WeCode

Brought to you by CitrixNChill

Authors: David Acevedo, Jessica Huang, Zhansaya Abdikarimova, Felicia Truong, Weimin Ouyang
Team: CitrixNChill
Revision History: 11/25: Added test cases, updated appendix, added more requirements, added system models

Intro & Goal

WeCode is a web application with real-time code editor and voice and text communication, that also incorporates quick access to documentation. Our goal is to make WeCode the goto product for programmers to code and collaborate with ease. With WeCode, we hope to address the collaboration obstacles faced by programmers, such as merge conflicts and scheduling meetings.

Who's it for?

Programmers are our main target. As programmers ourselves we understand first-hand how this can change the way programmers collaborate. With a single code editor and real time communication, WeCode can change the way we program together.

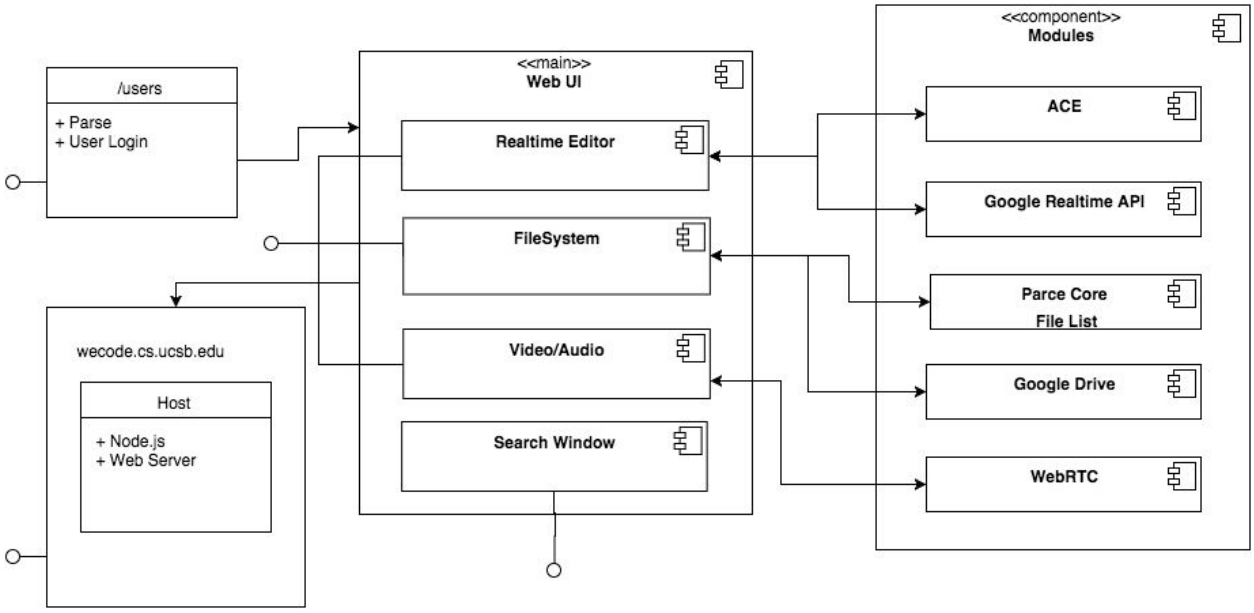
Why build it?

From the beginning of our programming days, we are taught to live and breath the concept of pair-programming so with WeCode, we can effectively and efficiently do so. Furthermore, working in groups and communicating with each other is a guarantee thing in the industry.

Glossary of Terms

WebRTC - Real-time communication using peer to peer connections.
In-Session - Portion of WeCode that encapsulates the real-time communication and code editor
IFrame - A frame where you can put html on

System Architecture Overview



Requirements

Use Cases:

- As a user, I can edit code with syntax highlighting.
- As a user, I can see other people editing the file in real time.
- As a user, I can work on a specific project from GitHub.
- As a user, I can send instant messages to other collaborators.
- As a user, I can receive instant messages from other collaborators.
- As a user, I can invite other people to collaborate on my code.
- As a user, I can toggle the chat panel.
- As a user, I can toggle the search panel.
- As a user, I can change the site address of the browser.
- As a user, I can press “Back" for the browser to go back in history.
- As a user, I can mute my voice chat.
- As a user, I can disable the camera.
- As a user, I can toggle the attendee panel and be able to see all the attendees.
- As a user, I can see all my project files and directories.
- As a user, if I click the search option, I can type ask a question and receive search results.
- As a user, I can drag the videos around the screen.
- As a user, I can create a new folder to organize my files.
- As a user, I can create a new file to edit on the code editor.

