**SQLite Database**

In the project we need a Database to store the Instructors and Students information, like ID, user name, semester, course number and the password.  
We used SQLite database, version three for our project. You can import it using : **import sqlite3**

**What is SQLite?**

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

**Why SQLite?**

* SQLite does not require a separate server process or system to operate (serverless).
* SQLite comes with zero-configuration, which means no setup or administration needed.
* SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
* SQLite is self-contained, which means no external dependencies.
* SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

**Tables:**

We create two tables for the Students and instructors that’s include:   
(ID INT PRIMARY KEY NOT NULL,   
Name TEXT NOT NULL,  
 Course TEXT NOT NULL,  
 Semester TEXT,  
 Email TEXT,  
 Passwd TEXT NOT NULL,  
 Salt TEXT NOT NULL)

We put database server in the server and communicate with the instructors and Students using TCP.

**Reference**: https://www.tutorialspoint.com/sqlite/sqlite\_quick\_guide.htm

# **MD5 Passwords with Salt**

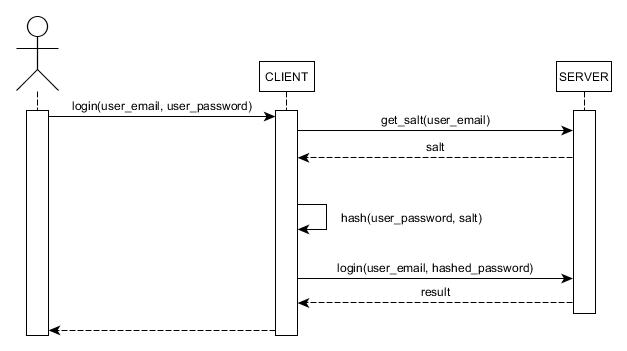
In order to save the passwords of the users we need hash algorithm, we use MD5 with salt.

First of all we create a random salt for each user in sign up process using SHA512: **salt=crypt.mksalt(crypt.METHOD\_SHA512)**  
You should import **: import hashlib** and **import crypt** to use this function.

Then we create a hash for the summation of password and the salt using md5 algorithm, and we save the result in the database.  
**str=salt+passwd\_field  
hashlib.md5(str.encode())**

**Verification:**

For verification we use the following diagram.



Steps:

1. The user send the ID to the server
2. Server send back the salt value to the client
3. The client make a hash to the password +salt and send the result to the server
4. The server check if the received hash equals the hash in the database and send the result to the user

**Reference:** https://docs.python.org/3.5/library/hashlib.html