

Republic of Cameroon

Peace- Work-Fatherland

First Semester

Academic Year 2021/2022

UNIVERSITY OF BUEA

P.O Box 63

South West Region

Tel: (237) 332 2134

Fax: (237) 332 2272

Telex: 5155KN

Faculty of Engineering and Engineering and Technology

* Department: COMPUTER ENGINEERING (FET)
* CEF 349: Analysis and Design of Information Systems

**DESIGNING A SIMPLE MESSAGING SYSTEM FOR UB**

**Under The Supervision of**: **Mme TIAKO FANI MICHELE**



|  |  |
| --- | --- |
| **NAMES** | **MATRICULES** |
| POKAM NGOUFFO TANEKOU | FE21A299 |
| DJEUTIO QUOIMON ANDERSON ROY | FE21A169 |
| NDI JEFFERSON NCHOTU | FE19A073 |
| TIANI PEKINS EBIKA | FE21A325 |
| AKENGNI KEANLI EMMANUEL | FE21A132 |
| TATA CHRISTIAN EJAFF | FE19A105 |
| TANDONGFOR SHALOM CHANGEH | FE20A111 |
| OJONG-ENYANG OYERE | FE21A297 |
| ALEANU NTIMAEH ENOW | FE21A134 |
| MEWOABI NGUEFACK DORE | FE21A239 |
| SIRRI THERESIA | FE21A306 |

**Problem definition:**

Develop a simple messaging system for UB, which has the following functionalities:

* 1. Students can sign up freely for an account.
  2. Students can send and receive messages from other friend.
  3. Messages should not have attachments.
  4. A message can be sent to only one person at a time.
  5. Only one person can be copied.
  6. Read messages are marked as ‘Read’ and unread as ‘Unread’.
  7. Add any interesting features to your database if any.

**What to do:**

We are to:

* 1. Design an ER model for the system
  2. Convert the model to a relational database model
  3. Draw a process flow diagram.

**Might require the following:**

* + - **Use case diagram**
    - **Activity diagram**
  1. Design a front end to interact with the database to meet the above described problem.

**Required Software:**

1. **MYSQL Workbench** will be required to create our Database
2. **EdrawMax** will be used to design the ER model, Relational database schema and process flow diagram of the system
3. **NetBeans JavaSwing** will be used to design the front end

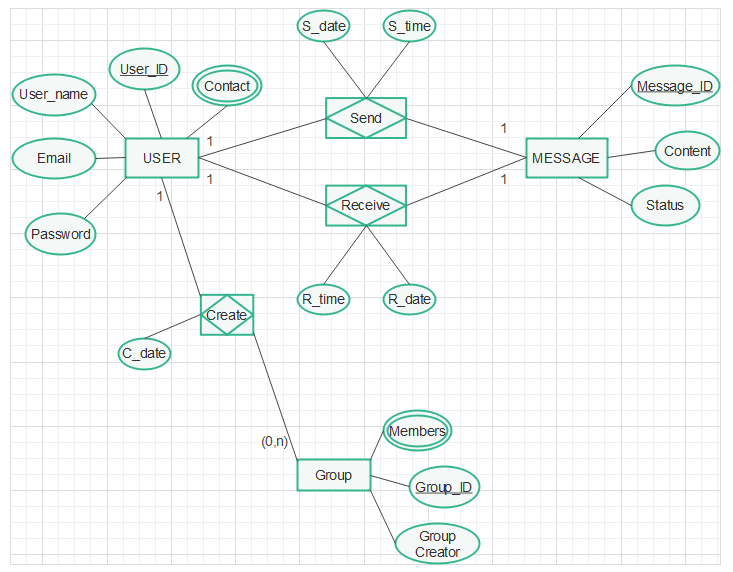
Throughout our design, we implemented the use of the **Waterfall Model** since we are to submit the finally prototype all together with the implementation processes

**DESIGNING**

As we earlier said we shall use different software’s for the implementation and design of our messaging app.

