**CEIT Major Selection Tool**

**Final Documentation**

**Development Team:**

Christopher Poole

Caelan Cooper

Charles Eaton

Michael Powell

Amanjot Johal

Software Engineering

CSCI 5530, Section A

Dr. Andrew Allen

November 28, 2017

**CEIT Major Selection Tool**

**Final Documentation**

**Development Team:**

Christopher Poole

Caelan Cooper

Charles Eaton

Michael Powell

Amanjot Johal

Software Engineering

CSCI 5530, Section A

Dr. Andrew Allen

November 28, 2017

**Table of Contents**

**Introduction……………………………………………………………...…………….…5**

**Feasibility Study………………………………………….………...………………….....6**

Clients……………………………………………………....…...…….………......7

Obstacle..…...…………………………………………...…………….………......7

Project Scope.………………………...………………………..……....……..…...7

Benefits.…...……………………………………...……………………………….8

Software Development Process..………….……....…………………………..…..8

Preliminary Research……....…………...………………………………….….…..9

Technical Requirements….…….………...………………………………..……....9

Risk Analysis………...…………………………………....…...…….…….….......9

Timeline…………...……....………………………………………....…………...11

Suggested Deliverables…….....…….…….……….……………………….……..11

Development Process………...……….………………....………………..……...12

Outline Plan.………………...………………………………………....…...…….12

Visibility Plan…………........………………………………………....……….....13

Business Considerations.………....…….………………....………………..…….13

Conclusion………..……..…..…….………………....……....……………..…….14

**Software Requirements…...…………………….…………………………………..…..16**

1. **Introduction……………………………………………………...……....….16**

1. Purpose……………………………………………………..………...16

2. Scope…………………………………………………………..……..16

3. Document Conventions………………………………………..……..17

2. **Overall Description………………………………………...……………….17**

1. Product Perspective………………………...…………...…….……...17

2. Product Functions..…….….…...…...….….……..…….……...…......17

3. User Class and Characteristics.………………………...……....….…19

4. Design and Implementation Constraints.…...…………..……….…...19

5. Operating Environment………….………………………..………….20

6. User Documentation……….………….………………...…….…...…20

7. Assumptions and Dependencies………….…………...………..…….21

3. **System Features…...…………………….……………………...……..…….22**

1. Account Creation…………………………………….......………...…22

2. Discovery………………………………………………...…………...22

3. Exploration……………..…………………………...…………….….22

4. School Links………....…………………………...………….....…….22

4. **Specific Requirements……………………………..……….……….……....23**

1. External Interface Requirements…………...……………..………….23

1. User Interface…………………………………...….…..….…23

2. Hardware Interface………………………………..…….........25

3. Software Interface.……………………………….………......26

4. Communications Interface……………………….…........…..26

2. Functional Requirements………………………………......…….…...27

1. Use Case 1…………………………………………..…....…..27

2. Use Case 2…………………………………………...…….....27

3. Nonfunctional Requirements…………………………...….…..…......28

1. Performance Requirements…………………………..………28

2. Usability Requirements…………………………………..…..28

3. Security Requirements…………………………….……....…28

4. Software Quality Attributes……………………………..…...29

4. Other Requirements……………………………………………..…...30

5. Key Milestones………………………………………...………....….31

**Project Schedule…………………………………………………………………….…..32**

1. Timeline…………………...………………………………………...….......….33

2. Meetings………………..………………..……………………..………….…..33

3. Work Breakdown Structure………..…………………………………..……....35

**Software Design ……………………………………..……….…………..…………......36**

**1. Introduction.………………………………………………………....…...…..37**

1. Purpose of System……………………………………………..…….37

2. Design Methodology……………………………………………..….38

3. Keywords and Acronyms.………………………………………..….39

4. Document Overview...…………………………………………..…...39

**2. Proposed Software Architecture………………………………………….....40**

1. Overview………………………………………………………….…41

2. Subsystem Decomposition……………………………………..…....42

3. Hardware and Software Mapping……………………………..…….42

4. Persistent Data Management………………………………..……….42

**3. Object Design………………………………………………………………....43**

1. Overview………………………………………………………….....44

2. Detailed Class Design…………………………………………….…45

**4. Use Cases……………………………………………………………………...51**

1. Use Case Diagram…………………………………………………..51

2. Use Cases………………………………………………………...…51

**SQA and Test Plans…………………………...………………………………………...54**

1. **Introduction………………………………………………………………..55**

**2. SQA Plan…………………………………………………………..………..55**

1. SQA Requirements……………………………………….…...….….55

2. SQA Checklist…………………………………………………....….57

**3. Test Plan…………………………………………………………………….58**

1. What will be tested?………………………………………………….58

2. Testing Environment………………………………………………....59

3. Test Phases…………………………………………………………...59

4. Validation…………………………………………………………….60

5. Test Cases…………………………………………………………….61

**Post-Mortem………………………………………………………………………….....67**

**1. Introduction………………………………………………………………..68**

**2. Report Goals……………………………………………………………….68**

**3. Project Parameters………………………………………………………...68**

**4. Performance………………………………………………………………..69**

**5. Key Lessons Learned……………………………………………….……...70**

**Appendix ……………………………………..…………………………………..……...72**

**A. User Manual …………………...………………………………….....….....73**

**B. Copyright Disclaimer…………………………………………….………...81**

**C. Glossary………………………………………………………….………….81**

**D. Diagrams……………………………………………………….…………...84**

1. Analysis Model………………………………………….…………...84

2. Detailed Class Diagram………………………………….….....…….85

3. Database ER Diagram…………………………………...….….…….86

4. Use Case Diagram……………………………………….…………...87

**E. Supplementary Materials………………………………………………….87**

1. Android Studio Study…………………………………….………….87

2. Requirements Statement…………………………….……………….88

3. Discovery Quiz Questions………………………..………………….90

4. Explore Information Text…………………………....……………….92

5. Weekly Reports……………………………………...……………….96

**F. Source Code…………………………………………………...………...…104**

**Introduction**

CEIT Major Selection Tool

This software was developed as a group project for Dr. Andrew Allen’s Software Engineering course at Georgia Southern University. At the beginning of the Fall 2017 semester our group met with Dr. John O’Malley, director of Regents’ Engineering Pathway and Co-op programs, and Dr. Adrian Gardiner, associate professor in the department of Information Systems. Dr. O’Malley and Dr. Gardiner posed a problem to the group: incoming students do not seem to be getting the information they need about the differences between computing majors offered at GSU. Our group’s role, with the assistance of the clients, was to develop an application that could potentially solve this problem and fill in any information that incoming students may not receive from advising. This document details the development process of the CEIT Major Selection Tool, an application that allows students to reflect on their personal interests to determine which major in the field of computing might be a perfect fit for them.

**Feasibility Study**

CEIT Major Selection Tool

**Introduction:**

This app was commissioned as part of the Software Engineering class at Georgia Southern University (CSCI 5530). The clients this group was assigned to requested an app to help incoming and existing college students that were interested in the computing disciplines offered at this school. In an effort to close the gap in counseling that some of these students may have not received, this app provides a way to discover new options and a way to explore those options.

**Clients:**

|  |  |
| --- | --- |
| Dr. John R O'Malley | jomalley@georgiasouthern.edu |
| Dr. Adrian Gardiner | agardiner@georgiasouthern.edu |

**Obstacle:**

Incoming college students in the CEIT fields often have little knowledge of what each major encompasses and usually default to CS or IT. As they go through the courses they realize that the field they picked is unsuitable for them and often leave the CEIT field. These students are not getting properly advised while in high school and do not know about the different computing majors nor the differences between them.

**Project Scope:**

The objective of this project is to develop an app to be used by incoming students in the CEIT fields to determine which major is best suited for them. One intended feature of the app is a test or game that potential students can interact with to learn which major best compliments their skills or interests. Another feature will allow students to access information about each CEIT major including a description of the program, related career paths, and colleges that offer the major. The app will have a database to store student and admin accounts and save previous attempts for the users. The app will also provide student information for advisors/recruiters at different colleges and possibly information about different colleges for the students.

**Benefits:**

The development of this app will lead to higher retention of students in the CEIT fields as well as less time and money spent on career choice. The app aims to fill in the lack of information about CEIT majors for high school students that should be received from their advisors. At the same time, the app will provide information about colleges to users and information about users to colleges in order to give the students somewhere to start on their college search.

**Software Development Process:**

The team will follow the agile development technique with the idea of design thinking. Because of the short timeline, the app will need to be developed quickly in order to deploy it on time. Usually this process does not focus on documentation heavily, however the team will provide documentation related to the project at least weekly. This process will facilitate a user centric design and keep the clients’ views and the team’s product in line with each other.

**Preliminary Research:**

The team will review each major to be included in the app in order to better understand what sets each of them apart as well as what skills and abilities are important for each. In addition, the team thought it would be beneficial to research existing apps and websites with similar functions in order to see how they were implemented and how they could be improved.

**Technical Requirements:**

*Platform* - The app will be programmed on the Android interface.

*Database* - The database will store student user results and information so that when the user logs in, they can access them again and the admin users can contact them based on their results.

*User/Admin Login* - A login to differentiate the user view from the administrator view.

*Security* - User information will need to be protected from outside parties.

*Reporting Structure* - A method to provide feedback and reporting of the results to the user and admin.

*Interface* - A clean and intuitive UI allowing the user to easily navigate between the test/game, pages displaying major information, and the about page.

**Risk Analysis:**

**Requirements Change**

**Risk:** There is a possibility that the client might have a few different ideas about the system during the project. Depending on the situation, changes/modifications that the Client wishes to have implemented may require subtle or significant changes to the architecture structure or development process.

**Solution:** Project manager and team members need a coherent plan and visual design with Client to reduce significant alterations to the software architecture and interface towards end of development.

**Specification Delay**

**Risk:** Potential for delays for specifications on interfaces that need to be delivered based on schedule.

**Solution:** Maintain constant contact with the clients and coordinate with them on all levels. If serious problems arise, the team will contact Dr. Allen.

**Impractical Features**

**Risk:** Maintaining a database for users of this app could prove to be more work than it’s addition is worth. There is no practical reason for a student to save their attempts at the selection quiz, as the quiz only suggests a program of study for them to enter, which the users can access information about from the main menu of the app. Including this feature would require the addition of a database and a security system to protect user info while offering very minimal benefits.

**Solution:** Change the scope of the project to call for a non-persistent app to eliminate the need to support accounts.

**Time Management**

**Risk:** The project must be completed before the end of the Fall semester and therefore no time extensions will be given. The risk involved is that it is possible that some non-essential features may not be fully implemented by the final deadline.

**Solution:** The team will adhere to a strict timeline to complete as much of the project as possible within the given time frame. As a worst case scenario, the app will be designed in a way that leaves it extendable so more work may be done to complete it.

**Timeline:**

|  |  |
| --- | --- |
| Week 5~6 | Feasibility Report Conducted |
| Week 6 | Research and initial mock-ups finish, begin coding prototype |
| Week 7 | Continue work on prototype app, begin designing method to get results from the quiz |
| Week 8 | Finish quiz design, begin implementing in the app, begin coding the login screen |
| Week 9 | Finish login screen, begin constructing database |
| Week 10 | Finish database design and begin implementing it |

**Suggested Deliverables:**

After the first meeting with the clients, the team will conduct a feasibility study and give a report to the clients and Dr. Allen. During the course of the semester we will be developing a software application to be used by incoming students in College of Engineering and Information Technology department at Georgia Southern. The final version of the deliverable will be presented in front of faculty and the client at the end of the semester. In addition, earlier version deliverables along with considerable documentation will be provided. The team will also provide a weekly progress report to Dr. Allen.

**Development Process:**

The development team will be following an iterative step by step process according to the design model after determining with the client whether the project is feasible or not.

1. Gather requirements

2. Initial design plan and timeline

3. Additional analysis

4. Prototype

5. Implementation

6. Testing

7. Debugging

8. Deployment

9. Maintenance

**Outline Plan:**

1. Feasibility study completion – September 22, 2017

2. Initial design and development – TBD

3. By early October further design and development will be done including the possibility for extension. The development should slowly start to coming together for first deliverable prototype.

**Visibility Plan:**

**External**

The team will plan in-person meetings with client and external stakeholders at least once a week for the purpose of discussing project status, potential problems/ risks, changed requirements and feedback. Our client is conveniently located on the third floor of the IT building in the dean’s office. We will regularly communicate by email or in-person meetings.

**Internal**

The team will regularly plan project meetings at least once per week either in the IT building or in class based on available schedule. Communication is key in a project of this scope so the team will be using the GroupMe app to stay in contact and up to date on meetings. All meeting will be recorded by a single group member to keep track of progress and milestone. In addition, all project documentation and reports will be shared on a google drive server.

**Business Considerations:**

As the Project Developers and Georgia Southern University students, the copyright for this software belongs to the team. The copyright will be transferred to the clients (listed above) with an unrestricted license for use of the system at the end of the semester.

**Conclusion:**

After conducting this feasibility study, the team finds that the project is feasible. The clients requirements are within the skill range of the team members. The team members are also familiar with the software and hardware to be used. Having a one semester time constraint is manageable and all the clients requirements can be fulfilled. The project risks can easily be kept to a minimum and the benefits far outweigh them.

Software Requirements Specification

CEIT Major Selection Tool

1. Introduction

**1.1 Purpose**

The purpose of this Software Requirements Specifications document is to give a detailed description of the functionalities and requirements for the CEIT Major Selection Tool. This document will encompass all of the systems planned features. This document will also give a preliminary glance into the structure of the software such as the back end database and the applications front end User Interface(UI). It will also go on to explain requirements and various technical dependencies.

**1.2 Project Scope**

The CEIT Major Selection Tool is an Android platform based application which helps incoming college freshmen in the CEIT fields to determine which major is best suited for them. The CEIT majors to be included in the project are Computer Science, Information Technology, Information Systems, and Computer Engineering. These will be the focus of the application however the application will allow for the expansion of supported majors in the future.

One intended feature of the app is a quiz or game that potential students can interact with to learn which major best compliments their skills or interests. Another feature will allow students to access information about each CEIT major including a description of the program, related career paths, and colleges that offer the major.

The app will have a database to store student and admin accounts and save previous attempts for the users. The app will also provide student information for advisors/recruiters at different colleges and possibly information about different colleges for the students.

**1.3 Document Conventions**

This SRS document contains some terminology which potential readers might be unfamiliar with. For a list of unfamiliar terms and their definitions, see Appendix C (Glossary).

2. Overall Description

**2.1 Product Perspective**

The CEIT Major Selection Tool is a new, self-contained system for use on the Android platform. The application will be used to help potential students in the CEIT fields to find a close career match based on interests and discover in depth information about each major. The mobile application’s front end user interface (UI) will communicate with a back end database to store user information and provide security.

**2.2 Product Functions**

The following list features a brief outline and description of the main features of the CEIT Major Selection Tool.

1. ACCOUNT REGISTRATION

o Allows user to create an account

o Can be accessed from the login screen

2. USER ACCOUNT LOGIN

o Allows user to login to access app

o Allows user to save progress on quiz

o Allows user to save results on quiz

3. PLAYABLE QUIZ GAME

o Quiz that users play to match them up with a major based on responses

o Emails the user their results

4. ABOUT SCREEN

o Displays information about the creators and clients

o Provides copyright information

5. CEIT MAJOR INFORMATION SCREEN

o Provides an in depth look into the CEIT majors and their respective careers

**2.3 User Classes and Characteristics**

There will be 5 types of anticipated users who will interact with the system:

|  |  |
| --- | --- |
| Incoming freshman interested in any CEIT field. | Incoming students can utilize the app’s quiz and information to discover alternatives to their chosen CEIT career field to determine which major is best suited for them. They can also create an account to save their progress. |
| Incoming freshman not interested in any CEIT field. | Incoming students with no interest in the CEIT fields can utilize the app’s information and quiz to garner interest when their was previously none due to lack of knowledge. They can also create an account to save their progress. |
| Current college students looking for a major change | Current college students can use the app to expand their knowledge about the CEIT majors offered at their respective colleges. They can also create an account to save their progress. |
| School administration | School administrators will manage the application to monitor student results from the quiz so that the respective colleges and universities can outreach to potential students. |
| System administrator | The administrator manages the overall system to ensure that all functions are properly working and that all information for the majors in the CEIT fields is up to date. |

**2.4 Design and Implementation Constraints**

Since the application is on a mobile platform, the small screen size and resolution will be a major design constraint. Too small text and complicated layout can cause users to abandon the app. In addition, since the app is being developed for touchscreen devices any interactive features such as buttons or sliders should be large enough to be easily controlled. The app is not graphic or resource heavy so their are no other hardware constraints.

Another constraint is connecting a full database from the cloud to the Android platform. Since the app will be transferring data to and from the cloud database, it is important that the user’s device be connected to the Internet in order to function properly. Security is also a concern any time user data is transmitted, therefore measures will be taken to prevent inadvertently exposing personal information to other parties. Data transfers will need to be as efficient as possible to avoid network processes using too many resources such as battery power or memory on the user’s device. In order for the app to connect to the SQL database, it will need to use a PHP function in JavaScript Object Notation (JSON) to retrieve and push information to it.

**2.5 Operating Environment**

The CEIT Major Selection Tool will run on the Android platform. In order to ensure that the app can run on most devices (62.6% of active devices on the google play store as of 10/13/17), the software will be implemented on the Lollipop Operating System (Android 5.1 | API 22). The application is entirely self contained and will not rely on any external software or other Android related software components.

**2.6 User Documentation**

The CEIT Major Selection Tool is designed to be as simple to use as possible. The software will come with a brief tutorial document on how to navigate the app and a description of it’s main functions. In addition, a more detailed step-by-step set of instructions can be found in Appendix A (User Manual).

**2.7 Assumptions and Dependencies**

**Time Dependencies**

The Major Selection Tool will have its core features and additional features planned out to later be implemented into the the tool. The core features are the basis for which The Major Selection Tool was built upon and are needed for the program to be useful for the situations on which it was created for.

Additional features are not currently implemented into The Major Selection Tool but have been planned out and structured within to be later added or potentially expanded upon. These features are done and added with discretion to the time spent developing and time remaining after development upon the core features is complete. The decision for which these features will be implemented will come in the final design phase.

**Hardware Dependencies**

In order to function properly, the app should be running on an Android device that supports Android 5.1 and up. In addition, the device should be able to connect to the Internet as it is required for the transfer of user data.

3. System Features

**3.1 Account Creation**

Users will be able to create an account to store their login information. With their account, the app will store their email and their most recent attempts at the discovery test.

**3.2 Discovery**

Users will have the option to answer questions about their interests that will allow the application to suggest a program of study.

**3.3 Exploration**

Users can see information about the available majors, including a description of the program of study and a list of possible occupations.

**3.4 School Links**

The app shall be able to link users to Georgia Southern Website pages including admissions and specific pages for the majors available in the app.

4. Specific Requirements

**4.1 External Interface Requirements**

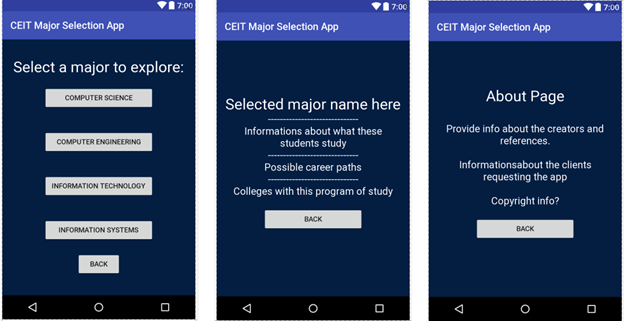
This section provides information about the various interfaces utilized by the application including those for the user, hardware, software, and communications. In addition, a prototype of the user interface is provided for reference.

**4.1.1 User Interfaces**

Upon starting the app, a splash screen will be displayed showing the name and logo. Next, the new account page will be displayed, providing the user with boxes to enter an email, password, and password confirmation. An option for users who have already made an account will allow them to travel to a login page where they are asked for the email and password they signed up with.

Upon signing in, the user is taken to the main menu where they are given the option to access the discovery feature, explore majors, visit the about page, or sign out of the application.





The discovery feature will begin by asking the user a series of questions to be answered with either a button or slider. A progress meter at the top of the page will display how many questions the user has completed and how many remain. When all questions have been answered, a results page will show the major recommendation(s) for the user along with justification for the results.

The explore feature will take the user to a page displaying each of the supported majors (CS, CE, IS, IT). Upon clicking one of the majors, the user is taken to an information page for that major. The information page displays the name of the major along with categorized details about the major such as the program of study, required skills, possible career paths, and a list of colleges that offer the program.

The about page will provide information about the creators of the app, the stakeholders, current build version, and potentially copyright information. In addition, the discovery, exploration, and about pages will each have a button to easily return to the main menu.

**4.1.2 Hardware Interfaces**

CEIT Major Selection Tool is at the moment intended for the Android Platform, with planning for future expansion into iOS. The exchange of information such as name, email address, and password is stored on a database separate from the Android device.

CEIT Major Selection Tool is being developed for Android 5.0 (Lollipop) and all of the versions that come after.

**4.1.3 Software Interfaces**

The Major Selection Tool when developed under the Android Platform is to use the Java JDK (Java Development Kit) and the Android SDK (Software Development Kit) tools.

For database usage and storage MySQL 5.7.14 will be used, along with the Java MySQL Connector API.

