General Statement

MyMathTutor™ is designed to provide users with an environment to learn and practice basic mathematics. Our goal is to replace in-person math tutors for early math by providing easy to understand math lessons and an unlimited number of practice opportunities while providing feedback and positive reinforcement. After launching MyMathTutor the user is presented with the options to either log in to the program or create a new account. If the user does not have an account the account sign up screen prompts the user to select a username and password that will be used to log in to the program. After providing a valid username and password a new account is created and the user and again sent to the log in screen to log in with their newly created username and password.

After successfully logging in the user is directed to the welcome screen of our program that displays a welcome message, various user statistics, an option to update personal contact information, and provides the user with various options to practice and test their knowledge of basic mathematics. The update info button launches a screen where the user can see their current contact information, and update their information and password. The quiz tab provides the user with several options that include: Starting a new quiz, continuing an old quiz that has not been submitted, and a practice option. Starting a new quiz begins a quiz with the problem types the user selects. A new quiz is graded and may be reviewed. If the user chooses to review a quiz after it has been submitted the review quiz screen is launched. The review quiz screen shows all the questions, the user’s answers, the correct answer, and an indication of whether the question was answered correctly or not. The review quiz screen also provides the user an opportunity to save the quiz to a text file. If the user fails to complete a quiz the quiz is stored externally when the user closes MyMathTutor and may be continued later by selecting “resume quiz”.

If practice is selected a new window is launched that provides the user with questions whose type can be selected as they go, users may choose to check their answers as they go or choose to skip a question and move on to the next question. No data about the practice quiz is saved. The lesson tab provides an experience similar to the practice quiz but focuses more on providing the user with positive feedback and reinforcement. Lessons are separated by subject that the user must select. In order for a lesson to be considered completed 10 questions must be answered correctly. Once a lesson is completed the user is updated to show lesson completion and the user will no longer be allowed to enter that subjects lesson. Additional functionality is included in the menu bar. The user may use the menu bar to close a current lesson, print a currently active quiz without completing it, and review past quizzes that have been completed and submitted.

Data

MyMathTutor™ will utilize ArrayLists, Strings, Integers, Doubles, ObservableLists, Booleans, .ser files as well as implement JavaFx functionality, and a number of custom objects tailored specifically for our program.

Polymorphic Behavior

Our program exhibits polymorphic behavior when generating quiz questions and lesson activates. Polymorphic behavior is exemplified when problems are created. Every specific problem type (addition, subtraction, multiplication, and division) is a sub class of the parent class Problem. The parent class creates and stores the variables that are used during problem generation while each sub classes uses these integers to generate questions and answers polymorphically.

Regular Expressions

Regular expressions are used to validate proper format of usernames, phone numbers, email addresses. Regular expressions ensure usernames are not duplicated, and that passwords are strong.

Memory

Our program stores data in memory by updating and passing a currentUser object of type User between controllers, all necessary data is stored within this User object including Quiz objects (Quiz objects contain Problem objects). The User object can be considered the parent class of the data storage hierarchy. The User and the externally stored ArrayList of users is updated whenever new information is stored as well as each time the program is closed.

.

Serialization

Serialization is done by a single class that is capable of both serializing and deserializing Array Lists containing User objects to and from a file “users.ser”. All necessary data including lesson completion, quiz history, user identification, contact details, etc. are all store within the User object. This externally stored ArrayList is accessed and updated whenever new information about a user is acquired or updated.

Unit Testing

Junit testing will include almost all functions of our program including: account creation, sign in validation, quiz and question generation, and proper data storage both internally and externally.

Exception Handling

Custom exceptions created for our program include InvalidUsernameException, InvalidPasswordException, and an UnansweredQuestionException. The InvalidUsernameException and InvalidPasswordException are thrown when attempting to create an account on the account creation screen with an invalid username or password. When these exceptions are caught the user is made aware that their username/password are invalid and prevents further method execution guaranteeing that a user is not created with invalid parameters. The UnansweredQuestionException is thrown if the user has left questions in a graded quiz unanswered. When caught the user is made aware that they have left questions unanswered and is gives an option whether or not to continue submitting their quiz. Exceptions are also handled when attempting to save a quiz that doesn’t exist and when grading a quiz where user answers are left blank. If the former occurs the continueQuiz value is simply set to null, and if the user fails to answer a question and a NumberFormatException occurs when grading the problem is simply set to incorrect.

JavaDoc Utility

Appropriate documentation has been created by using the “Generate JavaDoc” wizard found within Eclipse IDE found by navigating to Project → Generate Javadoc. Although this process is largely automated it is necessary to add appropriate JavaDoc comments before each method within each class in a format beginning with /\*\* and ending with \*/ additional information may be documented using appropriate tags such as @param, @author, and @throws @returns.

GUI

MyMathTutor’s GUI is simple and intuitive allowing the user to easily navigate between lessons and quizzes, account information, and user records and statistics. With an easy to use account creation screen, login controller, and easy to understand interface, MyMathTutor will have you learning simple mathematics in a matter of minutes.

