\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Dijkstra  
Crist – Howell – Ghodratnama

t15 – Software Specification and Installation Guide  
Email Server/Client System

Software Engineering II  
Spring 2013  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*“Elegance is not a dispensable luxury but a quality that decides between success and failure”   
-* Edsger Dijkstra

Table of Contents

[Project Overview 3](#_Toc353814591)

[The Application Design 3](#_Toc353814592)

[System and Operating Requirements 4](#_Toc353814593)

[Installation 5](#_Toc353814594)

[Obtaining the Software 5](#_Toc353814595)

[Basic Operation of the Software 6](#_Toc353814596)

[Server 6](#_Toc353814597)

[Client 6](#_Toc353814598)

[User Registration 7](#_Toc353814599)

[Sending an Email 7](#_Toc353814600)

[Verification and Getting Emails 7](#_Toc353814601)

[Additional Information 8](#_Toc353814602)

[Server 8](#_Toc353814603)

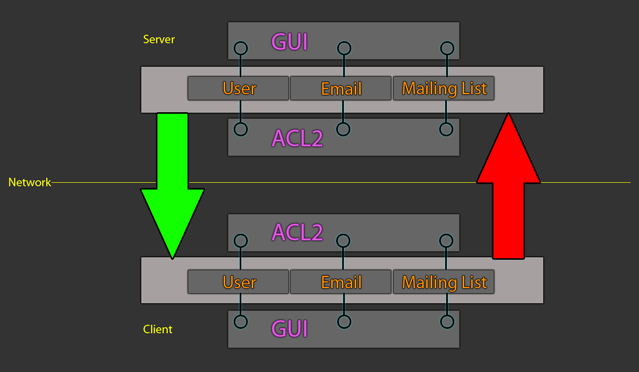
[Client 8](#_Toc353814604)

[Other 8](#_Toc353814605)

# Project Overview

The proposed application is that of an Email Client/Server transaction system that utilizes ACL2 in order to theoretically prove the correctness of the data transformations that occur on both the client and the server applications. While not all components of the system can be theoretically proven, we can provide the means to implement predicate based testing on the data acquired and formulate data structures in which we can inductively test, as opposed to the discrete methods currently implemented in the software industry. However, ACL2 is not without its shortcomings and as a result, we have had to implement some of the systems in a secondary language. For this application, we chose to use Java for interoperability. This also promotes a multilevel design that is easy to build upon and upgradable without detriment to the application.

# The Application Design

  
There are two separate applications to this program, which must be running in a synchronized fashion. One is the client, which is the program that invokes the actions upon the program, and the other is the server, which is an automaton after the user starts it on a machine. This allows the client to send information to the server and expect a response based on the type of transaction it invokes. Each program consists of a modules, which in itself, is a standalone application. These modules may be called individually without the GUI intervention. They are, however, dependent upon ACL2 for data transformation. The structure of both applications is the same, with a GUI layer for client interaction, an ACL2 layer for data processing and a module layer for integrating the two and providing a means for invocation.

# System and Operating Requirements

In order to operate the application, you need to have the following environments installed on your computer.

**ACL2 version 6.1** (<http://www.cs.utexas.edu/~moore/acl2/>) will be used for all the logic functionality of the program as well as generating IO and data processing of email messages and requests.

**Java SDK 1.7** (<http://www.oracle.com/technetwork/java/index.html>) will be used for all the invocation of ACL2 in the program as well as socket connections to the server.

**Dr. Racket with Dracula** (<http://www.ccs.neu.edu/home/cce/acl2/>) will be used for all Property-based test, check-expects, and theorems associated with the logic functions of the program.

**An SVN Client installed on your system:** Many SVN clients exist, but **Tortoise SVN** (<http://tortoisesvn.net/>) is a nice SVN client if you are using Windows.

Optional: The following application was used for development, but is not necessary for operating the application. However, other developers may find the following IDE useful for adding additional features to our application.

**Proof Pad** (<http://proofpad.org>) an alternative IDE for ACL2.

# Installation

## Obtaining the Software

In order to obtain the software and read additional software documentation on your computer, you need to navigate to our project page at: <https://code.google.com/p/spring-2013-se2-dijkstra/>

If you wish to download the software and use it on your machine, use the following SVN command located here:

svn checkout **http**://spring-2013-se2-dijkstra.googlecode.com/svn/ spring-2013-se2-dijkstra-read-only

Once the checkout is complete, you should navigate to the directory with your terminal for setup.

# Basic Operation of the Software

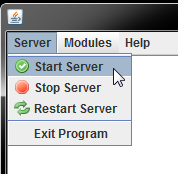
There are two programs in this application. There is a client program which will be used for the main interaction of the program and the Server which will handled all the transmissions and backend of the program. In order to operate the program successfully, you need to launch the Server first.

## Server

The Server executable is located in the /server/ root directory. Navigate to this directory with the console and issue the following command:

java Server

The Server GUI should now be open. To start the server operations, go to Menu -> Start Server.



