Conceptual Design

September 15, 2016

System Description

* **Point-of-View: Customer**
  + Calculator application
  + Pleasing GUI
  + Must perform the following operations:
    - Addition
    - Subtraction
    - Multiplication
    - Division
    - Modulus Division
  + Uses Shunting Yard Algorithm to convert expressions from infix to postfix
  + User login required
  + New user registration should be easy and streamlined
  + Users can access their account info once logged in
* **Point-of-View: User**
  + Calculator application
  + Must be able to do basic math problems
  + Easy to use
  + Answers should be displayed in large font
  + Should have option of typing expressions using keyboard and clicking buttons using mouse
* **Point-of-View: Developer**
  + Java Calculator application
  + Must perform the following operations:
    - Addition
    - Subtraction
    - Multiplication
    - Division
    - Modulus Division
  + Login
    - Email must be checked for correct format
    - Password must be length 8 and combination of numbers/letters
  + Uses Shunting Yard Algorithm to convert expressions from infix to postfix
  + GUI must include (Java Swing)
    - Text-field for expression
    - Button for “Calculate”
    - Calculator buttons (e.g. 1,2,3,4,5,6,7,8,9,0,+,-,\*,/, etc)
    - Close app upon exiting calculator window
    - Separate Login window
    - Separate New User Registration
    - Invalid entry notifications
  + MySQL database
    - Username
    - Password
    - Email
  + Java APIs
    - Java MySQL with JDBC
    - Java Gmail server API

Use Cases

**Register**

Preconditions:

* Application is already open
* Login window is open

Main Success Scenario:

1. User selects “Register New User” option
2. Registration window opens
3. User types in:
   1. Email
   2. Username
   3. Password
4. User selects “Register” to complete registration.

Variations:

1. Username is already in database
   * 1. User is notified that the username is already taken
2. Password does not comply with length/format requirements
   * 1. User is notified of password requirements
3. Email is not valid
   * 1. User is notified of invalid email format

Notes:

* Automatic feedback next to the corresponding field that is invalid
* “Register” button does not need to be selected to determine validity of fields

**Login**

Preconditions:

* Application is already open
* User has already registered
* Login window is open

Main Success Scenario:

1. User enters username & password
2. User selects “Login” to login.

Variations:

1. User enters username that is not in database
   * 1. User is notified that username or password is wrong
2. User enters username that is correct but a password that is incorrect
   * 1. User is notified that username or password is wrong
3. User enters username that is not in database and a password that is incorrect
   * 1. User is notified that username or password is wrong
4. User enters invalid information three times consecutively
   * 1. User is taken into **Password Recovery** (see “Password Recovery” use case)

**Change Password**

Preconditions:

* Application is already open
* User has already registered
* User is logged in

Main Success Scenario:

1. User selects “Account”
2. “Account Settings” window opens
3. Under “Change Password” user types current password
4. Under “Change Password” user types new password
5. Under “Change Password” user types new password a second time to confirm
6. User selects button to confirm password change
7. Password is updated in database

Variations:

1. User enters incorrect current password
   * 1. User is notified that “Current Password” field is invalid
2. User enters two different passwords
   * 1. User is notified that the two fields do not match

**Change Email**

Preconditions:

* Application is already open
* User has already registered
* User is logged in

Main Success Scenario:

1. User selects “Account”
2. “Account” window opens
3. Under “Change Email” user types new email
4. User selects button to confirm email change
5. Email is updated in database

Variations:

1. User enters invalid email format
   * 1. User is notified that “New Email” field is invalid

**Password Recovery**

Preconditions:

* Application is already open
* User has already registered
* User has entered login information incorrectly three times

Main Success Scenario:

1. User is “locked out”; they are not allowed to re-enter login credentials until password recovery is completed
2. User enters email address
3. User receives email with new password corresponding to the username that was attempted to login with
4. User can now login with the new password

Variations:

1. User enters invalid email
   * 1. User is notified of no matching email in database
2. Username from attempted login is not valid
   * 1. User is notified that the username entered does not exist
3. Username from attempted login is not associated with the email given at Password Recovery
   * 1. User is notified that the username entered is not associated to the email given

**Computations**

Preconditions:

* Application is already open
* User has already registered
* User is logged in

Main Success Scenario:

User can

1. Add
2. Subtract
3. Multiply
4. Divide
5. Modulo divide

Variations:

1. User enters letters instead of numbers
   * 1. Letter keystrokes will be ignored
2. User enters numbers that cause integer overflow
   * 1. Catch overflow and notify user that expression cannot be calculated
     2. Use long, or Java.math.BigInteger to perform large calculations
3. User divides by zero
   * 1. Return “Undefined”

**Logout**

Preconditions:

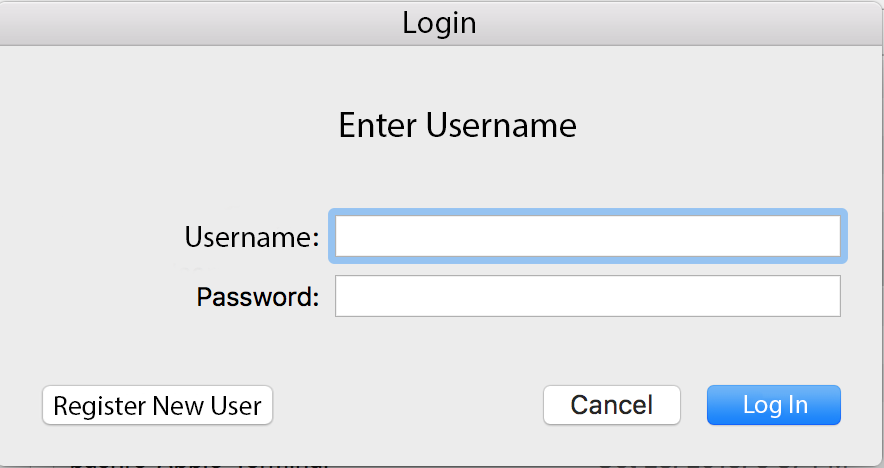
* Application is already open
* User has already registered
* User is logged in

Main Success Scenario:

1. User selects “Logout” from main calculator window
2. User is logged out
3. User is taken back to log in screen

GUI Design

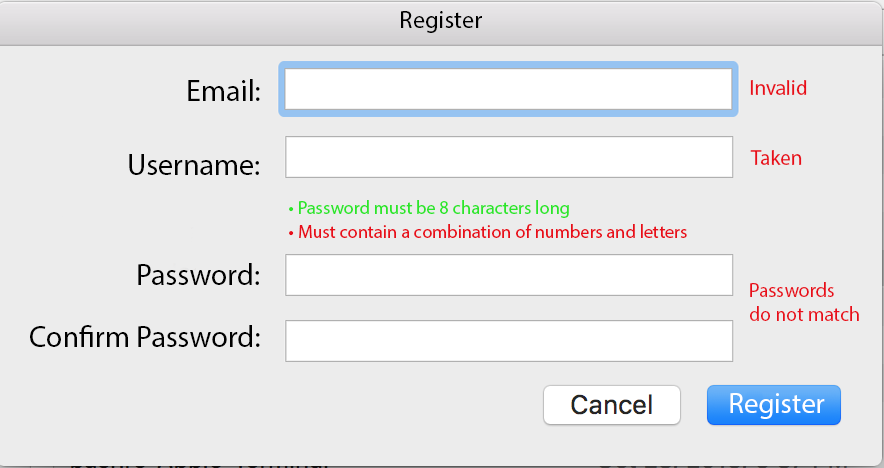
**Login**



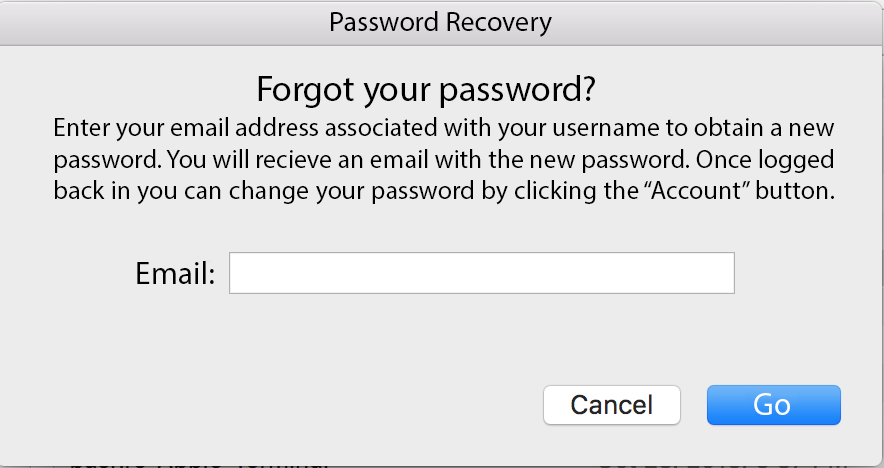
**Login Error**



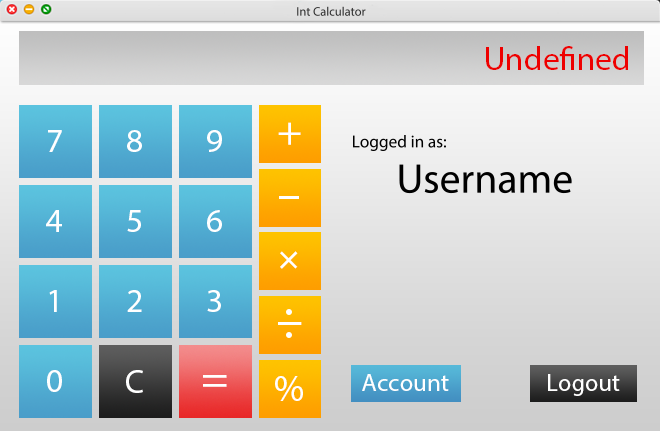
**Register**



**Password Recovery**



**Calculator**



**Account Settings**

