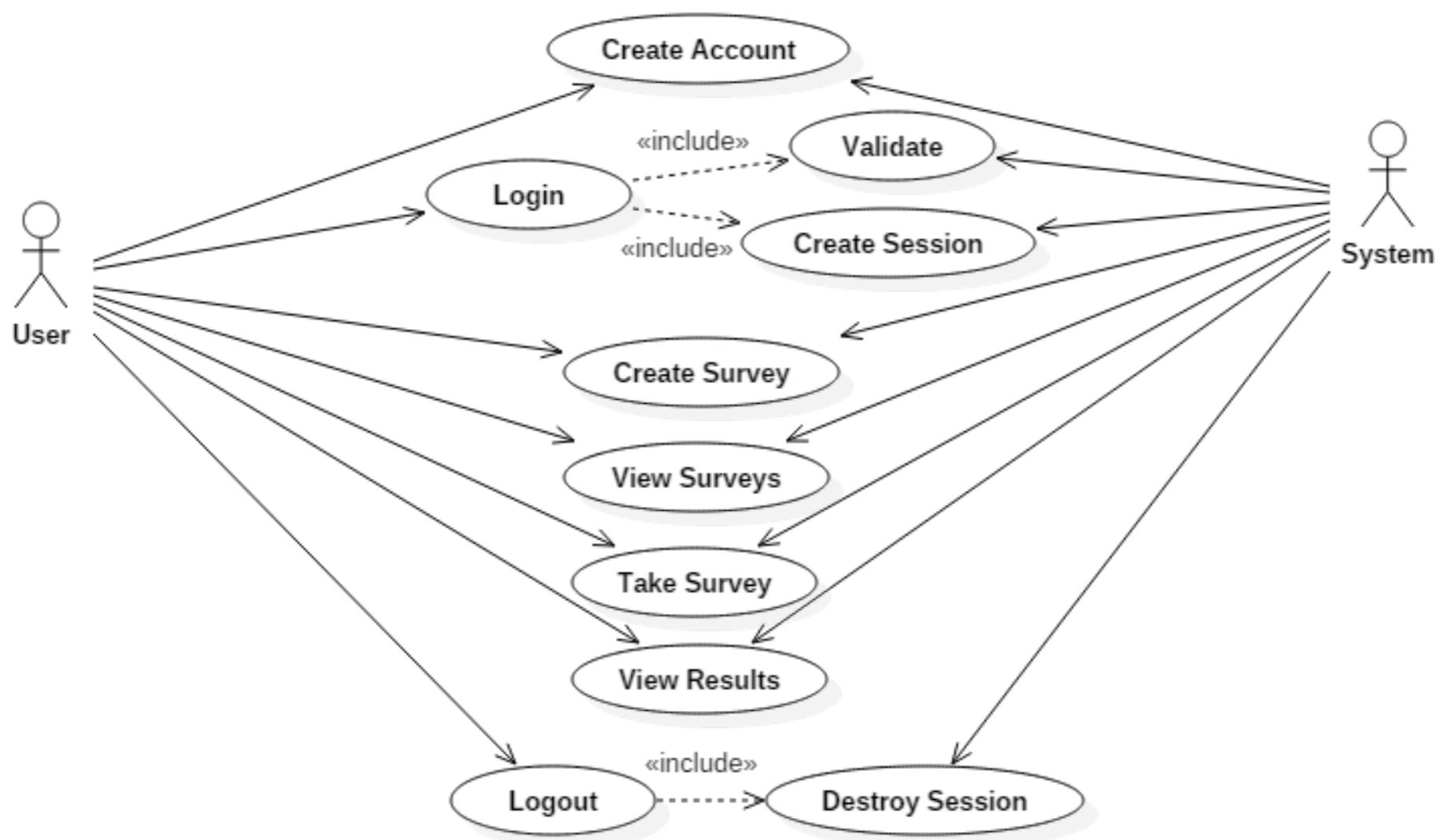
- Programming w/ new languages including PHP and MySQL
- ANDROID STUDIO !!!!!!
- Time constraints and workload
- Some UI issues are still present, freezes etc.

UML – Website Survey System

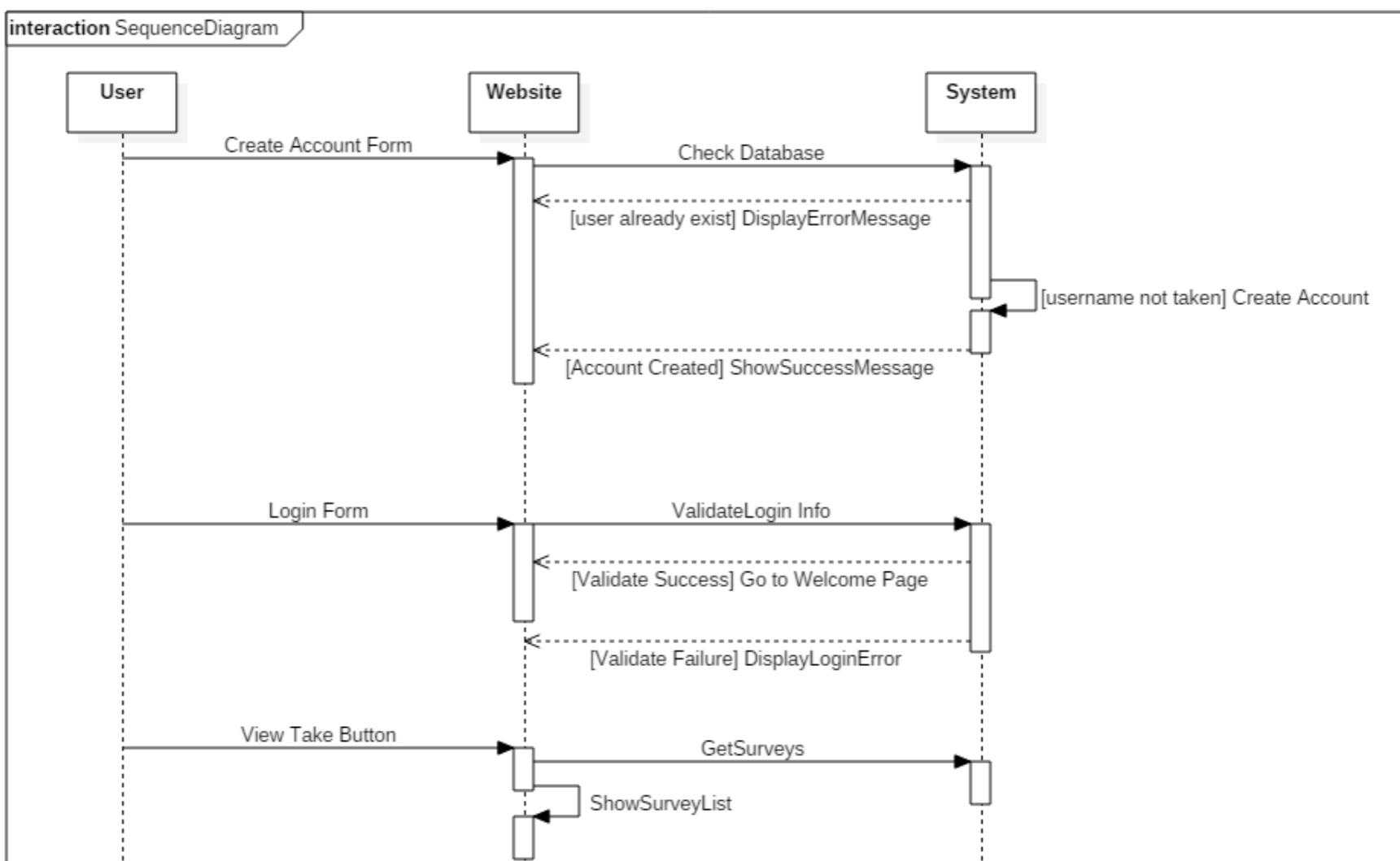
Database Structure

user_accts	survey	answers
<ul style="list-style-type: none">• username• password	<ul style="list-style-type: none">• user• sid• qnum• ques• ans1• ans2• ans3• ans4	<ul style="list-style-type: none">• user• sid• qnum• ques• answer

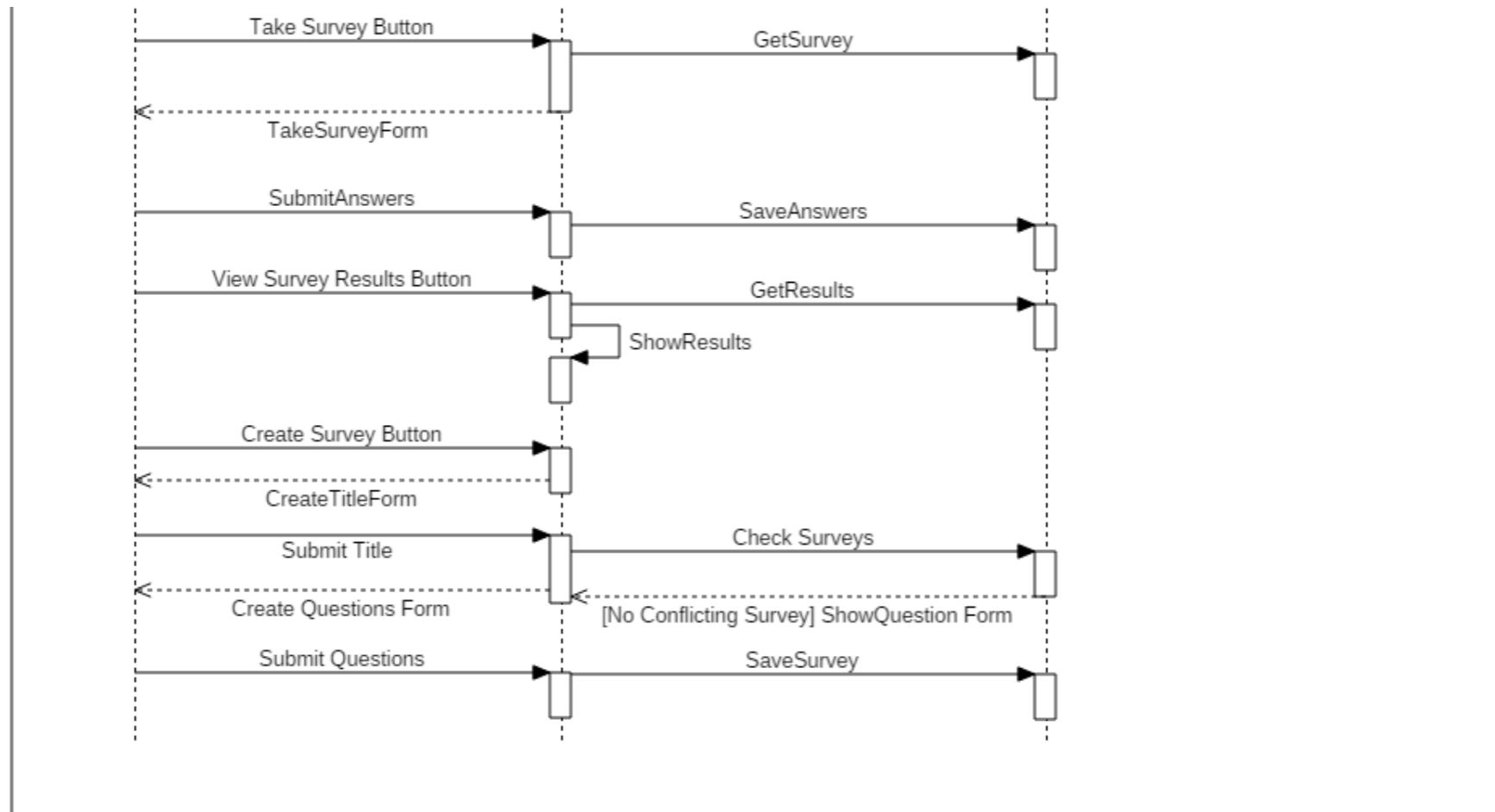
Use Case Diagram – Website Survey System