Outgoing data consists of registration informations such as Name, Email Address, and Password submitted by the user to the server containing the database. Also, the user’s results after using the discovery portion of the Major Selection Tool will be sent to the server.

Incoming data consists of progress session saved and sent to the user locally from their current device, as well as their user registration information sent to the user from the server containing the database.

**4.1.4 Communications Interfaces**

Communication between the user’s phone and the server containing the database will occur only for a few instances. Those instances are:

* When the user creates an account
* When the user logs into their existing account
* When the user completes one instance of the quiz
* When the user requests to see previous attempts at the quiz

These communications will be handled through PHP functions written in JSON.

The app will also have the ability to connect the user with links to the Georgia Southern website. These links will open in the user’s default browser using an HTTP protocol.

**4.2 Functional Requirements**

**4.2.1 Use case 1**

High School student using the website/app to reflect on their personal alignment (interests, abilities, goals) with each program, and being motivated by the application to seek more focused information (e.g., I did not have any idea what IT is about, but now I have used this application I want to know more as it seems a fit for me).

**4.2.2 Use case 2**

High School student already knows what major they are interested in, but they aren’t particularly knowledgable about that major as much as they think they should be, so using The Major Selection Tool they are able to read and learn about their selected major and what it requires instead of going in blind.

**4.3 Nonfunctional Requirements**

**4.3.1 Performance Requirements**

System performance should not be a major issue because queries from the server involves small sets of data. Changing screens will require very little computation resulting in very fast transition. The cost -division algorithms used by the mobile application will be highly efficient, taking only a split second to complete.

**4.3.2 Safety Requirements**

CEIT Major Selection Tool is cloud-based application which will not affect data stored outside its servers nor will it interfere with other installed applications on the user’s phone. It can not cause any damage to the phone hardware and software components. CMST should not be used while operating vehicles or dangerous equipment that requires the user’s attention.

**4.3.3 Security Requirements**

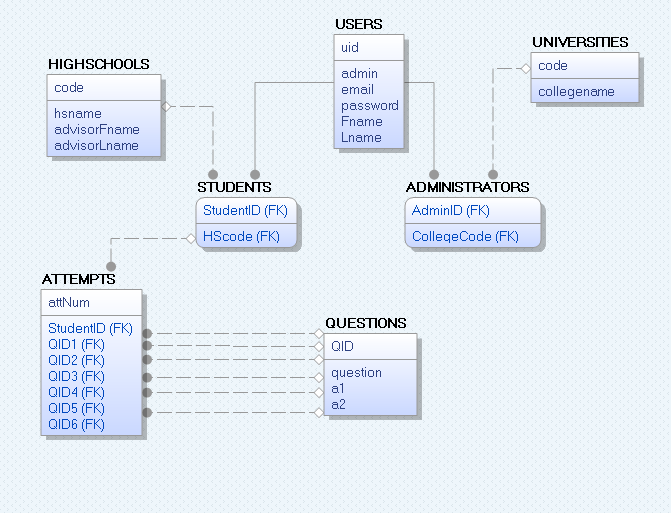
This application uses a login system where only a email address and password is required to verify the identity of the user upon opening the app. Thus, this method of persistence authenticates the user’s identity for privacy security and data safety concerns. In addition, measures will be taken to ensure that transferred user information is secure and not shared with any other parties. Microsoft Azure, the platform for hosting the server, already has an encryption with a rotating key built in, which protects all information within the database.

**4.3.4 Software Quality Attributes**

The graphical user interface of CEIT Major Selection Tool is to be designed with ease of usability and learning as the first priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate. In addition, the user interface and layout will be simple enough that users will learn features quite quickly for easy use and navigability. There will be feedbacks and visual cues such as notifications to inform users of updates and pop-ups to provide users with instructions. To ensure reliability and correctness, there will be zero tolerance for errors in the algorithm that determines a list of recommendations of majors for users. To maintain flexibility and adaptability, the app will take into account situations in which a user loses internet connection or for any reason and cannot establish a connection with the server. These users will still be able to use the application while disconnected will be cached until the connection is restored. With CMST being ported solely for the Android platform, this software application has the advantage of being portable and convenient to use.

**4.4 Other Requirements**

**4.4.1 Database Requirements**

****

The database will need to be housed on a cloud server for ease of access and affordability. It will have specified tables for student and admin users, students’ latest 4 discovery attempts, and schools associated to each admin. Theoretically, the database will support up to 550,000 users; however, it will only hold up to 20 users for the trial.

.

5. Key Milestones

**Milestone Deadline**

Feasibility Study September 22, 2017

GUI Design October 14, 2017

SRS Document October 14, 2017

Server Setup October 21, 2017

Software Design Document October 27, 2017

Implementation November 4, 2017

SQA and Test Plans November 10, 2017

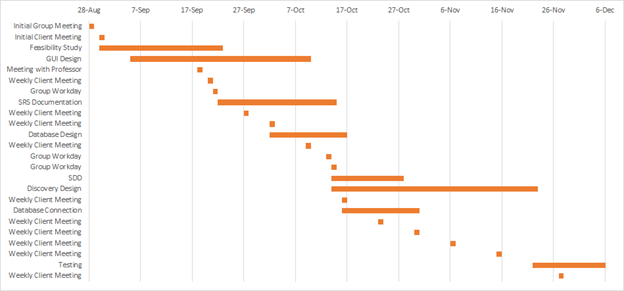
Test Reports November 28, 2017

Final Report December 7, 2017

Project Schedule

CEIT Major Selection Tool

1. Timeline



2. Meetings

|  |  |  |
| --- | --- | --- |
| **Date** | **Duration** | **Purpose** |
| 8/28/17 | 20 mins | * Initial meeting with group * Discussed regular meeting times and coordination |
| 8/30/17 | 50 mins | * Initial meeting with clients, discussed problem to solve * Began brainstorming and researching |
| 9/5/17 | 5 mins | * Began feasibility study |
| 9/18/17 | 15 mins | * Discussed concerns with professor |
| 9/20/17 | 50 mins | * Discussed scope and platform with clients |
| 9/21/17 | 40 mins | * Discussed potential GUI designs |
| 9/22/17 | 3 hrs 30 mins | * Drafted feasibility report |
| 9/27/17 | 50 mins | * Presented GUI prototypes to clients * Team roles assigned |
| 10/2/17 | 40 mins | * Database discussion * Began developing questions for the discover section of app |
| 10/2/17 | 45 mins | * Cemented upcoming tasks and deadlines |
| 10/9/17 | 35 mins | * Refined GUI prototypes presented * Discover questions further discussed |
| 10/13/17 | 3 hrs 30 mins | * Online collaboration for SRS |
| 10/14/17 | 3 hrs 30 mins | * Online collaboration for SDD |
| 10/16/17 | 35 mins | * Database schema presented * Discussed need for cloud server |
| 10/23/17 | 75 mins | * Progress report to clients |
| 10/30/17 | 30 mins | * Demonstrated connection to database to clients |
| 11/6/17 | 30 mins | * Progress report to clients |
| 11/15/17 | 30 mins | * Demonstration of discovery portion of app |
| 11/27/17 | 30 mins | * Discussed implementation of OpenAuth |

3. Work Breakdown

* Amanjot Johal
  + Front-end development
  + Connectivity
* Caelan Cooper
  + Front-end development
  + Documentation
  + Testing
* Charles Eaton
  + Documentation
  + Testing
* Chris Poole
  + Group lead
  + Back-end development
  + Documentation
* Michael Powell
  + Back-end development
  + Documentation

**Software Design Document**

CEIT Major Selection Tool

1.Introduction

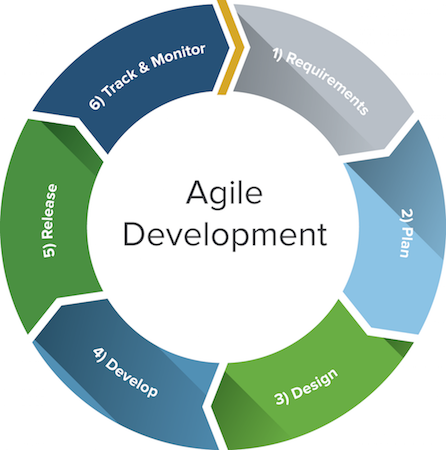
The CEIT Major Selection Tool is intended to supplement the educational advisement of students interested in computing and IT related majors. In order for the application to be as helpful as possible, its design should be detailed, accurate, and well-planned. The purpose of this document is to provide a thorough record of the software’s specifications and overall architecture to ensure that the application is well-defined and complete. This chapter will introduce the CEIT Major Selection Tool by more clearly defining the system’s purpose, outlining relevant design methodologies and keywords, and providing a general overview of this document. This document may contain terms, acronyms, or abbreviations that some readers may be unfamiliar with. Some keywords and acronyms can be found in Section 1.3. For more detailed clarification on any unfamiliar terms please refer to Appendix C (Glossary).

**1.1 Purpose of the System**

The reason for the creation of the CEIT Major Selection Tool is to fill any knowledge gaps prospective college students may have about majors within the computing and technology disciplines. The differences between majors such as Computer Science and Information Technology are not always made clear to those being advised. The CEIT Major Selection Tool intends to expound on those differences in a way that appeals to each particular user’s interests. The application is not meant to replace traditional college advisement but instead serves as a supplemental tool to help inform users of what makes each major unique. Utilizing a series of questions, the application will also allow users to discover which major best suits their individual skills and interests.

**1.2 Design Methodology**

The CEIT Major Selection Tool is being developed under the Agile software development process with an emphasis on design thinking. The Agile methodology was chosen for its flexibility, rapid delivery, and the benefit of constant feedback from stakeholders. Design thinking is a complementary thought process that focuses on research to understand the needs of users before generating ideas and putting them into action. For the CEIT Major Selection Tool, Trello is being used to manage project tasks and ensure that they are completed by a deadline.



**1.3 Keywords and Acronyms**

**API:** Application Programming Interface

**App:** Application

**CE:** Computer Engineering

**CEIT:** College of Engineering and Information Technology

**CS:** Computer Science

**Erwin:** Data modeling software

**GUI:** Graphical User Interface

**HTTP:** HyperText Transfer Protocol

**IDE:** Integrated Development Environment

**IS:** Information Systems

**IT:** Information Technology

**PHP**: Hypertext Preprocessor

**Trello:** project management software

**UML:** Unified Modeling Language

**XML:** Extensible Markup Language

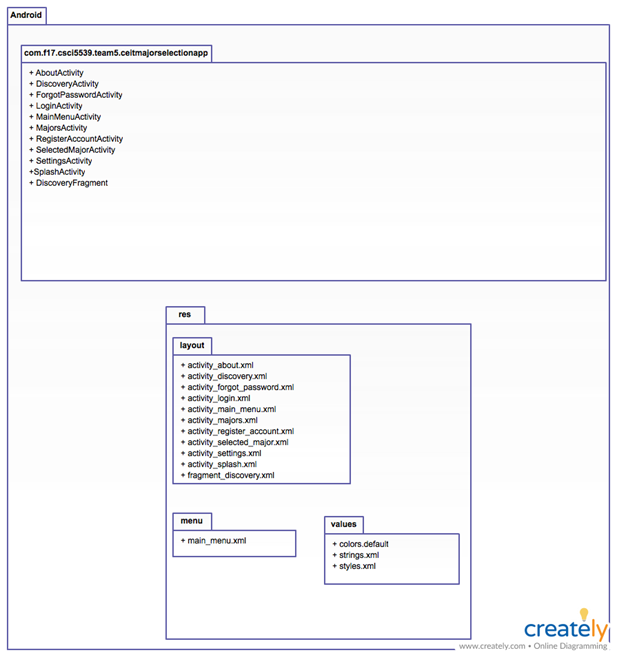
**1.4 Document Overview**

In the following chapters the design of the CEIT Major Selection Tool will be further detailed. Chapter 2 (Proposed Software Architecture) examines the overall architecture of the application. This includes an overview of the system structure, descriptions of the major subsystems, information about hardware and software mapping, and details on data management. Chapter 2 will also address security requirements and discuss how they will be resolved. Chapter 3 (Object Design) introduces the class structure and object interactions implemented within the CEIT Major Selection Tool. This chapter includes visual diagrams and descriptions of the purpose of each class as well as how they are connected to one another. By the end of this document all software design considerations pertaining to the CEIT Major Selection Tool will have been addressed.

2. Proposed Software Architecture

Planning and designing the architecture for a system is one of the most important steps in software engineering. Without proper forethought the system could be limited in scope, function, extendability, or some other critical component. A well-designed architecture will give a software system a strong foundation and will overall lead to more organized and thorough work. With this in mind, this chapter will provide an in-depth look at the structure of the CEIT Major Selection Tool. This includes a breakdown of each of the major subsystems as well as the use cases associated with them. In addition, the chapter will examine hardware and software mapping for the application, address data management persistence, and provide solutions for security concerns.

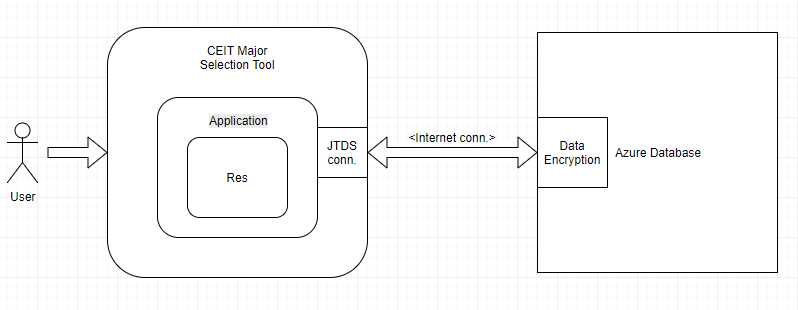
**2.1 Overview**



**2.2 Subsystem Decomposition**

The CEIT Major Selection Tool is split up into 2 previously shown packages. These packages are the res package, which contains resources such as styling and layout, and the application package, which contains the main processes of the application. The below figure shows how these packages interact.

**2.3 Hardware and Software Mapping**



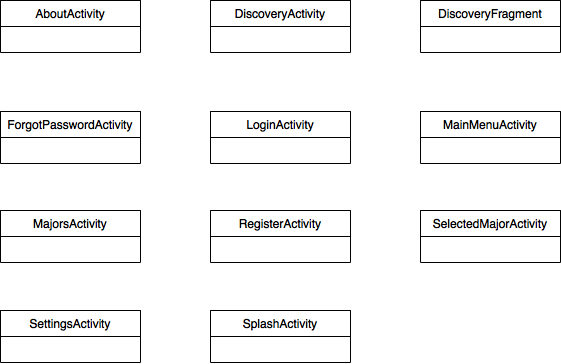
**2.4 Persistent Data Management**

The app will need store information about users, discovery attempts, and the colleges that would use different versions of the app. It will store all users’ full names and email addresses, assign them an ID number based on the order of accounts created, and differentiates them from students and administrators. For students, the app also stores what high school they attend/ attended, their advisor’s name, and the latest 4 attempts of the discovery quiz. For administrators it only stores the college that they are tied to. The app will also save the state it was last in if the user closes it during a discovery attempt. Soon, it will also include information about the colleges (such as icons and color schemes) for branding purposes.

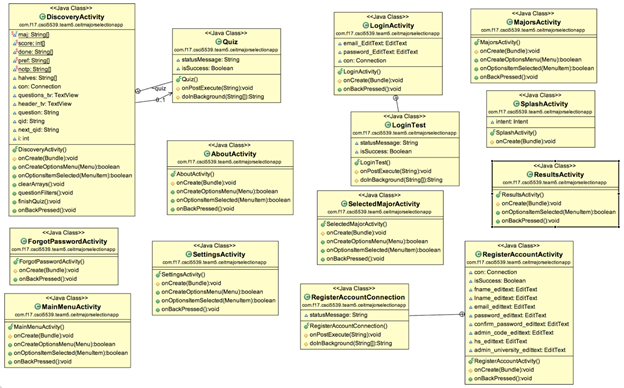
3. Object Design

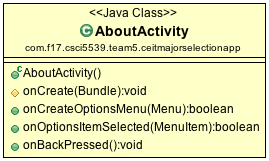
The design and structure of objects is another important aspect within software engineering. Through the use of proper object design, it can improve the intended software’s efficiency, maintainability and overall robustness. With that in mind, this chapter will discuss several object design diagrams for the CEIT major selection tool software such as UML diagrams, class diagrams, and sequence diagrams. The three sections will discuss structural relationships between classes that will be implemented in subsystems, object interactions relating to use cases and a description of each of the classes including methods and give an emphasis on details outlined in the overview.

**1. Overview**

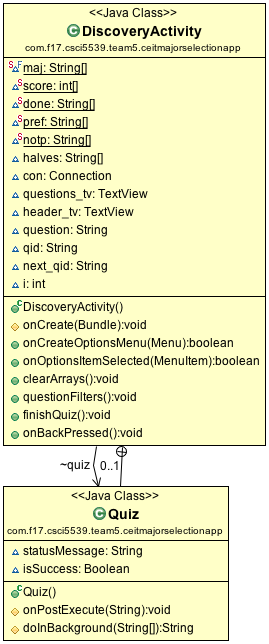
****

**2. Detailed Class Design**

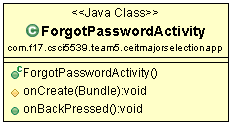
****



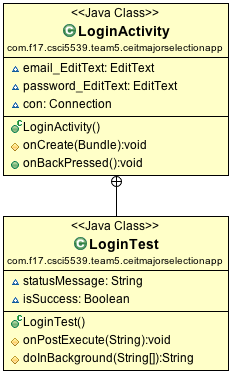
AboutActivity is simply changing to the About Page which tells about the application, the clients, and the developers.



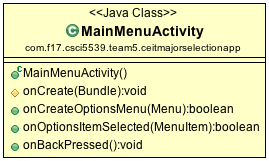
DiscoveryActivity is with Quiz as discovery is basically the quizzing aspect of the application. It processes the questions and answers it’s pulled from the database, and in the end outputs a major the user is leaning towards.



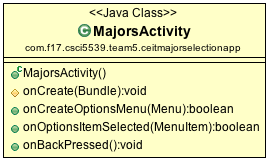
ForgotPasswordActivity finds the user from the database and sends the password to the email address they signed up with.



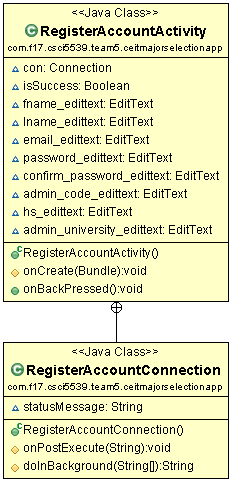
LoginActivity takes the email and password, along with LoginTest finds the user in the database with the matching values and logs that user in. If LoginTest returns false for isSuccess it does not log the user in.



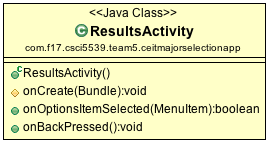
MainMenuActivity is the starting menu the user first sees with Discovery, Major Info, and About pages.



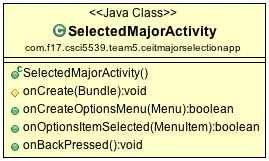
MajorsActivity explains the various majors that is currently implemented into the application.



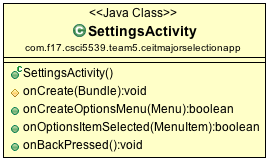
RegisterAccountActivity along with RegisterConnection takes the values submitted in the registration page, connects to the database and inserts that information if it does not already exist.



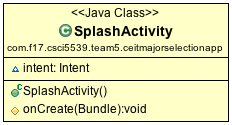
ResultsActivity is the results page after the Discovery process. Gives a primary major, and secondary major. As well as a way to the pages about those two specific majors.



SelectedMajorActivity will give the user the major for which they’ve selected. It gives information about that major such as potential careers and colleges.



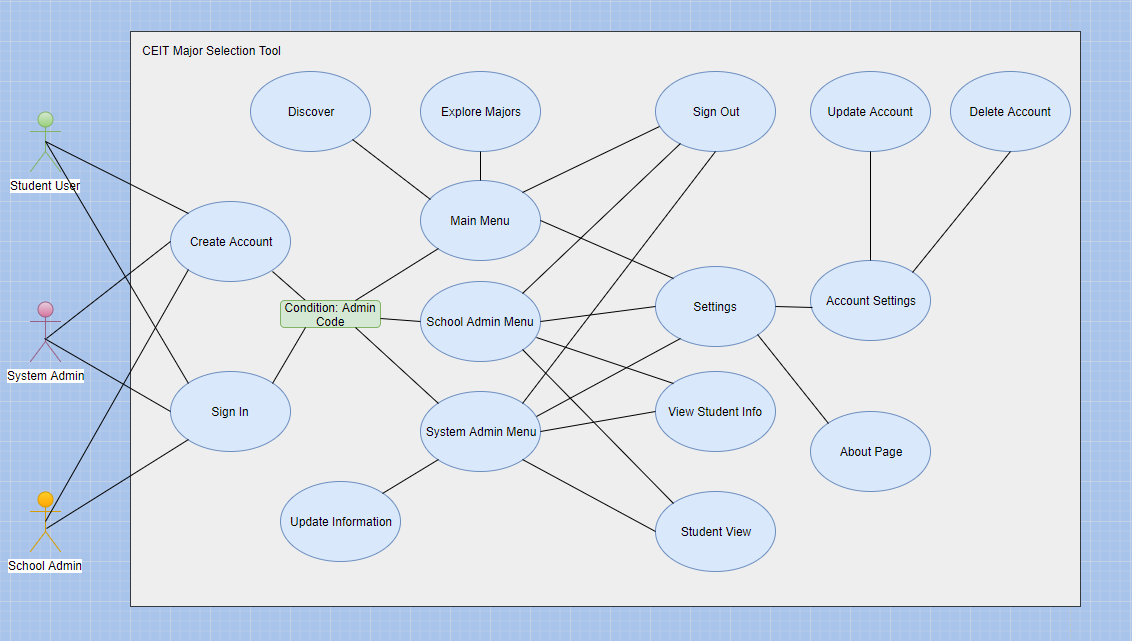
SettingsActivity takes the user to the Settings pages which allows for various things such as name, password, and email.



The splash screen which shows the Georgia Southern Logo.

4. Use Cases

**1. Use Case Diagram**

****

**2. Use Cases**

**Use Case 1**

High School student using the app to reflect on their personal alignment (interests, abilities, goals) with each program, and being motivated by the application to seek more focused information (e.g., I did not have any idea what IT is about, but now I have used this application I want to know more as it seems a fit for me).

**Use Case 2**

High School student already knows what major they are interested in, but they aren’t particularly knowledgable about that major as much as they think they should be, so using The Major Selection Tool they are able to read and learn about their selected major and what it requires instead of going in blind.

**Use case 3**

College advisor creates an administrator account. The advisor wants to update the color scheme and view what the students see. From the main menu, they select the app settings option and select the color that needs to be changed. Once they are done, they return to the main menu and select the student view. From here, they have all the same options as a student user. They can see how the test works, what information is displayed for the major.

**Anticipated Users**

|  |  |
| --- | --- |
| Incoming freshman interested in any CEIT field. | Incoming students can utilize the app’s quiz and information to discover alternatives to their chosen CEIT career field to determine which major is best suited for them. They can also create an account to save their progress. |
| Incoming freshman not interested in any CEIT field. | Incoming students with no interest in the CEIT fields can utilize the app’s information and quiz to garner interest when their was previously none due to lack of knowledge. They can also create an account to save their progress. |
| Current college students looking for a major change | Current college students can use the app to expand their knowledge about the CEIT majors offered at their respective colleges. They can also create an account to save their progress. |
| School administrator | School administrators will manage the application to monitor student results from the quiz so that the respective colleges and universities can outreach to potential students. |
| System administrator | The administrator manages the overall system to ensure that all functions are properly working and that all information for the majors in the CEIT fields is up to date. |

SQA and Test Plans

CEIT Major Selection Tool

1. Introduction

In order to ensure that a software application performs the way it is intended to, thorough testing and quality assurance is required. This document defines the software quality assurance and planned testing approach for the CEIT Major Selection Tool. These plans are intended to be executed for both the prototype and final version of the application. The purpose of this document is to outline all tools, procedures, and techniques used to ensure that the application meets all requirements specified in the software requirements specification. The following tests will verify that the application is complete, meets all necessary requirements, and provides a user-friendly experience.

2. SQA Plan

**1. SQA Requirements**

Software quality assurance requirements for the system are outlined as follows:·

* Documentation
  + Feasibility Study
    - Client identified
    - Project scope defined
    - Includes risk analysis
  + Software Requirements Specification
    - Overall description of the application
    - Intended user(s) identified
    - Specific requirements listed
    - Use cases specified
* Implementation
  + Code structure
    - Thoroughly commented code
    - Processes occur in correct order
    - Error handling present
  + User Interface
    - UI is consistent and easy to use
    - Buttons, menus, etc. are visible and easily accessed
  + Database
    - Application front-end is connected to database
    - Database transactions are successfully processed
* Testing
  + Testing methods
    - Tests cover all aspects of the application
    - Program logic is tested
    - Tests are documented and problems are resolved
  + Features
    - User is able to create account and login
    - Discovery quiz is functioning
    - Explore section lists accurate information
    - Settings menus function
    - User can create admin account
    - Account can be updated and deleted

**2. SQA Checklist**

|  |  |  |
| --- | --- | --- |
|  | **Pass/Fail** | **Comments** |
| **Documentation** |  |  |
| Feasibility Study |  |  |
| SRS Document |  |  |
| SDD Document |  |  |
| SQA & Test Plans |  |  |
| Client identified |  |  |
| Project scope defined |  |  |
| Audience defined |  |  |
| Specific requirements listed |  |  |
| **Implementation** |  |  |
| Code thoroughly commented |  |  |
| Processes handled correctly |  |  |
| Error handling present |  |  |
| UI is consistent and easy to use |  |  |
| Elements are visible and easily accessed |  |  |
| Application front and back end are connected |  |  |
| Database transactions processed successfully |  |  |
| **Testing** |  |  |
| Tests cover all aspects of application |  |  |
| Tests are documented and resolved |  |  |
| User can create account and login |  |  |
| Discovery quiz populates new questions |  |  |
| Explore section lists major information |  |  |
| Settings menu functions correctly |  |  |
| User can update or delete account |  |  |
| Admin functionality is working |  |  |

3. Test Plan

**1. What will be tested?**

* Application performance
* Login and registration
* Discovery section functionality
* Explore section functionality
* Settings for application and account
* Account updates and deletion
* Database transactions

**2. Testing environment**

Testing for the CEIT Major Selection Tool will be executed on the following platforms:

* Application - The CEIT Major Selection Tool will be installed and tested on a physical Android device as well as on devices emulated by Android Studio.
* Database - The Azure Database for MySQL will be queried to ensure transactions are completed.
* Android Studio - The code for the application will be tested and debugged using Android Studio 2.3.

**3. Test phases**

Testing for the CEIT Major Selection Tool will be carried out in three phases:

* Continuous - Important features of the application will be briefly tested during all phases of development.
* Alpha - Test plan will be carried out upon completion of a working prototype.
* Beta - Test plan will be carried out on final version of the application prior to submitting deliverables.

**4. Validation**

The following activities will be performed in order to validate project requirements:

* Plan acceptance testing and include criteria for:
  + Meeting all established performance requirements
  + Providing adequate documentation
  + Resolving errors and unexpected results
* Execute test plan during each testing phase
* Document tests and results
* Verify requirements are met with client feedback

**5. Test cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test**  **Environment** | **Test Date** | **Feature Tested** | **Required Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC\_01 | Application | 12/06/2017 | Startup | Open app by tapping icon. | After tapping icon, app loads successfully within a few seconds. | Application opens and loads to the login quickly and successful. | pass |
| TC\_02 | Application | 12/06/17 | Login | Attempt login with registered email address and password. | Existing user is able to login. | Existing user login([eatoncr3@gmail.com](mailto:eatoncr3@gmail.com))  password(1234) produces successful login. | pass |
| TC\_03 | Application | 12/06/17 | Login | Attempt login with unregistered account or incorrect password. | Error message displayed for invalid credentials. | Login input ([ce01893@georgiasouthern.edu](mailto:ce01893@georgiasouthern.edu)) & pass (1234)produces incorrect login.l | pass |
| TC\_04 | Application | 12/06/17 | Register | Register new user email address and password with app. | Account is successfully registered. | Email [eatoncr3@gmail.com](mailto:eatoncr3@gmail.com) and pass 1234 Account created! Please Login | pass |
| TC\_05 | Application | 12/06/17 | Register | Attempt to register already existing account. | Error message displayed for already registered account. | Account created! Please Login No error message. | fail |
| **Test Case** | **Test**  **Environment** | **Test Date** | **Feature Tested** | **Required Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC\_06 | Application | 12/06/17 | Login/Register | Login or register without filling in email or password fields. | Error message displayed for missing input. | Please fill out all fields message. | pass |
| TC\_07 | Application | 12/06/17 | Login/Register | Login or register with incorrect email format. | Error message displayed with correct email format. | Incorrect email or password error message | fail |
| TC\_08 | Application | 12/06/17 | Discovery | Open and answer questions in the discovery section of the app. | Questions displayed page by page. Once answered, next question is displayed on the page. | Questions displayed page by pages and transitions to the next page when answered. | pass |
| TC\_09 | Application | 12/06/17 | Discovery | Complete discovery section of the app. | Depending on question responses, resulting major is displayed. | Results page is displayed with primary(Computer Science and secondary major  (IT) | pass |
| TC\_10 | Application | 12/06/17 | Discovery | Ensure discovery results match intended results from decision algorithm. | Correct major is suggested when discovery complete. | Based on discovery quiz inputs two majors are matched on the results page. | pass |
| TC\_11 | Application | 12/06/17 | Explore | Open explore section of the app. | Buttons for each available major are displayed. | Buttons for each of the four majors are displayed and working. | pass |
| **Test Case** | **Test**  **Environment** | **Test Date** | **Feature Tested** | **Required Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC\_12 | Application | 12/06/17 | Explore | From explore section click button for desired major. | Scrollable information page about the chosen major is displayed. | When button clicked transitions to information major page which is scrollable. | pass |
| TC\_14 | Application | 12/06/17 | Settings | Click settings icon. | Settings menu is displayed. | Settings drop down icon menu works. | pass |
| TC\_15 | Application | 12/06/17 | About | From settings menu, select about. | About page is displayed with relevant information. | Scrollable about page with relevant app and development information. | pass |
| TC\_16 | Application | 12/06/17 | Home | Select home button. | User is returned to the main menu of the app. | Home button (identified as “Back”) brings user back to main menu page. | pass |
| TC\_17 | Application | 12/06/17 | Account Settings | Select account settings from menu. | Options to update or delete account displayed. | Settings menu gives options for updating or deleting account. | pass |
| TC\_18 | Application | 12/06/17 | Update Account | Select update account and change email or password.. | Account is successfully updated. | Account new password (4321) s saved and updated to database. | pass |
| **Test Case** | **Test**  **Environment** | **Test Date** | **Feature Tested** | **Required Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC\_19 | Application | 12/06/17 | Delete Account | Select delete account. | Account is successfully deleted and user is returned to login page.. | Account user information is deleted and redirected to login page. | pass |
| TC\_20 | Application | 12/06/17 | Sign Out | Select sign out button. | User is signed out of the app and returned to the login screen. | User signed out and returns to login screen page. | pass |
| TC\_21 | Application | 12/06/17 | Admin Login | Attempt login with registered email address and password. | Existing admin is able to login. | Feature will be implemented in the future. | fail |
| TC\_22 | Application | 12/06/17 | Admin Login | Attempt login with unregistered account or incorrect password. | Error message displayed for invalid credentials. | Feature will be implemented in the future. | fail |
| TC\_23 | Application | 12/06/17 | Admin Type | Login as with system admin account and with school admin account. | System admin and school admin will have different privileges available. | Feature will be implemented in the future. | fail |
| TC\_24 | Application | 12/06/17 | Update Questions | From systems admin menu update discovery questions. | Questions will be successfully updated. | Feature will be implemented in the future. | fail |
| TC\_25 | Application | 12/06/17 | Update Information | From systems admin menu update information in explore section. | Information page in explore successfully updated. | Feature will be implemented in the future. | fail |
| **Test Case** | **Test**  **Environment** | **Test Date** | **Feature Tested** | **Required Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC\_26 | Application | 12/06/17 | View User Info | From systems or school admin menu view student accounts and attempts. | Student accounts and discovery results displayed. | Feature will be implemented in the future. | fail |
| TC\_27 | Application | 12/06/17 | Student View | From systems or school admin menu select student view. | App allows admin account to access student actions such as explore and discovery. | Feature will be implemented in the future. | fail |
| TC\_28 | Application | 12/06/17 | Color Scheme | From school admin menu change color scheme of app. | Color scheme successfully changed. | Feature will be implemented in the future. | fail |
| TC\_29 | Application | 12/06/17 | Image Test | Access each page of the app to view embedded images. | Images displayed successfully with no delay. | Images are displayed instantly on each page. | pass |
| TC\_30 | Database | 12/06/17 | Account Creation | Create account in app and verify database transaction. | Account data is successfully added to database. | User account is created and updated to database for view. | pass |
| TC\_31 | Database | 12/06/17 | Account Deletion | Delete account in app and verify deletion from database. | Account data successfully deleted. | User account info is deleted from view in the database. | pass |
| **Test Case** | **Test**  **Environment** | **Test Date** | **Feature Tested** | **Required Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC\_32 | Database | 12/06/17 | Account Update | Update account in app and verify update in database. | Account successfully updated. | Password is saved and updated to the database | pass |
| TC\_33 | Application/Database | 12/06/17 | Transaction Performance | Create, update, or delete an account. | Database transaction should be executed with reasonable speed and efficiency. | Account information is updated within a couple seconds.Transaction is efficient and seamless between app and database. | pass |
| TC\_33 | Database | 12/06/17 | Unique ID | View accounts in database. | Each account has a unique ID generated upon creation. | Unique id generated for each user/admin account when created. | pass |
| TC\_34 | Android Studio | 12/06/17 | Error Handling | Review error handlers within app code. | Potential run-time errors are detected and/or resolved. | Duplicate quiz questions would surface when user is interacting with discovery mode but resolved. | pass |
| TC\_35 | Application | 12/06/17 | Exit Application | Close the app. | App is successfully closed and any background processes are terminated. | App closes and signs the user out terminating all background processes. | pass |

**Total defects: 1**

**Defects resolved: 1**

**Defects remaining: 0**

xxxx used as placeholder, fields will be completed upon testing

Post-Mortem

CEIT Major Selection Tool

1. Introduction

The post mortem report will give an overview of the CEIT Major Selection Tool. The report will briefly cover the overall goals of the application software. In addition to the overall progress with the project such as the aspects that went well and the challenges/issues encountered with the software, key functionality of the app, and lessons the team learned from working on the project.

2. Report Goals

The goals of the post mortem include in following:

* Review the success and failures of the project
* Problem evaluation and resolution
* The lessons the project team learned
* A brief summary detailing the project

3. Project Parameters

Project Name: CEIT Major Selection Tool

Department: Computer Science of College of Engineering and Information Technology

Project Clients: Dr. John O’Malley and Dr. Adrian Gardiner

Project Manager: Christopher Poole

Post Mortem Facilitator: Charles Eaton and Caelan Cooper

Target Completion Date: 11-24-2017

Actual Completion Date: 12-7-2017

**Project Overview**

The purpose of the CEIT Major Selection Tool is to provide incoming students, who may not have received thorough advisement about the differences between computing majors, a chance to discover which major might align with their personal interests, skills, and abilities. The objectives we wished to accomplish with this project are:

* Allow users to create an account and login/logout
* Provide users with a quiz that suggests a major they have the most affinity for
* Provide users with information about each of the computing majors
* Allow school administrators to reach out to student users via email
* Ensure account information is secure

4. Performance

**Key Accomplishments**

* Create an account and login/logout implementation was successful
* Cloud database is properly connected to the application
* Discovery quiz was implemented successfully
* Quiz questions are dynamic and change based on answers
* User can easily find information on each of the majors

**Key Issues and Challenges**

* Drafting questions for major suggestions falls more into psychology, however the group did our best to create questions that fairly represent each major
* Determining an algorithm for the quiz decision path
* Implementing administrator accounts
* Time management

5. Key Lessons Learned

**Lessons Learned**

1. Working with clients:

Prior to this project, none of the members of our group had any experience developing software for an actual client. We had each worked on multiple assignments for our college professors of course, but working alongside clients who share a stake in the project is different in many ways. For most course projects, the requirements are set from the beginning. You complete your project, turn it in, and get a grade. When working with a client however, requirements can change. Our group was lucky enough to be able to meet frequently with our clients and through the course of the semester we learned to accept changes and use feedback to adapt the project to fit new requirements. This led to better communication between the development group and the clients and allowed us to make better decisions as a whole.

2. Software engineering process

One of the main lessons we learned is how important the software engineering process is to delivering a complete and polished product. Without conducting a feasibility study, the group could have potentially ended up with a half complete project due to not properly assessing the risks. If the group had not spent significant time on the design phase we could have missed an important function of the application, requiring us to backtrack and fix our mistakes. Documenting each step of the development is painfully thorough at times, however being able to see visual representations of the system and its processes as well as being able to review our progress was critical to the success of our project.

**Future Considerations**

* Further testing to ensure quality
* Additional majors could be implemented with the addition of new questions
* Admin account functionality could be expanded

Appendix

CEIT Major Selection Tool

A. User Manual

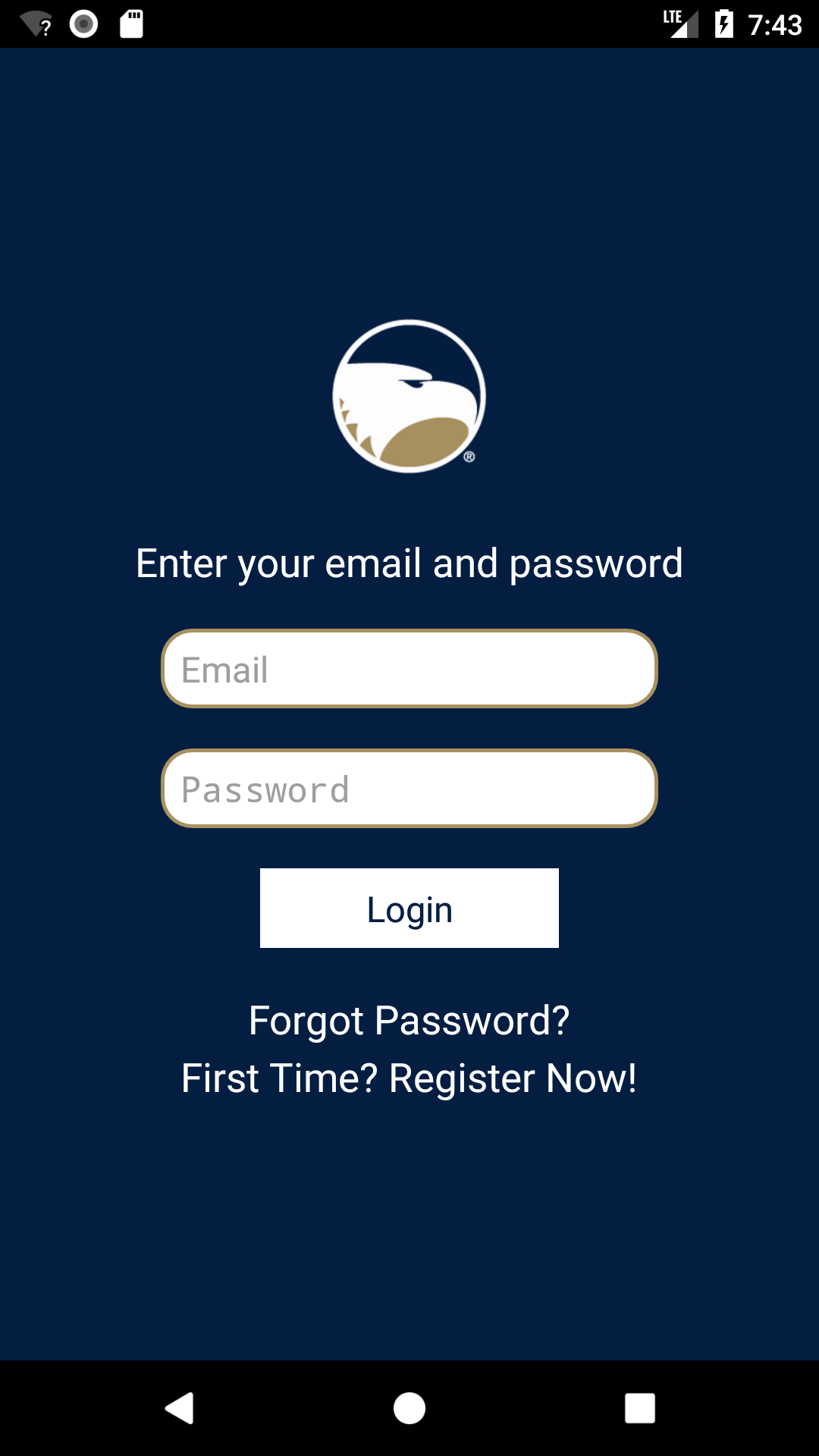
**Starting the Application**

In order to use the CEIT Major Selection Tool the user must first ensure that their device is connected to the internet. Once the application has been installed the user will simply click the icon and they should be brought to the initial splash screen:

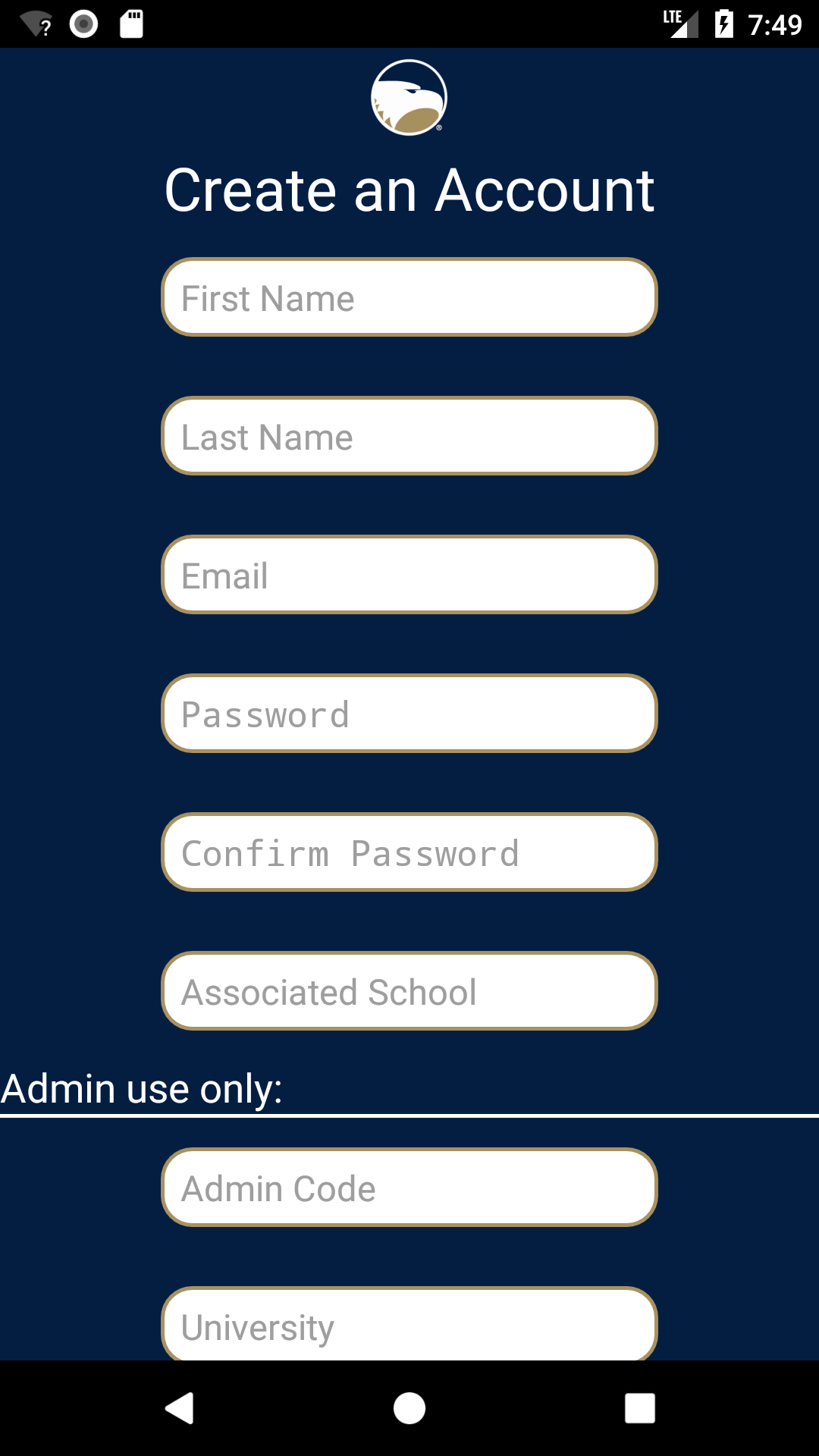


**Registering an Account**

After the splash screen the user will be brought to the login page for the application. Here the user can input their email address and password if they have already registered for an account. If not, the user will click the prompt to register:



The user will be asked to input a first name, last name, email address, password, and confirm their password. If the user is an administrator, they will enter an optional associated school and an admin code. Once these fields have been completed, the user is prompted to login with their new account.

