

IIT-Hyderabad

---

# Project Document

## [Collab]

Computer Network 2  
(CS3543)

---

Project Name	Collab	
Date	2019-03-31	
Author	Vedha Moorthy - CS16BTECH11040 Akshay Raghavan - CS16BTECH11041	

## Contents

1	SW Development Plan	4
1.1	Project Overview	4
1.1.1	Objective and Project Scope	4
1.2	Assumptions, Dependencies and Constraints	4
1.3	Roles and Responsibilities	4
1.4	Development Plan	5
1.4.1	Development Schedule	5
1.4.2	Development Environment	5
2	SW Requirements Specification	6
2.1	Major Functional Requirements	6
2.2	Non-Functional (Quality) Requirements	6
3	SW High & Detailed Level Design	7
3.1	Overall Architecture	7
3.2	SW System Operation Design	7
3.2.1	{Block 'n'} Structure Diagram	7
3.3	SW Code Structure	7
3.4	Requirement vs. Module Mapping	9
4	SW Unit Test Report	10
4.1	Bugs known at submission date	10
5	SW Development Completion Report	11
5.1	Project Result Analysis	11
5.1.1	Development Work Promotion Results	11
5.1.2	Development Results and Utilization	11
5.1.3	Deliverables List	11
	Terminology / Abbreviations	12

# 1 SW Development Plan

---

## 1.1 Project Overview

### 1.1.1 Objective and Project Scope

*To create a synchronized document where multiple people can collaborate across space and time. The document supports images and multiple formatting options. Also, edit and view rights can be implemented.*

*Scope:*

*This project can be used in multiple domains including schools, universities, workspaces where collaboration of a document is essential.*

*For example, in schools, students can collaborate on a group project, teachers can share marks or common study materials. This can also be used as a regular text formatter.*

*<Objective and Scope of Project in detail>*

Major review items
<i>&lt;&lt;List of Core Components&gt;&gt;</i>
Edit/view rights for clients.
Atomic edits to the synchronized document.
Server synchronizer.
Renderer and a chat program for clients.
Format options like font, size, color.

## 1.2 Assumptions, Dependencies and Constraints

*<<Identify and List down the Assumptions, Dependencies and Constraints of the SW System>>*

Item	Assumptions, Dependencies and Constraints	Remarks
1.	All clients and server are in the same network or in a statically configured network.	Established using static configuration of IP addresses. If this is not done, we might have to buy a server for the project.
2.	The server has the same IP and is known to the clients.	Statically configure IP for server using 'ifconfig'
3.	Secure network.	No encryption(we can do this if time permits) or any eavesdropper. So no need of using TLS.
4.	End to End reliable connection.	Use TCP.
5.	A limit on the maximum number of clients for QoS.	Maybe.
6.	Server always running.	Start server before clients, load saved files when it starts.
7.	People with view rights cannot edit.	well, duh.

## 1.3 Roles and Responsibilities

Student Name	Roles and Responsibilities
<i>Akshay Raghavan - CS16BTECH11041</i>	<b>Developer</b> <b>Software Requirements Analysis</b> Verifying requirements and performing analysis on requirements;
<i>Vedha Moorthy- CS16BTECH11040</i>	<b>Developer</b> <b>Software Architecture</b> -Mapping the requirements into Architecture
Software Development	<b>Akshay Raghavan V</b> <i>Module Names</i> <i>Client (or Renderer)</i> <i>Formatter</i>
	<b>Vedha Moorthy S</b> <i>Module Names</i>
	<i>Server (or synchronizer)</i> <i>Chat program</i>

## 1.4 Development Plan

### 1.4.1 Development Schedule

Estimated Project Period	25/03/2015 - 30/04/2015
Project Team Size	2
Estimated Man Months	2

Milestone	1 <sup>st</sup> Review	Final Review
Planned Schedule	2-April-2019	During final exam week.