Sequence Diagram – Website Survey System

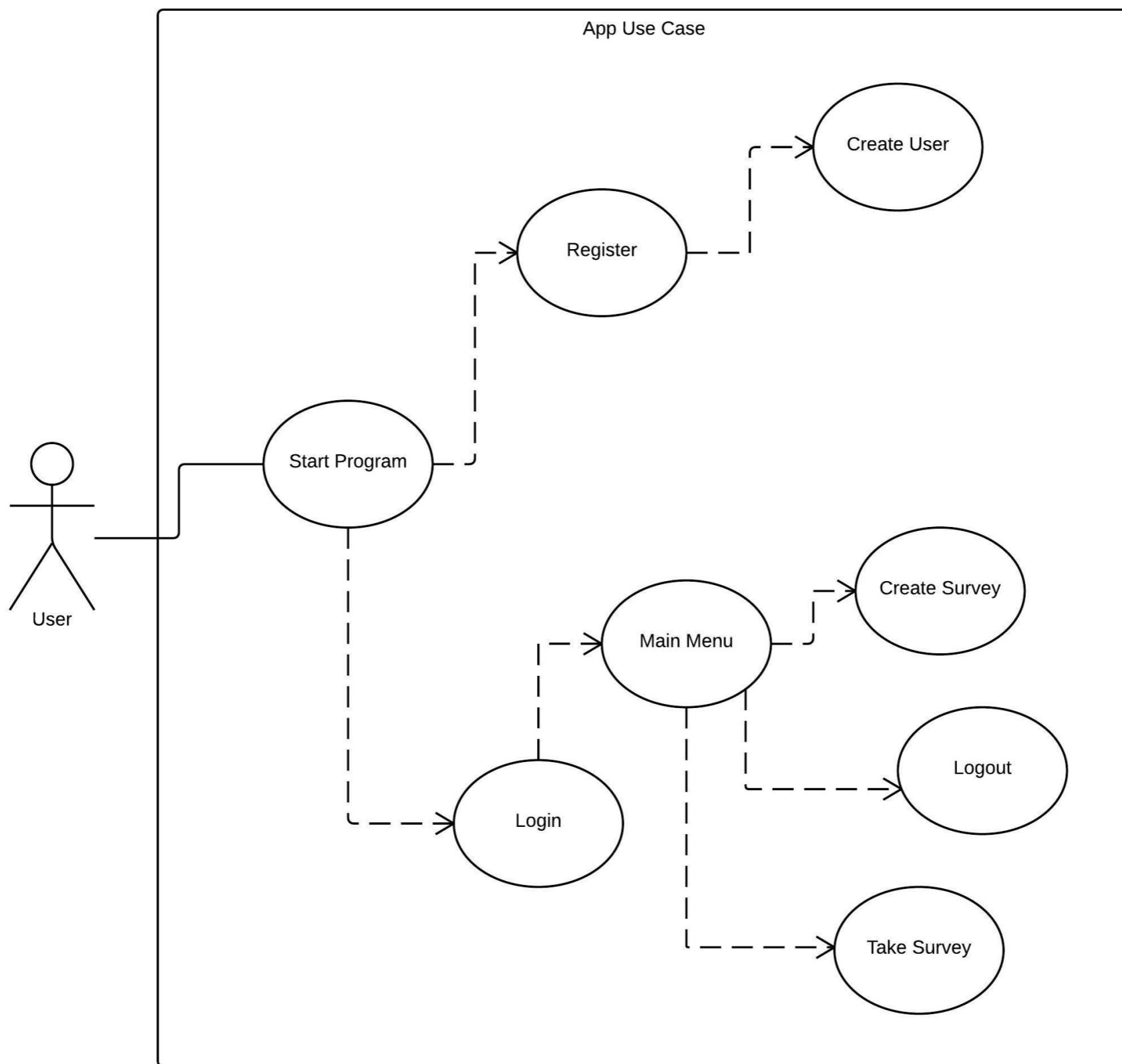


Sequence Diagram – Website Survey System

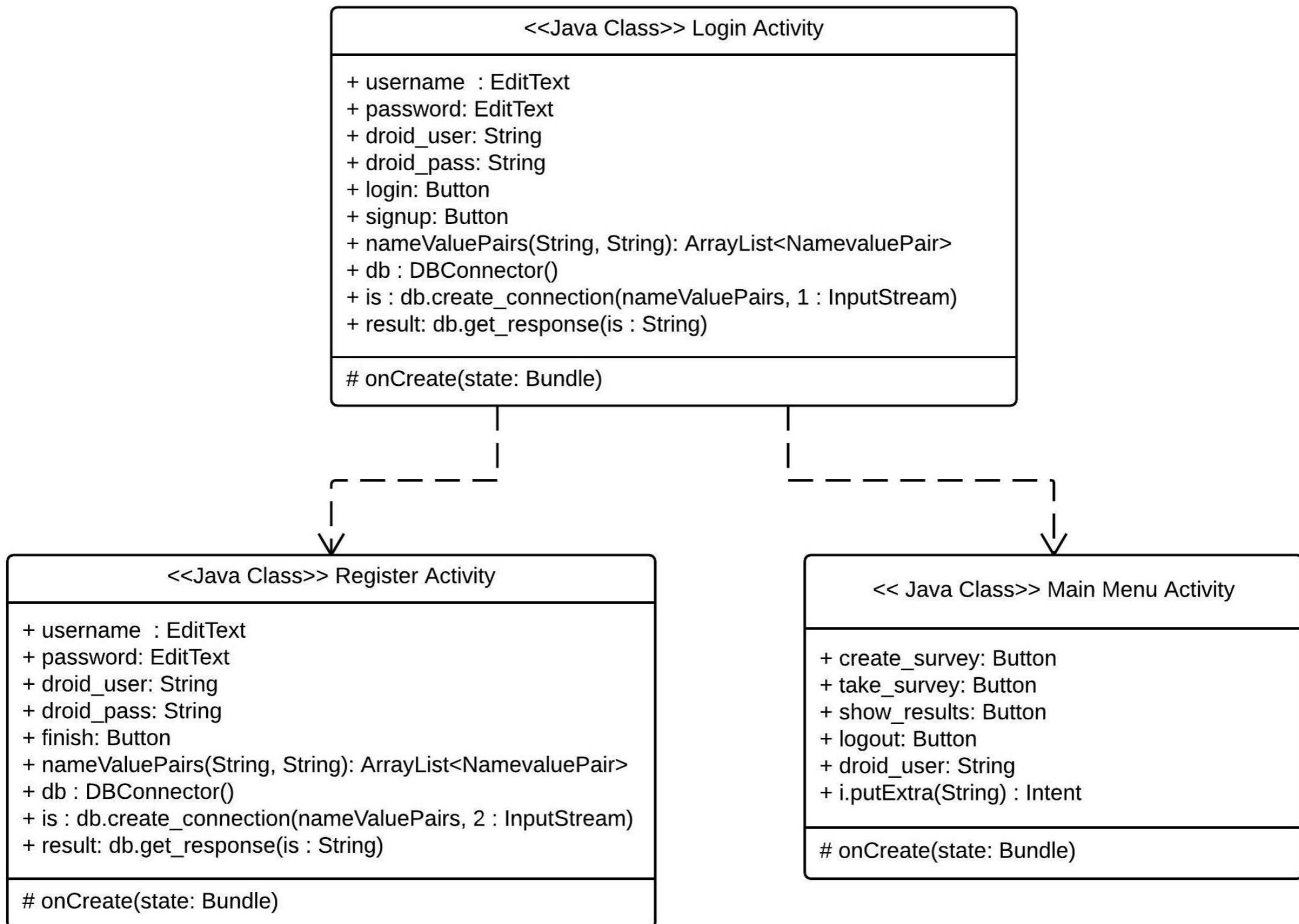


UML - Android Application

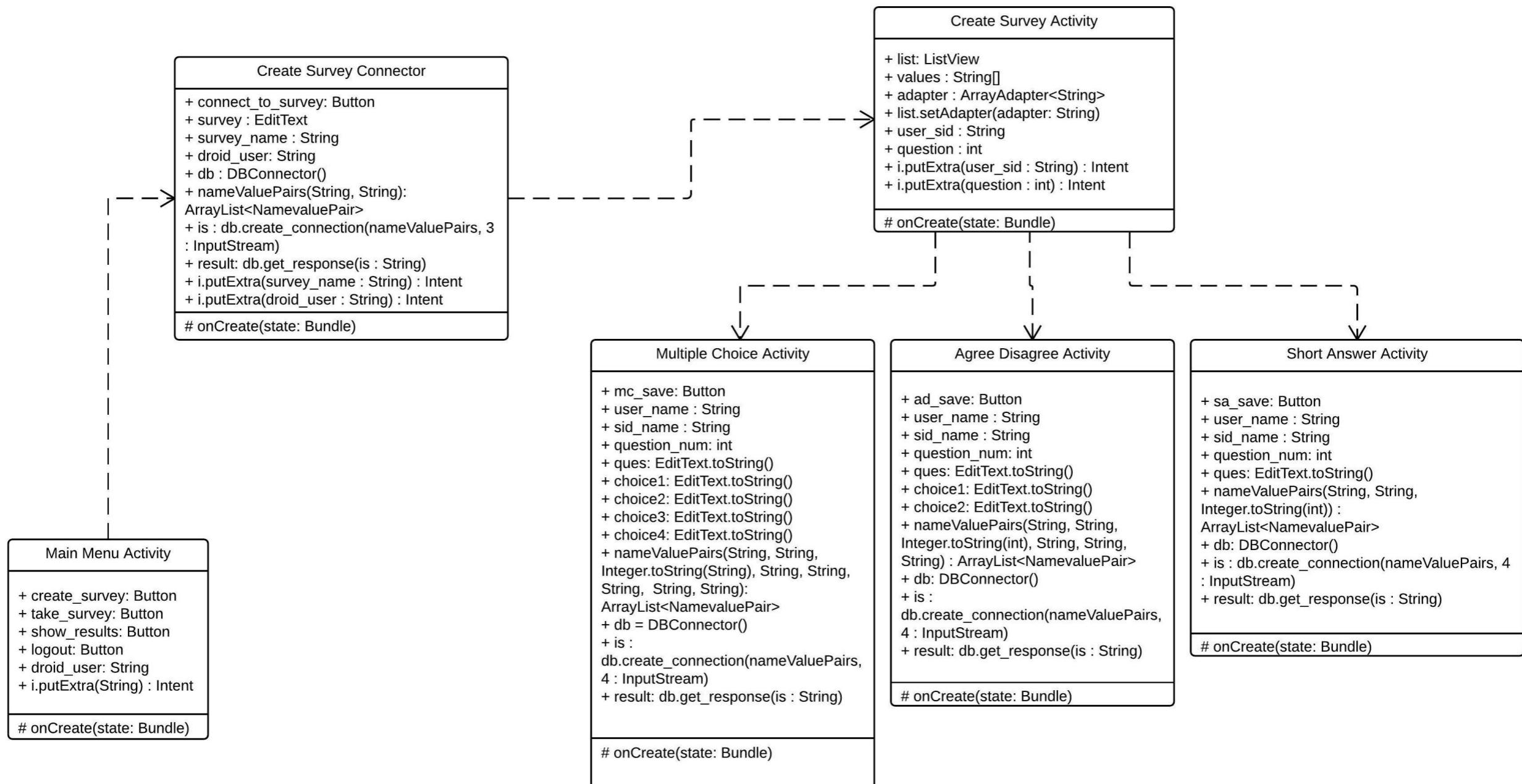
Android Application Use Case Diagram



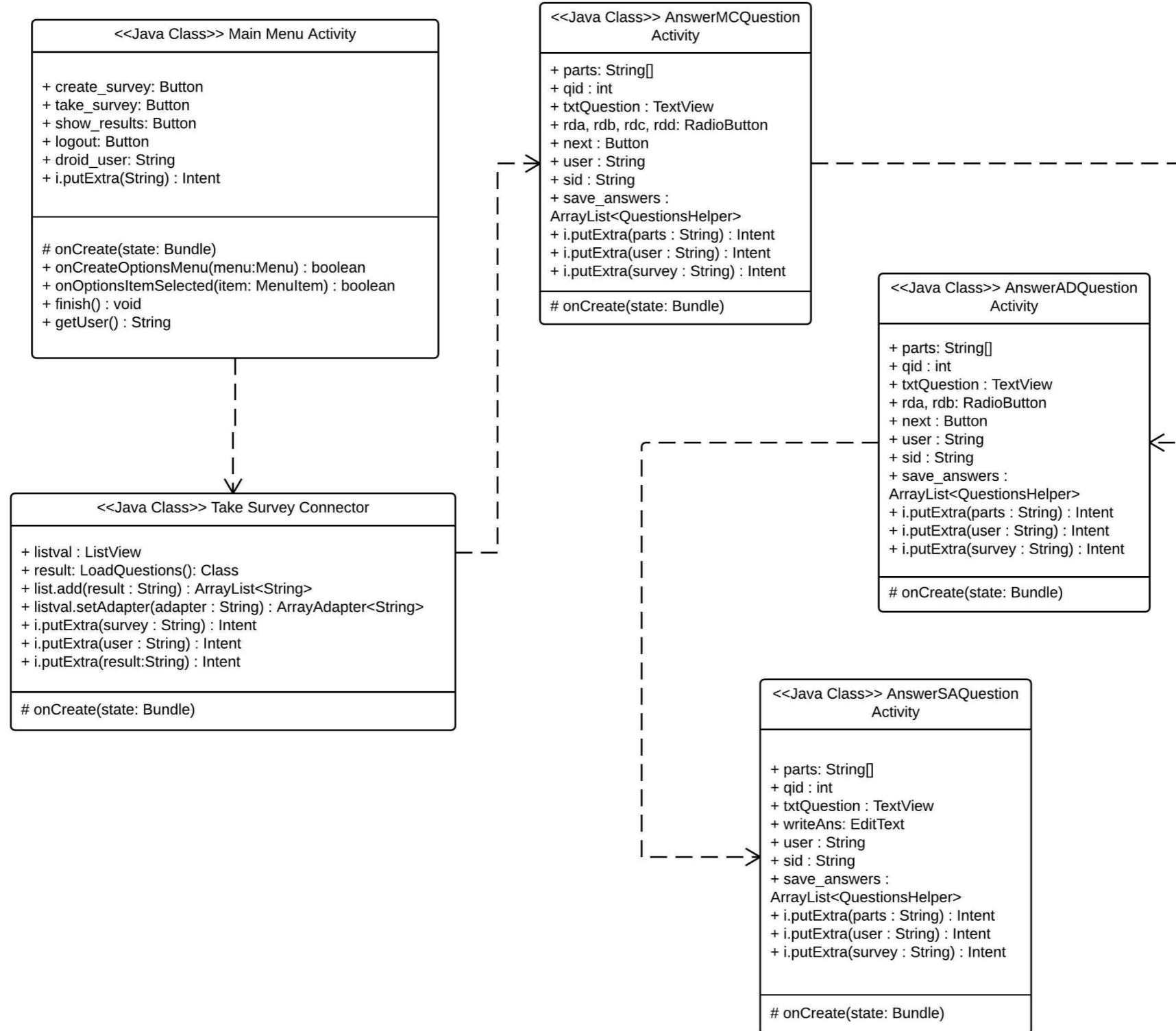
Login Process



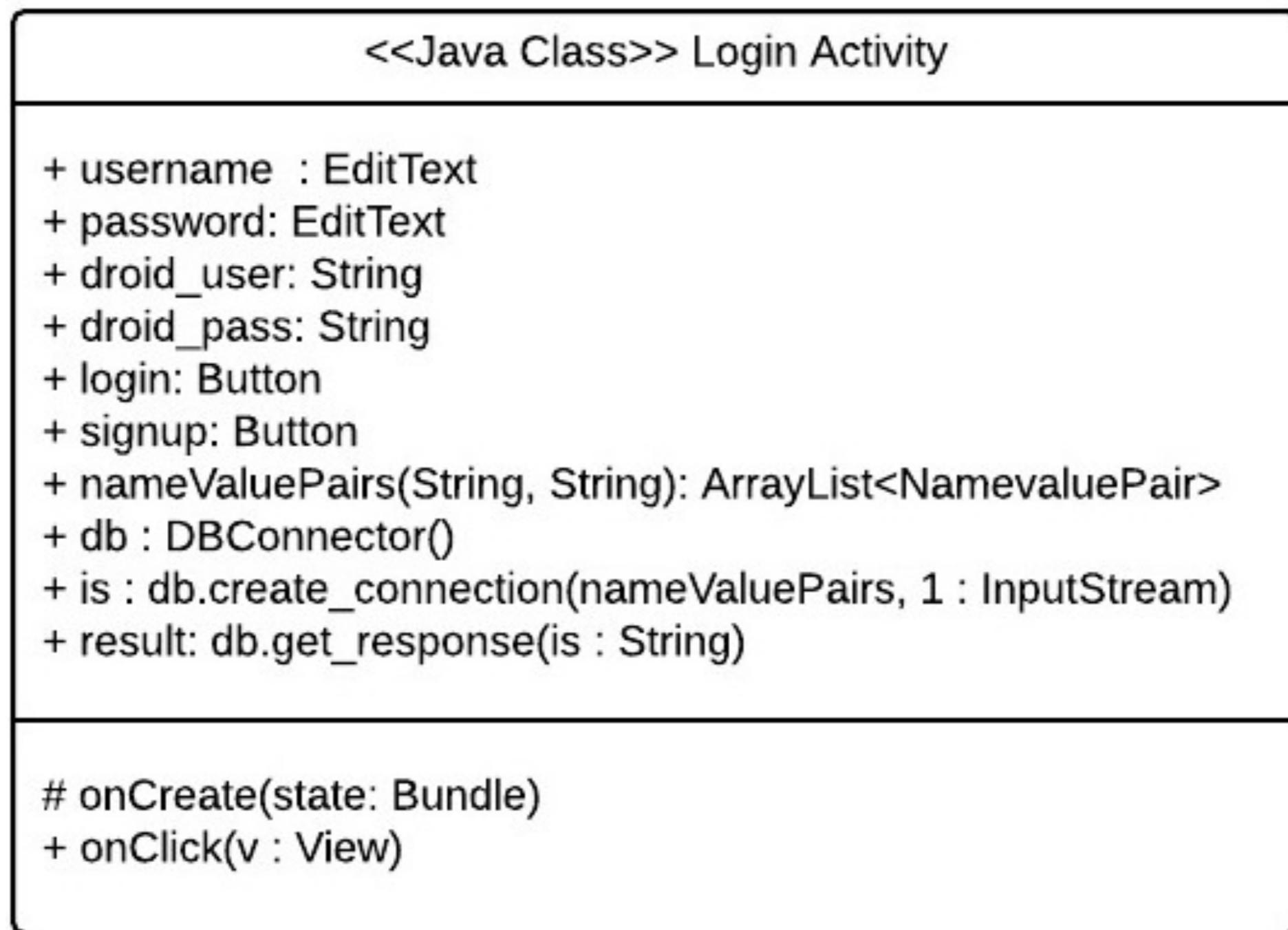
