

# Project Proposal: *Python 2FA GUI*

*Name: Matthew Fredericksen*

*Course: CSCE 3550.070*

*Date: 06/03/2020*

## Objective

I have chosen to complete the “Two Way Authentication System” project option. I chose this project primarily because I have written programs that send text messages in the past, so I am familiar with that process.

## Proposed Methodology

### Programming Language:

I will use Python 3.7 or 3.8 (depending on if all libraries support 3.8) for the entire project. I expect the program to be compatible with any version 3.6+.

### Tools and Libraries:

- [passlib](#): library for hash functions. I plan on using bcrypt.
- [cryptography](#): library for encrypting information that must be later decrypted (the phone number).
- [sqlite3](#): library for managing a sqlite database for storing user account information.
- [twilio](#): an API for sending text messages.
- [kivy](#): GUI creation library. I am already somewhat familiar with PyQt, but it seems that most people prefer kivy.

### Program outline:

1. Open a basic GUI with options for signing in and creating a new account.
  - Password fields will be obfuscated as the user types.
2. If creating a new account:
  - An entry will be created in the database containing the username (as primary key), the hashed password, and the encrypted phone number (probably encrypted using the unhashed password as the key, since that should be available later for decryption).

- Note: I don't believe the phone number can be "hashed" as specified in the Project Topics, since I would have no way of retrieving the phone number from its hashed value. However, it can be encrypted.

3. If attempting to sign in to an account:

- If username and hashed password do not both match an entry in the database, inform the user and allow them to retry.
- If the user name and hashed password find a match, retrieve the row from the database and decrypt the phone number using the unhashed password the user just entered. Generate (and hash, possibly) a random 6-digit number and text it to the phone number, then wait a certain amount of time for the user to enter the number.

4. Upon successful sign in, display a success indicator and a logout button.

**Development Process:**

I will start with implementing the necessary functionality (password hashing & verification, account creation, texting, etc.), and once that is complete I will hook everything into the GUI. This should work fine since the GUI will be simple. I may switch to developing them in tandem if I learn something unexpected about the GUI creation library.