CS3505 - Multi-User Spreadsheet Desktop Application

Richard Timpson Jabrail Ahmed Tyler Brewster
Conner Grimes Zack Muhlestein Peter Forsling

March 6, 2019

Contents

1 Introduction

The following document outlines a communications protocol for a client-server multi-user spreadsheet desktop application. The following protocol gives no information for either technologies used to implement the application on the client or server side, nor any details as to how the applications should be implemented. The protocol is split into two aspects, logging into the server and choosing a spreadsheet to edit, and editing a spreadsheet. All of the data transfer between client and server will happen using TCP and Sockets, and the data sent back and forth will be JSON.

1.1 Overview and Technologies

Put some more specific information here about the technologies

1.2 Connection Overview

Give an overview about the connection

1.3 Editing Overview

Give an overview about editing

2 Connection

2.1 UML Diagram

Figure 1 shows what the initial connection of the client to the server looks like as a process. The initial connection is broken up into two different phases; logging into the server and choosing a spreadsheet to edit.

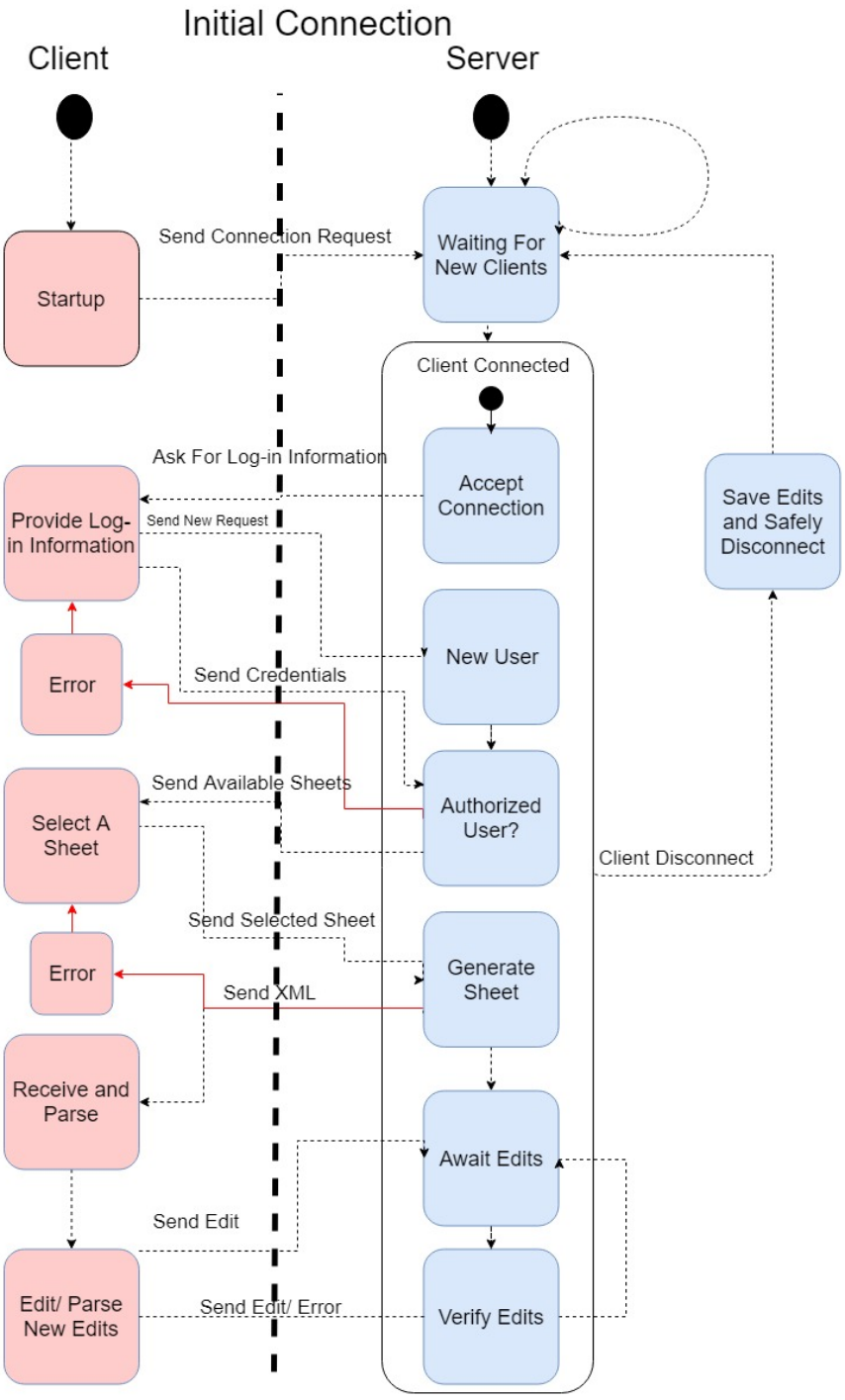

Figure 1: The UML diagram for the initial connection

2.2 Socket Connection

Before a user logs into the server, there must be an initial connection from the client to server so that they can begin communicating back and forth with each other. The connection will be made using standard networking practices with a socket connection built over TCP/IP. On the initial start of the server, it should begin listening for socket connections at whatever IP address and port are sufficient. It is important that the client knows what specific IP address and port number to connect to. With this information, the client should complete the socket connection with the server. If there are any errors with the connection, TCP should report them and both the client and server can handle the exceptions accordingly. If the connection is successful, the socket should stay connected to begin the transfer of data.

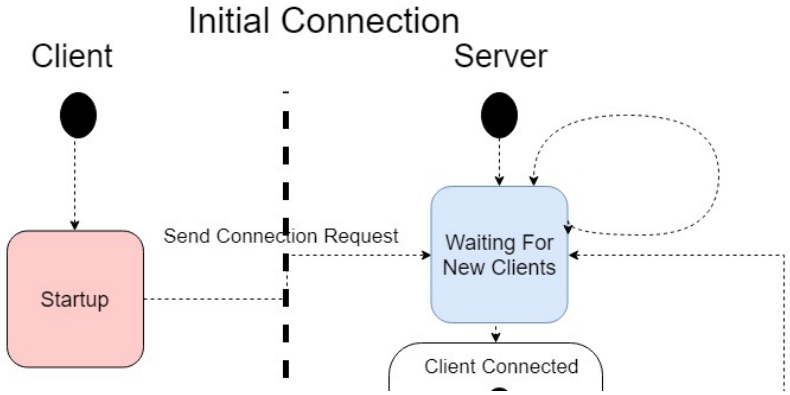

Figure 2: The socket connection in 1

2.3 Loggin In

Once the connection has been made, the server will expect the client, at some point in time, to send the information for either logging in, or creating a new user. This data will be sent as a JSON string terminated by a "n" character, as was discussed in the introduction (need to make sure to talk about our data representation in the beginning of the document). Because the user can either log in or create a new user, there will be two different types of strings. For logging in, it will be formatted like so.

3 Editing

4 Conclusion