Create a Survey Process



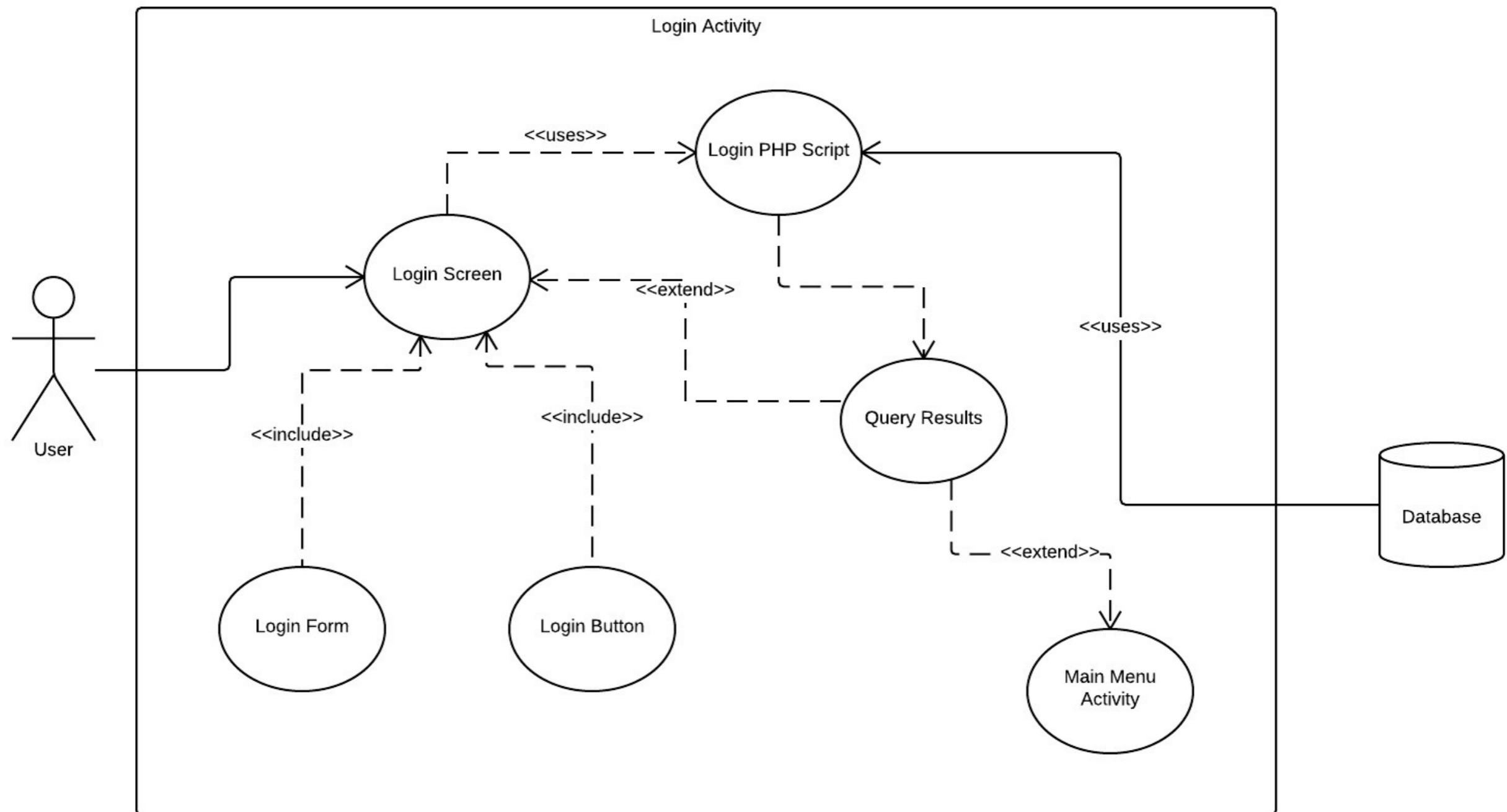
Take Survey Process



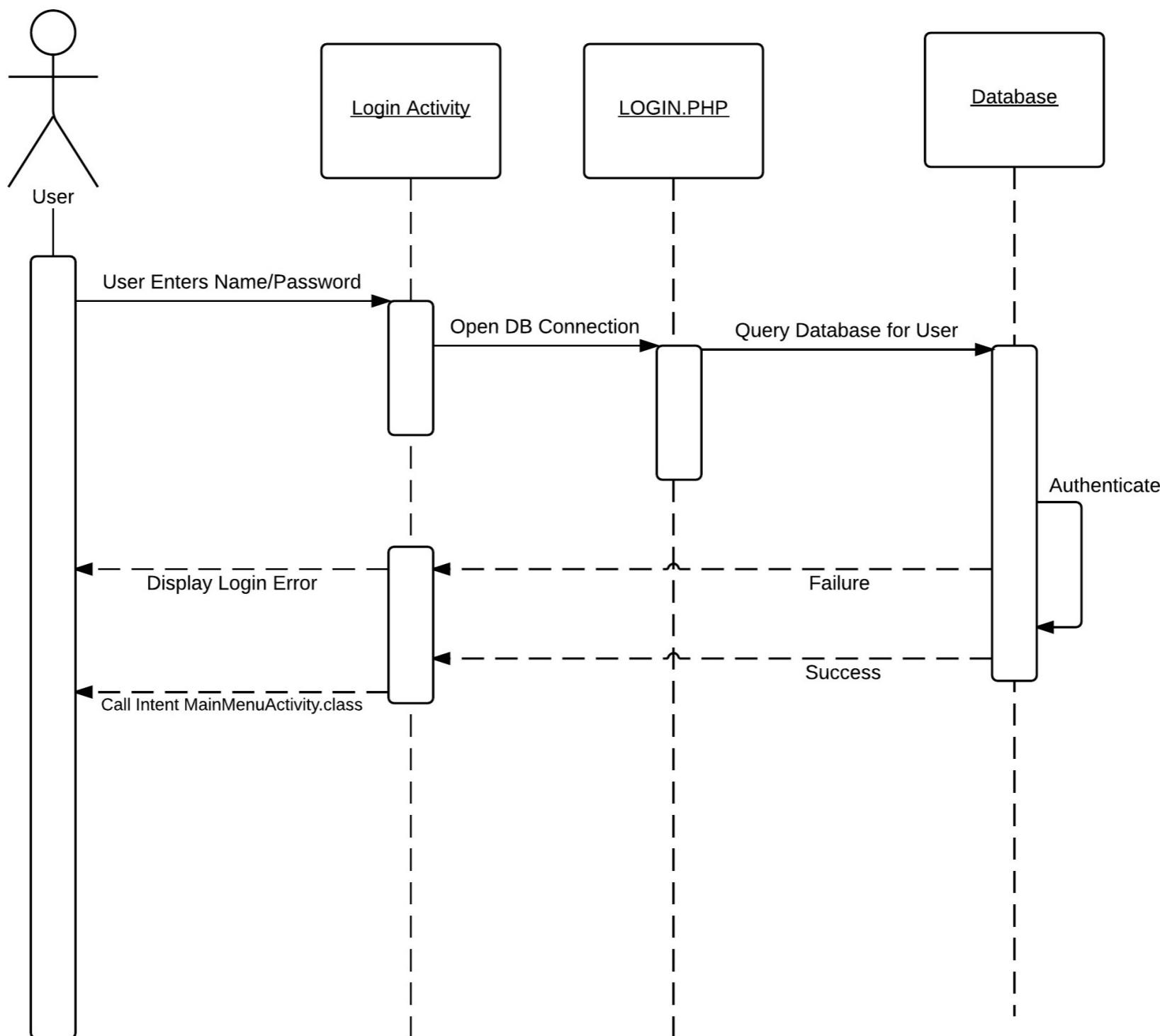
Login Activity Class Diagram



Login Activity Use Case Diagram



Login Sequence Diagram



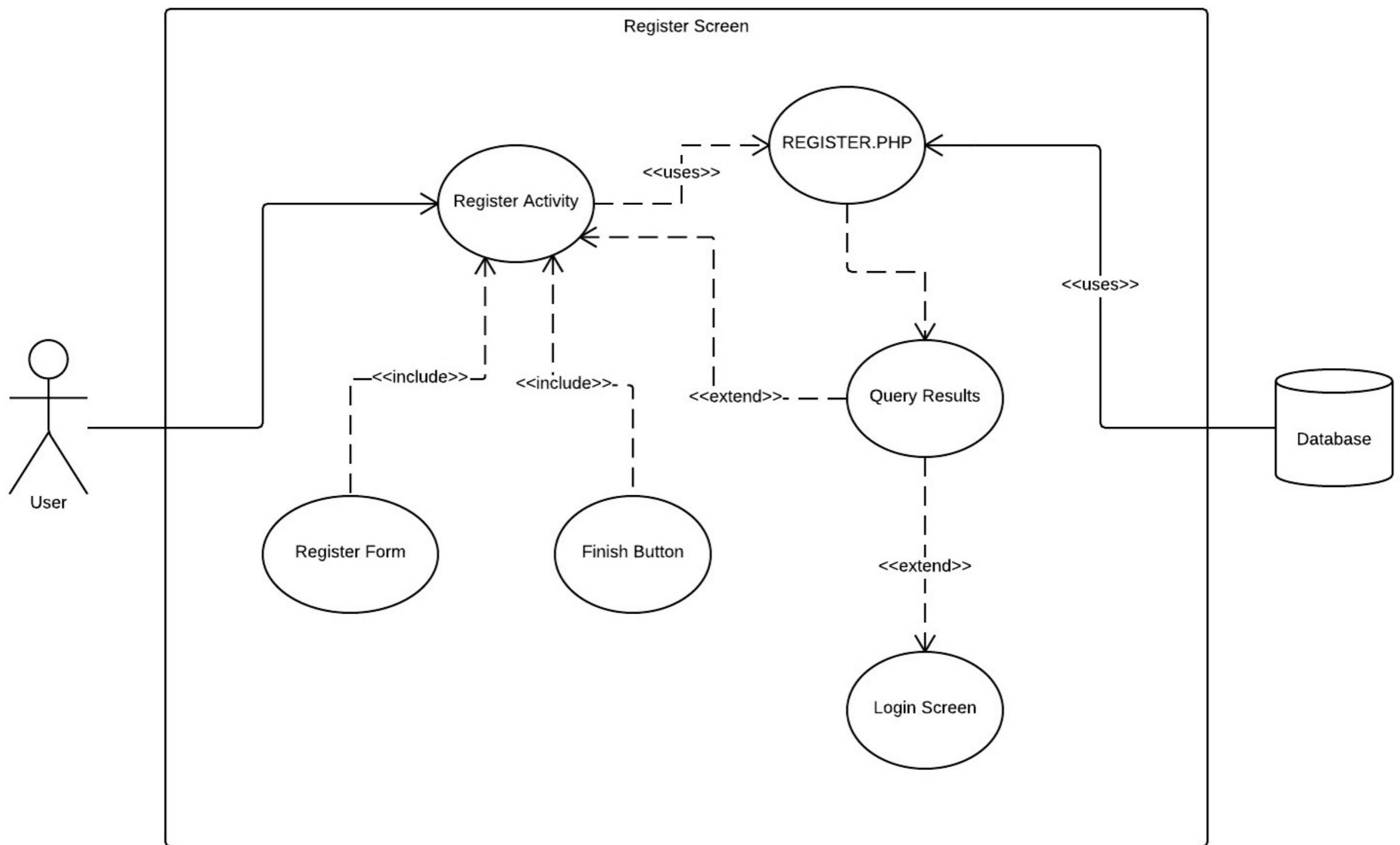
Register Activity Class Diagram

<<Java Class>> Register Activity

