****

**Project Report**

**Course Code:** CSE-427

**Course Name:** Software Quality Assurance and Testing

**Semester:** Spring’19

**Faculty Name:** Shaikh Shawon Arefin Shimon

**Faculty Initial: SAS3**

**Project Name:** Software testing for Client Server Communication

**Group Number: 13**

**Group Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial** | **Name** | **Student ID** | **NSU Email** |
| 1 | Khondaker Tanzim Alam | 1421422042 | tanzim.khondaker@northsouth.edu |
| 2 | Md Majedur Rahman | 1420497042 | majedur.rahman@northsouth.edu |

**GitHub URL:** <https://github.com/nsuspring2019cse427/Group13>

**Project Description:**

Now a days it’s very common to have two side of an application. One is frontend and another is backend. Basically, frontend is mainly used for user interaction and backend is hub for every interaction of frontend. That means we a such a system where frontend and backend need to perform their job continuously error and bug free.

That’s why we are decide to build such a system where we any ensure that error and bugs will be minimized. We will ensure that by testing our code. Therefore, have two components in our project. One is server (Backend) and other one is client (Frontend). Mainly we are trying to build a bug free end to end solution where a client can talk with server without any discomfort. For that on server side we will do Unit testing, Integration testing and for client side we perform mainly UI testing.

Client

Server

**System:**

**Client: Android Application**

**Language: Kotlin, Java, Android XML**

**Server: Spring Boot Application**

**Language: Java**

**Database: PostgreSQL**

**Tools/Frameworks:**

Unit testing: Junit, Mockito, JUnitParams

UI testing: Espresso, Kakao

**Input Space Portioning:**

We have approached to complete our Input Space Partitioning (ISP) by Functionality-Based Approach.

1. **ISP for validateUserName:**

**One parameter:**

user

**Characteristics:**

Username is empty

(true, false)

1. **ISP for validateUserPassword:**

**One parameter:**

user

**Characteristics:**

UserPassword is null

(true, false)

UserPassword is empty

(true, false)

UserPassword has length less than 7

(true, false)

1. **ISP for validateUserEmail:**

**One parameter:**

user

**Characteristics:**

Last part of UserEmail has “@northsouth.edu”

(true, false)

Email is valid

(true, false)

Email is invalid

(true, false)

UserEmail is empty

(true, false)

1. **ISP for signUpUser:**

**One parameter:**

user

**Characteristics:**

Username is empty

(new ResponseCommon<User>(true, "", savedUser),

new ResponseCommon<User>(false, result, null),

new ResponseCommon<User>(true, ex.getMessage(), null))

UserPassword is null

(new ResponseCommon<User>(true, "", savedUser),

new ResponseCommon<User>(false, result, null),

new ResponseCommon<User>(true, ex.getMessage(), null))

UserPassword is empty

(new ResponseCommon<User>(true, "", savedUser),

new ResponseCommon<User>(false, result, null),

new ResponseCommon<User>(true, ex.getMessage(), null))

UserPassword has length less than 7

(new ResponseCommon<User>(true, "", savedUser),

new ResponseCommon<User>(false, result, null),

new ResponseCommon<User>(true, ex.getMessage(), null))

UserName is present and active

(new ResponseCommon<User>(true, "", savedUser),

new ResponseCommon<User>(false, result, null),

new ResponseCommon<User>(true, ex.getMessage(), null))

