SecureChat4.me
End-to-End Secure Chat Application

Richard Lam
Mark Tsujimura

Dr. Mehrdad Aliasgari

California State University, Long Beach
CECS 478 (Section 01), Spring 2017

4 May 2017

GitHub Repository: https://github.com/richardvclam/CECS478-Project
Website/Server: https://securechat4.me

Introduction

SecureChat4.me is a secure end-to-end chat application. The application encrypts messages using AES-256 and checks for integrity using HMAC-SHA-256. Users obtain an AES and HMAC key through a Diffie-Hellman key exchange. The Diffie-Hellman keys are signed by the respected users' RSA private key, thus allowing for key authenticity. All of this information is relayed to the backend RESTful server. The backend only serves to retrieve and store data for users.

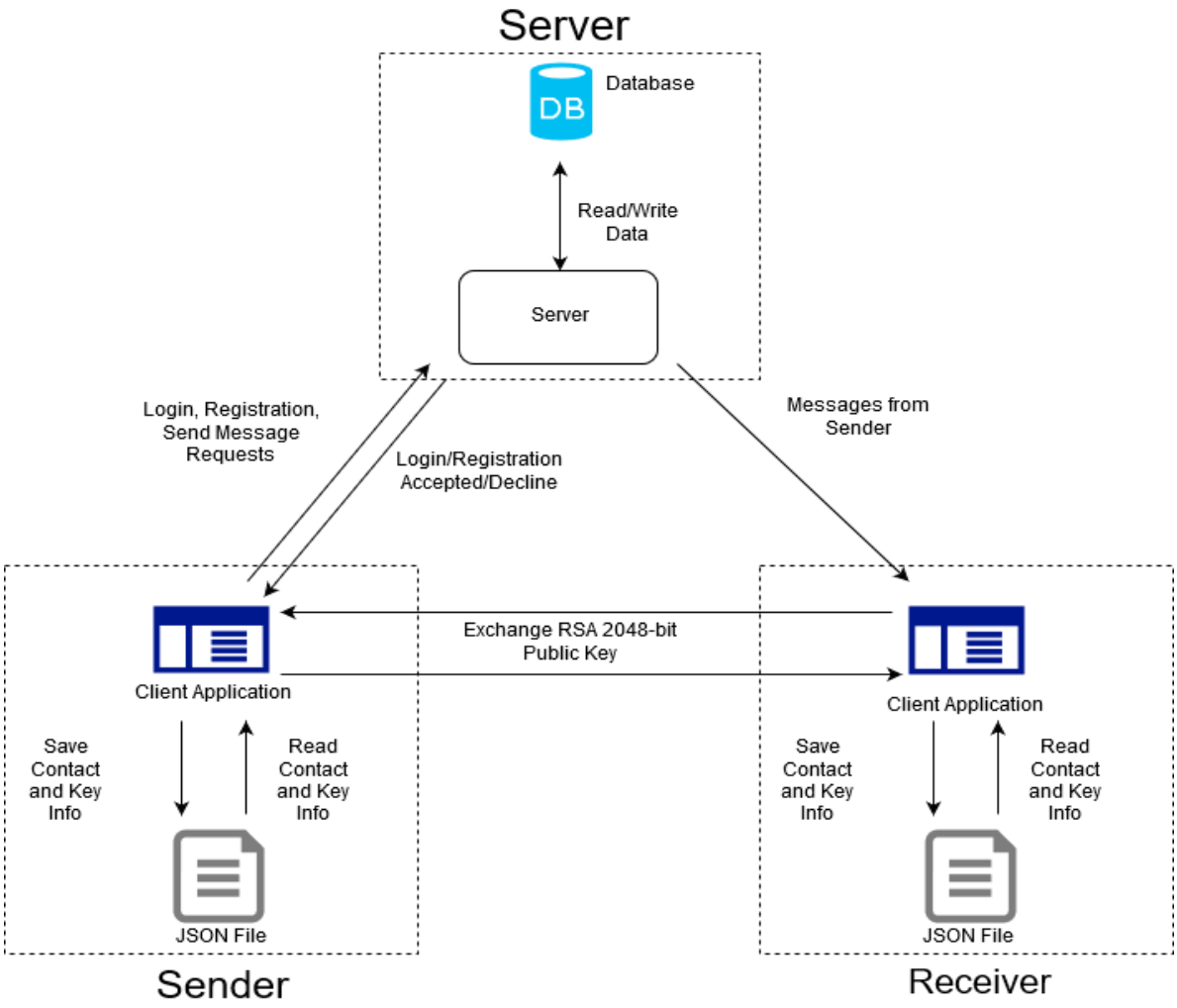
Technology and Languages Used

The server is hosted on Amazon Web Service (AWS) running on Ubuntu OS. The backend server uses the LAMP stack (Linux, Apache, MySQL, and PHP).