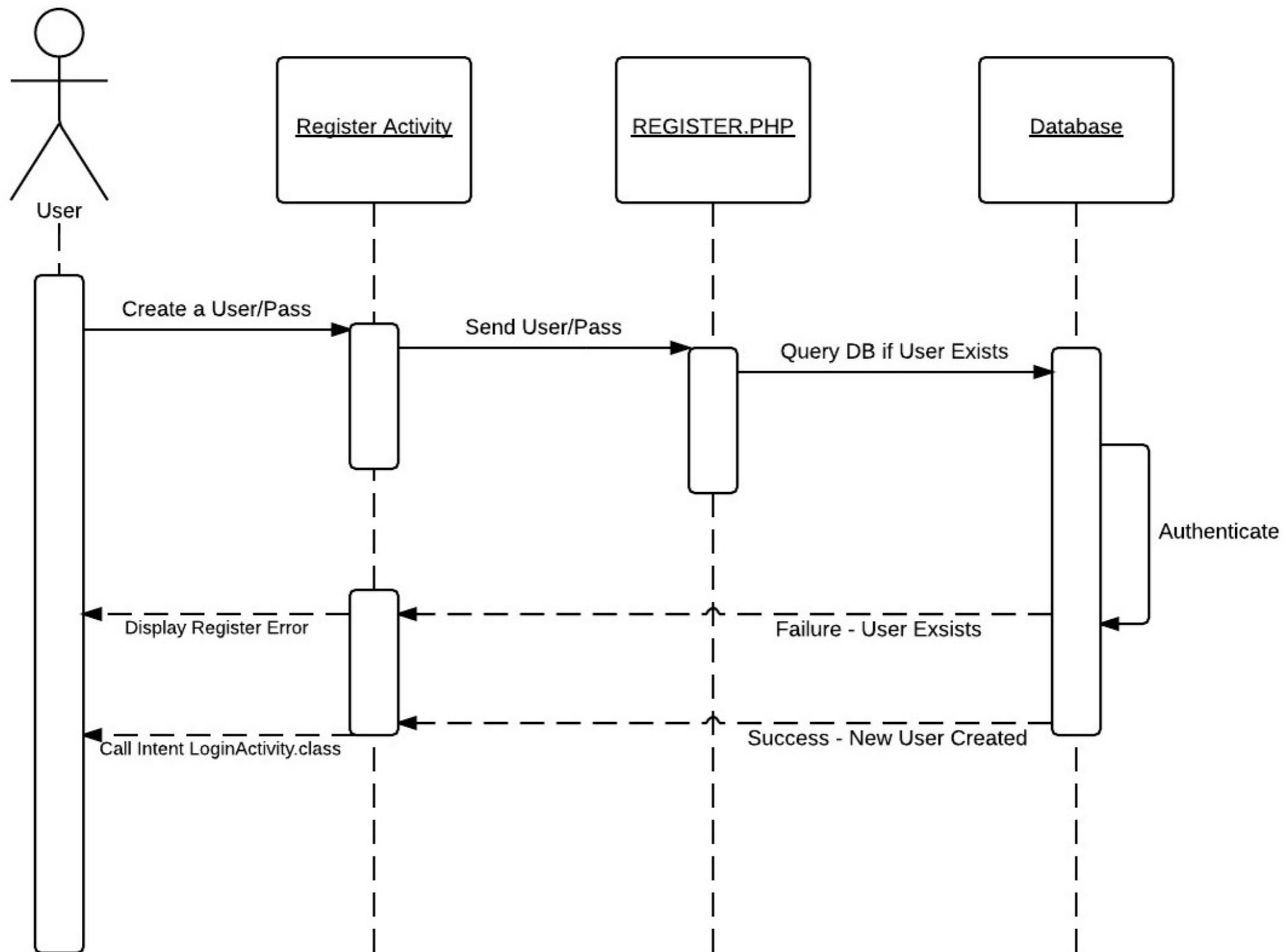
+ username : EditText
+ password: EditText
+ droid_user: String
+ droid_pass: String
+ finish: Button
+ nameValuePairs(String, String): ArrayList<NamevaluePair>
+ db : DBConnector()
+ is : db.create_connection(nameValuePairs, 2 : InputStream)
+ result: db.get_response(is : String)

onCreate(state: Bundle)
+ onClick(v : View)