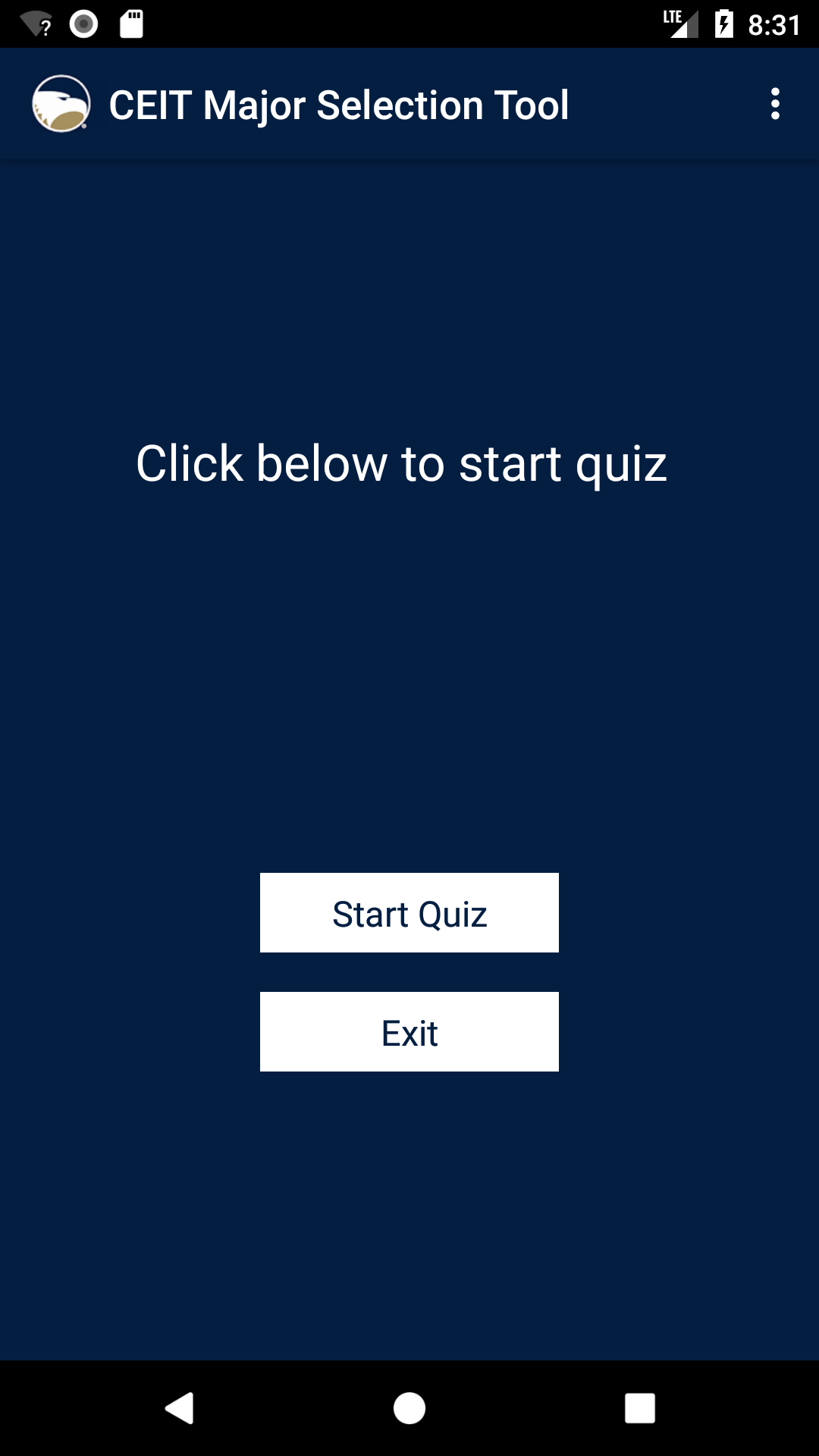


**Home Page**

Once the user has entered their login information and clicked the submit button, they will be taken to the home page of the application:

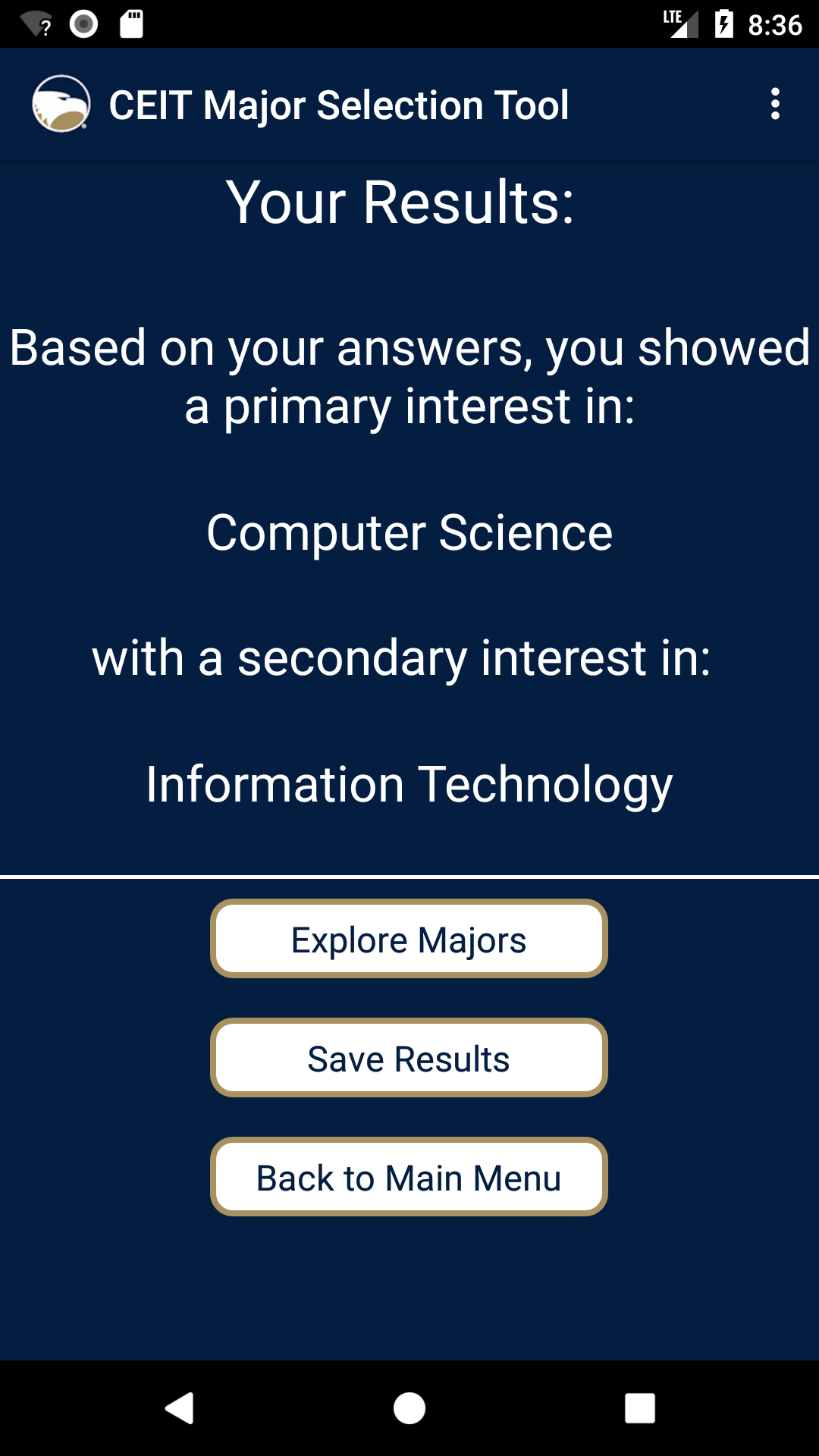


From here the user can access the main menu page and click on the Discovery button to take a short quiz to learn if their interests match with any CEIT majors. Once the user clicks on the Discovery button there are taken to the quiz page.

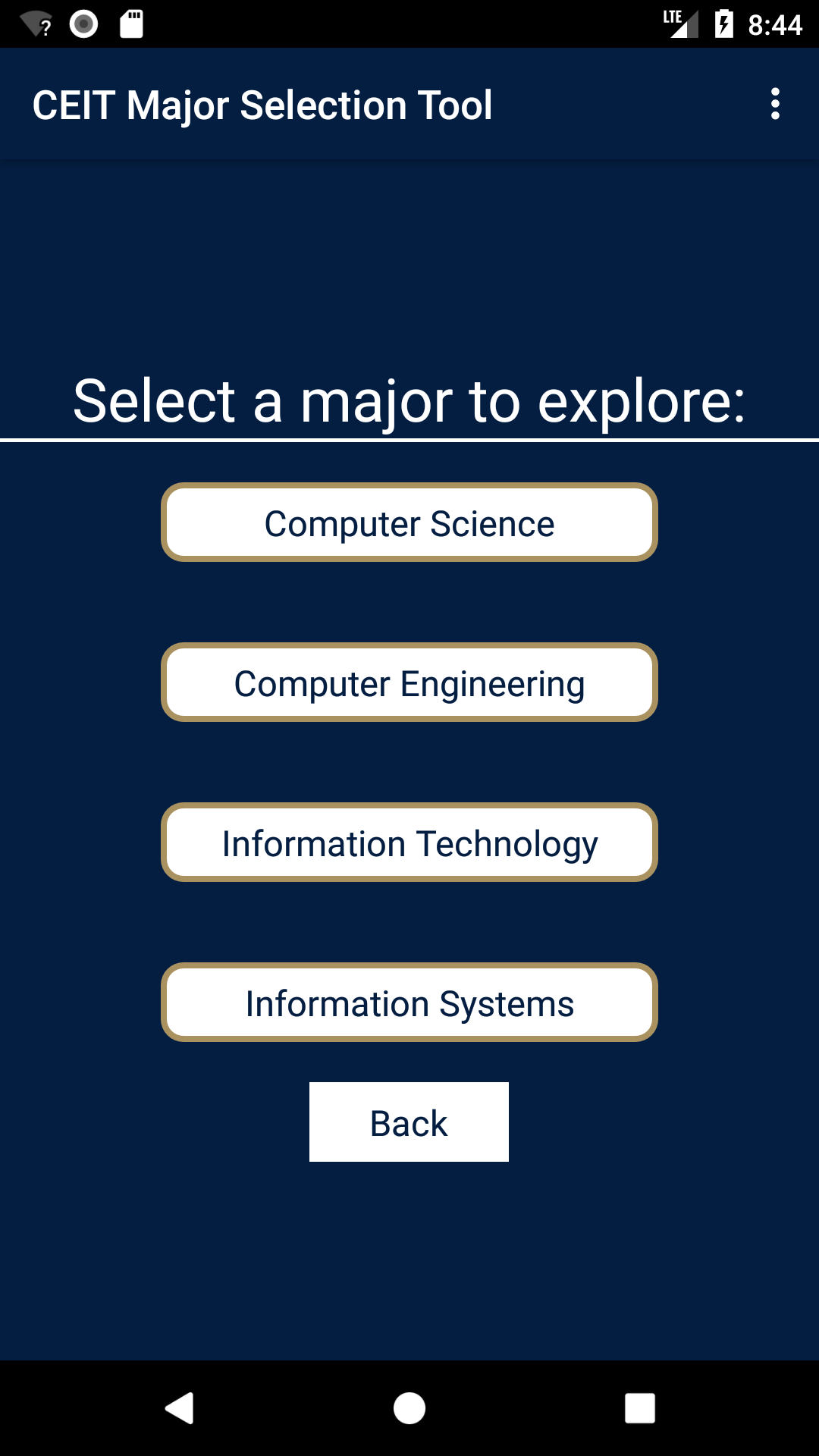


Then the user clicks on the Start Quiz button to proceed to the quiz or has the option of exiting the discovery quiz page.





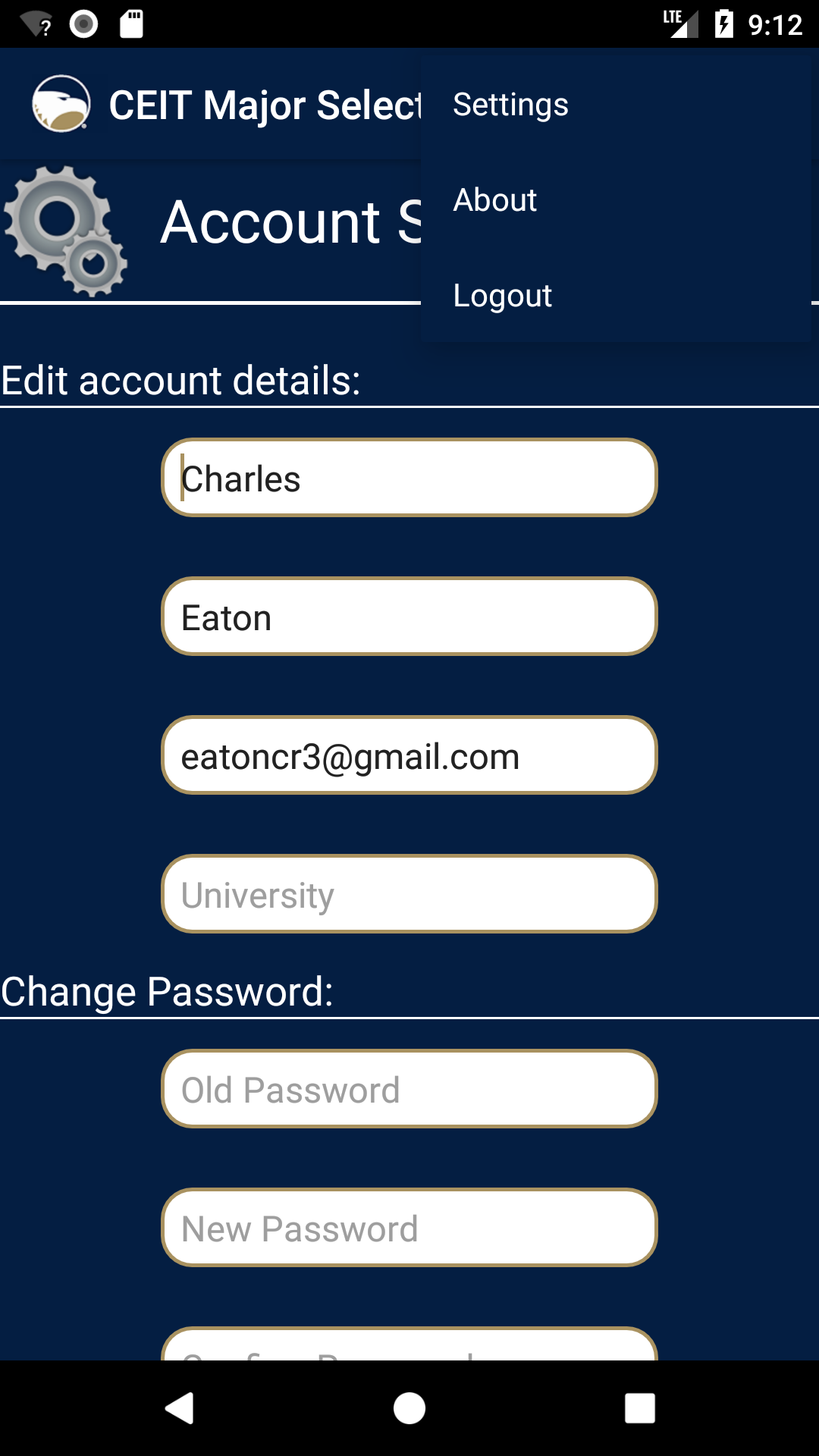
From here the user has the option of exploring more in depth relevant information about the majors that were recommended based on the quiz on the results page, saving quiz results/attempts or going back to the Main Menu. The Explore Button lists the current corresponding CEIT majors with info pages linked to the Georgia Southern website when clicked.

.

When a major button is clicked the user can view the corresponding info page detailing information about the major, possible career paths and which colleges have the major in Georgia.



In addition, the user can click on the three-dot drop down settings menu in the top right corner of the screen to view the settings page to edit account info or view the About page for app information.



B. Copyright Disclaimer

This software is an original creation and was developed by students of Georgia Southern University for Dr. Andrew Allen’s Software Engineering course. The work was a final project commissioned by Dr. John O’Malley and Dr. Adrian Gardiner and as such all intellectual property rights will belong to Dr. Allen, Dr. O’Malley, Dr. Gardiner, and Georgia Southern University.

C. Glossary

**Agile:** set of software development principles with an emphasis on collaboration, planning, and rapid delivery.

**API:** Application Programming Interface - a set of subroutines, protocols, and tools for building application  
software.

**App:** Application (In this context, a mobile application) - a computer program designed to perform a specific task.

**CE:** Computer Engineering - discipline that integrates Electrical Engineering and Computer Science to develop hardware and software.

**CEIT:** College of Engineering and Information Technology - academic department consisting of majors in computing, engineering, and technology fields.

**CS:** Computer Science - discipline that uses theory and information processing methods to design computer hardware and software.

**Erwin:** Data modeling software developed by Erwin Inc. that includes tools for transactional systems and data marts.

**GUI:** Graphical User Interface - visual component that allows users to interact with an electronic device or program.

**HTTP:** HyperText Transfer Protocol - underlying protocol for the entire World Wide Web that defines how messages are formatted and transmitted and what actions servers and browsers  
should take.

**IDE:** Integrated Development Environment - software that provides visual environment for programming and debugging.

**IS:** Information Systems - discipline concerned with the study of how hardware and software solutions are utilized by organizations such as businesses.

**IT:** Information Technology - discipline concerned with the application of hardware, software, and networks, usually within the context of a business.

**PHP**: Hypertext Preprocessor - general purpose scripting language used for dynamic web content and interactions with databases.