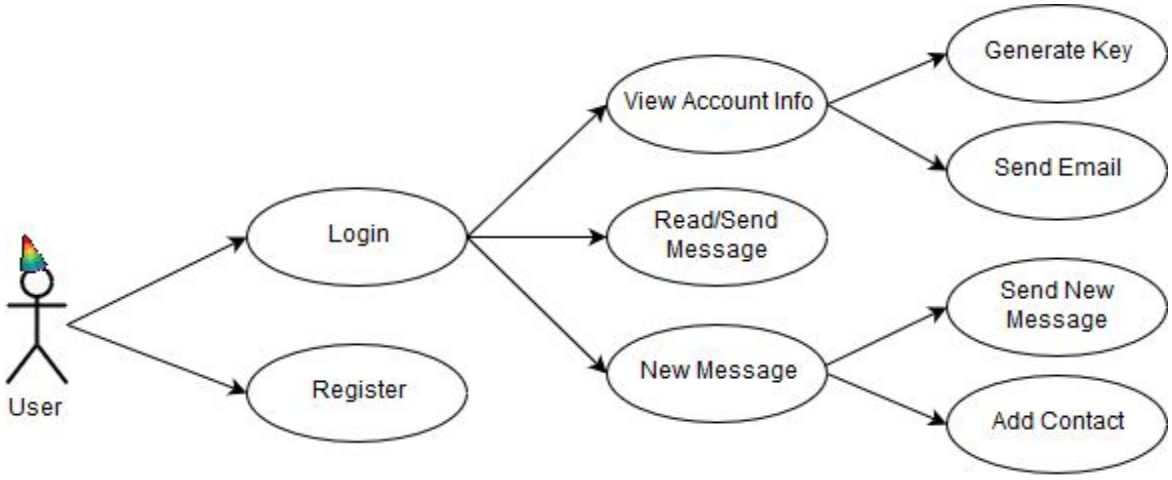
The client is a standalone application that runs on a Java 8 platform. It is developed and tested on Eclipse.

Additionally, we used GitHub as our repository of choice to collaborate between our partners.