For our system, the main users are STUDENTS

1. **ENTITY RELATIONSHIP MODEL**



From the above diagram we can see we’ve various entities and specific attributes associated. For our User entity we’ve the User\_ID as Primary key, the contact as a multivalued attribute since a user’s phone number can change with time. Also we have User\_name, email and password as basic attributes.

We also have a Message entity which has a Message\_ID as primary key with content and Status as other attributes.

For the case of the group, it has as primary key Group\_ID, Members as multivalued attribute since we’ll have more than a user in a group and we also have a Group creator which identifies the group admin.

Also, on our model we can find associative entities with their various attributes. We also included the different cardinalities on our main entities that is;

* A user can send one message at different time and a message can be sent by only one user
* A user can receive one message at different time and a message can be received by a user
* A user can create as many groups as possible but can decide not to create, any created group can have only one creator.

1. **RELATIONAL DATABASE MODEL**

For our relational database, we’ve what follows.

USER (**user\_ID**, Username, Fullname, contact, email, password)

GROUP (**Group\_ID**, Groupname, Group\_Cname)

MESSAGE (**Message\_ID**, Content, status, **user\_ID\***)

SEND (send\_time, send\_date, **user\_ID\***)

RECEIVE (receive\_time, receive\_date, **user\_ID\***)

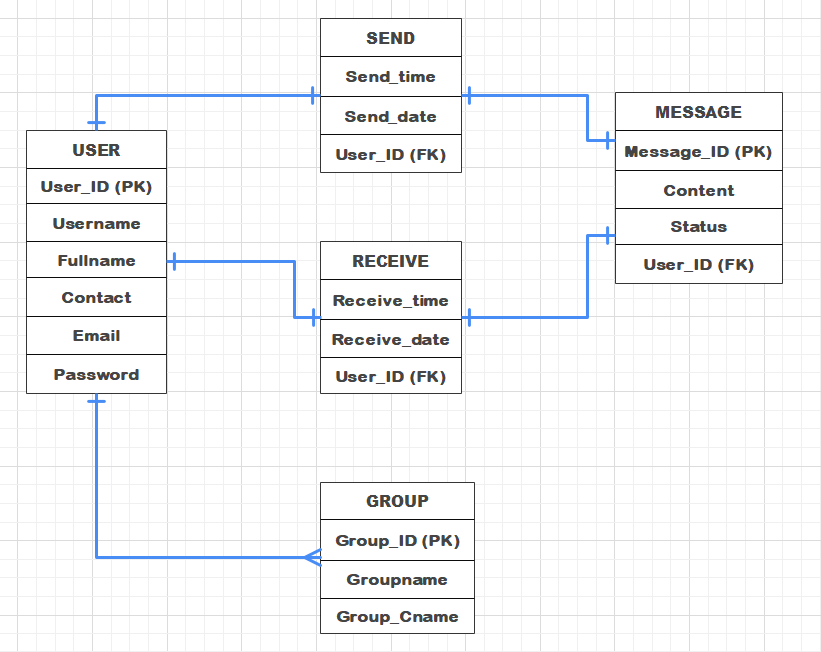
In the USER class, we’ve **user\_ID** as primary key that uniquely identifies the class and all its attributes are normal attribute

On the Group class, we’ve **Group\_ID** as a strong Primary key since it can be used to uniquely identify the whole class

For the message table, we’ve the **message\_ID** as primary key and we also included a **user\_ID**\* as foreign key since a message is sent to a user

For the Send class we included as foreign key **user\_ID**\* and since we didn’t have a primary key which could uniquely identify this class, we took the whole class a super key. We did same for the Receive class

Below is an illustration of our Relational Schema



The diagram above was designed to ease our understanding.

