**Introduction**

**Script: Hello everyone. In this video segment, we would like to talk about system modeling for our project. We will show you one sample of our use cases and test cases that we used to develop our class and sequence diagram.**

**Use Case**

* **Identifier:1**
* **Iteration: 1**
* **Summary: GSU email sign up and verification**
* **Actors: User, Database**
* **Basic Course of Events:**

1. **User signs up for an account to use the system**
2. **System sends verification email**
3. **User verifies GSU email via system-generated email**

* **Alternative Paths: None**
* **Exception Paths:**

**1. User enters non-GSU email.**

**3. User can’t reuse the email.**

* **Extension Points: User is allowed to create new account**
* **Trigger: User is in need of study community within GSU**
* **Assumptions: User is a GSU student**
* **Precondition: User must have a valid GSU email**
* **Postcondition: User requirements are sufficient to create account**
* **Author: All**
* **Date: 02/14/2018 (Revised: 3/5/2018)**

EVER:

Script: Our first use case is the sign up and verification process. The actors in this particular use case are the user who visited the website and the database that stores user information. The basic course of events are user signs up for an account to use our system. The system will then send a verification email to the email the user supplied to sign up with. The user will need to log into his or her email to click on the confirmation link, to verify that this is a valid email. There are a number of requirements and prerequisites placed in order be able to sign up. We will discuss those in our test case.

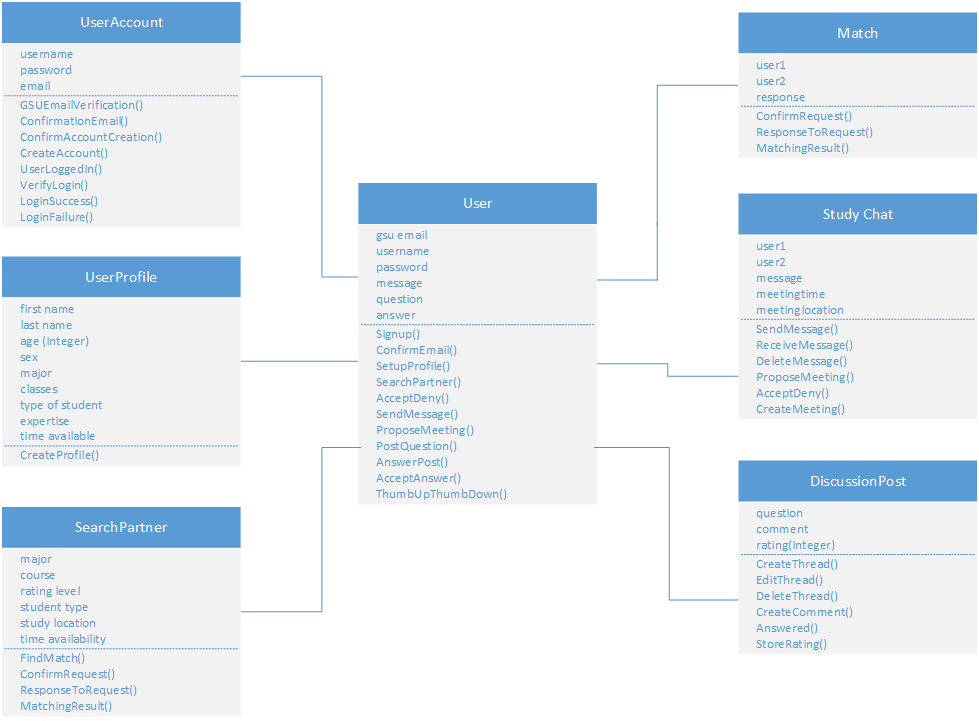
**Test Case**

1. **Sign Up and Verification**

|  |  |
| --- | --- |
| **Description:** | **Ensures that a valid GSU student can sign up.** |
| **Test Inputs:** | **GSU email to verify.** |
| **Expected Results:** | **Verification email is sent to the GSU email.** |
| **Dependencies:** | **None** |
|  |  |
| **Test Steps** | 1. **Form values are not left blank.** 2. **Entered email ends with student.gsu.edu** 3. **Password is valid.** 4. **Verify that the retrieved email doesn't already exist in database.** 5. **Send verification email.** 6. **Confirm sign up.** 7. **Save user into Users database.** |
| **Owner:** | **All team members (Revised: 3/718)** |

Script: Our service is available to GSU students only who have a valid GSU student email of @student.gsu.edu. For the sign-up form itself, we have a number of requirements and pre requisites in order to sign up. All fields must have a value and cannot be left blank. The email must associated with @student.gsu.edu address. Passwords must be at least 8 characters and email has not previously been registered in the database. User must log into their GSU email to confirm that the email is valid.