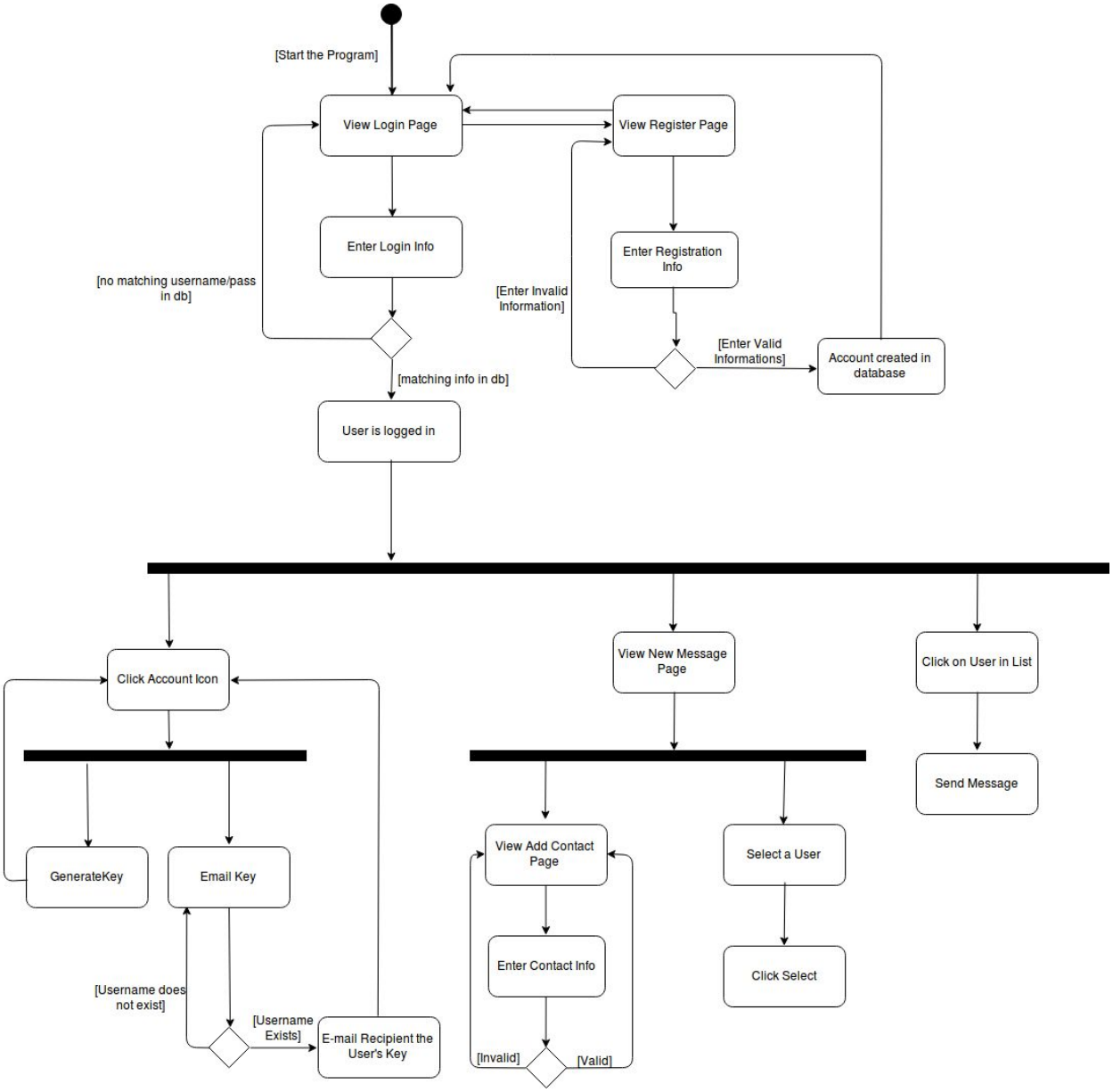
System Overview Diagram



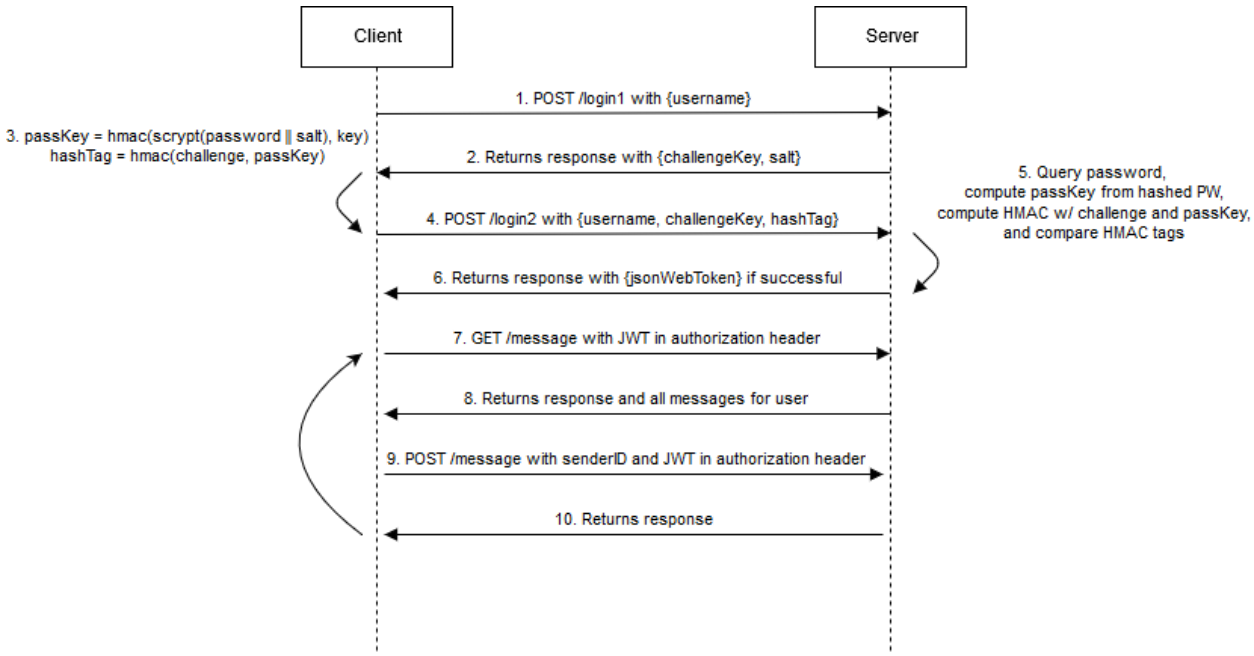
Use Case Diagram



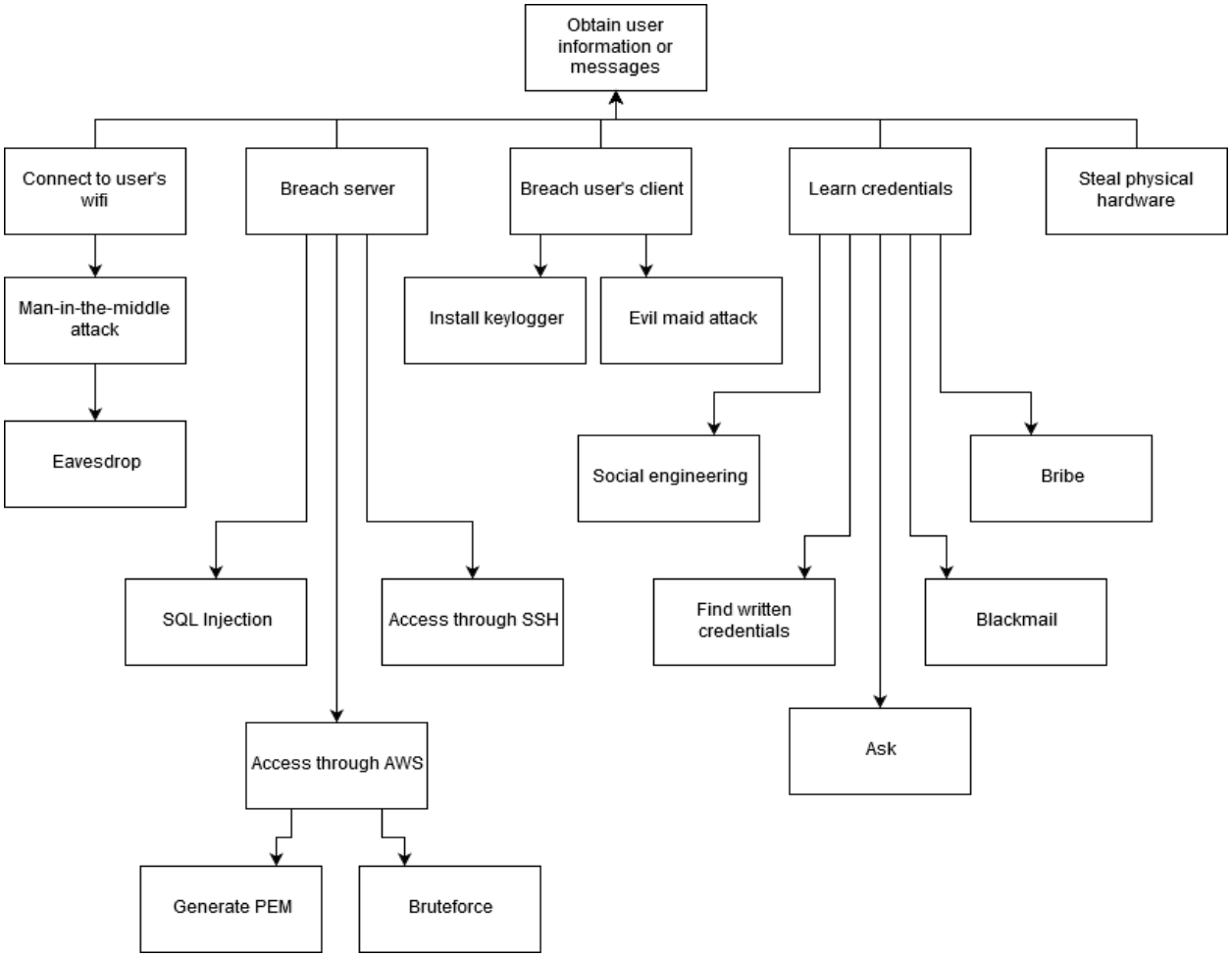
Activity Diagram



Message Sequence Diagram



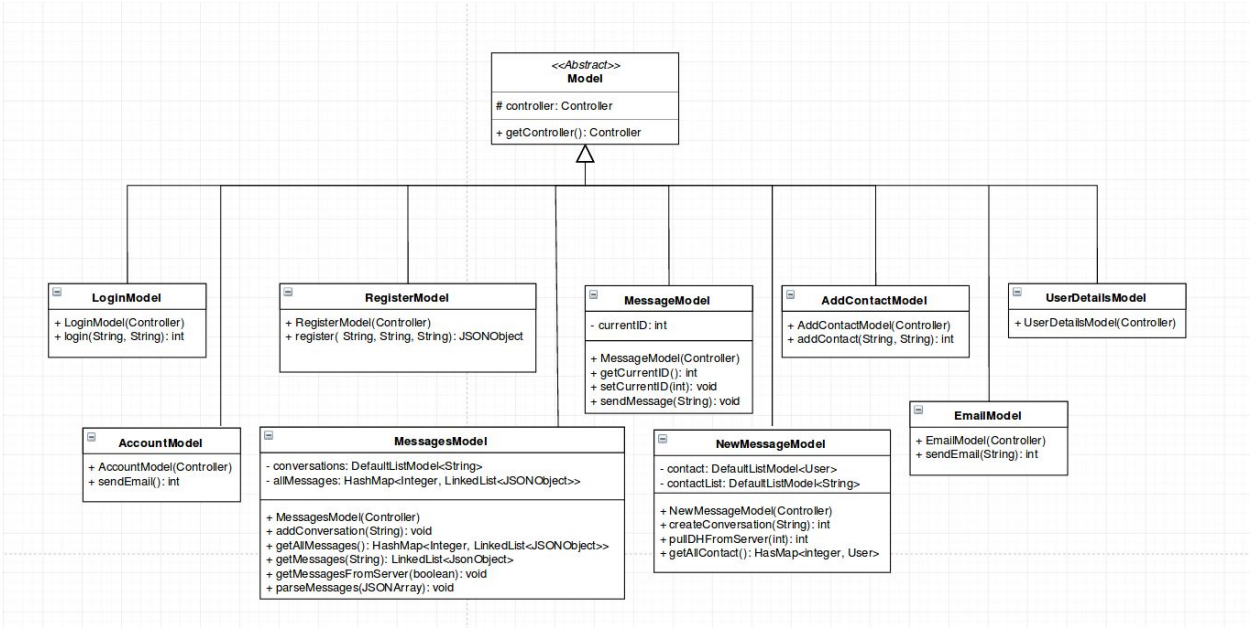
Attack Tree Diagram



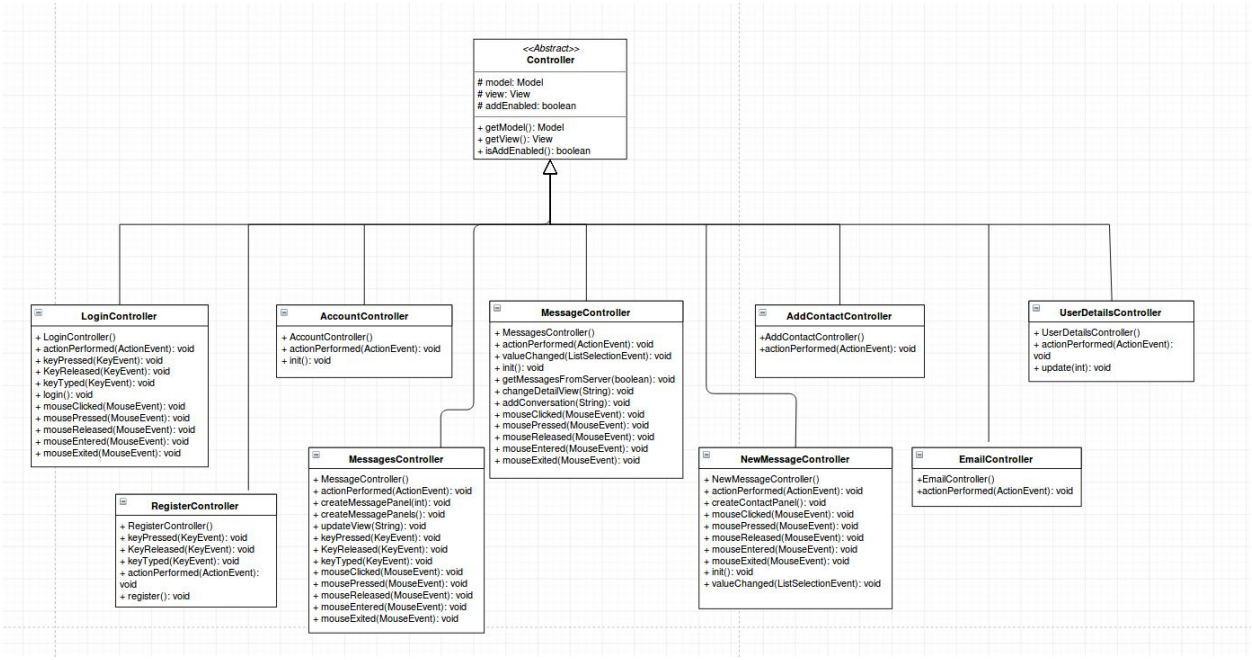
Attack Surfaces

Attack Surface:	**Defense Method:**
Modified Client Application	● Application Key