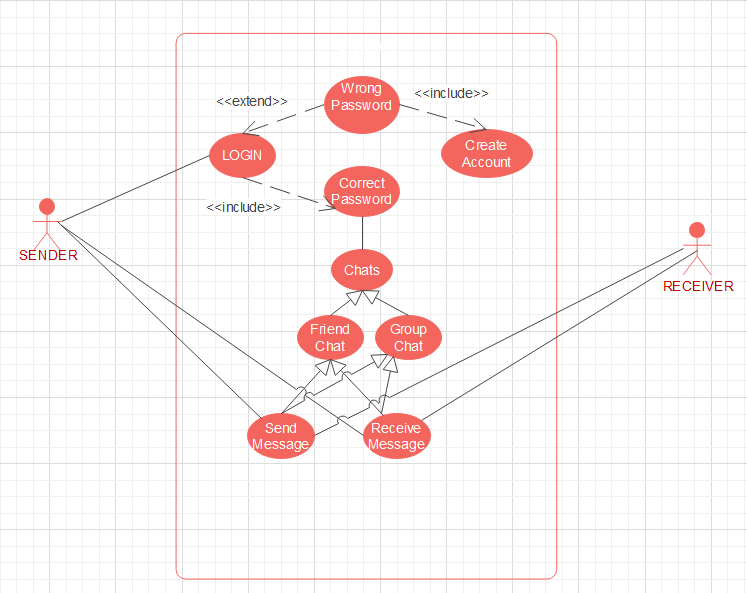
1. **PROCESS FLOW DIAGRAM**

For the process flow diagram, we included a simple use case diagram and an Activity diagram

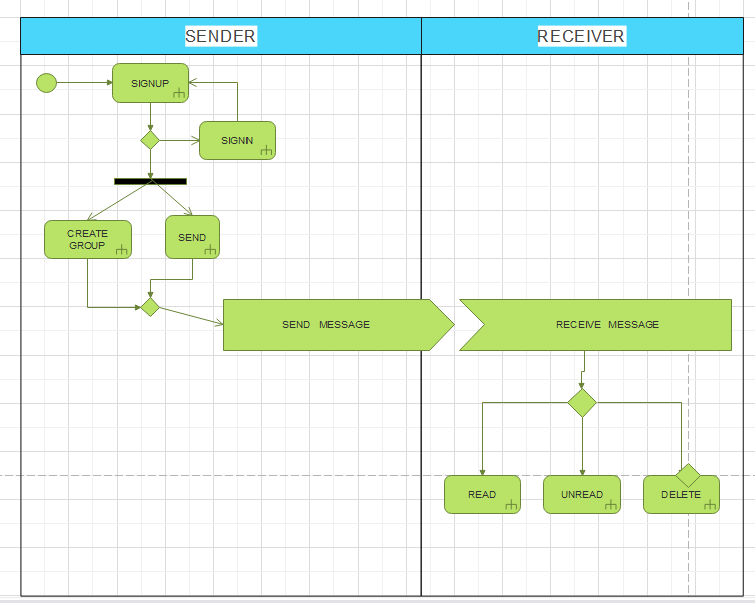
* **USECASE DIAGRAM**

For our use case diagram, we’ve two actors i.e. a Sender as primary actor and a Receiver as secondary actor. Here we made it simple, the sender can Login into his account if he has any and create an account if he doesn’t have one. Once into his account he can chat with friends or even have a group chat which includes sending and receiving messages and that goes as well for a receiver since being a receiver in a messaging system implies you that you can also be a sender.

Below is an illustration of our explanation



* **ACTIVITY DIAGRAM**



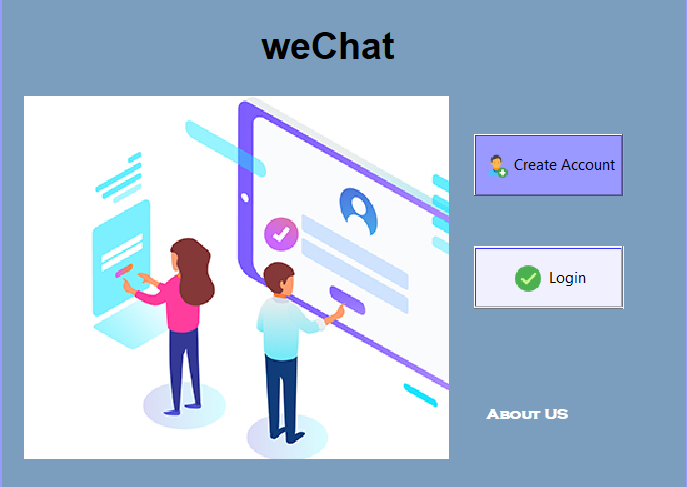
The activity diagram above shows the different process we’ve in our system i.e.