Register Activity Use Case Diagram



Register Activity Sequence Diagram



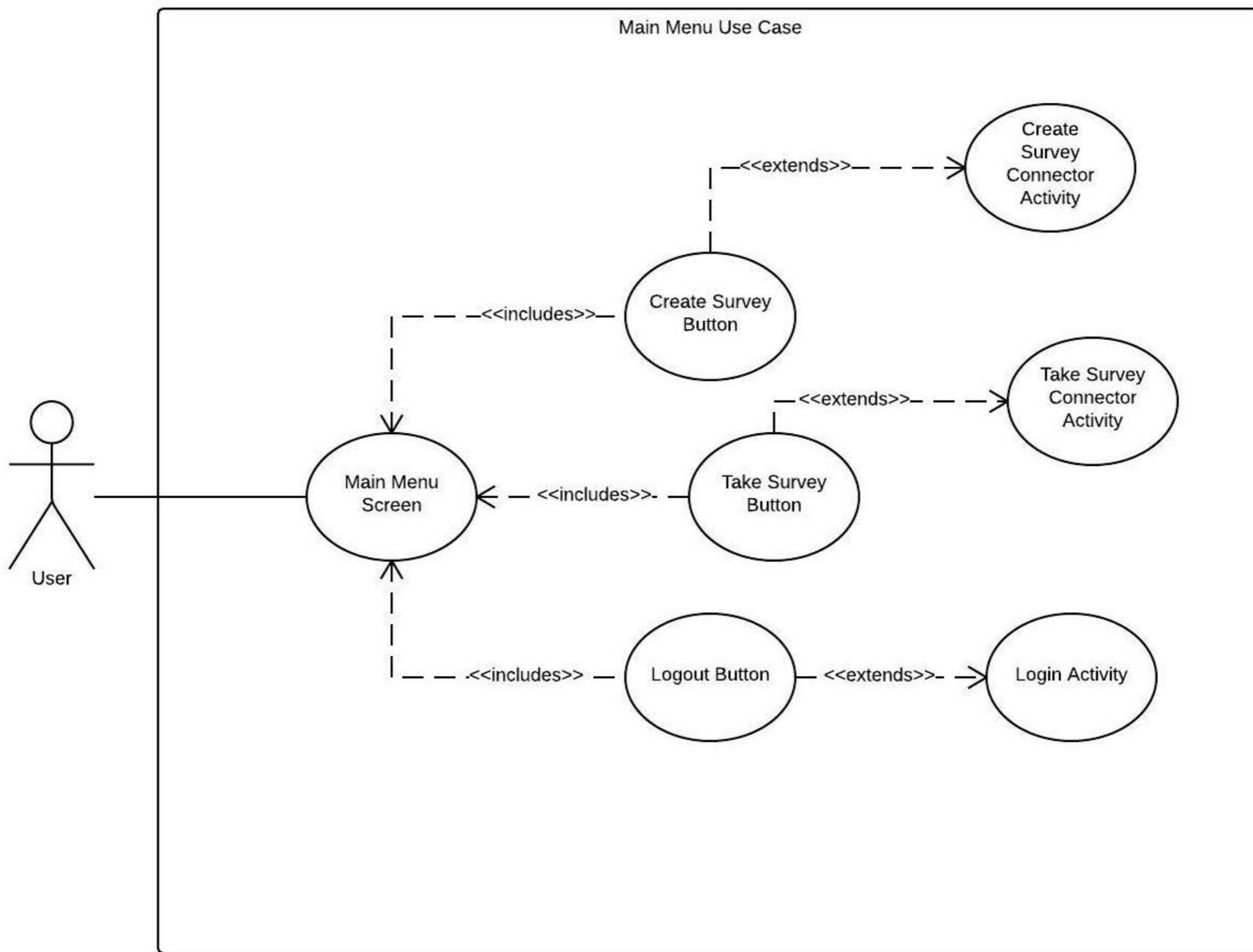
Main Menu Class Diagram

<<Java Class>> Main Menu Activity

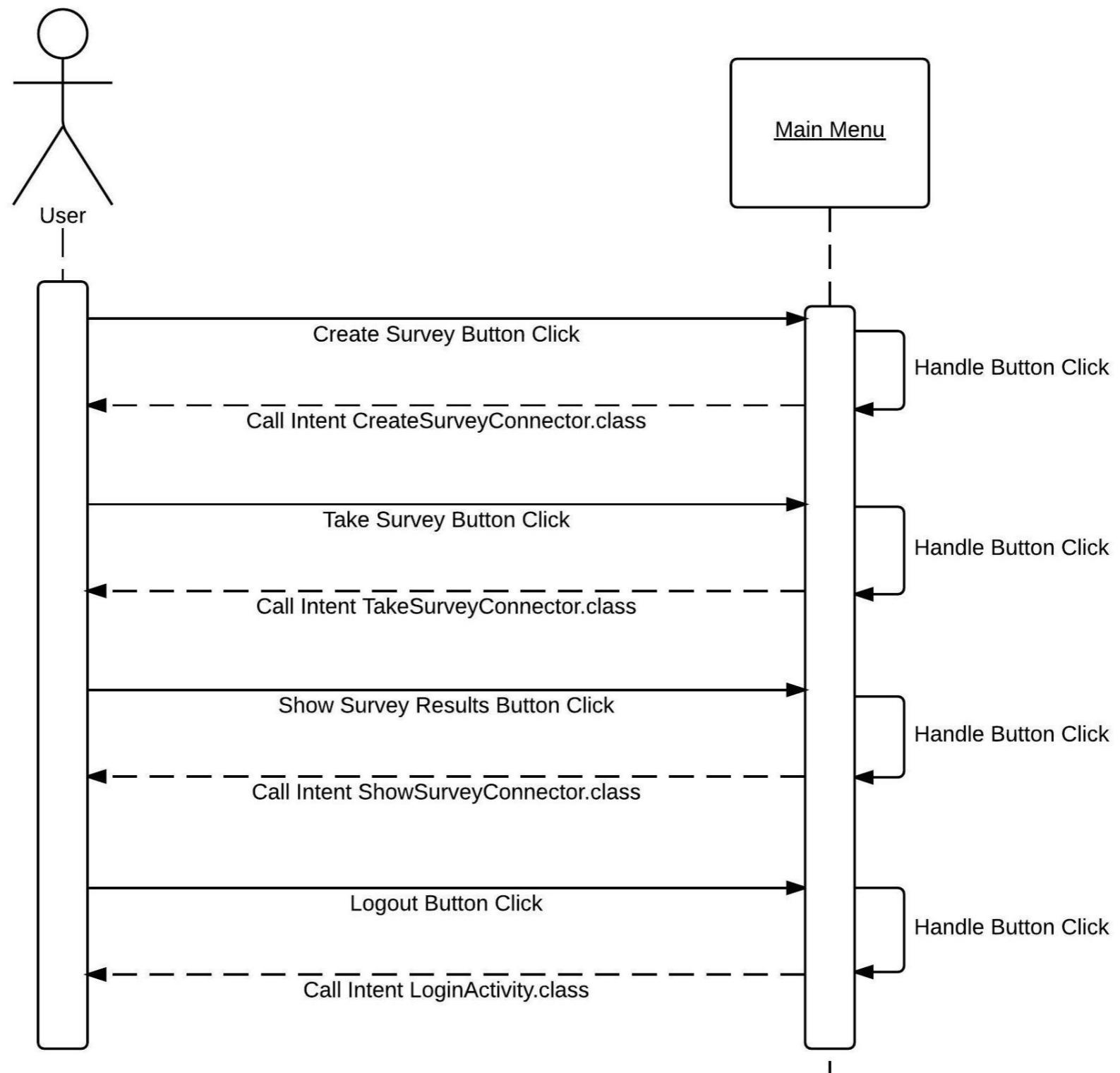
+ create_survey: Button
+ take_survey: Button
+ show_results: Button
+ logout: Button
+ droid_user: String
+ i.putExtra(String) : Intent

onCreate(state: Bundle)
+ onClick(v : View)