### 1.4.2 Development Environment

Item	Development Environment	Remarks
Program Languages	<i>Qt C++</i>	<i>Follow the OOP design rule</i>
Compiler, Build	<i>Qt compiler</i>	<i>A new version is expected if a chip is changed</i> <i>Specify compiler version.</i>
Target Kernel	<i>Linux 4.15.0-46-generic</i> <i>Ubuntu 18.04</i>	<i>The version is expected to be changed according to new chipset.</i>
Word Processor for Document Creation	<i>Google Docs, Google sheets etc.</i>	
Configuration Management		

## 2 SW Requirements Specification

---

### 2.1 Major Functional Requirements

No	Requirement Id	Function Requirement Name	Description
1	Client_01	Renderer	Renders the synchronized document on the clients along with formatted text.
2	Client_02	Chat program	Allows users(clients) to chat amongst themselves.
3	Server_01	Server/Synchroniser	Collect inputs from clients and synchronises the overall document.
4	Common_01	Formatter	Manages text fonts, size etc.

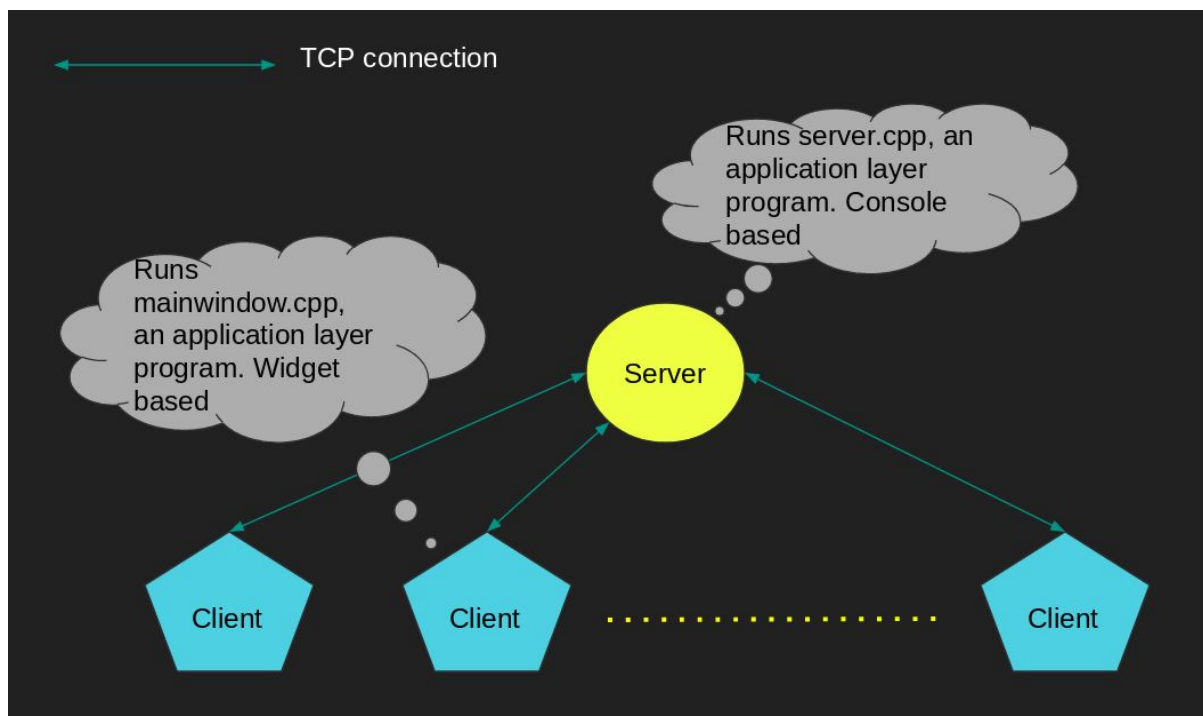
## 3 SW High & Detailed Level Design

---

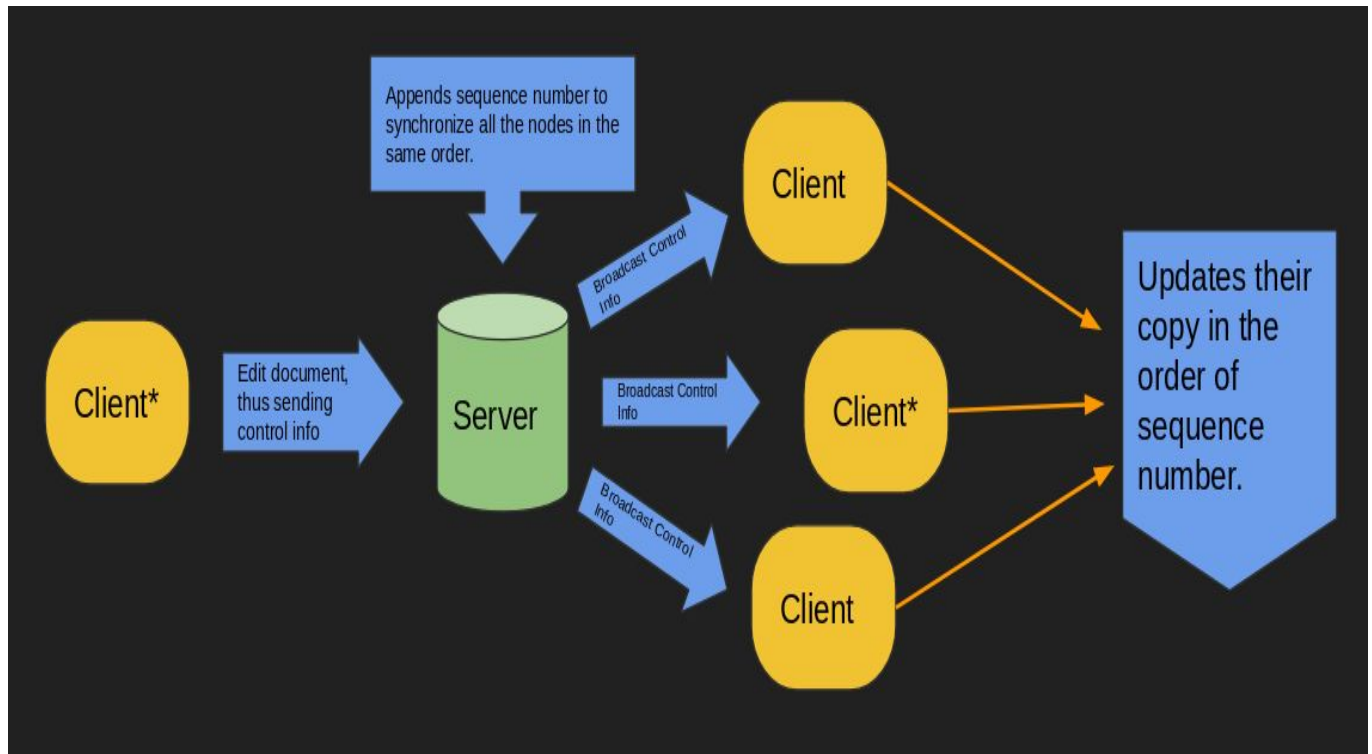
### 3.1 Overall Architecture

Represent the SW system to be developed, as a layered architecture or in a diagram by breaking it down to the block level.