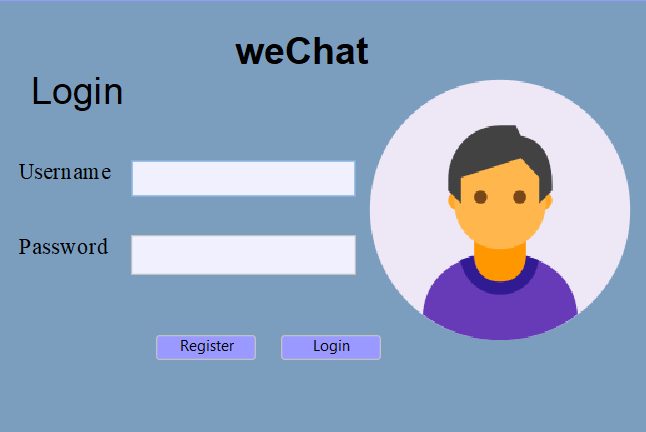
When you get into the system, you can sign up if you already have and account or if not you’re redirected to sign in in order to create an account before continuing. Once done, one can create a group if he/she wants one or just chat with their contact (friends).

When a message is sent and received on the other hand, the receive has 3 options i.e. He can Read a message, delete a message and the messages he didn’t read will be marked as unread. Also, as a receiver one can perform the same actions as a sender.

1. **FRONT END DESIGN**

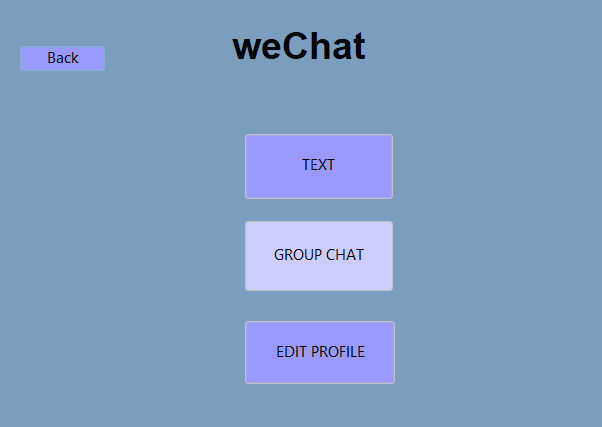
As earlier said, we used **NetBeans JavaSwing** to design a front end that can ease interaction with our database. Below is a sample of different pages of our system