**Trello:** project management system developed by Atlassian that allows seamless collaboration and assignment of tasks.

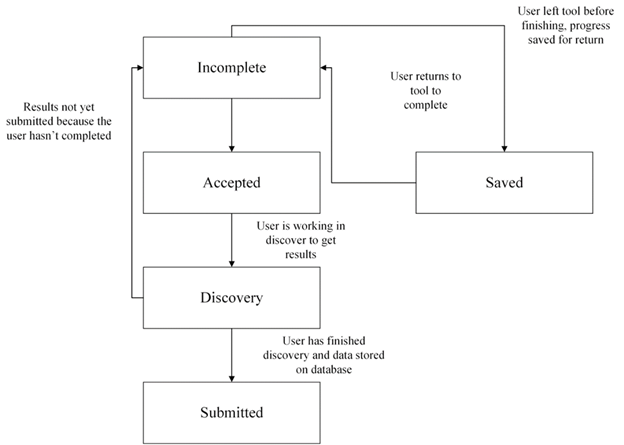
**UML:** Unified Modeling Language - visual modeling language used in the architecture, design, and implementation of complex software systems.

**Use Case:** in software design, example of a situation in which a user interacts with a program.

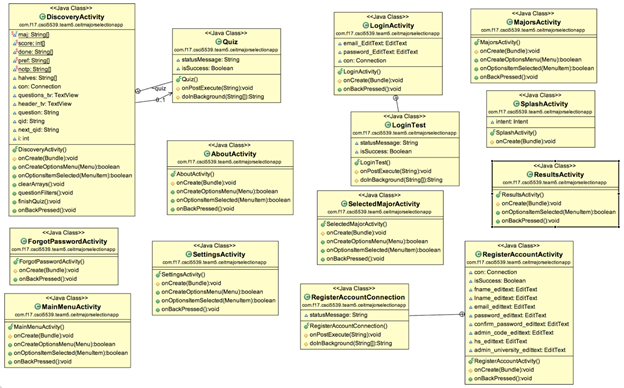
**XML:** Extensible Markup Language - language used to standardize layout in certain applications.

D. Diagrams

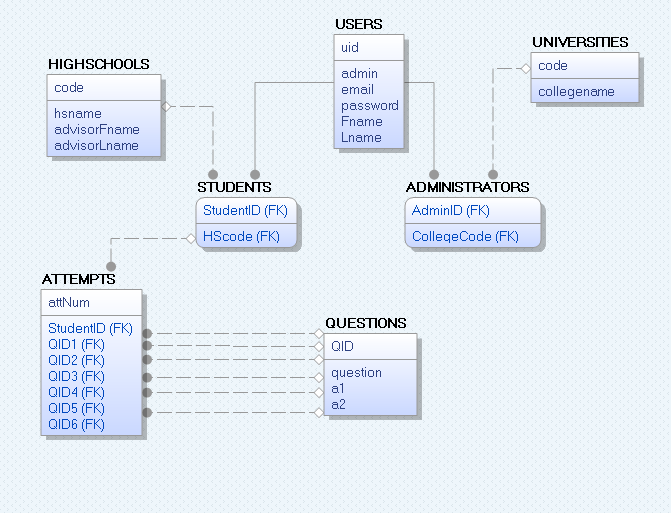
**1. Analysis Model**

****

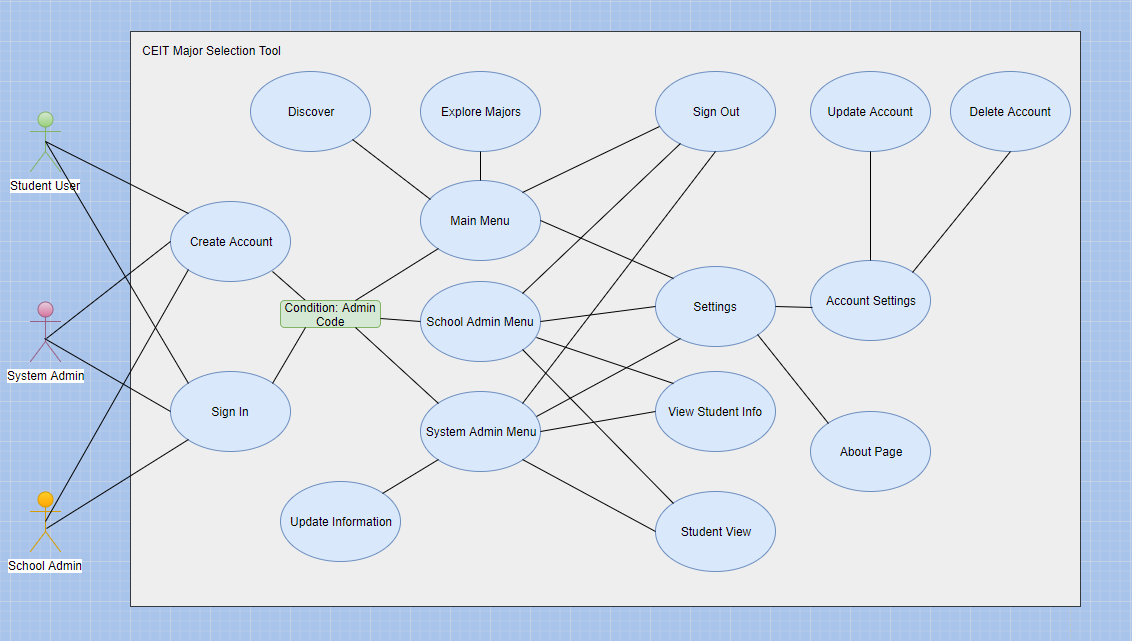
**2. Detailed Class Diagram**

****

**3. Database ER Diagram**

****

**4. Use Case Diagram**

****

E. Supplementary Materials

**1. Android Studio Study**

**Overview:**

Android Studio is Google’s official IDE for developing android applications and includes tools for editing code as well as design tools for creating a user interface. Our team chose to use Android Studio as our main development tool for several reasons. Some of these reasons include its ease of use, flexibility, and Android-specific features. Additionally, when it comes to Android programming our team has the most experience with this piece of software and the fact that it is available for free with continuous updates and support from Google further establishes it as the best option for this particular project.

**Features:**

* Gradle-based build support
* Android-specific refactoring
* Syntax highlighting with detailed annotations
* Tools to catch version compatibility, usability, and performance problems
* Layout editor with drag-and-drop UI components
* Emulator for application preview
* Google Cloud Platform Integration

**Version:**

Our team will be using Android Studio version 2.3, the latest official release of the software. A new version, 3.0, is currently in beta however the additional features do not warrant the use of experimental software for this particular project.

**2. Requirements Statement**

By Dr. John O’Malley and Dr. Adrian Gardiner

An Android application will be developed with the objective of being deployed for the use of Georgia high school students to use in order to be informed about or discover information relating to specific programs at Georgia Southern University namely the Bachelor of Science in Computer Science, Bachelor of Business Administration in Information Systems, Bachelor of Science in Computer Engineering, and the Bachelor of Science in Information Technology.   
  
The Android app will be cloud-based with the client installed on a smartphone and persistent data stored in the cloud within a database. When the user starts the app they will be presented with a screen to allow them to create an account or to resume a previously started session. Once they have signed into their account they will be presented with a preliminary menu of options that focus on the two main objectives of the application, namely gathering information about a specific program of study (both generalized and specific to GSU), or discovering which program offered at Georgia Southern is perhaps the most applicable given their talents, interests and future plans. The discovery part of the application will be the most challenging, with the user being presented with a series of preliminary questions, and based on their responses to these questions, a recommendation for a major(s) will be provided together with the rationale for making that recommendation. The app should also link this recommendation directly with the applicable Georgia Southern resources such as admission, department and program websites. Furthermore, the app will have a clean consistent brand throughout the screens.  
  
The app will need to include a privacy statement that will need to be agreed to prior to using the app. The confirmation information including but not limited to date, time, and user will be stored in the database. In addition, the app will allow users to delete the app and also delete the account information. Finally, security will be included in the systems (app, data communication, middleware, and backend including the database.

**3. Discovery Quiz Questions**

What interests you more?  
 (IS vs. IT)  
A) Designing the next version of a marketing and sales computer system  
B) Ensuring the computer network this application runs on is secure and always available  
  
 (IS vs. CS)  
A) Designing the next version of a computer system to be used by office workers for hiring employees  
B) Writing the computer code needed to implement this application  
  
(IS vs. CE)  
A) Working with top-level business executives on a plan to safeguard important files  
B) Creating a program that identifies security breaches within the business network  
  
(IT vs. IS)  
A) Installing servers to meet the needs of a growing business  
B) Designing a plan to ensure these servers operate at peak efficiency  
  
(IT vs. CE)  
A) Setting up and configuring a web server to enable people to share photos  
B) Designing the circuitry of the next smartphone camera to capture even better quality photos  
  
 (IT vs. CS)  
A) Installing and configuring networking equipment to allow wireless communication  
B) Using a programming language to create a user-interface for managing the wireless network  
  
(CS vs. CE)  
A) Using a programming language to create new video games  
B) Designing a new type of video card to run complex games faster  
  
(CS vs. IS)  
A) Programming the software for a point-of-sale system to be used in a retail store  
B) Analyzing the system to ensure it is user-friendly and accurate  
  
(CS vs. IT)  
A) Creating the next version of a computer operating system like Windows (Microsoft) or iOS (Apple)  
B) Ensuring the computer operating system is properly working on employees’ computers, and has the latest security updates  
  
(CE vs. IT)  
A) Design software for a heart rate monitor to be used in a hospital  
B) Perform repairs and maintenance on the monitor to keep it functioning  
  
(CE vs. CS)  
A) Developing a motion sensor for a home security system  
B) Programming the software that connects this sensor to a camera  
  
(CE vs. IS)  
A) Designing the hardware components for a new virtual reality system  
B) Creating a plan for the system to be used in schools as an educational tool

**4. Explore Information Text**

Computer Science  
Description:  
 Ranges from theoretical foundations to cutting-edge developments  
 Develop effective ways to solve computing problems  
 Devise new ways to use computers  
 Designing and implementing software  
  
Possible Career Paths:  
Software Developer  
Computer Programmer   
Systems Analyst  
Database Administrator   
Video Game Designer  
  
Colleges:  
Georgia Southern University, Berry College, Clark Atlanta University, Emory University, Fort Valley State University, Georgia Institute of Technology, Georgia Southwestern State University, Georgia State University, Kennesaw State University, Mercer University, Paine College, Piedmont College, Southern Polytechnic State University, University of Georgia, University of West Georgia

Computer Engineering   
Description:  
 Design and construction of computers, and computer based systems.  
 Design of digital hardware/software systems  
 Development of devices that have embedded systems  
 Integration of hardware and software  
  
Possible Career Paths:  
Computer Hardware Engineer  
Computer Systems Designer  
Software Engineer  
Network Architect  
Electronics Engineer  
  
Colleges:   
Georgia Southern University, Albany Technical College, Dalton State College, DeVry University, Georgia Institute of Technology, Georgia Piedmont Technical College, Mercer University, Omnitech Institute, Savannah State University, Southern Polytechnic State University, University of Georgia  
  
  
Information Technology   
Description:  
 Combination of knowledge and practical applications with hands-on expertise  
 Maintain an organization’s information technology structure  
 Installation and maintenance of computer systems  
 Current emphasis on networks  
  
Possible Career Paths:  
Network Administrator   
Web Developer  
IT Security Specialist  
Database Administrator   
IT Applications Designer  
  
Colleges:   
Georgia Southern University, Abraham Baldwin Agricultural College, Albany State University, American Intercontinental University, Armstrong Atlantic State University, Augusta State University, Brewton-Parker College, Clark Atlanta University, Clayton State University, College of Coastal Georgia, Columbus State University, Covenant College, Darton College, Fort Valley State University, Georgia College and State University, Georgia Gwinnett College, Georgia Institute of Technology, Georgia Military College, Georgia Southwestern State University, lifeGeorgia State University, Herzing University, Interactive College of Technology, Kennesaw State, LaGrange College, Life University, Lincoln College of Technology, Macon State College, Mercer University, Morehouse College, North Georgia College and State University, omnitech Institute, Savannah State University, Shorter University, South Georgia College, South University, Southern Polytechnic State University, Spelman College, University of Phoenix, University of West Georgia, Valdosta State University, WEsleyan College, Westwood College,   
  
  
Information Systems   
Description:  
 Satisfy informational needs of businesses and organizations  
 Emphasis on information rather than technology  
 Concerned with information provided by computer systems  
 Determines requirements and design of an organization’s information systems  
  
Possible Career Paths:  
Business Analyst  
IS Manager  
Systems Administrator   
Data Analyst  
Cloud Architect  
  
  
Colleges:  
Georgia Southern University, Georgia Institute of Technology, Georgia State university, Morehouse College, mercer University, Valdosta State University, University of West Georgia, Columbus State University, Clayton State University, Clark Atlanta University

**5. Weekly Reports**

**Weeks 1-2**

The class was still getting started and we had not been assigned our groups at this point in time.

**Week 3**

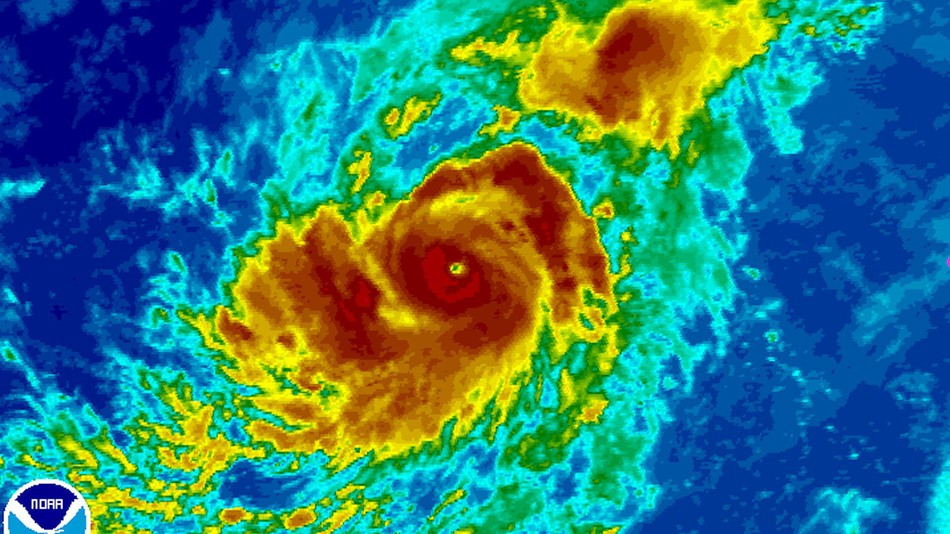
The group had 2 meetings during this week. The first meeting (Mon, 28 Aug) was only for the group. We discussed the potential layout of the project and group structure but decided to leave definite decisions until after we had met with our clients. The group leader set up a meeting with the clients via email for that Wednesday. Additionally, we discussed our class schedules to find the best time to have out entire group meet with the clients at the same time. Finally, we set up a Google Drive folder for the group.

The second meeting (Wed, 30 Aug) was our initial meeting with the clients. Unfortunately, the best meeting time had yet to be worked out and only Caelan, Amy, and Chris were able to attend this meeting. During this meeting, we discussed the problem our group was trying to solve and the scope of the project itself, as well as platforms for us to develop on. Towards the end, our clients got off topic and began giving us career advice. The clients’ first requests for the group were to have a copy of the syllabus and a set time to have a weekly meeting. We emailed them a copy of the syllabus and set the weekly meeting to be on Monday at 1030.

**Week 4**

This week we stayed after class on Tuesday (5 Sep) to discuss the feasibility study. We were concerned about whether or not we had enough information to complete the report and decided to ask for all of the required information during our next meeting (Fri, 8 Sep). Thanks to hurricane Irma, that meeting never happened. The group decided we would continue to research the computing majors offered at Georgia Southern for the time being.

**Week 5**

****

We spent the week doing research at home while we waited out the hurricane.

**Week 6**

This week we talked with you (Tues, 19 Sep) after class about the lack of information given to us by our clients. We decided that the feasibility report would be due later in the week.

The next day, we all met with the and discussed design thinking. Shortly after it started, you joined us and stressed that the design should be fed to us. We worked out a target audience and established that the application will need to be expandable. We decide that the platform for the app will be Android and work out some of the other requirements for the application. At the end we are left with an article and a webpage related to our project that we had to look over and critique for class.

On Thursday and Friday (21-22 Sep), we spent time discussing potential GUI prototypes and working on the feasibility report.

**Week 7**

The next client meeting (Wed, 27 Sep) provides us with more requirements (namely a reporting system to track demographics and preferences) and allows us to present some of the mockups we have. There is some debate about whether we should continue with an application or switch to a website design, but we decided to continue with the application. The clients stress the importance of persistence and flexibility and the meeting ends. During the week, we message each other and decide on individual roles within our group.

**Week 8**

This week’s client meeting (Mon, 2 Oct) focused on content in the app. The clients requested a document explaining the features and capabilities of Android Studios (which we used to develop the application) and then stress the importance of being able to connect a “full database” to the app. They also inform us that the focus of the questions and information should be on the majors CS, SE, CE, IS, and IT. By the next week, they want us to have a couple of questions that could be implemented in the app and have a Google Drive folder shared with them as well.

That Thursday (5 Oct) the group discussed upcoming tasks. Knowing what needed to be completed, we split up and started working on the tasks the group leader assigned to everyone throughout the week.

**Week 9**

The meeting (Mon, 9 Oct) starts with a presentation of the questions. The clients believe they might be too complicated and ask us to rethink them. Then we presented the wireframe of the GUI we developed. We also discussed the potential for branding the application so that it may be used by different schools. For the next meeting, the clients’ wanted to see new questions, an improved GUI wireframe, and a database schema.

That Friday (13 Oct), we met to finish the SRS document and worked into the next morning to get it complete.

**Week 10**

The meeting was a short one this week (Mon, 16 Oct). We presented the updated GUI and database schema to the clients and took feed back from them. During the week, we work individually to update the GUI, database, and questions.

**Week 11**

In this meeting (Mon, 23 Oct), we present more questions that are just not right for our clients. We are told to revise them some more and think about alternate ways of scoring them. We inform them that the database is up and running, hosted by Microsoft Azure.

**Week 12**

The client meeting for this week was also very short (Mon, 30 Oct). We update the clients on the questions and they provide us with another requirement for the application.

The next day, we met with you and spent some time discussing the scoring system for the questions. We then began looking at alternate ways to have the questions displayed and scored.

That Friday (3 Nov), the group meets with Bridges Adams. We asked him if he would be able to provide us with questions related to the different majors and he informed us that he had them in his office. He would look through them and send his best ones to us during the next week.

**Week 13**

Another short meeting (Mon, 6 Nov), the clients stress that there are only 3 weeks left of class in the semester. We inform them that Bridges Adams will be getting the questions to us later in the week and that we should have the questions implemented in the app by the next meeting. We plan to demonstrate the questions on either the next Monday or Wednesday.

Later in the week, we have trouble getting in contact with Bridges about the questions. He requests to meet one of the group members on Thursday or Friday but then stops answering his email and his phone. We inform the clients that we will have to do the demo on Wednesday as we are having communication issues with Bridges.

**Week 14**

On Monday (13 Nov) the group leader received an email from Bridges saying that providing us with these questions was not his job and that he would not do it. This was also the day we had previously set as the very latest we could take the questions from him.

Because of that, our meeting with the clients (Wed, 15 Nov) was short and nearly useless. We had no questions to demonstrate, but we were able to show the connection to the database. We demonstrated the process to create an account and login on the application, and they reminded us that users should be able to delete their accounts. We also discussed trying to demonstrate the questions over Thanksgiving break and decided on sending screenshots of the quiz Wednesday right before thanksgiving.

Since we had such trouble with the questions, the group leader decided to do the quiz in a very simple manner. When this was brought up in class on Thursday (16 Nov), you politely informed us that we would fail with going down that route. After class, we stayed behind and worked on a sufficiently complex algorithm for displaying the quiz. We presented it to you and began implementing it the next day.

**Week 15**

Over Thanksgiving break, we each had our hands full with either some form of code or some form of documentation. We didn’t meet, but worked together online to get our tasks completed. We sent a working prototype of the quiz to you as well as our clients.

**Week 16**

This Monday (27 Nov), we met with our clients and demonstrated the nearly finished application in person. While going through the quiz, we found a bug where it will sometimes display the same question twice and began to immediately try to fix it. We decided we would have our final meeting with the clients in the next week where we would hand over the last of our documentation and code.

During this week, we began to design and practice our presentation for the final presentations that Friday. We met on Tuesday, Wednesday, Thursday, and Friday morning (28 Nov-1 Dec) to prepare for the presentation and ensure we had all of our slides up and accurate.