Main Menu Use Case Diagram



Main Menu Sequence Diagram



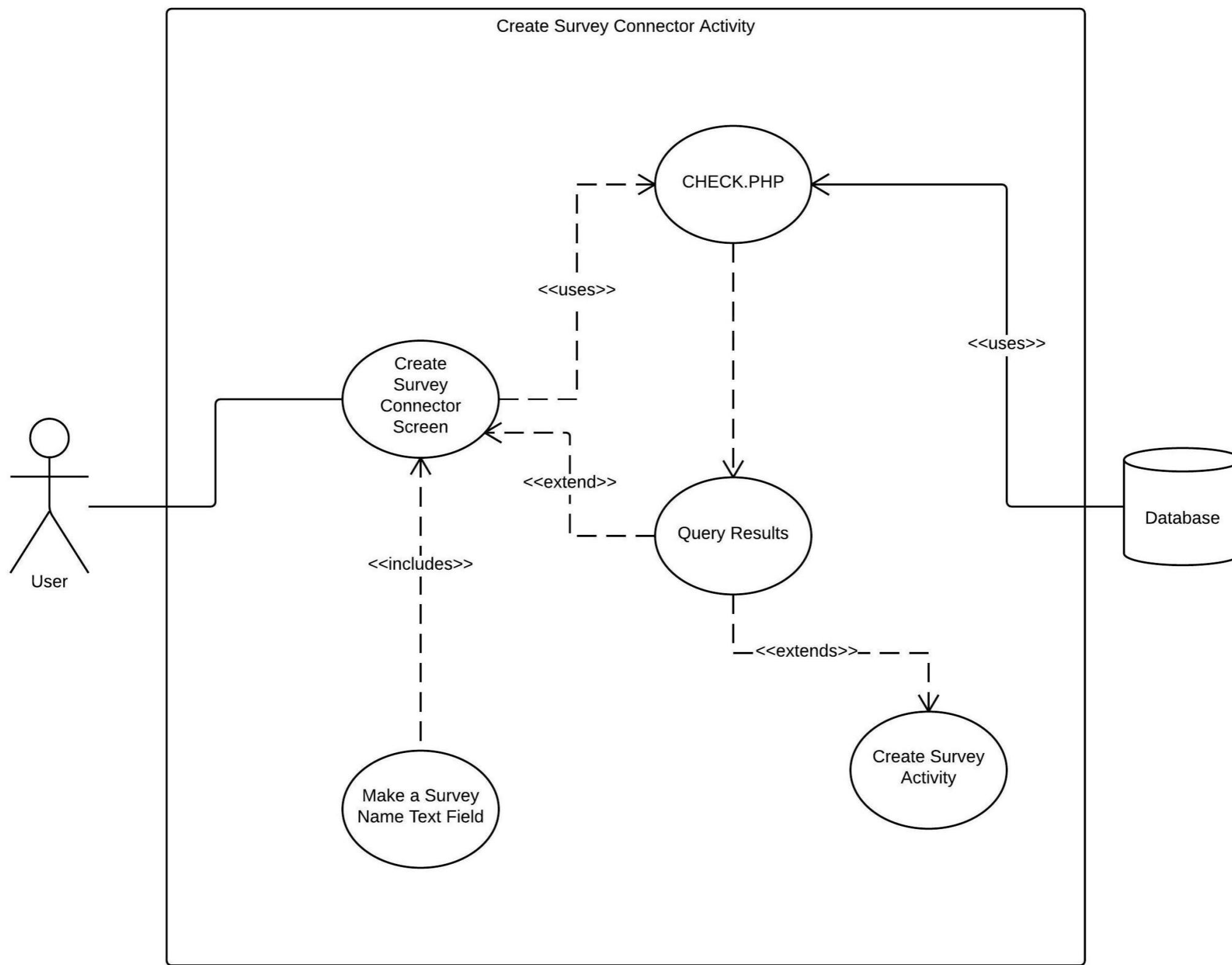
Create Survey Connector Class Diagram

<<Java Class>> Create Survey Connector

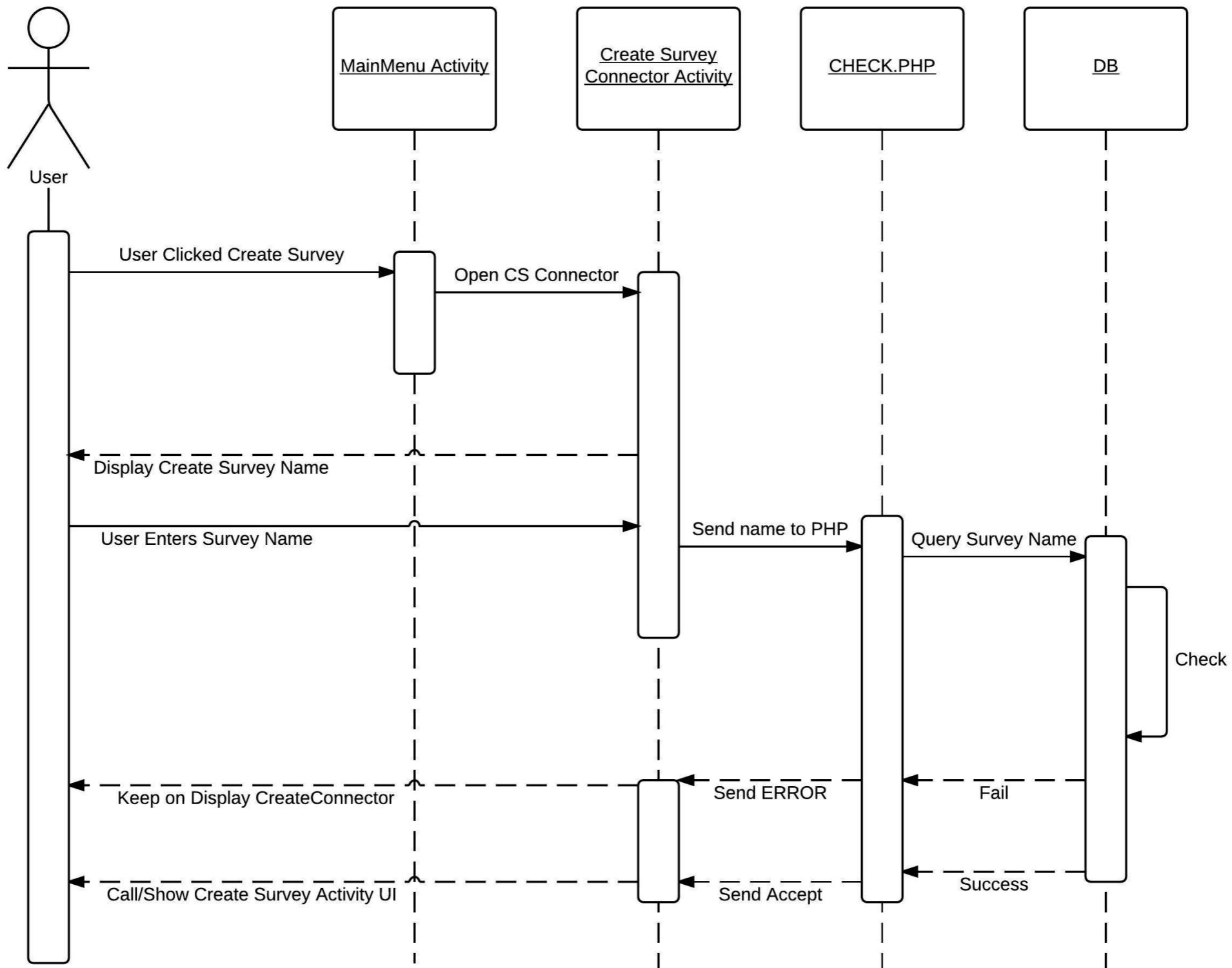
- + connect_to_survey: Button
- + survey : EditText
- + survey_name : String
- + droid_user: String
- + db : DBConnector()
- + nameValuePairs(String, String): ArrayList<NamevaluePair>
- + is : db.create_connection(nameValuePairs, 3 : InputStream)
- + result: db.get_response(is : String)
- + i.putExtra(survey_name : String) : Intent
- + i.putExtra(droid_user : String) : Intent

onCreate(state: Bundle)
+ onClick(v : View)

Create Survey Connector Use Case Diagram



Create Survey Connector Sequence Diagram



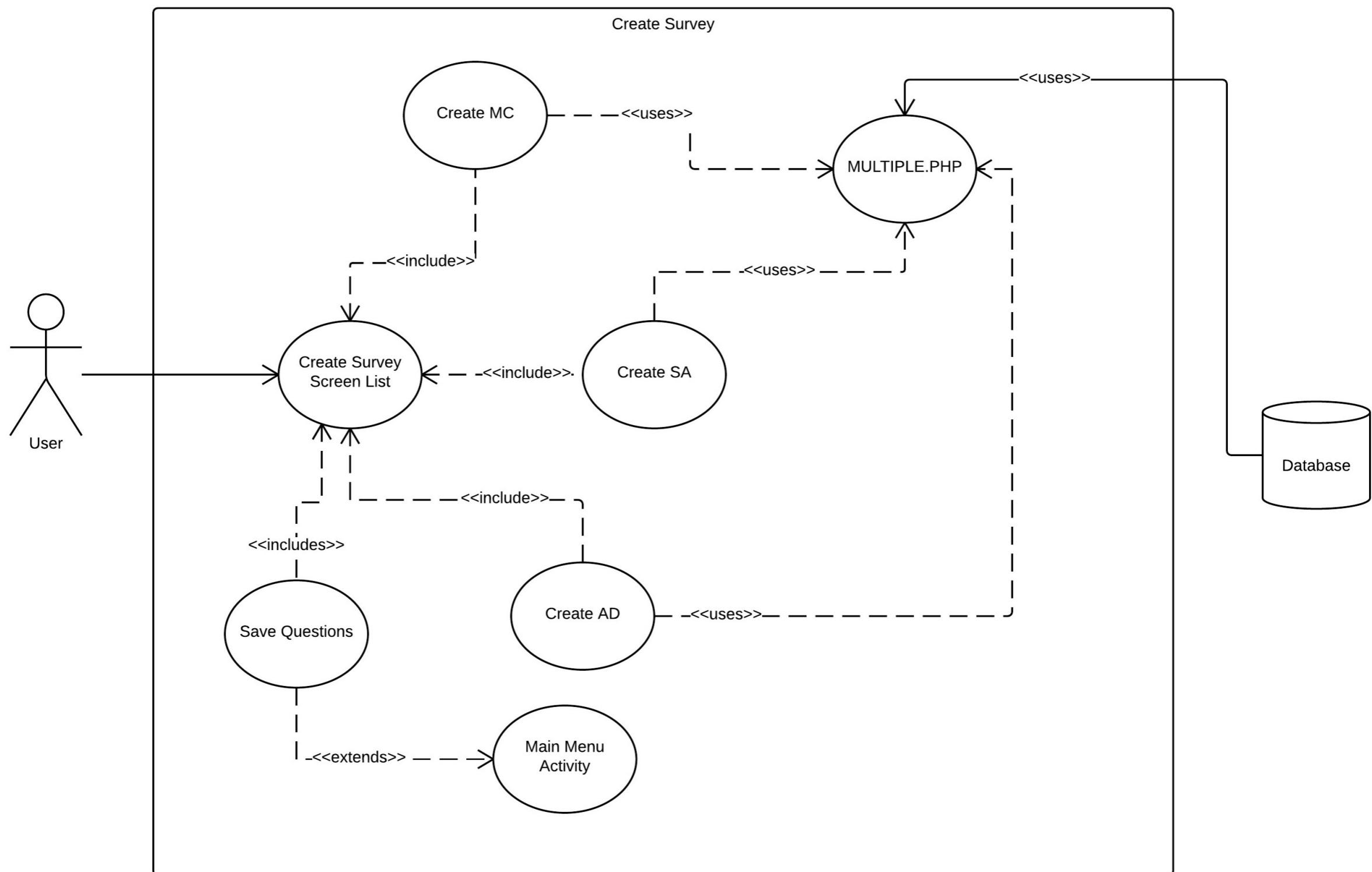
Create Survey Activity Class Diagram

<<Java Class>> Create Survey Activity

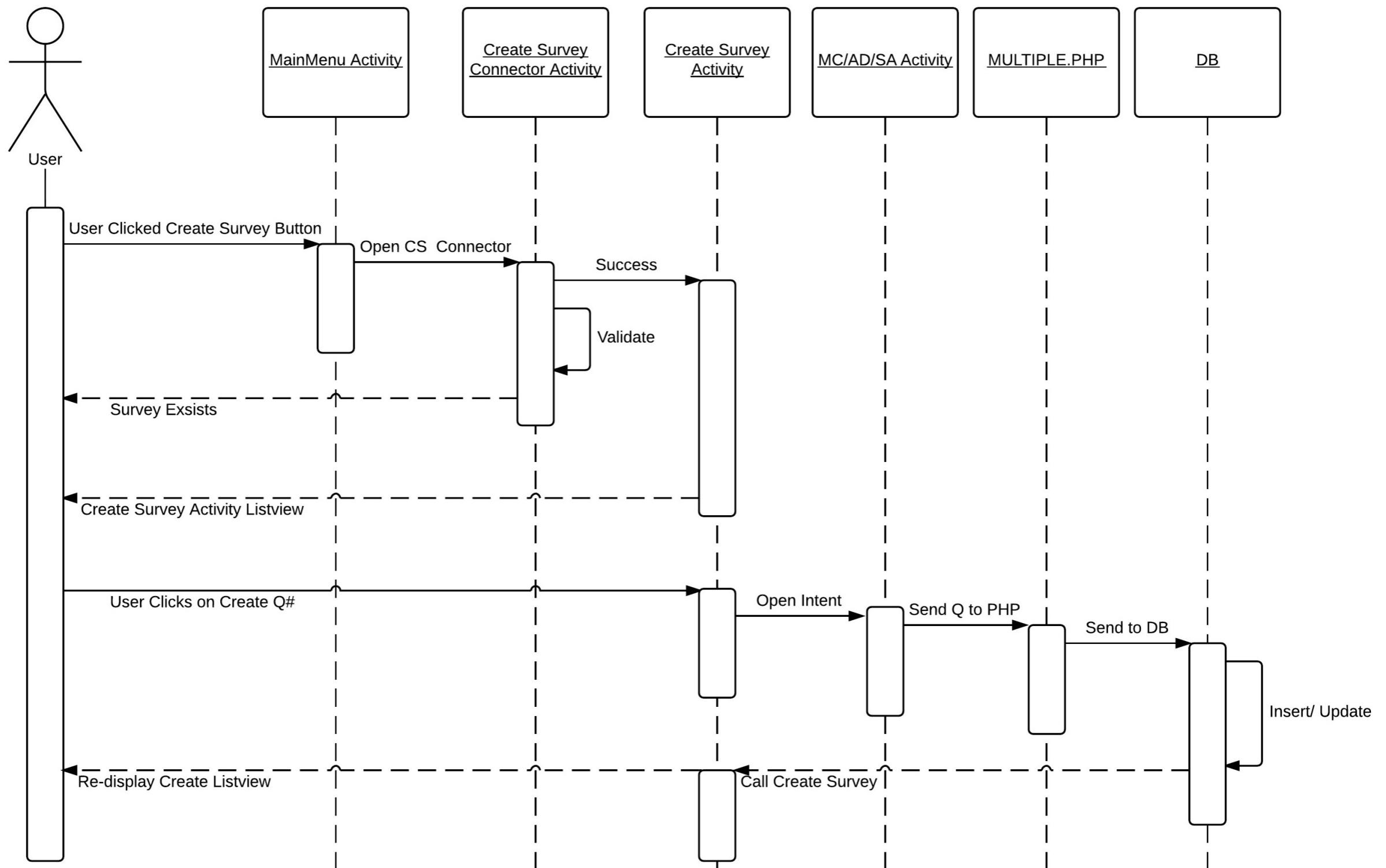
- + list: ListView
- + values : String[]
- + adapter : ArrayAdapter<String>
- + list.setAdapter(adapter: String)
- + user_sid : String
- + question : int
- + i.putExtra(user_sid : String) : Intent
- + i.putExtra(question : int) : Intent

onCreate(state: Bundle)