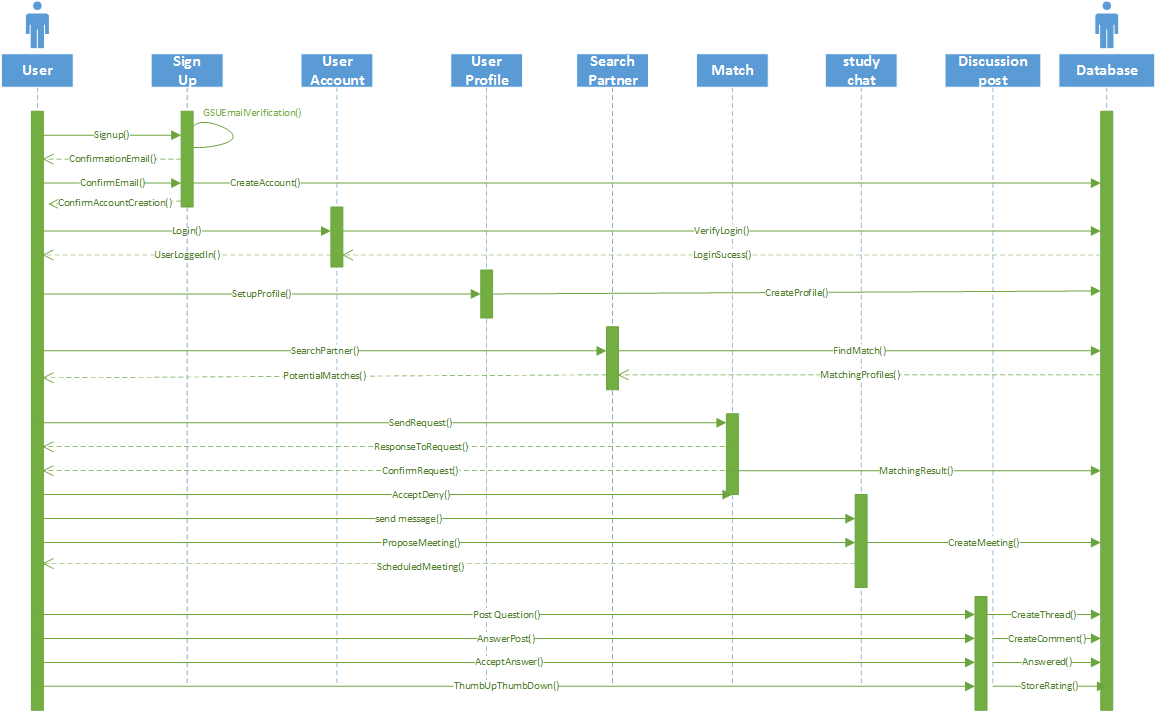
**Class Diagram**



Script: The class diagram is based on the use cases. As you can see, we have 9 objects: User, Database, SignUp, UserAccount, UserProfile, SearchPartner, Match, Study Chat, and DisscusionPost. In each of these objects, we have the attributes and the operations performed by these objects.

Take for example, the SignUp object. We have username, password, email as the attributes for this object and the operations performed by this object are GSUEmailVerification(), ConfirmationEmail(), ConfirmAccountCreation(). This object is linked with the User and Database object in an aggregation associations with multiplicity 1 to 1 relationships.

**Sequence Diagram**



Script: From the class diagram, we developed the sequence diagram to model the behaviors of these objects with each other. Let’s take a look at the Signup object once again. As you can see, the User initiates the Signup() operation on the Signup object. The Signup object will verify the user’s GSU student email address until a valid email is entered. The Signup object will then send a confirmation email back to the User. The user will need to confirm that the email is valid. Once confirmed, the Signup object will create an account for the user in the Database and confirms with the User that an accounted has been created to allow the User to interact with other objects.