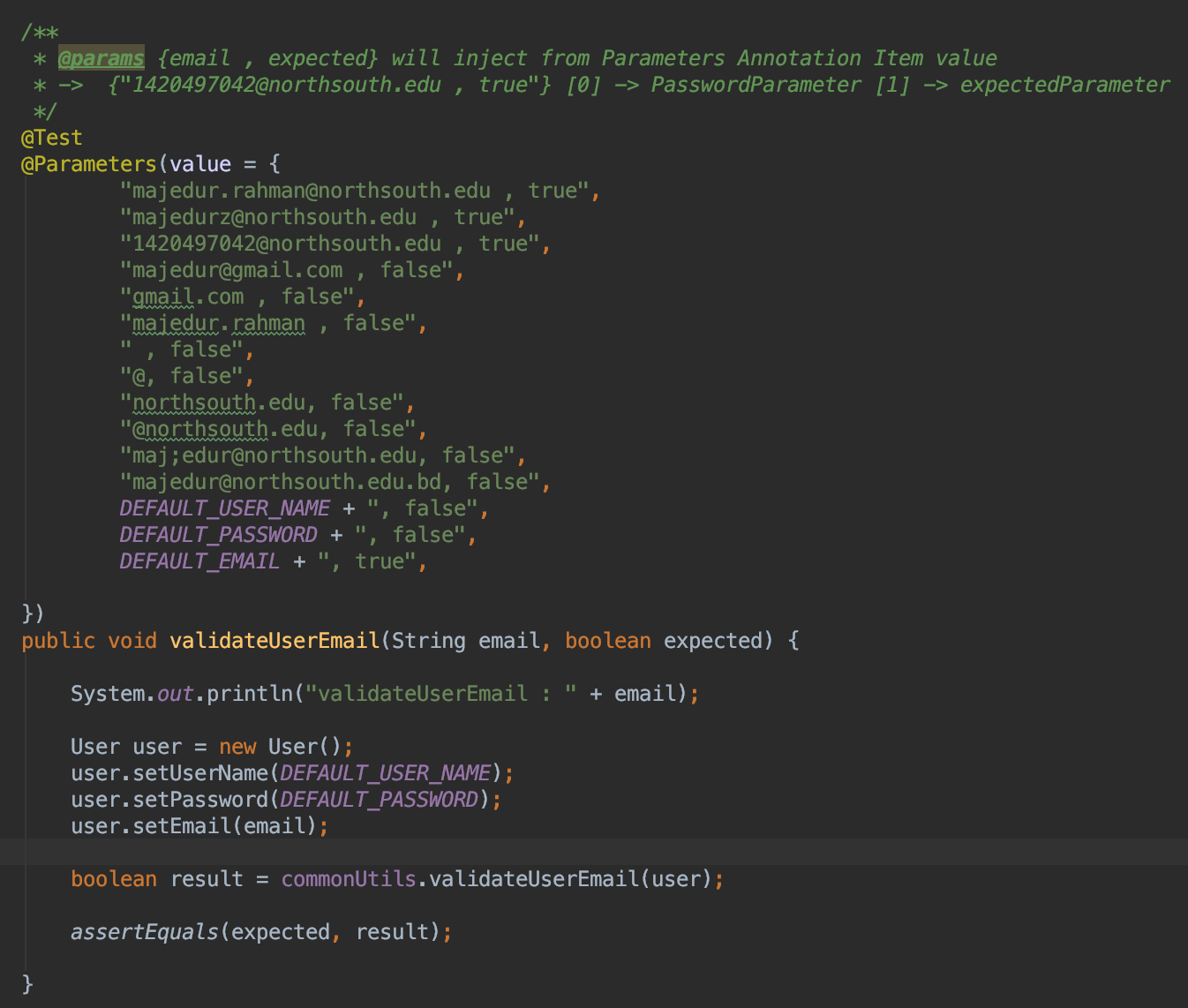
1. **ISP for getAllActiveUser:**

**No parameter**

**Characteristics:**

All the active user

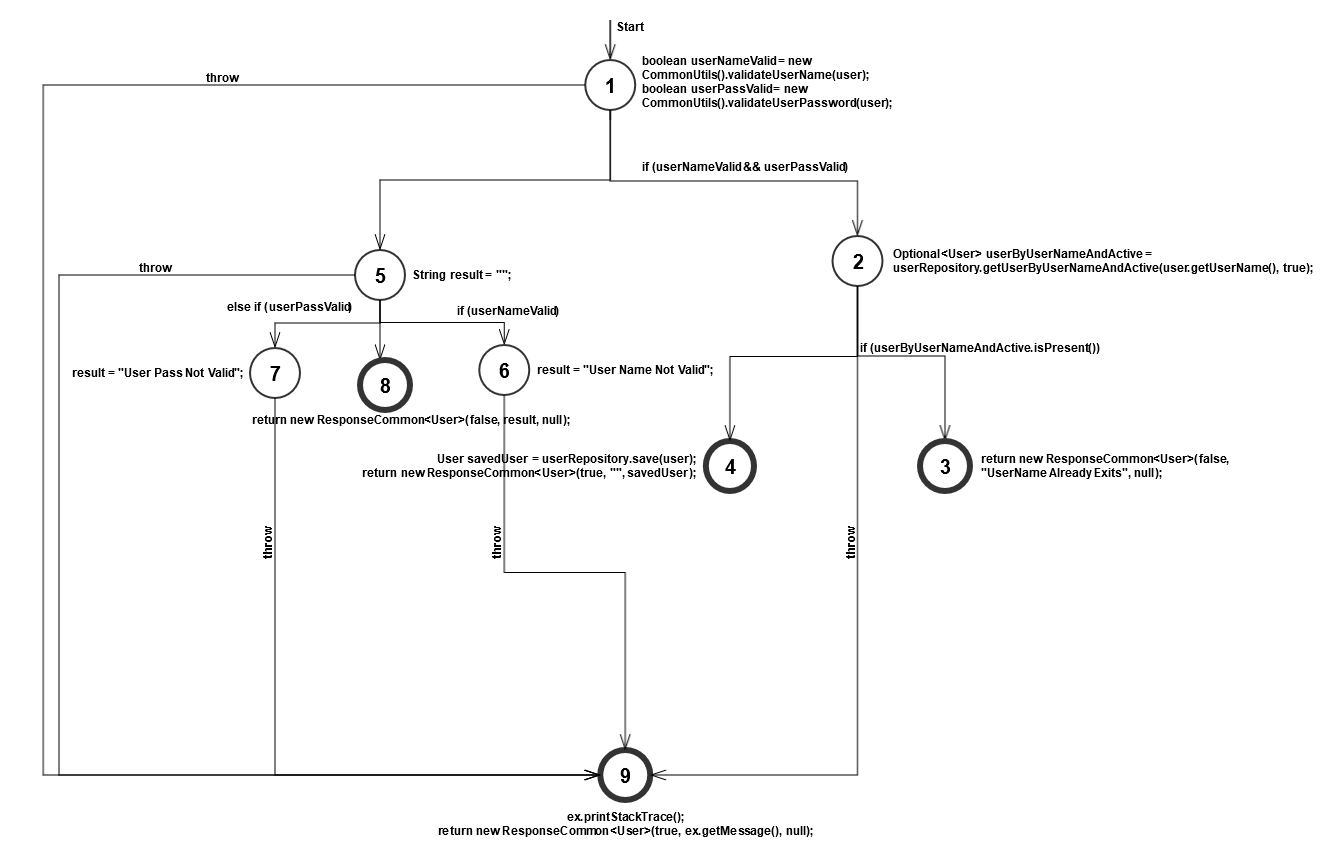
(userList, null)