Create Survey Activity Use Case Diagram



Create Survey Activity Sequence Diagram



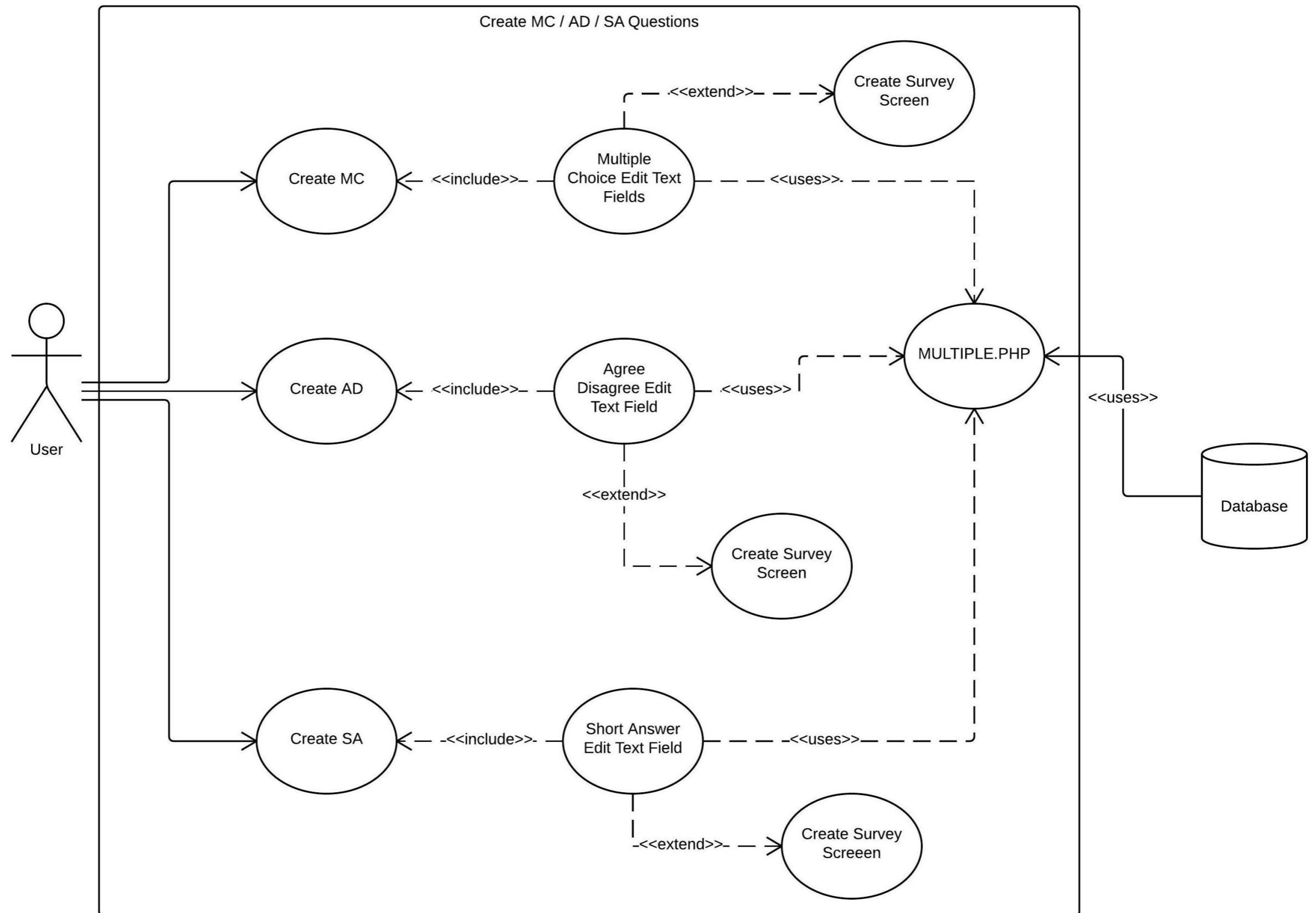
Create MC / AD / SA Activities Class Diagrams

<<Java Class>> Multiple Choice Activity
+ mc_save: Button + user_name : String + sid_name : String + question_num: int + ques: EditText.toString() + choice1: EditText.toString() + choice2: EditText.toString() + choice3: EditText.toString() + choice4: EditText.toString() + nameValuePairs(String, String, Integer.toString(String), String, String, String, String, String): ArrayList<NamevaluePair> + db: DBConnector() + is : db.create_connection(nameValuePairs, 4 : InputStream) + result: db.get_response(is : String)
onCreate(state: Bundle) + onClick(v : View)

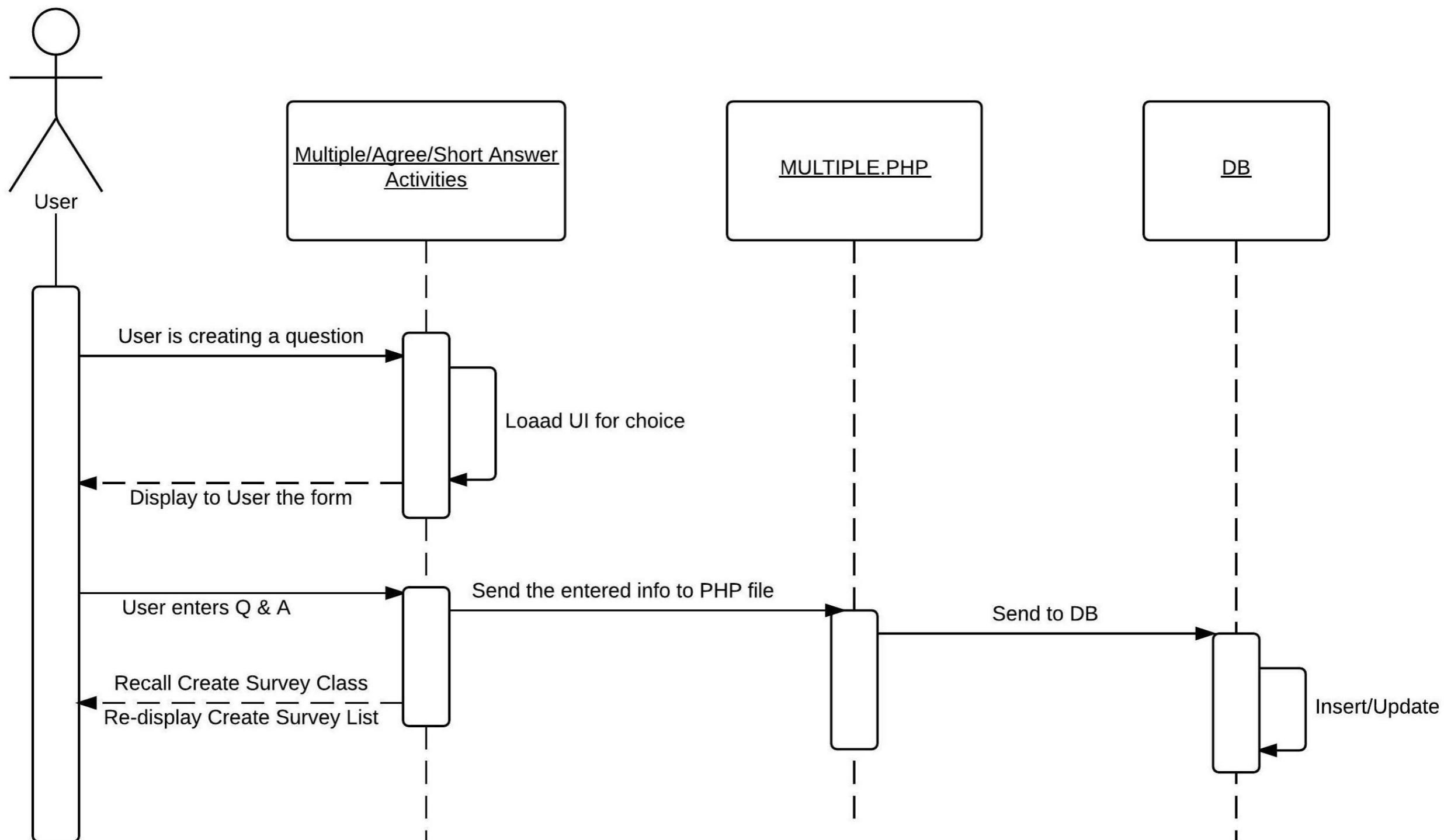
<< Java Class >> Agree Disagree Activity
+ ad_save: Button + user_name : String + sid_name : String + question_num: int + ques: EditText.toString() + choice1: EditText.toString() + choice2: EditText.toString() + nameValuePairs(String, String, Integer.toString(int), String, String, String) : ArrayList<NamevaluePair> + db: DBConnector() + is : db.create_connection(nameValuePairs, 4 : InputStream) + result: db.get_response(is : String)

<<Java Class>> Short Answer Activity
+ sa_save: Button + user_name : String + sid_name : String + question_num: int + ques: EditText.toString() + nameValuePairs(String, String, Integer.toString(int)) : ArrayList<NamevaluePair> + db: DBConnector() + is : db.create_connection(nameValuePairs, 4 : InputStream) + result: db.get_response(is : String)

Create MC / AD / SA Activities Use Case Diagram



Create MC / AD / SA Activities Sequence Diagram



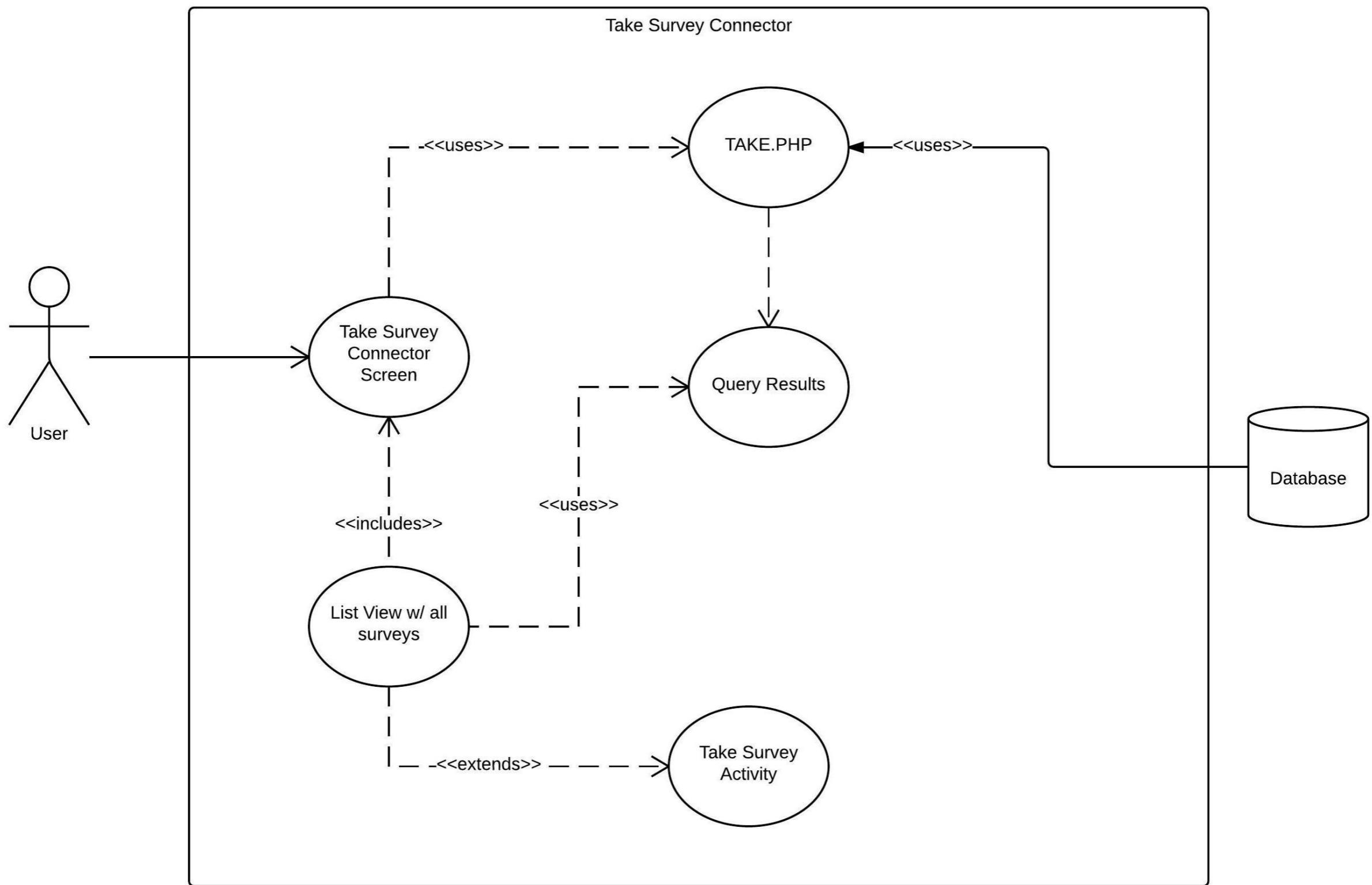
Take Survey Connector Class Diagram

<<Java Class>> Take Survey Connector

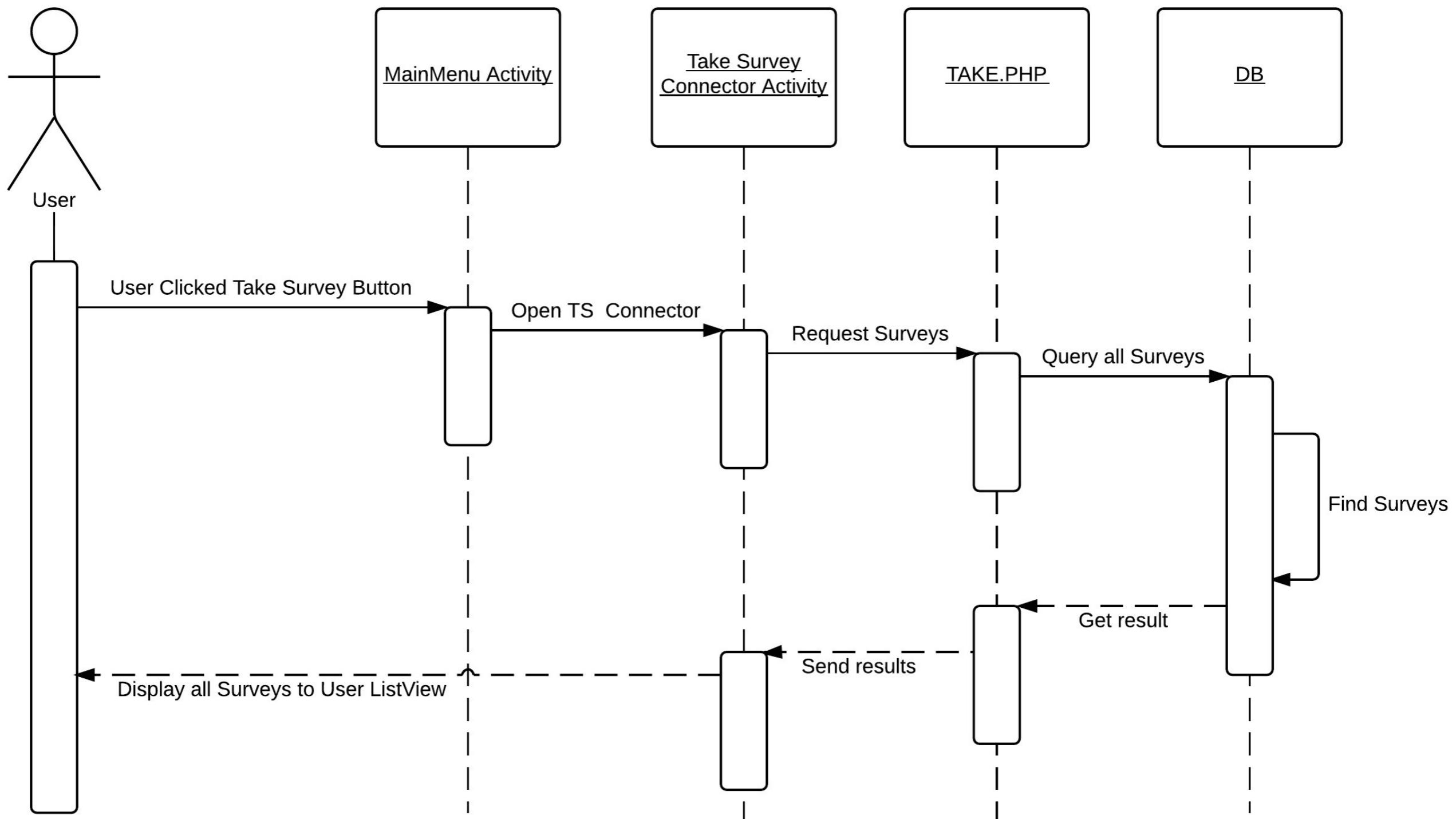
- + listval : ListView
- + result: LoadQuestions(): Class
- + list.add(result : String) : ArrayList<String>
- + listval.setAdapter(adapter : String) : ArrayAdapter<String>
- + i.putExtra(survey : String) : Intent
- + i.putExtra(user : String) : Intent
- + i.putExtra(result:String) : Intent

- # onCreate(state: Bundle)
- + TakeTask : Class

Take Survey Connector Use Case Diagram



Take Survey Connector Sequence Diagram



Answer MC / AD / SA Activities Class Diagram

<<Java Class>> AnswerMCActivity

```
+ parts : String[]
+ qid : int
+ txtQuestion : TextView
+ rda, rdb, rdc, rdd: RadioButton
+ next : Button
+ user : String
+ sid : String
+ save_answers : ArrayList<QuestionsHelper>
+ i.putExtra(parts : String) : Intent
+ i.putExtra(user : String) : Intent
+ i.putExtra(survey : String) : Intent

+ onCreate(save : Bundle)
+ onClick(v : View)
+ setQuestionView() : void
```

<<Java Class>> AnswerADActivity

```
+ parts : String[]
+ qid : int
+ txtQuestion : TextView
+ rda, rdb: RadioButton
+ next : Button
+ user : String
+ sid : String
+ save_answers : ArrayList<QuestionsHelper>
+ i.putExtra(parts : String) : Intent
+ i.putExtra(user : String) : Intent
+ i.putExtra(survey : String) : Intent

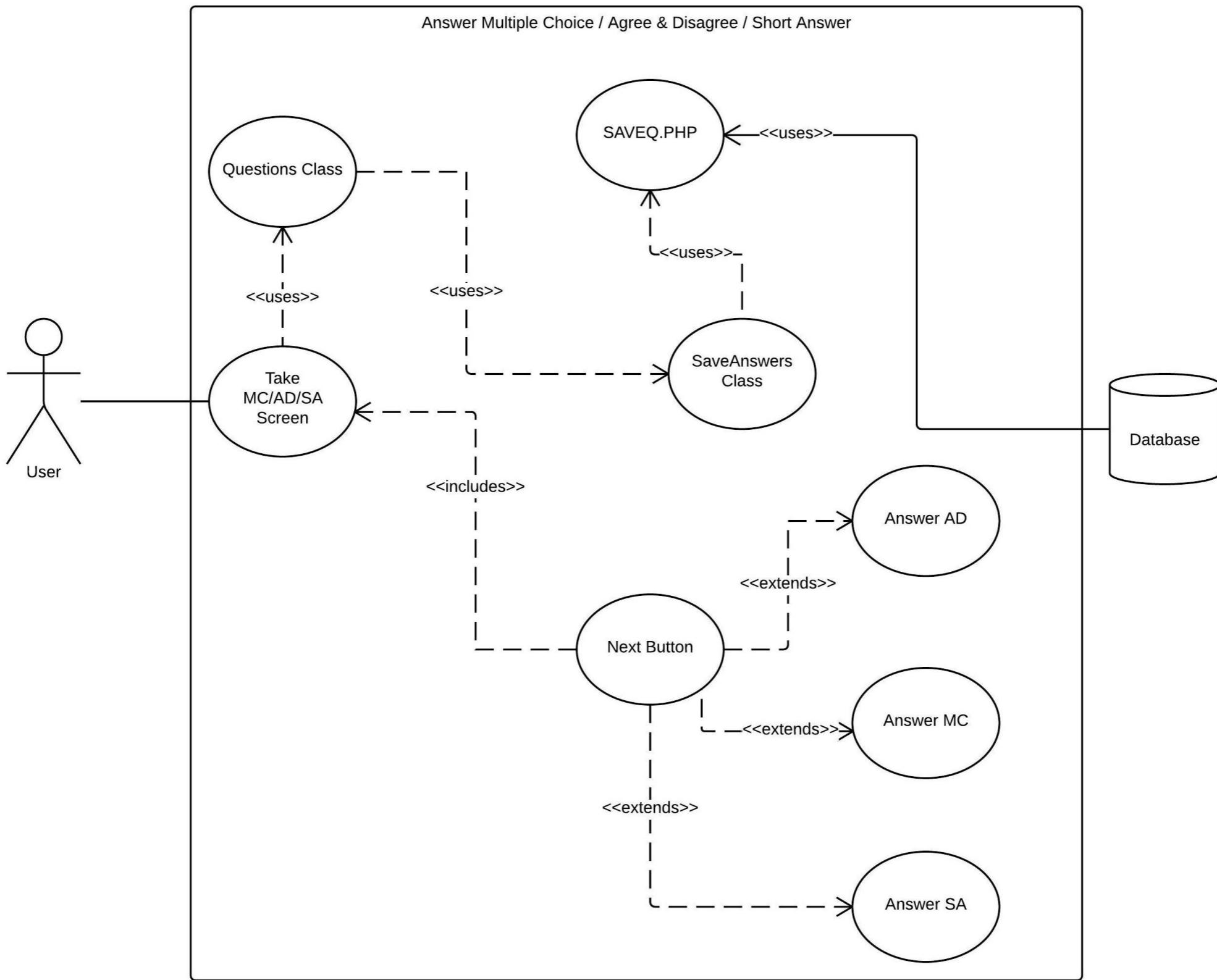
+ onCreate(save : Bundle)
+ onClick(v : View)
+ setQuestionView() : void
```

<<Java Class>> AnswerSAActivity

```
+ parts : String[]
+ qid : int
+ txtQuestion : TextView
+ writeAns: EditText
+ next : Button
+ user : String
+ sid : String
+ save_answers : ArrayList<QuestionsHelper>
+ i.putExtra(parts : String) : Intent
+ i.putExtra(user : String) : Intent
+ i.putExtra(survey : String) : Intent

+ onCreate(save : Bundle)
+ onClick(v : View)
+ setQuestionView() : void
```

Answer MC / AD / SA Activities Use Case Diagram



Answer MC / AD / SA Activities Sequence Diagram

