**Integration Test Plan and Test Record Outline**

**1. Introduction**

**1.1 Revision History**

For the practical part of the subject software engineering we have being asked to design a desktop application that will allow the creation of chat with a single person and with group of people. It will require to apply some principles in order to complete the application.

**1.2. Purpose and Scope.**

The purpose of this document is to list the test that must be executed so that the application is checked against the requirements.

**2. Integration Strategy**

**2.1 Entry Criteria.**

In other to start the integration, many things were taken into account:

* **The requirements:**
* Each user needs to create an account first
* The application allows exchanging messages between users.
* Message exchange must be available at no cost
* The software is available on desktop
* The software will allow exchanging messages in a group of users which is created by the administrator of the group
* One can send text messages with the software
* **The constraints criteria**

The different constraints criteria are:

* The system must be written in java
* The system must have a database which can store each user data
* The final product is a desktop software and will be available on any kind computer
* Every password will be encrypted to assure security

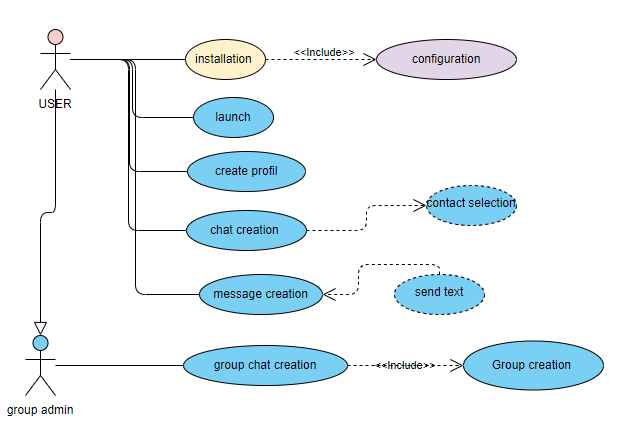
**2.2 Elements to be Integrated.**

The elements that must be integrated are:

* The group window
* The Message window
* The connection window
* The user test
* Register GUI
* The main function

**2.3 Integration Strategy.**

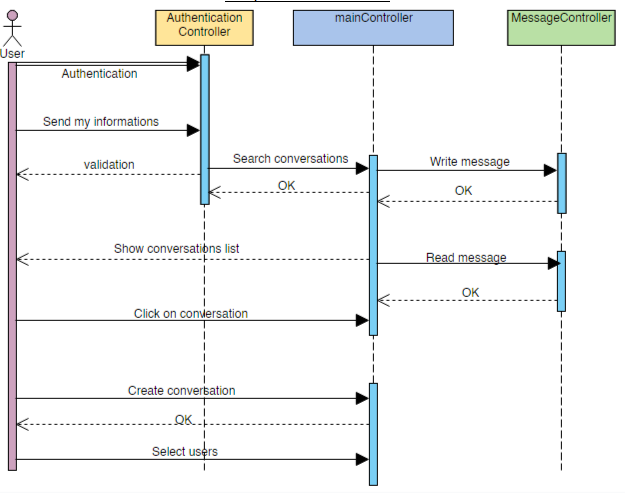
To start the integration, an UML grouping all the objects and methods were made according to the sequence diagram, the use case digram and the entry criteria.



After the completion of the UML with follows the Client server and the MVC design, each developer took classes to be implement. As the integration management was done with the Agile Method which focus on delivering codes faster and testing them.

**2.4 Sequence of Feature/Function and Hardware/Software Integration**

**2.4.1 Software Integration Sequence.**



**.**

**3. Individual Steps and Test Description**

The tests were made for each methods of the classes

* The group window test

The test made in this class are

* The test to get the contact name
* The test to get the group name
* The test to get the contact list
* The test to send the first message to the group
* The test to get the group ID
* The test to create a group
* The test to check the action performed
* The test to check the key pressed
* The test to check the key released
* The test to check the key typed
* **The message window test**
* The test to get the group member
* The test to load contacts
* The test to load messages
* The test to get pseudo by Id
* The test to get Id from pseudo
* The test to get group name
* The test to check contact exits
* The test to add contact
* Test to get group ID from group name
* Test to send message
* Test for action performed
* **The user test**
* Test for user
* Test to get pseudo
* Test to set pseudo
* Test to get user Id
* Test to set user ID
* Test to get password
* Test to set password
* **Connection window test**
* Test to get pseudo
* Test to get password
* Test for connection window
* Test to check connexion
* Test for action performed
* **Register GUI test**
* Test to get pseudo
* Test to get password
* Test for the Register GUI
* Test to check if a user already exists
* Test to add user
* Test for action perform
* **Main**
* Test the main

**3.1 Subsystem Integration Test Description in order**

* **1st:**  the creation of the database and the test of the database
* **2:** the design of the GUI and the test of the GUI
* **3:** the creation of the User class who is identified by the pseudo and the password and the test of the getter and setter of the pseudo and password which identifies the user
* **4:** the creation of the register GUI which requires the pseudo and password of the user and test if the user exits or not if does not exist we test the add User and test if the action is performed
* **5:** During the register We test the connection window which by his turn test the connexion, test the pseudo and the password from the database.
* **6:** After testing all the previous classes we then test the message window and group test. That include the test of load contacts, load message, the test of either adding a user or checking if the contact exists, the test of sending message to one person or in group by testing if we can get contact name and contact list to create groupe.

**Y**

**3.2 Final Functional Tests.**

These tests are intended to confirm that the system has been successfully integrated and is now ready for full System Test by Engineering.

**4. Tools and Test Equipment Required**

The equipment use for the integration testing are:

* Hardware: a computer
* Software: Eclipse,JUnit, , Github

The additional tool required is internet to able to launch the final application and to access the database

**5. Program Stubs and Test Data Required**

In order to connect to database, we chose to use a functionality named ‘jdbc’ which allows us to connect our code to a server, which is a database in our case.

**6. Problem Recording and Resolution**

We encountered only one main problem which was to find a website that can give us a free database. We found multiple free services but all of them did not allow connections with ‘jdbc’, except one which is ‘freehostingsql.net’. Therefore, we chose this service to host our database.

**7. Suspension, Restart, and Exit Criteria.**

The implementation of the GUI was suspended and restarted with new features (button, frame,…) as the classes were implemented