Concordia University

SOEN 6471 – Advanced software Architecture

*Professor Peter Rigby*

Facebook chat Instant Messenger (FBCIM)

|  |  |  |
| --- | --- | --- |
| Student ID | Name | E-mail Ids |
| 6228569 | Bala Arun Reddy Vatti | [balaarunreddyv@gmail.com](mailto:balaarunreddyv@gmail.com) |
| 5972515 | Deepu Kumar | Kumar.deep@live.in |
| 6249922 | Pankaj Kapania | [pankajkapania@yahoo.com](mailto:pankajkapania@yahoo.com) |
| 1370189 | Paramjeet Singh | [er.paramjeetsingh@yahoo.com](mailto:er.paramjeetsingh@yahoo.com) |
| 1255185 | Gilles Desrochers | [gilles.desrochers@hotmail.com](mailto:gilles.desrochers@hotmail.com) |

# Project Description

## A general overview of the project and why it interests us:

Facebook chat IM (FBCIM) is a free and open source desktop instant messenger (IM) application that allows chatting with all Facebook contacts in real time with its unique, swift and easy to use tabbed message window interface. You can see and respond to chats right from your desktop. Whether you're browsing other websites or using another app, you don't have to click away to stay connected. It supports the open-source XMPP protocol, connects primarily between Facebook users and synchronizes with the website. It’s written in Java and released under the GNU LGPL.

The tabbed interface supports chatting with various friends at once. You can minimize the application and be notified with instant alerts even if your web browser is not open in Facebook. The application will require you to disclose your location when sendind messages. FBCIM currently supports all Windows operating systems (including Windows 98, SE, ME, server, XP, Vista, 7 and Metro).

FBCIM is simple and convenient to use and offers many great features such as emoticons for Facebook, text formatting, status update notification, sound alerts, chat history and many more. With FBCIM you'll never miss a friend and you'll have more fun while chatting with them. This innovative instant messaging application, which connects to Facebook Chat, provides the convenience and ease of an instant messenger. This novel application caught our attention and has a potential for continued growth and use, that is why we are interested and have selected FBCIM for our project.

## Maturity of the project (e.g., alpha, stable), developers are listed as working on the project

Currently FBCIM is a stable release (the latest release is 2.1.4651 according to its website (<http://www.filehorse.com/download-facebook-messenger/> ) and is in widespread use over globe. Throughout the entire period of development, the project has been consistent in terms of the number of commits. The last commit was made in Jan 2013. This project contains approximately 7071 SLOC.

## Project Domain

FBCIM has emerged from the realm of Instant Messaging which is a form of synchronous communication over the internet. Instant Messaging offers quick transmissions of text messages from sender to receiver by offering certain advantages over e-mail based communication. Internet messaging applications are not limited to messaging but it further allows functions such as file transfer, maintaining a contact list, and simultaneous conversations. Various instant messaging protocols, both open source and proprietary, are available. These include SIP, APEX, and XMPP.

FBCIM is built using open source Extensible Messaging and Presence Protocol (XMPP). FBCIM allows you keep a list of contacts you interact with. You can interact with anyone on your friend list or contact list as long as that person is online. You can also retrieve history of conversations.

# Project Size and Scope

In FBCIM we have a total of **8** Packages, a total **69** classes and the Physical SLOC is **7071**. This project is built using **Smack,** which is an open-source Java library for **XMPP** instant messaging. It also uses other frameworks such as **Xstream** (a simple library to serialize objects to XML and back again), **Google gson** and **restfb**.

## Is the project of reasonable size?

We believe that our project is of reasonable size given the number of packages (8) and number of classes (69) it has. The application builds on the aforementioned frameworks. The following graph depicts the SLOC distribution among packages. It is evident from the graph that ther UI package has the most percentage of SLOC. Components like Chat, Update, and Core are relatively of the same size. All in all, a manageable project for a team of 5 as far as exploring architecture and performing refactoring is concerned.

## Execution of the program/application

We have compiled and assessed this project and believe it to be of a reasonable size for a term project.

# Group Members

**Bala Arun Reddy**

I have been developing various chatting/messenger application for many year. I developed a chatting system which can be used via local LAN/W-LAN suing Java Applets; Working on a desktop application which has billion of user would improve my skill to enhance to the next level. I will observe, learn, experiment, re-factor the architectural design of the application which help me a lot in the real world for developing more efficient, enhanced, reliable and user friendly applications, I believe that's the main intent of this course as-well.

**Deepu Kumar**

I am indulging with software development field since two years only and I mostly work with object-oriented programming languages, however I have worked on IPhone application based on my academic projects so I am aware of xcode as well. Therefore I am familiar with designing an architecture and implementation of few design patterns with xcode. I firmly believe that these skills will help me in understanding FBCIM’s current architecture and I’ll be able to contribute by helping understand the code architecture and behavior. Apart from this I would really go in depth of re-factoring in context with this project.

**Paramjeet Singh**

I have worked as software professional at various positions for around 5 years. I worked in PLM Domain and has been working on development of PLM solutions and also providing consulting to various clients in Europe and North America. During my career path I have never been involved in the team working on the architecture of the software; most probably I was fresh in my career. As now my career has grown much long ahead, it would be great to have hands on by working on domain layer, refactoring, design patterns and other vital software architecture concepts. This project on which we would be going to work is based on Communication domain thus giving me great opportunity to create new skills.

**Pankaj Kapania**

I have been in software development for more than 5 years now. During this tenure, I worked in various web development projects using J2EE technologies and other related frameworks like struts and hibernate etc. I have particular interest in working on the domain layer. During all these years I did not get much to work on desktop based application. So I think this is my opportunity. In this project I would be striving for identifying the architecture specifically at the domain layer consisting of the core business logic. Also, I would strive for applying the re-factoring that we will be learning in this course.

**Gilles Desrochers**

In my work as a software developer, I have worked mostly on C/C++ application in a regulated environment (IEC 62304 compliant). My expertise has been more on desktop application development which should prove useful to that portion of the project. My experience in C++ is adaptable to Object Oriented languages such as Java. I look forward to applying techniques to improve a software application’s quality and I anticipate expanding my knowledge of the communication domain.

# Appendices:

