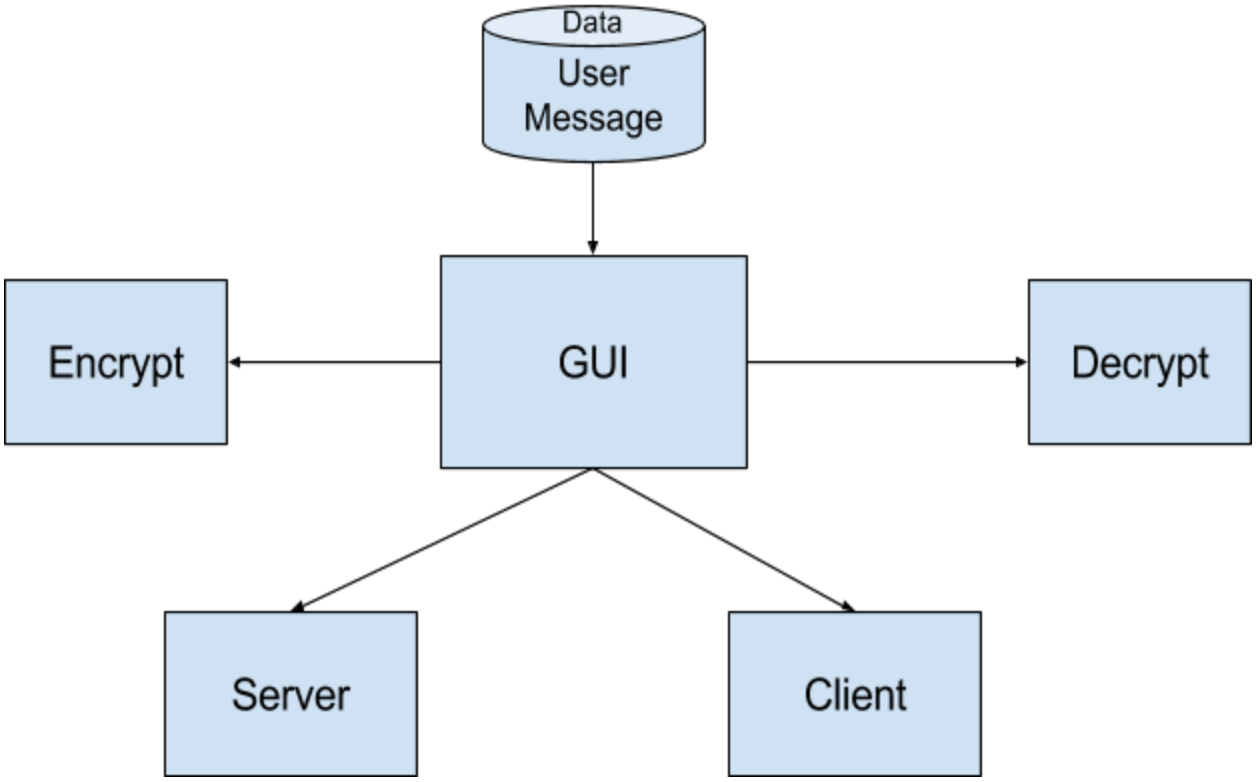
Ahmed Khan
Ryan Moran
Edgar Martinez-Ayala
Project 5 - Networked Chat with RSA Encryption/Decryption

Design Document

Section 1 – Purpose of your project:

The purpose of this project is to create a GUI based program in Java swing that's main feature is a communications chat which allows multiple users to connect to a central server and send encrypted messages to a specific user or users that are also connected to the server. The messages are then decrypted by the user receiving the message and displayed on that user's GUI.

Section 2 – High level entities in design:



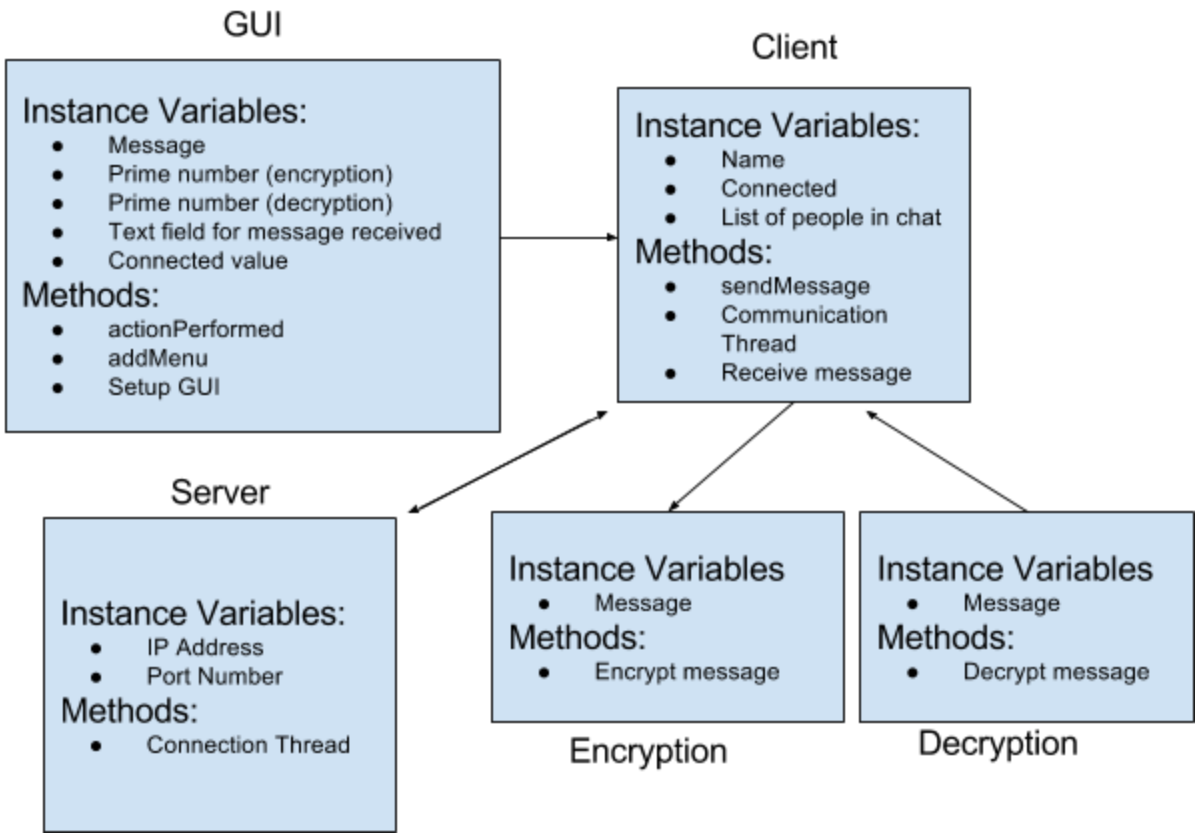
Section 3 – Low level design:

Usage:

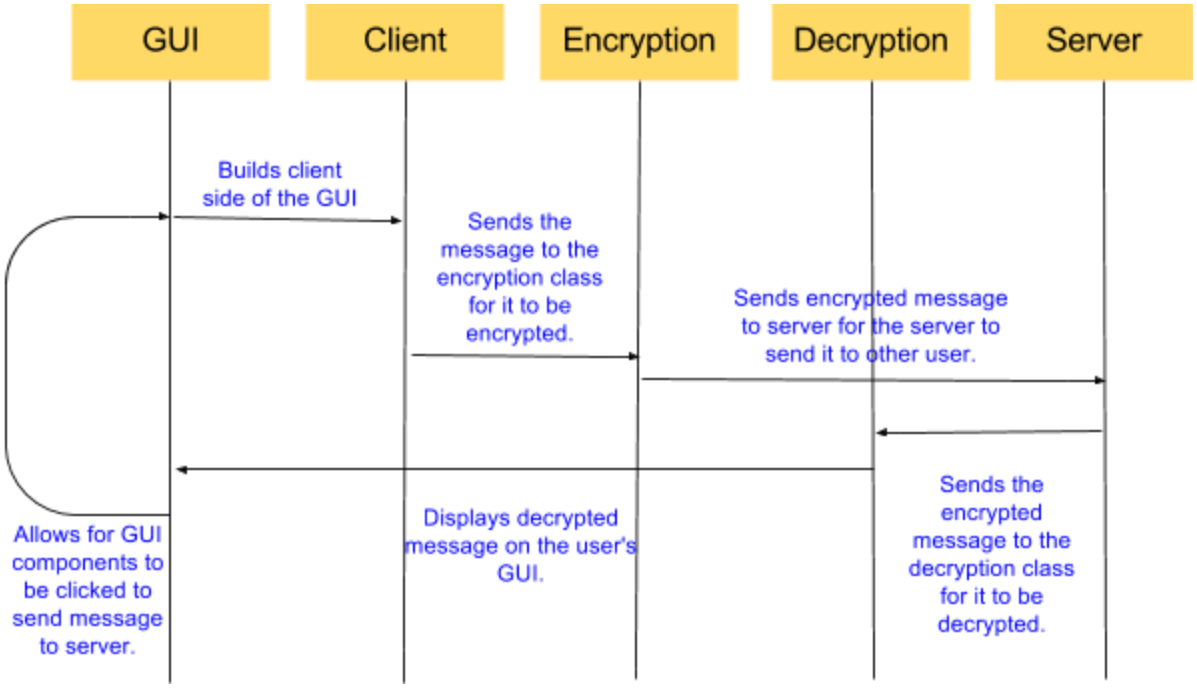
GUI:

The GUI class will handle all of the user interactions including entering a message, adding the the prime numbers for making the private and public key. It will also display who is connected to the server and is in the chat. It will also display messages that are sent to the user from other people.

Model:

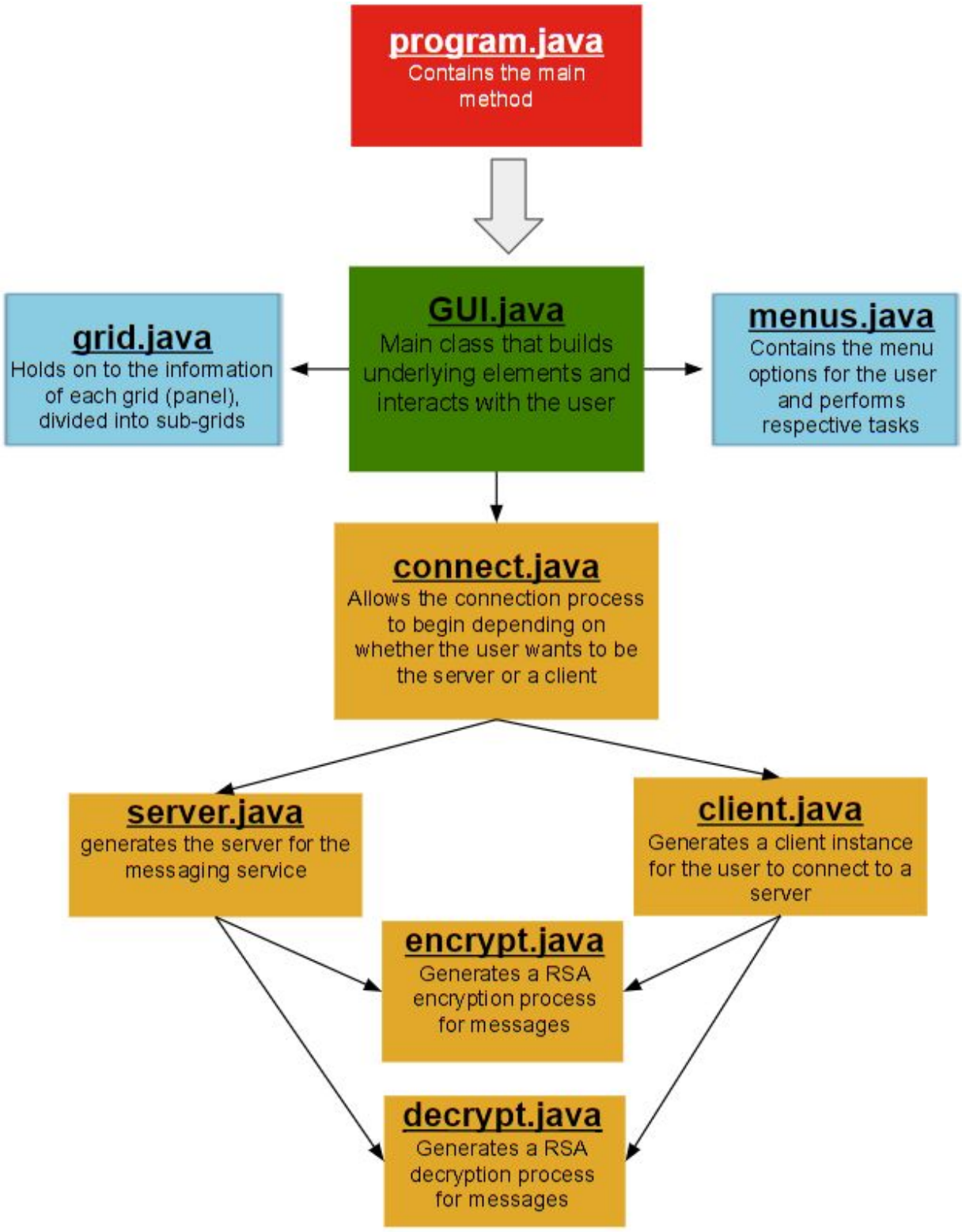


Interaction:



Component Diagram:

1. Program.java
   The main method is contained it it's own java file in order to make readability easier for any designer.
2. GUI.java
   The main interaction between the program and the user will be done in this component. All other components are derived and dependent on this file.
3. Grid.java
   Panels are important for front-end development, but also allow the back-end development to progress faster due to similar interface components being grouped together.
4. Menus.java
   Any type of user application has menu options to make usability more efficient, therefore the cleanest way to implement said options is within it's own component.
5. Connect.java
   When establishing a connection, it is of paramount importance to know where the server is located and which clients are attempting to connect. Although this information is important for the application, good software design makes the user experience as simple as possible; therefore information needed for the server and/or the client will be provided by the connection component.
6. Server.java
   All clients connect to a central server component. All messages are sent to this component and is then forwarded on to the desired client(s). The server should be able to encrypt and decrypt information being passed to the client(s).
7. Client.java
   The client component lists all persons connected to the central server component, while also showing a list of messages received and who sent those messages. The client should be able to encrypt and decrypt information that the client is sending and receiving.
8. Encrypt.java
   Normal messages can be "stolen" from packets during transmission; therefore it is of utmost importance that any data/information passed from one component to another be encrypted in some manner. The RSA encryption algorithm is used in this software. This component provides the encryption of data.
9. Decrypt.java
   Encryption is necessary in order to safeguard sensitive client information. If a message is encrypted, by logic it must also be able to be decrypted. The decryption component allows the client to receive and make sense of any data/information that is passed to him/her. The RSA decryption algorithm is used in this software. This component provides the decryption of data.


program.java
Contains the main method
GUI.java
Main class that builds underlying elements and interacts with the user
grid.java
Holds on to the information of each grid (panel), divided into sub-grids
menus.java
Contains the menu options for the user and performs respective tasks
connect.java
Allows the connection process to begin depending on whether the user wants to be the server or a client
server.java
generates the server for the messaging service
client.java
Generates a client instance for the user to connect to a server
encrypt.java
Generates a RSA encryption process for messages
decrypt.java
Generates a RSA decryption process for messages

Section 4 – Benefits, assumptions, risks/issues:

Benefits:

- Inheritance is used to reuse code
- Makes the code more scalable
- Classes are smaller and more organized
- Each class has a purpose and doesn't do all

Assumptions:

- The numbers used in the private and public key are valid
- There is a valid server to connect to with open ports

Risks/Issues:

- If the encryption and decryption aren't written correctly then the message could be received wrong, or even compromised

Section 5 - Conclusion

This documents highlights the design we are planning on using for our network chat project. It highlights what classes we plan on implementing and also how the different classes, components, and objects interact with each other and the user. Our design has many benefits, but there are also a few risks/issues simply due to the fact that no design is perfect. After consultation between the different software engineers taking part in this application we came up with the design where the pros outweigh the cons significantly.