A real-time collaborative (code) editor

with centralized server infrastructure

The project we propose is a real-time collaborative (code) editor based a centralized self-hosted server infrastructure. The two main components will be a client-side application used to interact with the project files and a server-side application which will be responsible for hosting the files and coordinating the real-time collaboration between the various connected clients.

Since this editor is meant for collaborative projects, we want to focus on implementing specific features supporting a collaborative coding effort. We believe that, given the numerous discrete parts that can be added to our minimum viable product, our proposed project is suitable for a group of our size. We have specified multiple stretch-goals as well in case the minimal product is working properly and we have some time left for implementing more sophisticated features.

The planned components of the application are laid out in the following table:

Precedence	**Local**	**Remote**
Minimum viable product	● A basic terminal application capable of connecting to the proposed server infrastructure and capable of editing remote files.	● A basic server-side application allowing multiple clients to be connected, presenting the project files to remote clients and allowing real-time editing for all connected clients.
Important additional features	● Support for non-in-file comments/suggestions ● Hexedit mode, handle large files (>2GB) ● Show code in different codestyle (e.g. indentation, placement of curly braces) ● Communication with other collaborators (chat) ● Code completion ● Emacs/VI keybindings ● Working locally (git-integration)	● Remote execution of compiled project ● Read from and write to file script triggers (e.g. automated build) ● Local synchronisation of files on client-side file system ● Client login infrastructure
Extra additional features	● Version history diffs ● The option for peer-to-peer real-time collaboration ● Caller/callee graphs, definition/implementation for expression under cursor	● P2P connection vs. centralized system ● Git-integration such that multiple collaborators can work on a different or the same branch ● Plan board integration

In order to successfully engineer the collaborative editor, the following hardware and software equipment is required for a minimal viable product:

Proposed hardware requirements:

Component	Client	Server
Operating System	Linux, Windows (WSL only in case of shell application), (Apple)	Linux, (Apple?)

Proposed software requirements:

Component	Client	Server
Text-based	VT100 compliant, colors	VT100 compliant
Graphical-based	Target must be supported by Simple Directmedia Library. At least Windows and Linux are supported	N/A

Proposed software libraries:

Component	Library	Version
Graphical-based event and window handling	Simple Directmedia Library	2.0.9
Image subsystem (text)	Ncurses	libncurses5
Image subsystem (graphical)	Simple Directmedia Image Library, libpng, libjpeg, OpenGL	SDL: 2.0 OpenGL: 1.2 and better
Audio subsystem	SDL, OpenAL	2.0, 1.2
Filesystem event handling	inotifywait	3.14

This proposal is presented to you by the following students:

Mark Bebawy
Martijn Besamusca
Maxim van den Berg
Chris Bras
Sam van Kampen
Frederick Kreuk
Sverre Kroesen
Folkert van Verseveld
Mund Vetter
Robin Wacanno
Jim Wagemans