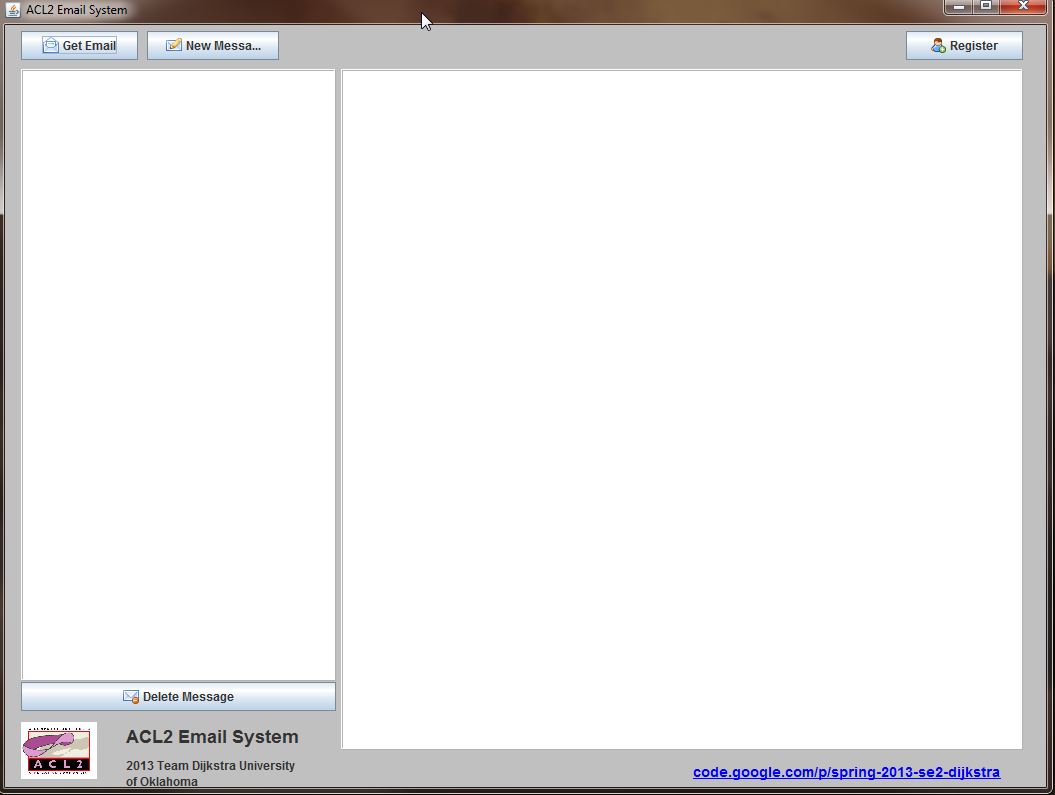
The server should now be running and listening on ports 20002, 20003, and 20005. These ports will wait and process User Registration Request, User Verification, and Send Email commands from connected clients.

## Client

Once the server is running you can execute the client. The client is executed in a similar fashion as the server. First, you need to navigate to the /client/ root directory in the console and issue the following command:

java ACL2Email

This will launch the client GUI which should look like this:



Once the client GUI is launched, you can perform three options.

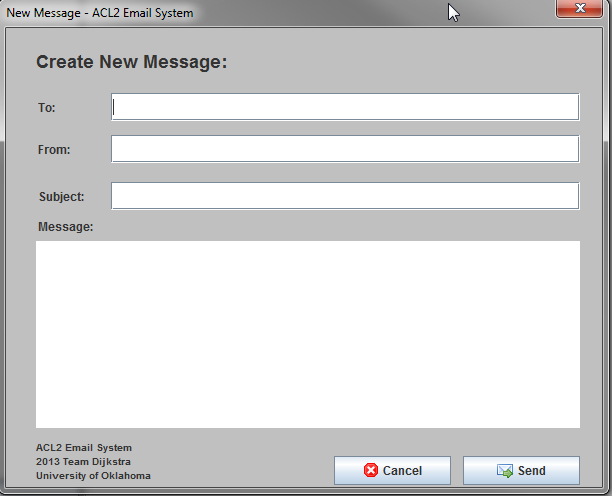
1. Register a User on the Server
2. Send an Email
3. Verify a user and get email messages.

### User Registration

To register a user, click on the *Register* button in the client interface. Three prompts will follow. The prompts will ask you to enter your desired username, domain, and password. Once you fill out each prompt, a request is generated in ACL2 and sent to the server.

### Sending an Email

To send an email, click on the *New Message* button. The following window will appear:



In order to send out an email, fill out the fields in the form and then press the *Send* button. This will invoke the system and send an email message to the server for processing

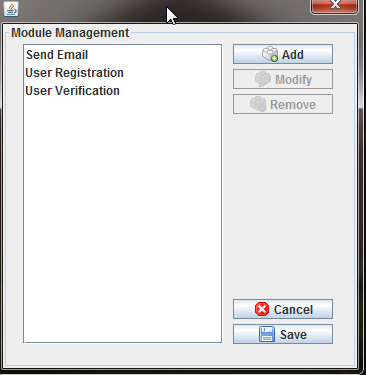
### Verification and Getting Emails

To get your messages on the server, you need to have the client applications already registered on the server. To get messages waiting for your client, press the *Get Email* button. This will invoke a verification request in ACL2 and send the request to the server. If the server accepts your request, waiting messages on the server will be sent back to you as a response.

# Additional Information

## Server

In order for the server to operate, each module needs to be registered with the server. This is done through the module management window in the Server GUI. To access this, go *Modules -> Management* menu item in the GUI. This will bring up the following window.



As you can see, we have already registered Send Email, User Registration, and User Verification. In order to change the listening port, or other features of pre-registered modules, select the module and use the modify button to change the values. To add a module that you have developed, use the Add button and registered the newly developed module. Once you close the server, the registered modules are stored as an XML document in the *server/config* directory. This means that once you register a module, you will not need to do it again when you re-launch the server.

## Client

The client also has the capability to be expanded. In order to do this, some knowledge of the Java Swing library is assumed. This leaves the client GUI open to the developer and allows them to implement features in a way they see necessary. However, when a module is created for the client. You should package it as a Java library and import the library into the GUI interface. This eliminates dependences on the GUI and allows the modules to be modifies without changing the GUI.

## Other

In order to find more information on the design of the application, you can view our detailed design document on the development for both the client and server applications. This document is located at:

<https://code.google.com/p/spring-2013-se2-dijkstra/downloads/list>