-----



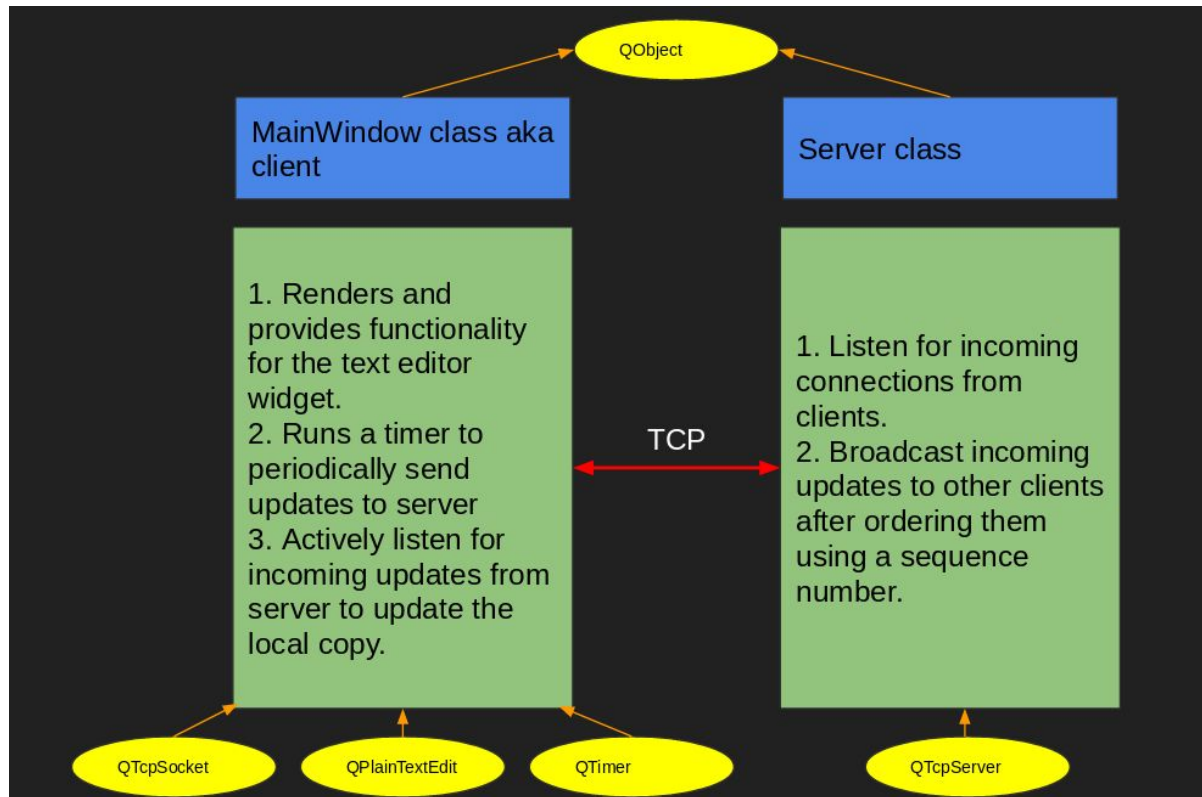
All the clients and server exist in the same network and all clients know the IP address of the server. There can be more than 2 clients.



Control information include information like the cursor position where text has to be added or removed, what characters have to be inserted or how many characters have to be removed etc. We check periodically if there is a local modification. If yes, we send the update to other clients connected to the server. The server basically acts as a forwarder of control information. It also orders the edits using sequence number. The clients actively listen for any incoming control information/update from the server. Updates happen in the order of sequence number and are displayed when the user pauses typing.

### 3.2 SW System Operation Design

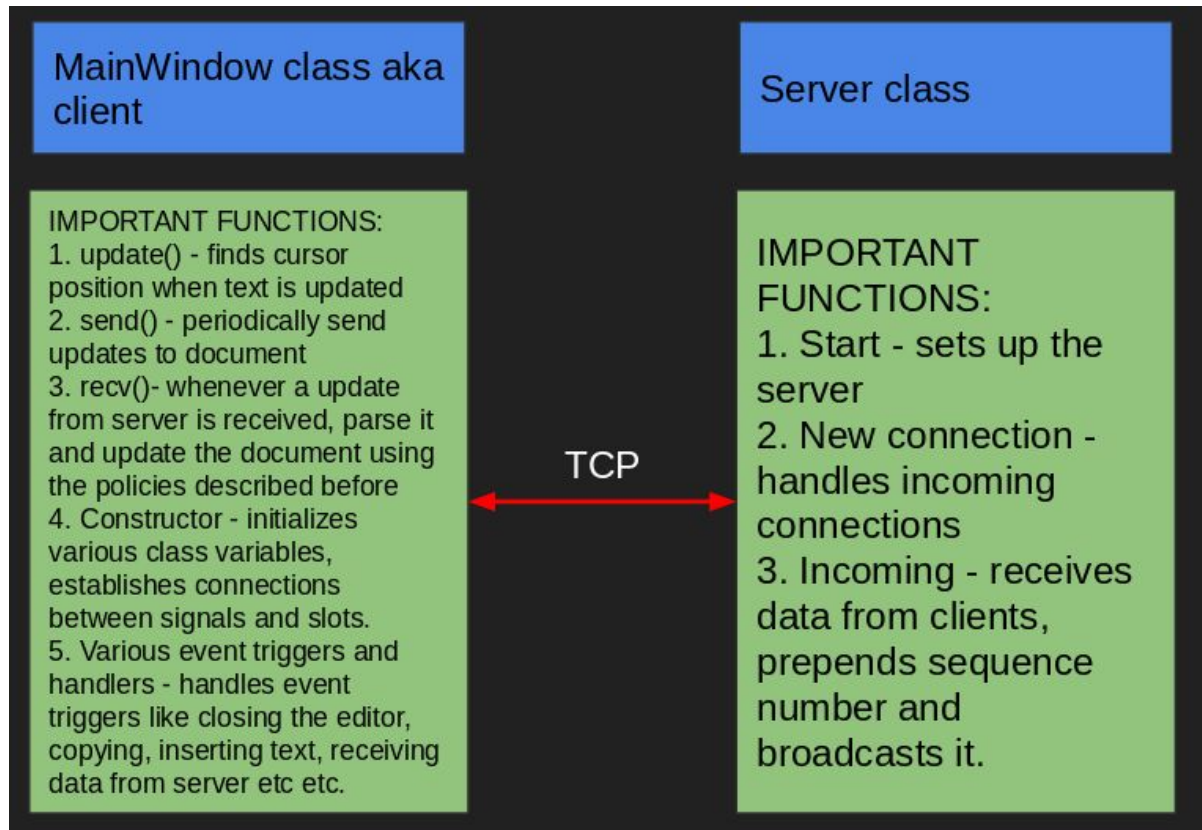
Represent the SW system Operation Design using Overall Class Diagram



The orange lines represent inheritance.

### 3.2.1 {DesignID} Structure Diagram

Describe the basic operation within each of the blocks of the corresponding SW system. Represent the basic operation design using a diagram such as a DFD and Class Diagram and describe each of the operations. In addition, describe the major functions, interfaces and the operation sequence of the corresponding block in sequence.



Main window is created. connection with server is established. periodically check for updates and send information to server if there is a change. Constantly check for any message from server. if yes, update document. When an update is sent to the server, it appends the sequence number to preserve order of updates. the server also broadcasts all the updates to all the clients. the clients then updates their copy of the document. Also, there are many other functions to handle events to close, save, open documents etc.

### 3.2.1.1 {Module 'n'} Component Design

#### 3.2.1.1.1 Module Description

*Describe the functions of the corresponding block.*

Component	Module	Description
Client	send	periodically send if there is a change in the document
Client	recv	Receive whenever there is an message from the server to update the document. Parse the message, update the document in the order of sequence number.
Client	update	updates cursor position when text is inserted.
Server	incoming	Gets update information from clients and broadcasts to others after prepending seqNum

#### 3.2.1.1.2 Interfaces

*Define the interfaces when the corresponding block requires interfaces to other blocks, or other blocks require an interface to the corresponding block, and describe how the interface works.*

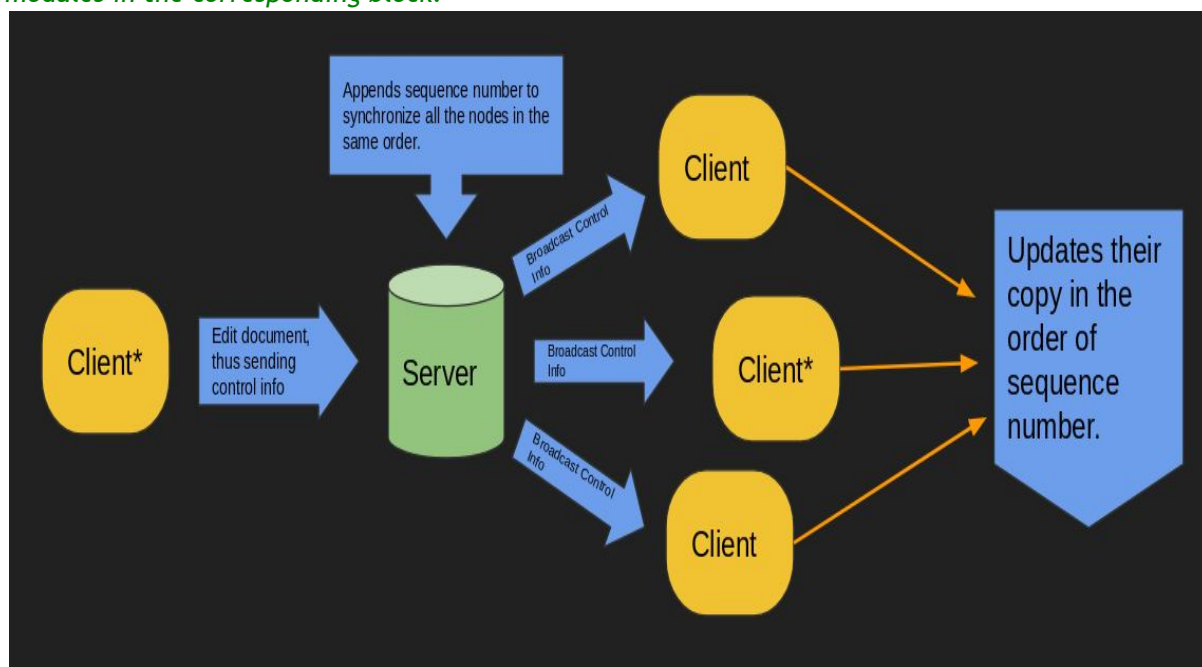
Component providing interface	Related Module	Interface ID	Description
-------------------------------	----------------	--------------	-------------



