|  |  |
| --- | --- |
| PROJECT DESIGN – MODULAR STUDY APPLICATION | ABSTRACT  A Project Design for the development of a Modular Study Application for students. The design outlines the interface and functionality, application structure, build instructions and UML Diagram for the application.  Kory Bennett / Brock Allton / Dylan Sawyer / Justin Casey / Dan Page / Cale Ward  UMUC 495 6380 09 Feb 2020 |

TABLE OF CONTENTS

|  |  |
| --- | --- |
| 1. Revision History | 3 |
| 1. User Interface And Functionality | 3 |
| 1. Application Structure | 3 |
| 1. Build Instructions | 4 |
| 1. Sequence Diagram | 5 |
| 1. UML Diagram | 6 |

1. **Revision History**

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 03 Feb 2020 | Initial Draft | K. Bennett |
| 05 Feb 2020 | Second Draft | D. Sawyer |
| 06 Feb 2020 | User Interface | C. Ward |
| 07 Feb 2020 | ERD | J. Casey |
| 9 Feb 2020 | Review | B. Allton |
| 09 Feb 2020 | Final Review | K. Bennett / Brock Allton |

Project Design

1. **User Interface and Functionality**

Team Modular Study created a Graphic User Interface (GUI) software that allows a user to access a database with questions and be tested with multiple choice options to answer. A score is provided to tell the user the success of the test. The application works by taking user input and importing questions from a database, then displaying them for the user to be tested on in a random order. Functionality includes:

* A drop-down pane that holds the subjects that a User can be tested on
* An area for the User to input how many questions they want to be tested on
* Multiple choice questions in random order with four options the User can select from
* The results of the test shown
* An option for the User at the end of the test to exit or restart the application

1. **Application Structure**

**Main.java**

* Started by the user
* Initializes and handles GUI.java
* Queries Maria DB for above information
* Parses database results for optimal viewing in GUI
* Receives requests from GUI.java
* Sends results to GUI.java
* Serves as primary interface to database
* Responsible for randomizing questions
* Tracks questions completed out of total
* Tracks number of correct answers out of total

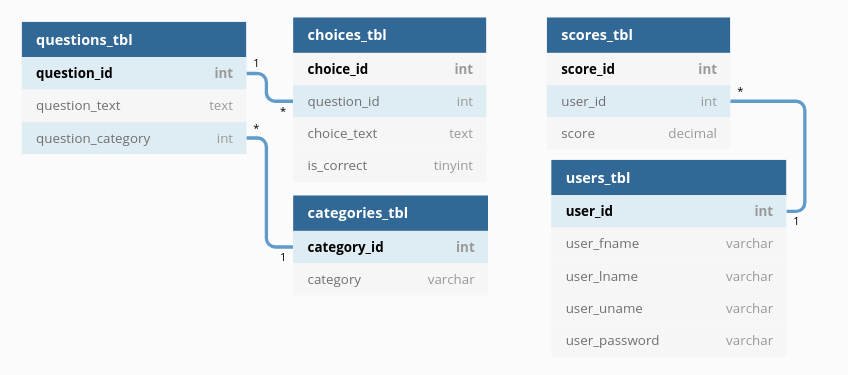
**GUI.java**

* Contains JFrame user interface
* Accepts user input via button and text
* Outputs questions, answers and results to user
* Sends requests to Main.java
  + Available Databases
  + Number of Questions within each database
  + Individual Questions
* Displays resulting scores to user

**SQL**

* Stores all subject and question information
* Only interfaced by MiddleMan.java
* Hosted remotely to prevent setup complicated database set up