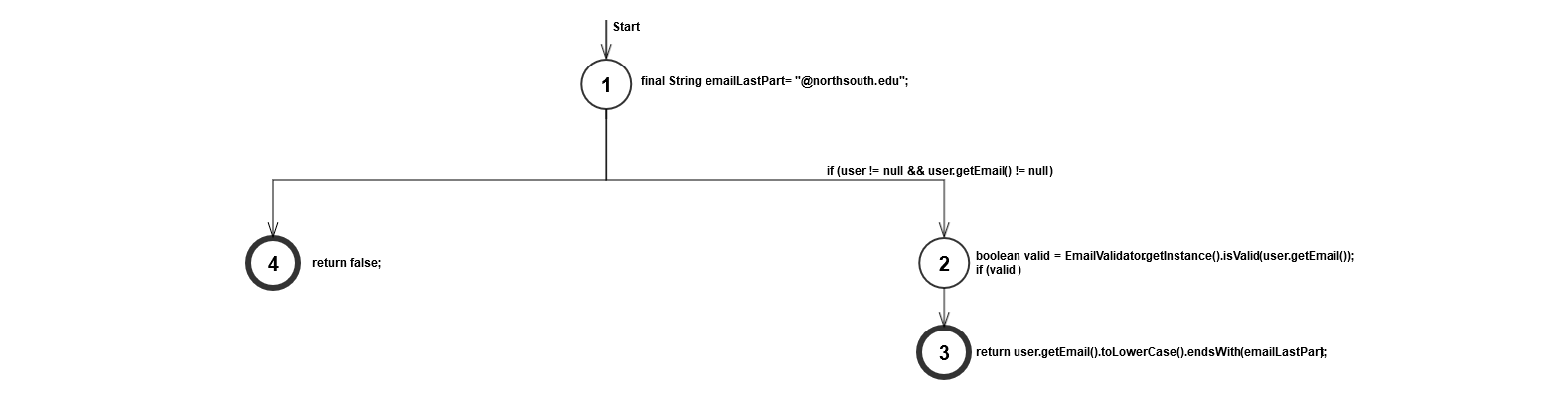
**Figure1: Sample screenshot of one of the tested methods.**