mainwindow.cpp	MainWindow()	TCP_conn1	Connects the client to the server when client starts the communication.
QPlainTextEdit::text Changed	MainWindow::update()	signal1	updates cursor position when text is inputted or deleted
QTimer::timeout	MainWindow::send()	signal2	calls to send periodically(0.5s) to send updates to the local document to the server to synchronize other clients.
QTcpSocket::readReady	MainWindow::recv	signal3	Constantly monitors if there is an update information to the local document from the server.

### 3.2.1.1.3 Sequence Diagram

*Describe the operation sequence in order to explain the sequence of the executions of the modules in the corresponding block.*



*Also when the user presses the new file button, newFile function is called, when document was modified wasDocumentModified function is called and so on.*

## 3.3 SW Code Structure

*Describe the code structure of the block. Insert a drawing or table that represents mapping modules to files.*

- Mapping list of modules and files (or folders)

Module name	File name (or folder name)
MainWindow	CN2_Project/Client/mainwindow.h
	CN2_Project/Client/mainwindow.cpp
Client	CN2_Project/Client/main.cpp

Server	CN2_Project/Server/main.cpp CN2_Project/Server/server.h CN2_Project/Server/server.cpp

### 3.4 Requirement vs. Module Mapping

Map the requirements defined in the SW requirements and the functional and non-functional requirements of the block.

*Requirement ID:* Refer to the description about the Requirement ID of the higher-level document, the SW requirements.

*SW Design Elements:* List the names of the elements implementing the corresponding requirements in the blank cells of the table. For low level elements, also list the name of the high-level element containing the low level element in the corresponding cell of the table.

Describe a design element as necessary, if it is below the package level.

Requirement ID	Design ID	SW Design Elements				Notes
		Component	Module	Class	Function	
1	1	Client	mainwindow.cpp	MainWindow	constructor, send recv	renders the window, establishes TCP connection, send and receives updates to local document
	2		main.cpp		main()	creates main window
	3		mainwindow.h	MainWindow		Declares class variables and functions
2	1	Server	server.cpp	Server	incoming,	listens and accepts TCP

						connec tion. stores local copy of shared docum ent,for wards update info.
--	--	--	--	--	--	--

## 4 SW Unit Test Report

---

<Please attach the Unit Test Report. >

### 4.1 Bugs known at submission date

DATE: 2019.04.30

S.No	Bug List	Description
1	minor synchronisation problem	typing 'in-humanly' fast leads to problems

## 5 SW Development Completion Report

---

### 5.1 Project Result Analysis

#### 5.1.1 Development Work Promotion Results

Describe learned technologies and technology utilization plans as well as the basic results

Item	Result

#### 5.1.2 Development Results and Utilization

Describe the effects of the development and the fields where the developed product can be utilized.  
e.g.

This is important as this demonstrates that Web-centered home multimedia can be constructed.

Our project is useful when multiple people have to collaborate on the same document. Scenarios where our project is useful:

- *Students collaborating on the same project.*
- *Time slot division.*
- *Chat program*
- *Achieving consensus etc etc.*

### 5.1.3 Deliverables List

S.No	Executable Name	Description
1	<i>Server.pro</i>	<i>runs server</i>
2	<i>CN2_Project.pro</i>	<i>runs client</i>

## Terminology / Abbreviations

<i>Terminology / Abbreviations</i>	<i>Description</i>
-	-

## ■ References

*[1] Qt Documentation*