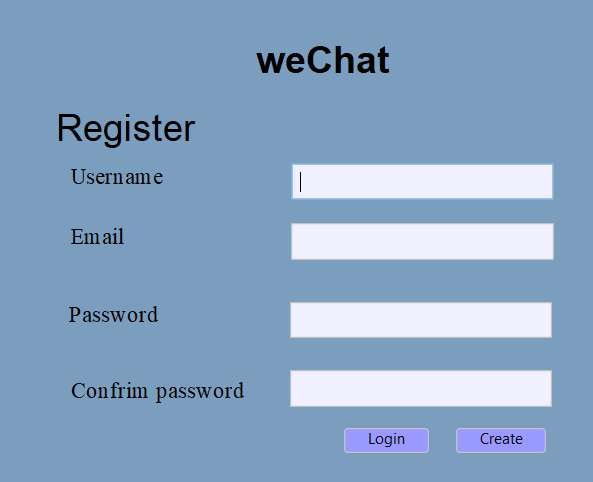




**Here we’ve the login page**

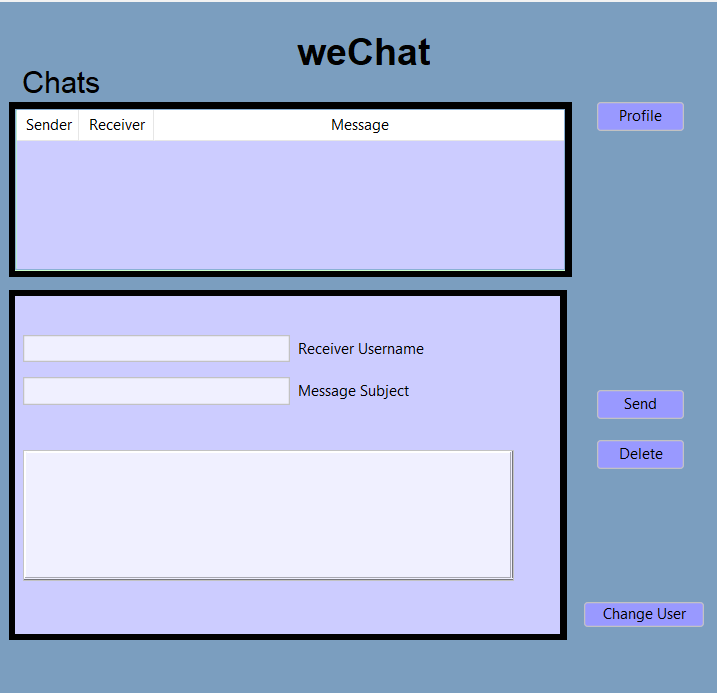
**Here we’ve the Home screen where we can Login or create an account**

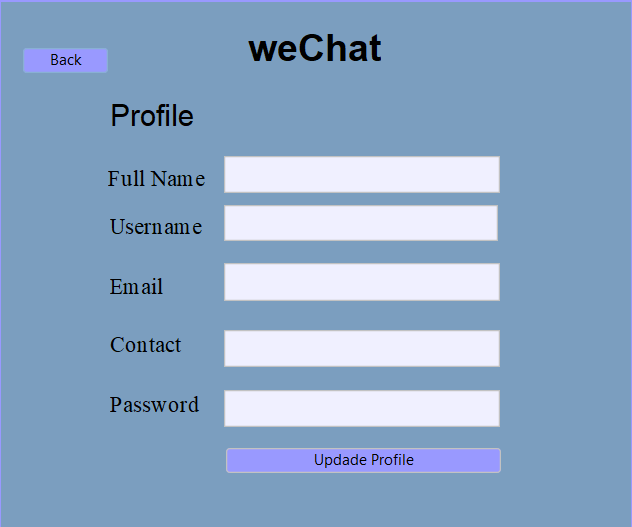




**Here we can switch to friend or group chat and edit your profile**

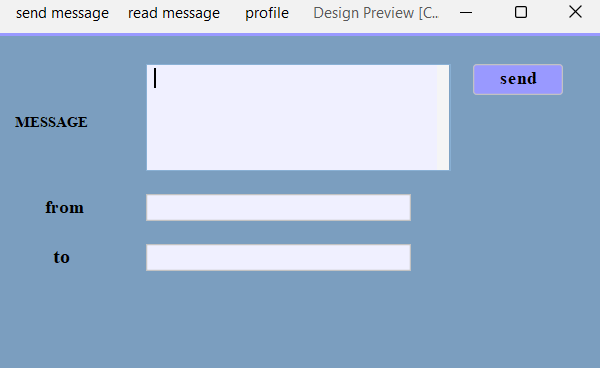
**We’ve the registration page**

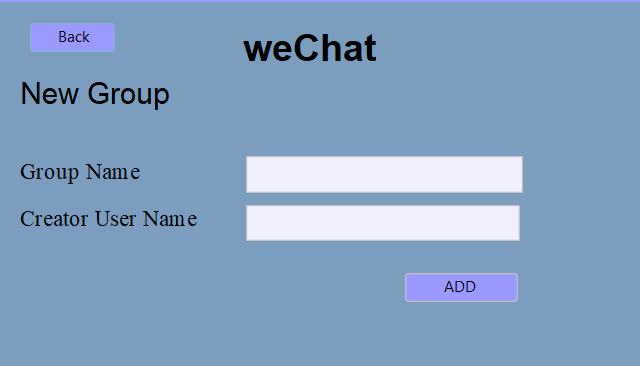




**In this page, the student can change his information**

**Here the Students chat page**





**This page is to create a group**

**Here we’ve the chat page**

**Here we’ve the view message page**