**Control Flow Graphs (CFGs):**  


**Graph1:** CFG for validateUserName

****

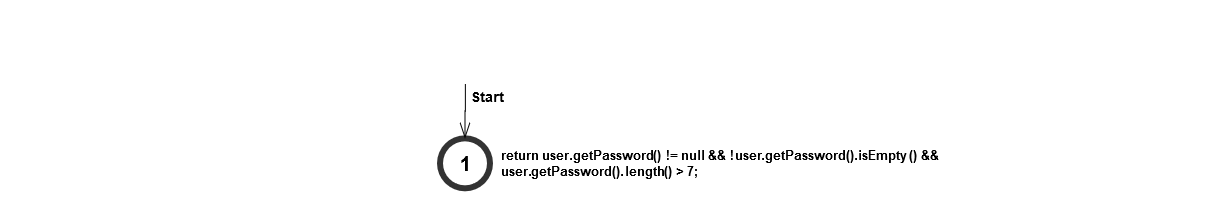
**Graph2:** CFG for signUpUser



**Graph3:** CFG for validateUserEmail



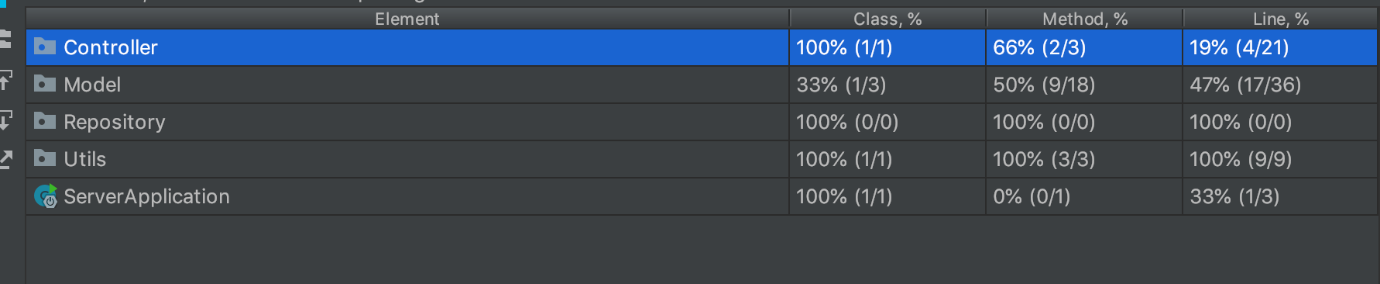
**Graph4:** CFG for getAllActiveUser



**Graph5:** CFG for validateUserPassword

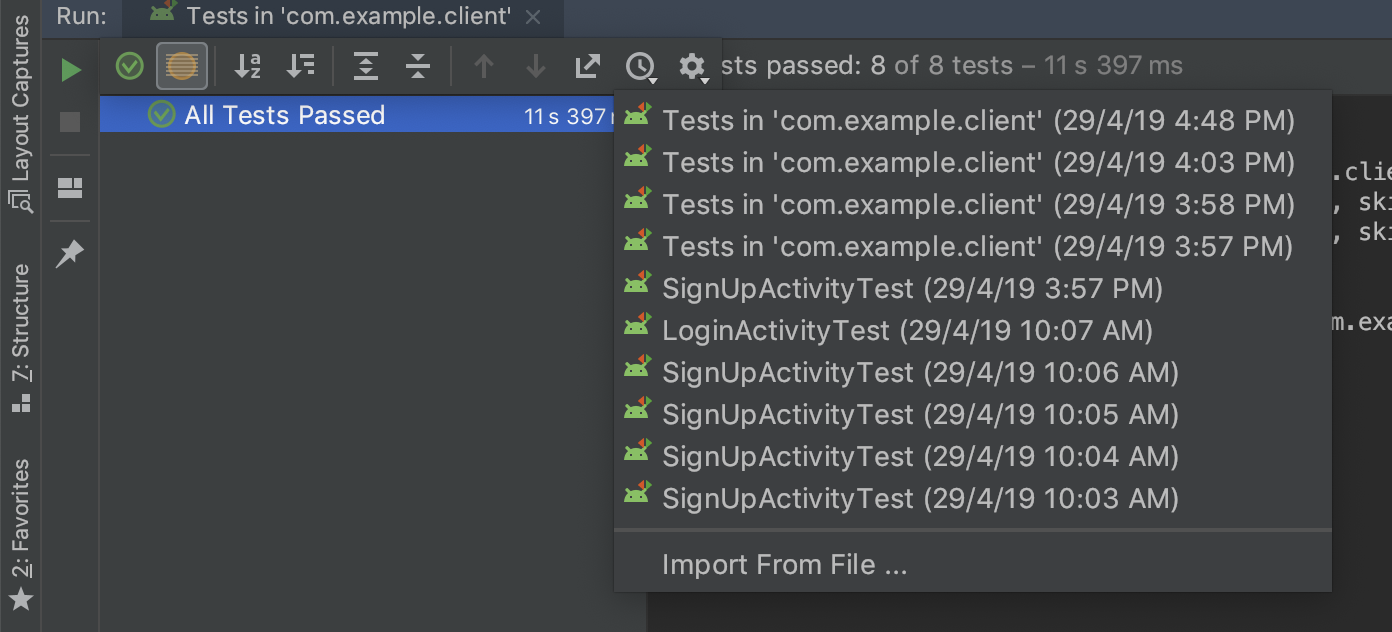
**Result:**

**Unit Testing:**



**Figure2: Sample screenshot of Unit Testing.**

**UI Testing:**



**Figure3: Sample screenshot of UI Testing.**