Entity Relationship Diagram



1. **Build Instructions**

**cd into the ‘Code’ directory**

* 1. **Compile your Java code, generating class files**

*javac -cp “.:./mysql-connector-java-8.0.19.jar” <ALL\_JAVA\_FILES>*

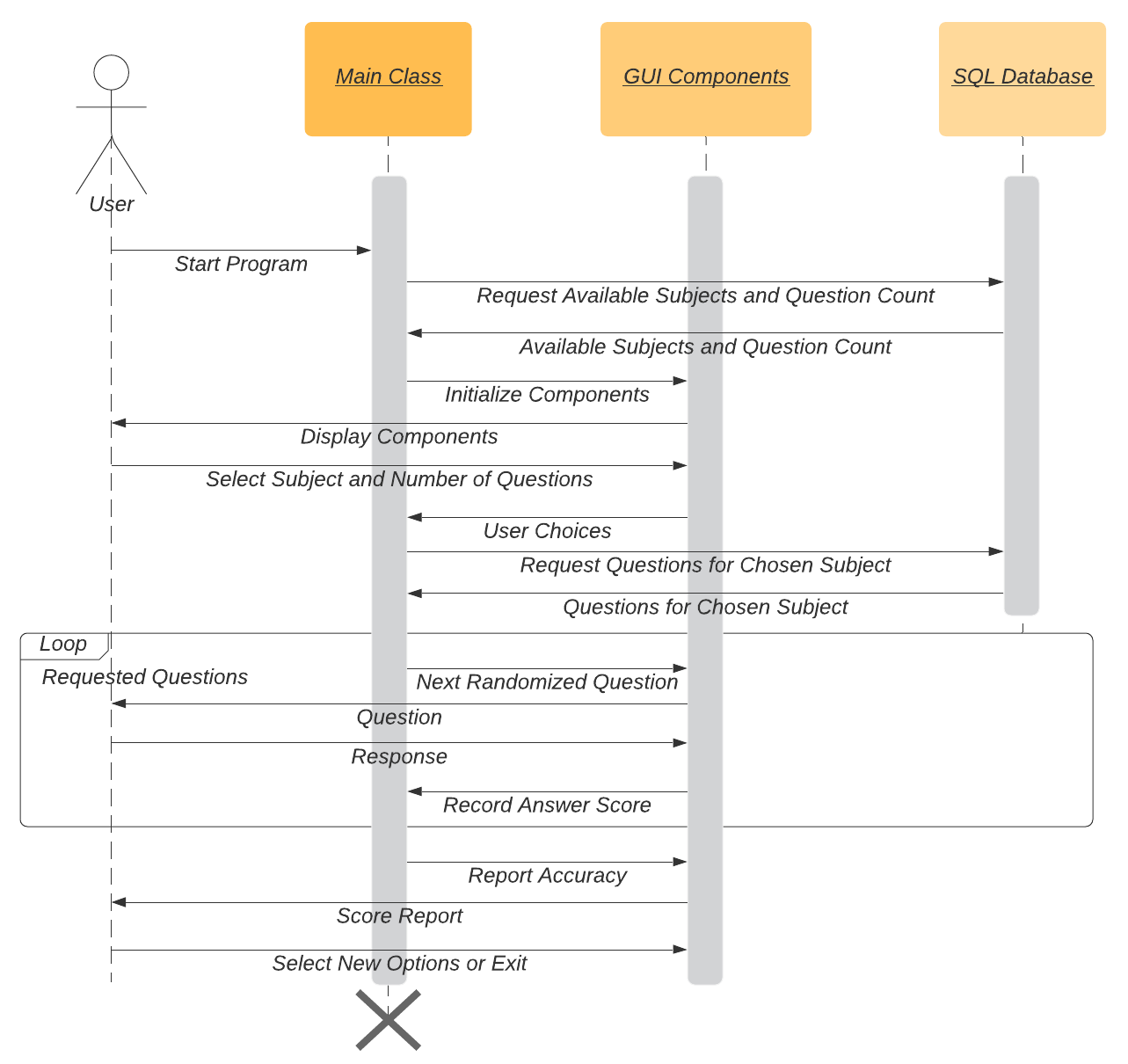
* 1. **Create a JAR file called study**

*jar cmf manifest study <ALL\_CLASS\_FILES>*

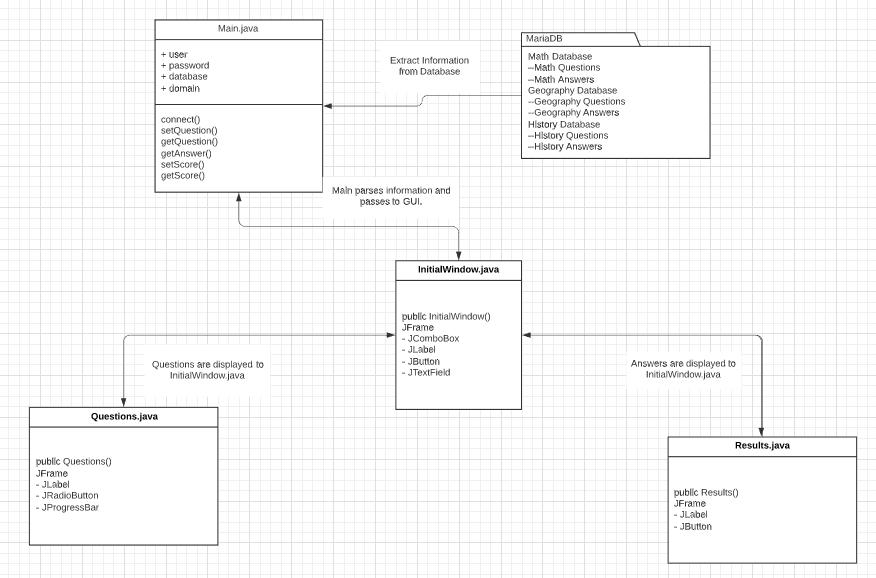
* 1. **Execute the JAR file**

*java -jar study*

1. **Sequence Diagram**

****

1. **UML Diagrams**

****