F. Source Code

**XMLS:**

activity\_splash.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:gravity="center"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**ImageView**  **android:layout\_width="150dp"**  **android:layout\_height="150dp"**  **android:src="@drawable/gsulogo"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/name"**  **android:textColor="@android:color/white"**  **android:textSize="27sp"** />  </**LinearLayout**> |

activity\_login.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:gravity="center"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**ImageView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:src="@drawable/gsulogo"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="\nEnter your email and password"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**EditText**  **android:id="@+id/email\_edit\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="20dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Email"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/password\_edit\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Password"**  **android:inputType="textPassword"**  **android:paddingLeft="10dp"** />  <**Button**  **android:id="@+id/login\_btn"**  **android:layout\_width="150dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:text="Login"** />  <**TextView**  **android:id="@+id/forgot\_password\_text\_view"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="2dp"**  **android:text="Forgot Password?"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**TextView**  **android:id="@+id/register\_text\_view"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="First Time? Register Now!"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  </**LinearLayout**> |

activity\_forgot\_password.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:gravity="center"**  **android:orientation="vertical"**>  <**ImageView**  **android:layout\_width="250dp"**  **android:layout\_height="150dp"**  **android:src="@drawable/eagle\_forgot"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="55dp"**  **android:text="Forgot Password?\n"**  **android:textAlignment="center"**  **android:textColor="@android:color/white"**  **android:textSize="35sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Enter your email below so we can send you your password."**  **android:textAlignment="center"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**EditText**  **android:id="@+id/forgot\_email\_edit\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="20dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Enter Email"**  **android:paddingLeft="10dp"** />  <**Button**  **android:id="@+id/send\_email\_pw\_btn"**  **android:layout\_width="155dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:text="Send Password"** />  <**TextView**  **android:id="@+id/remember\_password\_text\_view"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Remember Password? Login here"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**TextView**  **android:id="@+id/fp\_register\_text\_view"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="First Time? Register Now!"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  </**LinearLayout**> |

activity\_register\_account.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:layout\_gravity="center"**  **android:fillViewport="true"**>  <**LinearLayout**  **android:layout\_width="match\_parent"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**ImageView**  **android:layout\_width="50dp"**  **android:layout\_height="50dp"**  **android:src="@drawable/gsulogo"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Create an Account"**  **android:textColor="@android:color/white"**  **android:textSize="30sp"** />  <**EditText**  **android:id="@+id/reg\_fname\_edit\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="First Name"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/reg\_lname\_edit\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Last Name"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/reg\_email\_edit\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Email"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/reg\_password\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Password"**  **android:inputType="textPassword"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/reg\_confirm\_password\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Confirm Password"**  **android:inputType="textPassword"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/reg\_high\_school\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="School - Optional"**  **android:paddingLeft="10dp"** />  *<!--*  *<TextView*  *android:layout\_width="wrap\_content"*  *android:layout\_height="wrap\_content"*  *android:layout\_gravity="left"*  *android:text="Admin use only:"*  *android:textColor="@android:color/white"*  *android:textSize="20sp" />*  *<View*  *android:layout\_width="fill\_parent"*  *android:layout\_height="2dip"*  *android:background="#FFFFFF" />*  *<EditText*  *android:id="@+id/reg\_admin\_code\_text"*  *android:layout\_width="250dp"*  *android:layout\_height="40dp"*  *android:layout\_margin="*  *android:background="@drawable/edit\_text\_style"*  *android:hint="Admin Code"*  *android:paddingLeft="10dp" />*  *<EditText*  *android:id="@+id/reg\_university\_text"*  *android:layout\_width="250dp"*  *android:layout\_height="40dp"*  *android:layout\_margin="15dp"*  *android:background="@drawable/edit\_text\_style"*  *android:hint="University"*  *android:paddingLeft="10dp" /> -->*  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**Button**  **android:id="@+id/register\_btn"**  **android:layout\_width="200dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:text="Register"** />  <**TextView**  **android:id="@+id/login\_exists\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Already have an account? Login"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  </**LinearLayout**>  </**ScrollView**> |

activity\_main\_menu.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:layout\_gravity="center"**  **android:fillViewport="true"**>  <**LinearLayout**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:gravity="center"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**TextView**  **android:id="@+id/welcome\_name"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Welcome:---"**  **android:textColor="@android:color/white"**  **android:textSize="40dp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**ImageView**  **android:layout\_width="400dp"**  **android:layout\_height="150dp"**  **android:src="@drawable/pexels\_photo"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/CEIT\_welcome"**  **android:textColor="@android:color/white"**  **android:textSize="18sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="\nSelect an option below:"**  **android:textColor="@android:color/white"**  **android:textSize="25sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/CEIT\_discover"**  **android:textColor="@android:color/white"**  **android:textSize="18dp"** />  <**LinearLayout**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:orientation="horizontal"**>  <**ImageView**  **android:layout\_width="50dp"**  **android:layout\_height="50dp"**  **android:src="@drawable/quiz\_icon"** />  <**Button**  **android:id="@+id/discover\_btn"**  **android:layout\_width="200dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="Discover"** />  </**LinearLayout**>  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/CEIT\_explore"**  **android:textColor="@android:color/white"**  **android:textSize="18dp"** />  <**LinearLayout**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:orientation="horizontal"**>  <**ImageView**  **android:layout\_width="50dp"**  **android:layout\_height="50dp"**  **android:src="@drawable/explore\_icon"** />  <**Button**  **android:id="@+id/major\_info\_btn"**  **android:layout\_width="200dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="Explore Majors"** />  </**LinearLayout**>  </**LinearLayout**>  </**ScrollView**> |

activity\_discovery.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:gravity="center"**  **android:orientation="vertical"**>  <**TextView**  **android:id="@+id/header\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:text="Click below to start quiz\n"**  **android:textSize="25sp"** />  <**TextView**  **android:id="@+id/question\_textview"**  **android:layout\_width="350dp"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:text="------------------------"**  **android:textSize="15sp"**  **android:visibility="invisible"** />  <**LinearLayout**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:orientation="horizontal"**>  <**Button**  **android:id="@+id/a\_btn"**  **android:layout\_width="75dp"**  **android:layout\_height="50dp"**  **android:layout\_margin="40dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="A"**  **android:visibility="invisible"** />  <**Button**  **android:id="@+id/b\_btn"**  **android:layout\_width="75dp"**  **android:layout\_height="50dp"**  **android:layout\_margin="40dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="B"**  **android:visibility="invisible"** />  </**LinearLayout**>  <**Button**  **android:id="@+id/start\_btn"**  **android:layout\_width="150dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:text="Start Quiz"** />  <**Button**  **android:id="@+id/back\_btn"**  **android:layout\_width="150dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:text="Exit"** />  </**LinearLayout**> |

activity\_results.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:layout\_gravity="center"**  **android:fillViewport="true"**>  <**LinearLayout**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:orientation="vertical"**>  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Your Results:\n"**  **android:textSize="30sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:text="Based on your answers, you showed a primary interest in:\n"**  **android:textSize="25sp"** />  <**TextView**  **android:id="@+id/results\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:text="-----------"**  **android:textSize="25sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="\nwith a secondary interest in:\n"**  **android:textSize="25sp"** />  <**TextView**  **android:id="@+id/results\_sec\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_marginBottom="30dp"**  **android:gravity="center"**  **android:text="-----------"**  **android:textSize="25sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**Button**  **android:id="@+id/prim\_major\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="---"** />  <**Button**  **android:id="@+id/sec\_major\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="---"** />  <**Button**  **android:id="@+id/major\_info\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="Explore All Majors"** />  <**Button**  **android:id="@+id/save\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="Save Results"** />  <**Button**  **android:id="@+id/main\_menu\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="10dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="Back to Main Menu"** />  </**LinearLayout**>  </**ScrollView**> |

activity\_majors.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:gravity="center"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Select a major to explore:"**  **android:textAlignment="center"**  **android:textColor="@android:color/white"**  **android:textSize="30sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**Button**  **android:id="@+id/CS\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="@string/cs"** />  <**Button**  **android:id="@+id/CE\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="@string/ce"** />  <**Button**  **android:id="@+id/IT\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="@string/it"** />  <**Button**  **android:id="@+id/IS\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:text="@string/is"** />  <**Button**  **android:id="@+id/back\_btn"**  **android:layout\_width="100dp"**  **android:layout\_height="wrap\_content"**  **android:text="@string/back"** />  </**LinearLayout**> |

activity\_selected\_major.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:layout\_gravity="center"**  **android:fillViewport="true"**>  <**LinearLayout**  **android:layout\_width="match\_parent"**  **android:layout\_height="wrap\_content"**  **android:gravity="start"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**TextView**  **android:id="@+id/major\_name\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="center"**  **android:gravity="center"**  **android:text="Selected major name here"**  **android:textSize="35sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**ImageView**  **android:id="@+id/selected\_major\_image\_view"**  **android:layout\_width="285dp"**  **android:layout\_height="200dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="10dp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="\nDescription:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:id="@+id/major\_info\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="------------"**  **android:textSize="14sp"** />  <**TextView**  **android:id="@+id/link\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="------------"**  **android:textSize="14sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="\nPossible Careers:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:id="@+id/major\_careers\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="center"**  **android:gravity="center"**  **android:text="------------"**  **android:textSize="14sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="\nDegree Also Available at:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:id="@+id/colleges\_with\_major\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="------------"**  **android:textSize="14sp"** />  <**Button**  **android:id="@+id/back\_btn"**  **android:layout\_width="200dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="20dp"**  **android:text="@string/back"** />  </**LinearLayout**>  </**ScrollView**> |

activity\_about.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:layout\_gravity="center"**  **android:fillViewport="true"**>  <**LinearLayout**  **android:layout\_width="match\_parent"**  **android:layout\_height="wrap\_content"**  **android:gravity="center"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="About Page"**  **android:textColor="@android:color/white"**  **android:textSize="35sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**ImageView**  **android:layout\_width="250dp"**  **android:layout\_height="170dp"**  **android:layout\_margin="10dp"**  **android:background="@drawable/it\_walkway"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="General:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:id="@+id/about\_general\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/about\_general"**  **android:textSize="14sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="\nGuide:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/about\_guide"**  **android:textSize="14sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="\nClients:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:id="@+id/about\_clients\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/about\_clients"**  **android:textSize="14sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="Development team:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:id="@+id/about\_dev\_team\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/about\_dev\_team"**  **android:textSize="14sp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="start"**  **android:text="Copyright:"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:id="@+id/about\_copyright\_tv"**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="@string/about\_copyright"**  **android:textSize="14sp"** />  <**Button**  **android:id="@+id/back\_btn"**  **android:layout\_width="200dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_margin="20dp"**  **android:text="@string/back"** />  </**LinearLayout**>  </**ScrollView**> |

activity\_settings.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>*  <**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"**  **xmlns:tools="http://schemas.android.com/tools"**  **android:layout\_width="match\_parent"**  **android:layout\_height="match\_parent"**  **android:layout\_gravity="center"**  **android:fillViewport="true"**>  <**LinearLayout**  **android:layout\_width="match\_parent"**  **android:layout\_height="wrap\_content"**  **android:orientation="vertical"**  **tools:context="com.f17.csci5539.team5.ceitmajorselectionapp.MainMenuActivity"**>  <**LinearLayout**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:orientation="horizontal"**>  <**ImageView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:src="@drawable/settings\_icon"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="55sp"**  **android:paddingLeft="15dp"**  **android:paddingTop="10dp"**  **android:text="Account Settings"**  **android:textColor="@android:color/white"**  **android:textSize="30sp"** />  </**LinearLayout**>  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="\nEdit account details:"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Click and update any info you would like to change"**  **android:textColor="@android:color/white"**  **android:textSize="15sp"** />  <**EditText**  **android:id="@+id/update\_fname\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="First Name"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/update\_lname\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Last Name"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/update\_email\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Email"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/update\_hs\_uni\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="High School/University"**  **android:paddingLeft="10dp"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Change Password:"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Please input current password to save changes. If you want to change your password too, fill in the new password fields"**  **android:textColor="@android:color/white"**  **android:textSize="15sp"** />  <**EditText**  **android:id="@+id/old\_password\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Password"**  **android:inputType="textPassword"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/new\_password\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="New Password"**  **android:inputType="textPassword"**  **android:paddingLeft="10dp"** />  <**EditText**  **android:id="@+id/confirm\_new\_password\_text"**  **android:layout\_width="250dp"**  **android:layout\_height="40dp"**  **android:layout\_gravity="center"**  **android:layout\_margin="15dp"**  **android:background="@drawable/edit\_text\_style"**  **android:hint="Confirm Password"**  **android:inputType="textPassword"**  **android:paddingLeft="10dp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**Button**  **android:id="@+id/save\_btn"**  **android:layout\_width="100dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="center"**  **android:layout\_margin="10dp"**  **android:gravity="center"**  **android:text="Save"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="\nView Previous Attempts:"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**Button**  **android:id="@+id/attempt\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="center"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:gravity="center"**  **android:text="View Attempts"** />  <**TextView**  **android:layout\_width="wrap\_content"**  **android:layout\_height="wrap\_content"**  **android:text="Delete Option:"**  **android:textColor="@android:color/white"**  **android:textSize="20sp"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="1dip"**  **android:background="#FFFFFF"** />  <**Button**  **android:id="@+id/delete\_btn"**  **android:layout\_width="250dp"**  **android:layout\_height="wrap\_content"**  **android:layout\_gravity="center"**  **android:layout\_margin="20dp"**  **android:background="@drawable/button\_layout\_style"**  **android:gravity="center"**  **android:text="Delete Account"** />  <**View**  **android:layout\_width="fill\_parent"**  **android:layout\_height="2dip"**  **android:background="#FFFFFF"** />  </**LinearLayout**>  </**ScrollView**> |

**Classes:**

SQLConnection.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.annotation.SuppressLint;  **import** android.os.StrictMode;  **import** android.util.Log;  **import** java.sql.Connection;  **import** java.sql.DriverManager;  **import** java.sql.SQLException;  **public class** SQLConnection {  *//DATABASE CONNECTION METHOD*  **public static** Connection connectionclass() {  StrictMode.ThreadPolicy policy = **new** StrictMode.ThreadPolicy.Builder().permitAll().build();  StrictMode.*setThreadPolicy*(policy);  Connection connection = **null**;  **try** {  Class.*forName*(**"net.sourceforge.jtds.jdbc.Driver"**);  String hostName = **"ceit-major-selection.database.windows.net"**;  String dbName = **"CEIT\_APP"**;  String user = **"ceit\_admin"**;  String password = **"Softteam5"**;  String url = String.*format*(**"jdbc:jtds:sqlserver://%s:1433/%s;user=%s;password=%s;"**, hostName, dbName, user, password);  Log.*d*(**"URL: "**, url);  connection = DriverManager.*getConnection*(url);  } **catch** (SQLException se) {  Log.*e*(**"Error 1: "**, se.getMessage());  } **catch** (ClassNotFoundException e) {  Log.*e*(**"Error 2: "**, e.getMessage());  } **catch** (Exception e) {  Log.*e*(**"Error 3: "**, e.getMessage());  }  **return** connection;  }  } |

SplashActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.content.Intent;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** java.util.Timer;  **import** java.util.TimerTask;  **public class** SplashActivity **extends** AppCompatActivity {  Intent **intent**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_splash***);  **intent** = **new** Intent(**this**, LoginActivity.**class**);  **new** Timer().schedule(**new** TimerTask() {  @Override  **public void** run() {  *// buffer*  startActivity(**intent**);  finish();  }  }, 2000);  }  } |

LoginActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.annotation.SuppressLint;  **import** android.content.Intent;  **import** android.content.SharedPreferences;  **import** android.os.AsyncTask;  **import** android.os.StrictMode;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.util.Log;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.EditText;  **import** android.widget.TextView;  **import** android.widget.Toast;  **import** java.sql.Connection;  **import** java.sql.DriverManager;  **import** java.sql.PreparedStatement;  **import** java.sql.ResultSet;  **import** java.sql.SQLException;  **import** java.sql.Statement;  **public class** LoginActivity **extends** AppCompatActivity {  EditText **email\_EditText**;  EditText **password\_EditText**;  Connection **con**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_login***);  **email\_EditText** = (EditText) findViewById(R.id.***email\_edit\_text***);  **password\_EditText** = (EditText) findViewById(R.id.***password\_edit\_text***);  *//LOGIN BUTTON LISTENER*  Button login\_btn = (Button) findViewById(R.id.***login\_btn***);  login\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *// if (email\_EditText.getText().toString().equals("") || password\_EditText.getText().toString().equals("")) {*  *// Toast.makeText(LoginActivity.this, "Please fill out all fields", Toast.LENGTH\_SHORT).show();*  *//*  *// } else if (email\_EditText.getText().toString().equals("")) {*  *// Toast.makeText(LoginActivity.this, "Email field is blank", Toast.LENGTH\_SHORT).show();*  *// } else if (password\_EditText.getText().toString().equals("")) {*  *// Toast.makeText(LoginActivity.this, "Password field is blank", Toast.LENGTH\_SHORT).show();*  *// } else {*  *// LoginTest logintest = new LoginTest();*  *// logintest.execute();*  *// }*  startActivity(**new** Intent(LoginActivity.**this**, MainMenuActivity.**class**));  }  });  *//FORGOT PASSWORD TEXTVIEW LISTENER*  TextView forgot\_password\_tv = (TextView) findViewById(R.id.***forgot\_password\_text\_view***);  forgot\_password\_tv.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(LoginActivity.**this**, ForgotPasswordActivity.**class**));  }  });  *//REGISTER TEXTVIEW LISTENER*  TextView register\_tv = (TextView) findViewById(R.id.***register\_text\_view***);  register\_tv.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(LoginActivity.**this**, RegisterAccountActivity.**class**));  }  });  }  @Override  **public void** onBackPressed() {  **if** (getIntent().getBooleanExtra(**"EXIT"**, **false**)) {  finishAffinity();  }  **this**.finishAffinity();  }  *//CONNECT AND LOGIN TEST*  **public class** LoginTest **extends** AsyncTask<String, String, String> {  String **statusMessage** = **""**;  Boolean **isSuccess** = **false**;  @Override  **protected void** onPostExecute(String r) {  Toast.*makeText*(LoginActivity.**this**, r, Toast.***LENGTH\_SHORT***).show();  **if** (**isSuccess**) {  Toast.*makeText*(LoginActivity.**this**, **"Login Successful"**, Toast.***LENGTH\_LONG***).show();  }  }  @Override  **protected** String doInBackground(String... params) {  String email = **email\_EditText**.getText().toString().trim();  String password = **password\_EditText**.getText().toString().trim();  **try** {  **con** = SQLConnection.*connectionclass*();  **if** (**con** == **null**) {  **statusMessage** = **"Check Your Internet Connection."**;  } **else** {  *// USES PREPARED STATEMENT TO PREVENT SQL INJECTION*  PreparedStatement stmt = **con**.prepareStatement(**"select \* from users u join students s on u.uid = s.StudentID join HIGHSCHOOL h on s.HScode = h.code where email=? and password=?"**);  stmt.setString(1, email);  stmt.setString(2, password);  ResultSet rs = stmt.executeQuery();  **if** (rs.next()) {  **isSuccess** = **true**;  String firstName = rs.getString(**"fname"**);  String lastname = rs.getString(**"lname"**);  String hs = rs.getString(**"hsname"**);  **int** is\_admin = rs.getInt(**"admin"**);  **int** uid = rs.getInt(**"uid"**);  *//Store user info for settings*  SharedPreferences sp = getSharedPreferences(**"Login"**, ***MODE\_PRIVATE***);  SharedPreferences.Editor Ed = sp.edit();  Ed.putString(**"email"**, email);  Ed.putString(**"Psw"**, password);  Ed.putString(**"fname"**, firstName);  Ed.putString(**"lname"**, lastname);  Ed.putInt(**"isAdmin"**, is\_admin);  Ed.putInt(**"uid"**, uid);  Ed.putString(**"hs\_uni"**, hs);  Ed.apply();  *//Start Main Menu Activity*  Intent i = **new** Intent(getApplicationContext(), MainMenuActivity.**class**);  startActivity(i);  **con**.close();  } **else** {  **statusMessage** = **"Incorrect Email or Password"**;  **isSuccess** = **false**;  **con**.close();  }  }  } **catch** (Exception ex) {  **isSuccess** = **false**;  **statusMessage** = ex.getMessage();  }  **return statusMessage**;  }  }  } |

ForgotPasswordActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.content.Intent;  **import** android.content.SharedPreferences;  **import** android.os.AsyncTask;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.util.Log;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.EditText;  **import** android.widget.TextView;  **import** android.widget.Toast;  **import** java.sql.Connection;  **import** java.sql.PreparedStatement;  **import** java.sql.ResultSet;  **import** java.util.Timer;  **import** java.util.TimerTask;  **import** com.creativityapps.gmailbackgroundlibrary.BackgroundMail;  **public class** ForgotPasswordActivity **extends** AppCompatActivity {  EditText **email\_et**;  String **pw**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_forgot\_password***);  *//INITIALIZE VIEWS*  TextView remember\_password\_tv = (TextView) findViewById(R.id.***remember\_password\_text\_view***);  TextView register\_tv = (TextView) findViewById(R.id.***fp\_register\_text\_view***);  **email\_et** = (EditText) findViewById(R.id.***forgot\_email\_edit\_text***);  Button send\_pw\_btn = (Button) findViewById(R.id.***send\_email\_pw\_btn***);  SharedPreferences sp1 = **this**.getSharedPreferences(**"Login"**, ***MODE\_PRIVATE***);  **pw** = sp1.getString(**"Psw"**, **null**);  *//SEND EMAIL BUTTON LISTENER*  send\_pw\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *//TO SEND THIS EMAIL WE IMPORTED A LIBRARY DEVELOPED BY yesidlazaro ON GITHUB*  *//https://github.com/yesidlazaro/GmailBackground*  BackgroundMail.*newBuilder*(ForgotPasswordActivity.**this**)  .withUsername(**"amanjotjo@gmail.com"**)  .withPassword(**"Pepsi112"**)  .withMailto(**email\_et**.getText().toString().trim())  .withType(BackgroundMail.***TYPE\_PLAIN***)  .withSubject(**"Retrieve Password"**)  .withBody(**"Your password is: "** + **pw** + **"\nIt is highly recommended to reset your password upon logging in."**)  .send();  }  });  *//REMEMBER PASSWORD TEXTVIEW LISTENER*  remember\_password\_tv.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(ForgotPasswordActivity.**this**, LoginActivity.**class**));  }  });  *//REGISTER TEXTVIEW LISTENER*  register\_tv.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(ForgotPasswordActivity.**this**, RegisterAccountActivity.**class**));  }  });  }  @Override  **public void** onBackPressed() {  startActivity(**new** Intent(ForgotPasswordActivity.**this**, LoginActivity.**class**));  **this**.finish();  }  } |

RegisterAccountActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.content.Intent;  **import** android.os.AsyncTask;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.util.Log;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.EditText;  **import** android.widget.TextView;  **import** android.widget.Toast;  **import** java.sql.Connection;  **import** java.sql.ResultSet;  **import** java.sql.Statement;  **public class** RegisterAccountActivity **extends** AppCompatActivity {  Connection **con**;  Boolean **isSuccess** = **false**;  EditText **fname\_edittext**, **lname\_edittext**, **email\_edittext**, **password\_edittext**, **confirm\_password\_edittext**,  **admin\_code\_edittext**, **hs\_edittext**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_register\_account***);  *//ASSIGN EDITBOXES TO VIEW*  **fname\_edittext** = (EditText) findViewById(R.id.***reg\_fname\_edit\_text***);  **lname\_edittext** = (EditText) findViewById(R.id.***reg\_lname\_edit\_text***);  **email\_edittext** = (EditText) findViewById(R.id.***reg\_email\_edit\_text***);  **password\_edittext** = (EditText) findViewById(R.id.***reg\_password\_text***);  **confirm\_password\_edittext** = (EditText) findViewById(R.id.***reg\_confirm\_password\_text***);  **hs\_edittext** = (EditText) findViewById(R.id.***reg\_high\_school\_text***);  *//admin\_code\_edittext = (EditText) findViewById(R.id.reg\_admin\_code\_text);*  *//ACCOUNT EXISTS - BACK TO LOGIN TEXTVIEW LISTENER*  TextView login\_exists\_tv = (TextView) findViewById(R.id.***login\_exists\_tv***);  login\_exists\_tv.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(RegisterAccountActivity.**this**, LoginActivity.**class**));  }  });  *//REGISTER BUTTON LISTENER*  Button register\_account = (Button) findViewById(R.id.***register\_btn***);  register\_account.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *//CHECK FOR EMPTY FIELDS*  **if** (**fname\_edittext**.getText().toString().equals(**""**) || **fname\_edittext**.getText().toString().equals(**""**) || **email\_edittext**.getText().toString().equals(**""**)  || **password\_edittext**.getText().toString().equals(**""**) || **confirm\_password\_edittext**.getText().toString().equals(**""**)) {  Toast.*makeText*(RegisterAccountActivity.**this**, **"Please fill out all fields.\n(Admin field is optional. Leave blank)"**, Toast.***LENGTH\_LONG***).show();  } **else if** (!**password\_edittext**.getText().toString().trim().equals(**confirm\_password\_edittext**.getText().toString().trim())) {  Toast.*makeText*(RegisterAccountActivity.**this**, **"Passwords do not match"**, Toast.***LENGTH\_LONG***).show();  } **else** {  *//CONNECT AND QUERY*  RegisterAccountConnection registeraccountconnection = **new** RegisterAccountConnection();  registeraccountconnection.execute();  Toast.*makeText*(RegisterAccountActivity.**this**, **"Account created! Please login."**, Toast.***LENGTH\_LONG***).show();  startActivity(**new** Intent(RegisterAccountActivity.**this**, LoginActivity.**class**));  }  }  });  }  @Override  **public void** onBackPressed() {  startActivity(**new** Intent(RegisterAccountActivity.**this**, LoginActivity.**class**));  }  **public class** RegisterAccountConnection **extends** AsyncTask<String, String, String> {  String **statusMessage** = **""**;  @Override  **protected void** onPostExecute(String r) {  **if** (**isSuccess**) {  Toast.*makeText*(RegisterAccountActivity.**this**, **"Account Created"**, Toast.***LENGTH\_LONG***).show();  *//finish();*  }  }  @Override  **protected** String doInBackground(String... params) {  *//GET USER INFO FROM EDITTEXT BOXES AND STORE IN STRINGS*  String first\_name = **fname\_edittext**.getText().toString().trim();  String last\_name = **lname\_edittext**.getText().toString().trim();  String email = **email\_edittext**.getText().toString().trim();  String new\_password = **password\_edittext**.getText().toString().trim();  String confirm\_password = **confirm\_password\_edittext**.getText().toString().trim();  *//String admin\_code = admin\_code\_edittext.getText().toString().trim();*  String hs = **hs\_edittext**.getText().toString().trim();  **try** {  **con** = SQLConnection.*connectionclass*();  **if** (**con** == **null**) {  **statusMessage** = **"Account couldn't be created, check internet connection and try again."**;  } **else** {  String query = **"Select max(uid) as uid from USERS "**;  Statement stmt = **con**.createStatement();  ResultSet rs = stmt.executeQuery(query);  **if** (rs.next()) {  **int** uid = rs.getInt(**"uid"**);  **int** next\_uid = uid + 1;  *//int admin = 1;*  **int** non\_admin = 0;  *//CHECK FOR MATCHING PASSWORDS*  *//FILTER FOR ADMIN USER OR STUDENT USER*  *//if (admin\_code.equals("")) {*  String studentuserquery = **""**;  *//CHECK IF SCHOOL IS PROVIDED*  **if** (hs.equals(**""**)) {  studentuserquery = **"insert into users(uid, admin, email, password, fname,lname)\n"** +  **"values("** + next\_uid + **", "** + non\_admin + **", '"** + email + **"', '"** + new\_password + **"', '"** + first\_name + **"', '"** + last\_name + **"')"**;  } **else** {  studentuserquery = **"insert into users(uid, admin, email, password, fname,lname)\n"** +  **"values("** + next\_uid + **", "** + non\_admin + **", '"** + email + **"', '"** + new\_password + **"', '"** + first\_name + **"', '"** + last\_name + **"')"**  + **"insert into HIGHSCHOOL(code, hsname, advisorFname, advisorLname)\n"** +  **"values("** + 100 + next\_uid + **", '"** + hs + **"', null, null)"** +  **"insert into students(studentid, hscode, attnum)"** +  **"values("** + next\_uid + **",'"** + 100 + next\_uid + **"', null)"**;  }  ResultSet rs1 = stmt.executeQuery(studentuserquery);  Log.*d*(**"INSERT QUERY: "**, studentuserquery);  *// } else {*  *// String adminuserquery = "insert into users(uid, admin, email, password, fname,lname)\n" +*  *// "values(" + next\_uid + ", " + admin + ", '" + email + "', '" + new\_password + "', '" + first\_name + "', '" + last\_name + "')";*  *//*  *// String adduniversityquery = "";*  *//*  *// ResultSet rs2 = stmt.executeQuery(adminuserquery);*  *// Log.d("INSERT QUERY: ", adminuserquery);*  *//*  *// }*  **new** Thread() {  **public void** run() {  RegisterAccountActivity.**this**.runOnUiThread(**new** Runnable() {  **public void** run() {  Toast.*makeText*(RegisterAccountActivity.**this**, **"Account created!"**, Toast.***LENGTH\_LONG***).show();  }  });  }  }.start();  **isSuccess** = **true**;  **con**.close();  } **else** {  **statusMessage** = **"Account couldn't be created, try again"**;  **isSuccess** = **false**;  **con**.close();  }  }  } **catch** (Exception ex) {  **isSuccess** = **false**;  **statusMessage** = ex.getMessage();  }  **return statusMessage**;  }  }  } |

MainMenuActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.app.Activity;  **import** android.content.DialogInterface;  **import** android.content.Intent;  **import** android.content.SharedPreferences;  **import** android.support.v7.app.ActionBar;  **import** android.support.v7.app.AlertDialog;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.TextView;  **import static** java.security.AccessController.*getContext*;  **public class** MainMenuActivity **extends** AppCompatActivity {  String **firstName**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main\_menu***);  ActionBar menu = getSupportActionBar();  menu.setDisplayShowHomeEnabled(**true**);  menu.setIcon(R.drawable.***gsulogosmall***);  *//Retrieve User first name form SharedPreferences*  SharedPreferences sp1 = **this**.getSharedPreferences(**"Login"**, ***MODE\_PRIVATE***);  **firstName** = sp1.getString(**"fname"**, **null**);  *//GREET USER BY NAME*  TextView name\_tv = (TextView) findViewById(R.id.***welcome\_name***);  name\_tv.setText(**"Hello "** + **firstName** + **"!"**);  *//DISCOVER BUTTON LISTENER*  Button discover\_btn = (Button) findViewById(R.id.***discover\_btn***);  discover\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(MainMenuActivity.**this**, DiscoveryActivity.**class**));  }  });  *//MAJOR BUTTON LISTENER*  Button major\_info\_btn = (Button) findViewById(R.id.***major\_info\_btn***);  major\_info\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(MainMenuActivity.**this**, MajorsActivity.**class**));  }  });  }  *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }  *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }  **return true**;  }  @Override  **public void** onBackPressed() {  AlertDialog.Builder dlgAlert = **new** AlertDialog.Builder(**this**);  dlgAlert.setMessage(**"Would you like to logout?"**);  dlgAlert.setTitle(**"Logout?"**);  dlgAlert.setPositiveButton(**"Logout"**,  **new** DialogInterface.OnClickListener() {  **public void** onClick(DialogInterface dialog, **int** which) {  Intent intent = **new** Intent(getApplicationContext(), LoginActivity.**class**);  intent.setFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP***);  intent.putExtra(**"EXIT"**, **true**);  startActivity(intent);  }  });  dlgAlert.setNegativeButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  **public void** onClick(DialogInterface dialog, **int** which) {  *//DO NOTHING*  }  });  dlgAlert.create().show();  }  } |

DiscoveryActivity.Java

|  |
| --- |
|  |

ResultsActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;    **import** android.content.Intent;  **import** android.support.v7.app.ActionBar;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.TextView;    **public class** ResultsActivity **extends** AppCompatActivity {    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_results***);  ActionBar menu = getSupportActionBar();  menu.setDisplayShowHomeEnabled(**true**);  menu.setIcon(R.drawable.***gsulogosmall***);    *//INITIALIZE COMPONENTS*  TextView results\_tv = (TextView) findViewById(R.id.***results\_tv***);  TextView results\_sec\_tv = (TextView) findViewById(R.id.***results\_sec\_tv***);  Button majors\_btn = (Button) findViewById(R.id.***major\_info\_btn***);  Button save\_btn = (Button) findViewById(R.id.***save\_btn***);  Button main\_menu\_btn = (Button) findViewById(R.id.***main\_menu\_btn***);    *//GET EXTRAS FROM DISCOVERY ACTIVITY*  Intent i = getIntent();  **int** result\_index = i.getIntExtra(**"max\_score"**, 0);  **int** sec\_result\_index = i.getIntExtra(**"sec\_maj\_index"**, 0);    *//SET PRIMARY MAJOR*  **if** (result\_index == 0) {  results\_tv.setText(**"Informations Systems"**);  } **else if** (result\_index == 1) {  results\_tv.setText(**"Information Technology"**);  } **else if** (result\_index == 2) {  results\_tv.setText(**"Computer Science"**);  } **else if** (result\_index == 3) {  results\_tv.setText(**"Computer Engineering"**);  }    *//SET SECONDARY MAJOR*  **if** (sec\_result\_index == 0) {  results\_sec\_tv.setText(**"Informations Systems"**);  } **else if** (sec\_result\_index == 1) {  results\_sec\_tv.setText(**"Information Technology"**);  } **else if** (sec\_result\_index == 2) {  results\_sec\_tv.setText(**"Computer Science"**);  } **else if** (sec\_result\_index == 3) {  results\_sec\_tv.setText(**"Computer Engineering"**);  }    *//MAJOR BUTTON LISTENER*  majors\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(ResultsActivity.**this**, MajorsActivity.**class**));  }  });    *//SAVE BUTTON LISTENER*  save\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  *//startActivity(new Intent(ResultsActivity.this, MajorsActivity.class));*  }  });    *//BACK TO MAIN MENU BUTTON LISTENER*  main\_menu\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(ResultsActivity.**this**, MainMenuActivity.**class**));  }  });  }    *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {    getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }    *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {    **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }    **return true**;  }    @Override  **public void** onBackPressed() {  startActivity(**new** Intent(ResultsActivity.**this**, MainMenuActivity.**class**));    }  } |

MajorsActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;    **import** android.content.Intent;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;    **public class** MajorsActivity **extends** AppCompatActivity {    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_majors***);    *//CS BUTTON LISTENER*  Button CS\_btn = (Button) findViewById(R.id.***CS\_btn***);    CS\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {    String selection = **"cs"**;  Intent i = **new** Intent(getApplicationContext(), SelectedMajorActivity.**class**);  i.putExtra(**"selection"**, **"cs"**);  startActivity(i);    }  });    *//SE BUTTON LISTENER*  Button SE\_btn = (Button) findViewById(R.id.***CE\_btn***);    SE\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {    String selection = **"ce"**;  Intent i = **new** Intent(getApplicationContext(), SelectedMajorActivity.**class**);  i.putExtra(**"selection"**, **"ce"**);  startActivity(i);    }  });    *//IT BUTTON LISTENER*  Button IT\_btn = (Button) findViewById(R.id.***IT\_btn***);    IT\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {    String selection = **"it"**;  Intent i = **new** Intent(getApplicationContext(), SelectedMajorActivity.**class**);  i.putExtra(**"selection"**, **"it"**);  startActivity(i);  }  });    *//IS BUTTON LISTENER*  Button IS\_btn = (Button) findViewById(R.id.***IS\_btn***);    IS\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {    String selection = **"is"**;  Intent i = **new** Intent(getApplicationContext(), SelectedMajorActivity.**class**);  i.putExtra(**"selection"**, **"is"**);  startActivity(i);  }  });    *//BACK BUTTON LISTENER*  Button back\_btn = (Button) findViewById(R.id.***back\_btn***);    back\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(MajorsActivity.**this**, MainMenuActivity.**class**));  }  });  }    *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {    getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }    *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {    **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }    **return true**;  }    @Override  **public void** onBackPressed() {  startActivity(**new** Intent(MajorsActivity.**this**, MainMenuActivity.**class**));    }  } |

SelectedMajorActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;    **import** android.content.Intent;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.ImageView;  **import** android.widget.TextView;  **import** android.widget.Toast;    **public class** SelectedMajorActivity **extends** AppCompatActivity {    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_selected\_major***);    *//OBTAIN PASSED VALUE / MAJOR SELECTED BY USER*  String passed\_major = **""**;  Bundle extras = getIntent().getExtras();  **if** (extras != **null**) {  passed\_major = extras.getString(**"selection"**);  }    *//Initialize TextViews*  TextView major\_name = (TextView) findViewById(R.id.***major\_name\_tv***);  TextView major\_info = (TextView) findViewById(R.id.***major\_info\_tv***);  TextView major\_careers = (TextView) findViewById(R.id.***major\_careers\_tv***);  TextView major\_colleges = (TextView) findViewById(R.id.***colleges\_with\_major\_tv***);  ImageView major\_pic = (ImageView) findViewById(R.id.***selected\_major\_image\_view***);    *//CS SELECTED*  **if** (passed\_major.equals(**"cs"**)) {  major\_name.setText(getResources().getString(R.string.***cs***));  major\_info.setText(getResources().getString(R.string.***cs\_info***));  major\_careers.setText(getResources().getString(R.string.***cs\_careers***));  major\_colleges.setText(getResources().getString(R.string.***cs\_colleges***));    major\_pic.setBackgroundResource(R.drawable.***cs\_image***);  }    *//SE SELECTED*  **if** (passed\_major.equals(**"ce"**)) {  major\_name.setText(getResources().getString(R.string.***se***));  major\_info.setText(getResources().getString(R.string.***se\_info***));  major\_careers.setText(getResources().getString(R.string.***se\_careers***));  major\_colleges.setText(getResources().getString(R.string.***se\_colleges***));    major\_pic.setBackgroundResource(R.drawable.***se\_image***);  }    *//IT SELECTED*  **if** (passed\_major.equals(**"it"**)) {  major\_name.setText(getResources().getString(R.string.***it***));  major\_info.setText(getResources().getString(R.string.***it\_info***));  major\_careers.setText(getResources().getString(R.string.***it\_careers***));  major\_colleges.setText(getResources().getString(R.string.***it\_colleges***));    major\_pic.setBackgroundResource(R.drawable.***it\_image***);  }    *//IS SELECTED*  **if** (passed\_major.equals(**"is"**)) {  major\_name.setText(getResources().getString(R.string.***is***));  major\_info.setText(getResources().getString(R.string.***is\_info***));  major\_careers.setText(getResources().getString(R.string.***is\_careers***));  major\_colleges.setText(getResources().getString(R.string.***is\_colleges***));    major\_pic.setBackgroundResource(R.drawable.***is\_image***);  }    *//BACK BUTTON LISTENER*  Button back\_btn = (Button) findViewById(R.id.***back\_btn***);    back\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(SelectedMajorActivity.**this**, MajorsActivity.**class**));  }  });  }    *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {    getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }    *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {    **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }    **return true**;  }    @Override  **public void** onBackPressed() {  startActivity(**new** Intent(SelectedMajorActivity.**this**, MajorsActivity.**class**));  }  } |

AboutActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;    **import** android.content.Intent;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;    **public class** AboutActivity **extends** AppCompatActivity {    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_about***);    *//BACK BUTTON LISTENER*  Button back\_btn = (Button) findViewById(R.id.***back\_btn***);    back\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(AboutActivity.**this**, MainMenuActivity.**class**));  }  });  }    *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {    getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }    *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {    **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }    **return true**;  }    @Override  **public void** onBackPressed() {  startActivity(**new** Intent(AboutActivity.**this**, MainMenuActivity.**class**));  }  } |

SettingsActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.content.DialogInterface;  **import** android.content.Intent;  **import** android.content.SharedPreferences;  **import** android.os.AsyncTask;  **import** android.support.v7.app.ActionBar;  **import** android.support.v7.app.AlertDialog;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.util.Log;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.EditText;  **import** android.widget.Toast;  **import** java.sql.Connection;  **import** java.sql.PreparedStatement;  **import** java.sql.ResultSet;  **public class** SettingsActivity **extends** AppCompatActivity {  Connection **con**;  String **userEmail**, **pass**;  String **curr\_fname** = **""**, **curr\_lname**, **curr\_email**, **curr\_school**, **curr\_password**;  EditText **new\_pw**, **confirm\_new\_pw**, **fname**, **lname**, **email**, **hs\_uni**;  **boolean deletable** = **false**;  **boolean updatable** = **false**;  **boolean pw\_updatable** = **false**;  **int uid**;  **int is\_admin**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_settings***);  ActionBar menu = getSupportActionBar();  menu.setDisplayShowHomeEnabled(**true**);  menu.setIcon(R.drawable.***gsulogosmall***);  dataFetcher datafeetcher = **new** dataFetcher();  datafeetcher.execute();  *// while(curr\_fname == null){*  *//*  *// }*  *//Retrieve User info*  SharedPreferences sp1 = **this**.getSharedPreferences(**"Login"**, ***MODE\_PRIVATE***);  **userEmail** = sp1.getString(**"email"**, **null**);  **pass** = sp1.getString(**"Psw"**, **null**);  **while** (**curr\_fname**.equals(**""**)) {  Log.*d*(**"W"**, **"Loading "**);  }  *//SETTINGS VIEW INITIALIZE*  **fname** = (EditText) findViewById(R.id.***update\_fname\_text***);  **lname** = (EditText) findViewById(R.id.***update\_lname\_text***);  **email** = (EditText) findViewById(R.id.***update\_email\_text***);  **hs\_uni** = (EditText) findViewById(R.id.***update\_hs\_uni\_text***);  **final** EditText old\_pw\_et = (EditText) findViewById(R.id.***old\_password\_text***);  **new\_pw** = (EditText) findViewById(R.id.***new\_password\_text***);  **confirm\_new\_pw** = (EditText) findViewById(R.id.***confirm\_new\_password\_text***);  *//Append ORIGINAL User Values to Edittext*  **fname**.setText(**curr\_fname**);  **lname**.setText(**curr\_lname**);  **email**.setText(**curr\_email**);  **hs\_uni**.setText(**curr\_school**);  *// if(is\_admin == 0){*  *// hs\_uni.setHint("Highschool");*  *// hs\_uni.setText("");*  *// }else{*  *// hs\_uni.setHint("University");*  *// hs\_uni.setText("");*  *// }*  *// SAVE/UPDATE SETTINGS BUTTON*  Button save\_btn = (Button) findViewById(R.id.***save\_btn***);  save\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **updatable** = **true**;  **if** (old\_pw\_et.getText().toString().trim().equals(**pass**)) {  **if** (!**new\_pw**.getText().toString().trim().equals(**""**) & !**confirm\_new\_pw**.getText().toString().trim().equals(**""**)) {  **pw\_updatable** = **true**;  }  SettingsHandler sh = **new** SettingsHandler();  sh.doInBackground();  Toast.*makeText*(SettingsActivity.**this**, **"Account Updated"**, Toast.***LENGTH\_LONG***).show();  **updatable** = **false**;  } **else if** (old\_pw\_et.getText().toString().trim().equals(**""**)) {  Toast.*makeText*(SettingsActivity.**this**, **"Password field is blank"**, Toast.***LENGTH\_LONG***).show();  } **else** {  Toast.*makeText*(SettingsActivity.**this**, **"Incorrect Password"**, Toast.***LENGTH\_LONG***).show();  }  }  });  *//ATTEMPTS BUTTON*  Button att\_btn = (Button) findViewById(R.id.***attempt\_btn***);  att\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(SettingsActivity.**this**, ViewResultsActivity.**class**));  }  });  *//DELETE BUTTON LISTENER*  Button delete\_btn = (Button) findViewById(R.id.***delete\_btn***);  delete\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **deletable** = **true**;  *//ACTION TO DELETE ACCOUNT*  DialogInterface.OnClickListener dialogClickListener = **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  **switch** (which) {  **case** DialogInterface.***BUTTON\_POSITIVE***:  *//Yes button clicked*  SettingsHandler sa = **new** SettingsHandler();  sa.execute();  Toast.*makeText*(SettingsActivity.**this**, **"Account Successfully Deleted"**, Toast.***LENGTH\_LONG***).show();  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  SettingsActivity.**this**.finish();  **break**;  **case** DialogInterface.***BUTTON\_NEGATIVE***:  *//No button clicked*  **deletable** = **false**;  **break**;  }  }  };  AlertDialog.Builder builder = **new** AlertDialog.Builder(SettingsActivity.**this**);  builder.setMessage(**"Are you sure you want to delete this account?"**).setPositiveButton(**"Yes"**, dialogClickListener)  .setNegativeButton(**"No"**, dialogClickListener).show();  }  });  }  *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }  *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  SettingsActivity.**this**.finish();  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }  **return true**;  }  *//CONNECT to DB in ASYNC*  **public class** SettingsHandler **extends** AsyncTask<String, String, String> {  String statusMessage = **""**;  Boolean **isSuccess** = **false**;  @Override  **protected void** onPostExecute(String r) {  Toast.*makeText*(SettingsActivity.**this**, r, Toast.***LENGTH\_SHORT***).show();  **if** (**isSuccess**) {  Toast.*makeText*(SettingsActivity.**this**, **"Loaded"**, Toast.***LENGTH\_LONG***).show();  }  }  @Override  **protected** String doInBackground(String... params) {  **try** {  **con** = SQLConnection.*connectionclass*();  **if** (**con** == **null**) {  statusMessage = **"Check Your Internet Connection."**;  } **else if** (**deletable**) { *//DELETE ACCOUNT*  PreparedStatement stmt = **con**.prepareStatement(**"delete from students where studentid = "** + **uid** + **"\n"** +  **"delete from HIGHSCHOOL where code = 100"** + **uid** + **"\n"** +  **"delete from users where uid = "** + **uid** + **""** +  **"select \* from users"**);  stmt.executeUpdate();  **con**.close();  } **else if** (**updatable**) { *//EXECUTE QUERY OPERATIONS*  *// USES PREPARED STATEMENT TO PREVENT SQL INJECTION*  PreparedStatement stmt1 = **con**.prepareStatement(**"select \* from users u join students s on u.uid = s.StudentID join HIGHSCHOOL h on s.HScode = h.code where email=? and password=?"**);  stmt1.setString(1, **userEmail**);  stmt1.setString(2, **pass**);  ResultSet rs = stmt1.executeQuery();  **if** (rs.next()) {  **isSuccess** = **true**;  PreparedStatement stmt\_update = **con**.prepareStatement(**"update HIGHSCHOOL set hsname = '"** + **hs\_uni**.getText().toString().trim() + **"' where code = 100"** + **uid** + **"\n"** +  **"update users set fname = '"** + **fname**.getText().toString().trim() + **"', lname = '"** + **lname**.getText().toString().trim() + **"', email = '"** + **email**.getText().toString().trim() + **"' where uid = "** + **uid** + **""**);  Log.*d*(**"UPDATE USER STRING"**, **"------"** + stmt\_update.toString() + **""**);  stmt\_update.executeUpdate();  **if** (**pw\_updatable**) {  **if** (**new\_pw**.getText().toString().trim().equals(**confirm\_new\_pw**.getText().toString().trim())) {  PreparedStatement stmt\_pw\_update = **con**.prepareStatement(**"update users set password = '"** + **confirm\_new\_pw** + **"' where uid = "** + **uid** + **""**);  stmt\_pw\_update.executeUpdate();  Log.*d*(**"UPDATE USER STRING"**, **"------"** + stmt\_pw\_update.toString() + **""**);  }  }  **con**.close();  } **else** {  statusMessage = **"Failed to update data"**;  **isSuccess** = **false**;  **con**.close();  }  }  } **catch** (Exception ex) {  **isSuccess** = **false**;  statusMessage = ex.getMessage();  }  **return** statusMessage;  }  }  *//CONNECT to DB in ASYNC*  **public class** dataFetcher **extends** AsyncTask<String, String, String> {  String **statusMessage** = **""**;  Boolean **isSuccess** = **false**;  @Override  **protected void** onPostExecute(String r) {  Toast.*makeText*(SettingsActivity.**this**, r, Toast.***LENGTH\_SHORT***).show();  **if** (**isSuccess**) {  *//Toast.makeText(SettingsActivity.this, "Data Load successfull", Toast.LENGTH\_LONG).show();*  }  }  @Override  **protected** String doInBackground(String... params) {  **try** {  **con** = SQLConnection.*connectionclass*();  **if** (**con** == **null**) {  **statusMessage** = **"Check Your Internet Connection."**;  } **else** { *//EXECUTE QUERY OPERATIONS*  *// USES PREPARED STATEMENT TO PREVENT SQL INJECTION*  PreparedStatement stmt1 = **con**.prepareStatement(**"select \* from users u join students s on u.uid = s.StudentID join HIGHSCHOOL h on s.HScode = h.code where email=? and password=?"**);  stmt1.setString(1, **userEmail**);  stmt1.setString(2, **pass**);  ResultSet rs = stmt1.executeQuery();  **if** (rs.next()) {  **curr\_fname** = rs.getString(**"fname"**);  **curr\_lname** = rs.getString(**"lname"**);  **curr\_email** = rs.getString(**"email"**);  **curr\_school** = rs.getString(**"hsname"**);  **curr\_password** = rs.getString(**"password"**);  **uid** = rs.getInt(**"uid"**);  **isSuccess** = **true**;  **con**.close();  } **else** {  **statusMessage** = **"Failed to fetch data"**;  **isSuccess** = **false**;  **con**.close();  }  }  } **catch** (Exception ex) {  **isSuccess** = **false**;  **statusMessage** = ex.getMessage();  }  **return statusMessage**;  }  }  @Override  **public void** onBackPressed() {  **this**.finish();  }  } |

ResultsActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.content.Intent;  **import** android.os.AsyncTask;  **import** android.support.v7.app.ActionBar;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.util.Log;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.TextView;  **import** android.widget.Toast;  **import** java.sql.Connection;  **import** java.sql.ResultSet;  **import** java.sql.Statement;  **import** java.util.Random;  **public class** ResultsActivity **extends** AppCompatActivity {  Button **majors\_btn**, **save\_btn**, **main\_menu\_btn**, **prim\_major\_btn**, **sec\_major\_btn**;  TextView **results\_tv**, **results\_sec\_tv**;  String[] **done\_arr**;  String **primary**;  String **secondary**;  Connection **con**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_results***);  ActionBar menu = getSupportActionBar();  menu.setDisplayShowHomeEnabled(**true**);  menu.setIcon(R.drawable.***gsulogosmall***);  *//INITIALIZE VIEWS*  **results\_tv** = (TextView) findViewById(R.id.***results\_tv***);  **results\_sec\_tv** = (TextView) findViewById(R.id.***results\_sec\_tv***);  **majors\_btn** = (Button) findViewById(R.id.***major\_info\_btn***);  **save\_btn** = (Button) findViewById(R.id.***save\_btn***);  **main\_menu\_btn** = (Button) findViewById(R.id.***main\_menu\_btn***);  **prim\_major\_btn** = (Button) findViewById(R.id.***prim\_major\_btn***);  **sec\_major\_btn** = (Button) findViewById(R.id.***sec\_major\_btn***);  *//GET EXTRAS FROM DISCOVERY ACTIVITY*  Intent intent = getIntent();  **int** result\_index = intent.getIntExtra(**"max\_score"**, 0);  **int** sec\_result\_index = intent.getIntExtra(**"sec\_maj\_index"**, 0);  **done\_arr** = intent.getStringArrayExtra(**"done\_arr"**);  *//SET PRIMARY MAJOR*  **if** (result\_index == 0) {  **results\_tv**.setText(getResources().getString(R.string.***is***));  } **else if** (result\_index == 1) {  **results\_tv**.setText(getResources().getString(R.string.***it***));  } **else if** (result\_index == 2) {  **results\_tv**.setText(getResources().getString(R.string.***cs***));  } **else if** (result\_index == 3) {  **results\_tv**.setText(getResources().getString(R.string.***ce***));  }  *//SET SECONDARY MAJOR*  **if** (sec\_result\_index == 0) {  **results\_sec\_tv**.setText(getResources().getString(R.string.***is***));  } **else if** (sec\_result\_index == 1) {  **results\_sec\_tv**.setText(getResources().getString(R.string.***it***));  } **else if** (sec\_result\_index == 2) {  **results\_sec\_tv**.setText(getResources().getString(R.string.***cs***));  } **else if** (sec\_result\_index == 3) {  **results\_sec\_tv**.setText(getResources().getString(R.string.***ce***));  }  *//SET BUTTON NAMES*  **prim\_major\_btn**.setText(**results\_tv**.getText());  **sec\_major\_btn**.setText(**results\_sec\_tv**.getText());  *//PRIMARY BUTTON LISTENER*  **prim\_major\_btn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  String selection = **results\_tv**.getText().toString();  Intent i = **new** Intent(getApplicationContext(), SelectedMajorActivity.**class**);  i.putExtra(**"selection"**, selection);  startActivity(i);  }  });  *//SECONDARY BUTTON LISTENER*  **sec\_major\_btn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  String selection = **results\_sec\_tv**.getText().toString();  Intent i = **new** Intent(getApplicationContext(), SelectedMajorActivity.**class**);  i.putExtra(**"selection"**, selection);  startActivity(i);  }  });  *//MAJOR BUTTON LISTENER*  **majors\_btn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(ResultsActivity.**this**, MajorsActivity.**class**));  }  });  *//SAVE BUTTON LISTENER*  **save\_btn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  saveQuiz savequiz = **new** saveQuiz();  savequiz.execute();  Toast.*makeText*(ResultsActivity.**this**, **"Results saved!"**, Toast.***LENGTH\_SHORT***).show();  startActivity(**new** Intent(ResultsActivity.**this**, ViewResultsActivity.**class**));  }  });  *//BACK TO MAIN MENU BUTTON LISTENER*  **main\_menu\_btn**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  startActivity(**new** Intent(ResultsActivity.**this**, MainMenuActivity.**class**));  }  });  }  *//CONNECT TO PULL QUIZ QUESTIONS*  **public class** saveQuiz **extends** AsyncTask<String, String, String> {  String **statusMessage** = **""**;  Boolean **isSuccess** = **false**;  @Override  **protected void** onPostExecute(String r) {  *//Toast.makeText(LoginActivity.this, r, Toast.LENGTH\_SHORT).show();*  **if** (**isSuccess**) {  *// Toast.makeText(LoginActivity.this, "Login Successful", Toast.LENGTH\_LONG).show();*  *//finish();*  }  }  @Override  **protected** String doInBackground(String... params) {  **try** {  **con** = SQLConnection.*connectionclass*();  **if** (**con** == **null**) {  **statusMessage** = **"Check Your Internet Connection."**;  } **else** {  String query = **""**;  query = **"Select \* from attempts order by attnum desc"**;  Log.*d*(**"UPDATE QUESTION STRING"**, query);  Statement stmt = **con**.createStatement();  ResultSet rs = stmt.executeQuery(query);  **if** (!rs.next()) {  String attemptQuery = **""**;  **isSuccess** = **true**;  **for** (**int** i = 0; i < **done\_arr**.**length**; i++) {  attemptQuery = **"insert into attempts (attNum, qid, qid2, qid3, qid4, qid5, qid6, [primary], [secondary])\n"** +  **"values(1,'"** + **done\_arr**[0] + **"', '"** + **done\_arr**[1] + **"', '"** + **done\_arr**[2]  + **"', '"** + **done\_arr**[3] + **"','"** + **done\_arr**[4] + **"', '"** + **done\_arr**[5] + **"','"**  + **results\_tv**.getText().toString() + **"','"** + **results\_sec\_tv**.getText().toString() + **"')"**;  Log.*d*(**"QUESTION "**, attemptQuery);  }  ResultSet rs1 = stmt.executeQuery(attemptQuery);  **con**.close();  } **else** {  **int** attempt\_num = rs.getInt(**"attNum"**);  **int** next\_attempt\_num = attempt\_num + 1;  **if** (next\_attempt\_num == 5) {  next\_attempt\_num = 1;  }  String attemptQuery2 = **""**;  **for** (**int** i = 0; i < **done\_arr**.**length**; i++) {  attemptQuery2 = **"Delete from attempts where attNum = "** + next\_attempt\_num + **" "** +  **"insert into attempts (attNum, qid, qid2, qid3, qid4, qid5, qid6, [primary], [secondary])\n"** +  **"values("** + next\_attempt\_num + **",'"** + **done\_arr**[0] + **"', '"** + **done\_arr**[1] + **"', '"** + **done\_arr**[2]  + **"', '"** + **done\_arr**[3] + **"','"** + **done\_arr**[4] + **"', '"** + **done\_arr**[5] + **"','"**  + **results\_tv**.getText().toString() + **"','"** + **results\_sec\_tv**.getText().toString() + **"')"**;  }  ResultSet rs1 = stmt.executeQuery(attemptQuery2);  **con**.close();  }  *// else {*  *// statusMessage = "Something went wrong with the connection";*  *// isSuccess = false;*  *// con.close();*  *// }*  }  } **catch** (Exception ex) {  **isSuccess** = **false**;  **statusMessage** = ex.getMessage();  }  **return statusMessage**;  }  }  *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }  *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }  **return true**;  }  @Override  **public void** onBackPressed() {  startActivity(**new** Intent(ResultsActivity.**this**, MainMenuActivity.**class**));  }  } |

ViewResultsActivity.Java

|  |
| --- |
| **package** com.f17.csci5539.team5.ceitmajorselectionapp;  **import** android.content.Intent;  **import** android.content.SharedPreferences;  **import** android.os.AsyncTask;  **import** android.support.v7.app.ActionBar;  **import** android.support.v7.app.AppCompatActivity;  **import** android.os.Bundle;  **import** android.util.Log;  **import** android.view.Menu;  **import** android.view.MenuItem;  **import** android.view.View;  **import** android.widget.Button;  **import** android.widget.TextView;  **import** java.sql.Connection;  **import** java.sql.ResultSet;  **import** java.sql.Statement;  **import** java.util.Random;  **public class** ViewResultsActivity **extends** AppCompatActivity {  TextView **attempt\_1**, **attempt\_2**, **attempt\_3**, **attempt\_4**;  Connection **con**;  String **attempt1** = **""**;  String **attempt2** = **""**;  String **attempt3** = **""**;  String **attempt4** = **""**;  String **QID** = **""**;  String **QID2** = **""**;  String **QID3** = **""**;  String **QID4** = **""**;  String **QID5** = **""**;  String **QID6** = **""**;  String **prim** = **""**;  String **sec** = **""**;  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_view\_results***);  ActionBar menu = getSupportActionBar();  menu.setDisplayShowHomeEnabled(**true**);  menu.setIcon(R.drawable.***gsulogosmall***);  Button back\_btn = (Button) findViewById(R.id.***back\_btn***);  **attempt\_1** = (TextView) findViewById(R.id.***attempt\_1\_tv***);  **attempt\_2** = (TextView) findViewById(R.id.***attempt\_2\_tv***);  **attempt\_3** = (TextView) findViewById(R.id.***attempt\_3\_tv***);  **attempt\_4** = (TextView) findViewById(R.id.***attempt\_4\_tv***);  PullQuizResults pqr = **new** PullQuizResults();  pqr.execute();  **while**(**attempt4** == **""** ){  Log.*d*(**"WAIT"**, **"Waiting for load"**);  }  **attempt\_1**.setText(**attempt1**);  **attempt\_2**.setText(**attempt2**);  **attempt\_3**.setText(**attempt3**);  **attempt\_4**.setText(**attempt4**);  *//EXIT BUTTON BACK TO MAIN MENU*  back\_btn.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  ViewResultsActivity.**this**.finish();  }  });  }  *//CONNECT TO PULL QUIZ QUESTIONS*  **public class** PullQuizResults **extends** AsyncTask<String, String, String> {  String **statusMessage** = **""**;  Boolean **isSuccess** = **false**;  @Override  **protected void** onPostExecute(String r) {  *//Toast.makeText(LoginActivity.this, r, Toast.LENGTH\_SHORT).show();*  **if** (**isSuccess**) {  *// Toast.makeText(LoginActivity.this, "Login Successful", Toast.LENGTH\_LONG).show();*  *//finish();*  }  }  @Override  **protected** String doInBackground(String... params) {  **try** {  **con** = SQLConnection.*connectionclass*();  **if** (**con** == **null**) {  **statusMessage** = **"Check Your Internet Connection."**;  } **else** {  String query = **"Select \* from attempts"**;  Statement stmt = **con**.createStatement();  ResultSet rs = stmt.executeQuery(query);  **if** (rs.next()) {  **isSuccess** = **true**;  String query1 = **"Select \* from attempts where attnum = 1"**;  ResultSet rs1 = stmt.executeQuery(query1);  **if** (rs1.next()) {  **QID** = rs1.getString(**"QID"**);  **QID2** = rs1.getString(**"QID2"**);  **QID3** = rs1.getString(**"QID3"**);  **QID4** = rs1.getString(**"QID4"**);  **QID5** = rs1.getString(**"QID5"**);  **QID6** = rs1.getString(**"QID6"**);  **prim** = rs1.getString(**"primary"**);  **sec** = rs1.getString(**"secondary"**);  **attempt1** = **prim** + **"\n"** + **sec**;  } **else** {  **attempt1** = **"Not Attempted Yet"**;  }  String query2 = **"Select \* from attempts where attnum = 2"**;  ResultSet rs2 = stmt.executeQuery(query2);  **if** (rs2.next()) {  **QID** = rs2.getString(**"QID"**);  **QID2** = rs2.getString(**"QID2"**);  **QID3** = rs2.getString(**"QID3"**);  **QID4** = rs2.getString(**"QID4"**);  **QID5** = rs2.getString(**"QID5"**);  **QID6** = rs2.getString(**"QID6"**);  **prim** = rs2.getString(**"primary"**);  **sec** = rs2.getString(**"secondary"**);  **attempt2** = **prim** + **"\n"** + **sec**;  } **else** {  **attempt2** = **"Not Attempted Yet"**;  }  String query3 = **"Select \* from attempts where attnum = 3"**;  ResultSet rs3 = stmt.executeQuery(query3);  **if** (rs3.next()) {  **QID** = rs3.getString(**"QID"**);  **QID2** = rs3.getString(**"QID2"**);  **QID3** = rs3.getString(**"QID3"**);  **QID4** = rs3.getString(**"QID4"**);  **QID5** = rs3.getString(**"QID5"**);  **QID6** = rs3.getString(**"QID6"**);  **prim** = rs3.getString(**"primary"**);  **sec** = rs3.getString(**"secondary"**);  **attempt3** = **prim** + **"\n"** + **sec**;  } **else** {  **attempt3** = **"Not Attempted Yet"**;  }  String query4 = **"Select \* from attempts where attnum = 4"**;  ResultSet rs4 = stmt.executeQuery(query4);  **if** (rs4.next()) {  **QID** = rs4.getString(**"QID"**);  **QID2** = rs4.getString(**"QID2"**);  **QID3** = rs4.getString(**"QID3"**);  **QID4** = rs4.getString(**"QID4"**);  **QID5** = rs4.getString(**"QID5"**);  **QID6** = rs4.getString(**"QID6"**);  **prim** = rs4.getString(**"primary"**);  **sec** = rs4.getString(**"secondary"**);  **attempt4** = **prim** + **"\n"** + **sec**;  } **else** {  **attempt4** = **"Not Attempted Yet"**;  }  **con**.close();  } **else** {  **statusMessage** = **"Something went wrong with the connection"**;  **isSuccess** = **false**;  **con**.close();  }  }  } **catch** (Exception ex) {  **isSuccess** = **false**;  **statusMessage** = ex.getMessage();  }  **return statusMessage**;  }  }  *//INFLATE MENU*  @Override  **public boolean** onCreateOptionsMenu(Menu menu) {  getMenuInflater().inflate(R.menu.***main\_menu***, menu);  **return true**;  }  *//MENU ITEM ACTIONS*  @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** R.id.***Settings***:  startActivity(**new** Intent(getApplicationContext(), SettingsActivity.**class**));  **break**;  **case** R.id.***About***:  startActivity(**new** Intent(getApplicationContext(), AboutActivity.**class**));  **break**;  **case** R.id.***Logout***:  startActivity(**new** Intent(getApplicationContext(), LoginActivity.**class**));  **break**;  **default**:  **return super**.onOptionsItemSelected(item);  }  **return true**;  }  @Override  **public void** onBackPressed() {  **this**.finish();  }  } |