User Input: SQL Injection	● Prepared statement and sanitize input
User Input: Keylogger	● Install anti-virus detection
Man in the Middle: Compromise SSL	● Get certificate from trusted certificate authority
Man in the Middle: Intercept Message on Wifi	● Recommend user to only use on secure wifi network
Man in the Middle: Packet Logger	● Make sure no one else has access to device ● Ensure that you only use SSL access for email and web browsing ● VPN
DDOS	● Install DDOS protection ● Upgrade to stronger servers and increase bandwidth
Hack Servers/Database: Acquire SSH Certificate	● Encrypt certificate ● Put a password on the file ● Hide the file in a safe location on the device ● Authenticate IP address for only certain devices
Hack Servers/Database: Acquire AWS credentials/account info	● Put information in a secure location ● Memorize it and never write it down
Physical: Evil-Maid Attack	● Session Timeout ● Hire a nicer maid
Physical: Someone Watching Over Shoulder	● Ask them to leave
Physical: Stealing Phone/Computer	● Password Lock the device

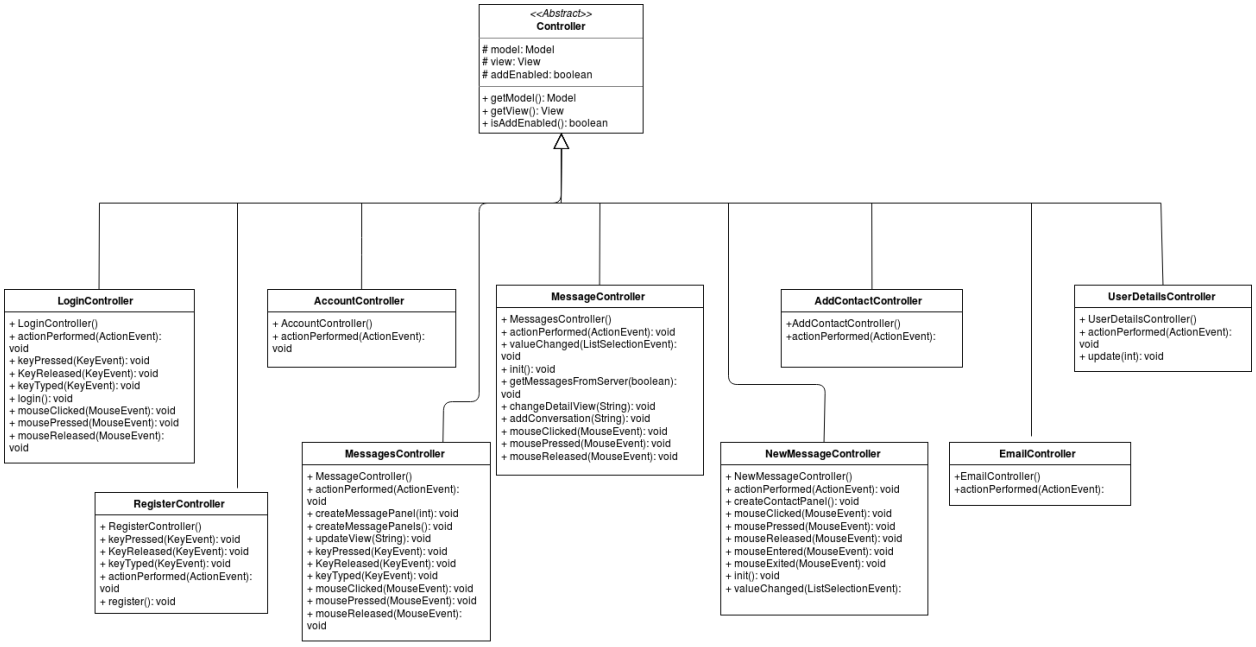
Class Diagram - Model



Class Diagram - View



Class Diagram - Controller



Class Diagram - Other