- As a user, I can switch between files to edit on the code editor.
- As a user, I can toggle the workspace panel.
- As a user, I can view all my workspaces.
- As a user, I can change my workspace to work on a different project.
- As a user, I can create a new workspace.
- As a user, I can login to Google Drive.
- As a user, I can sign up for an account to use WeCode.
- As a user, I can login to WeCode.
- As a user, I can change the size of the side bar.
- As a user, I can save the file I edit.
- As a user, I can delete a file from my project.
- As a user, I can return to the home page.
- As a user, I can see other people's cursors in different colors.

Prototyping and Test Cases

Prototyping Code: https://github.ucsb.edu/CitrixNChill/WeStudy
Test Code: https://github.ucsb.edu/CitrixNChill/WeStudy/tree/master/tests
Test Cases:

Feature: Login Page

Scenario: Logging in

- Given I navigate to the login page
- Then I type in the user name
- Then I type in the password
- When I click on login
- Then I should see that I am logged in

Scenario: Choosing a workspace

- Given that I am logged in
- When I see an available workspace
- Then I click on workspace 1
- Then I should see be redirected to the editor page

Feature: Video Chat

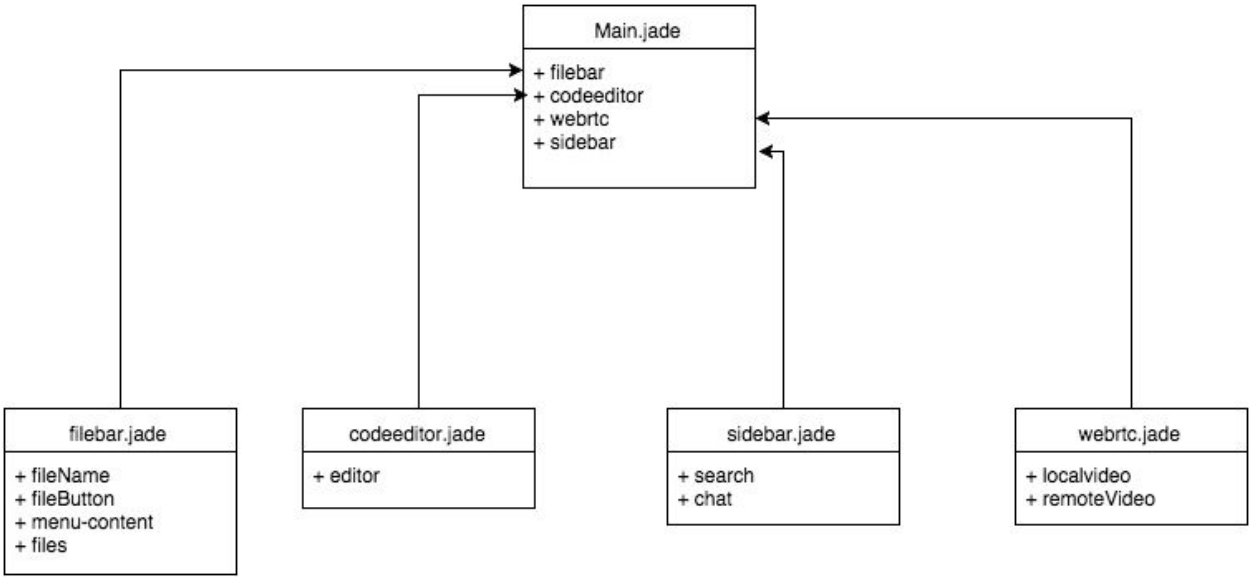
Scenario: 1 to 1 chat

- Given user1 logs into the session
- Then user1 should see their own video
- Given user2 logs into the session
- Then user2 should see their own video
- Then user2 should see user1's video

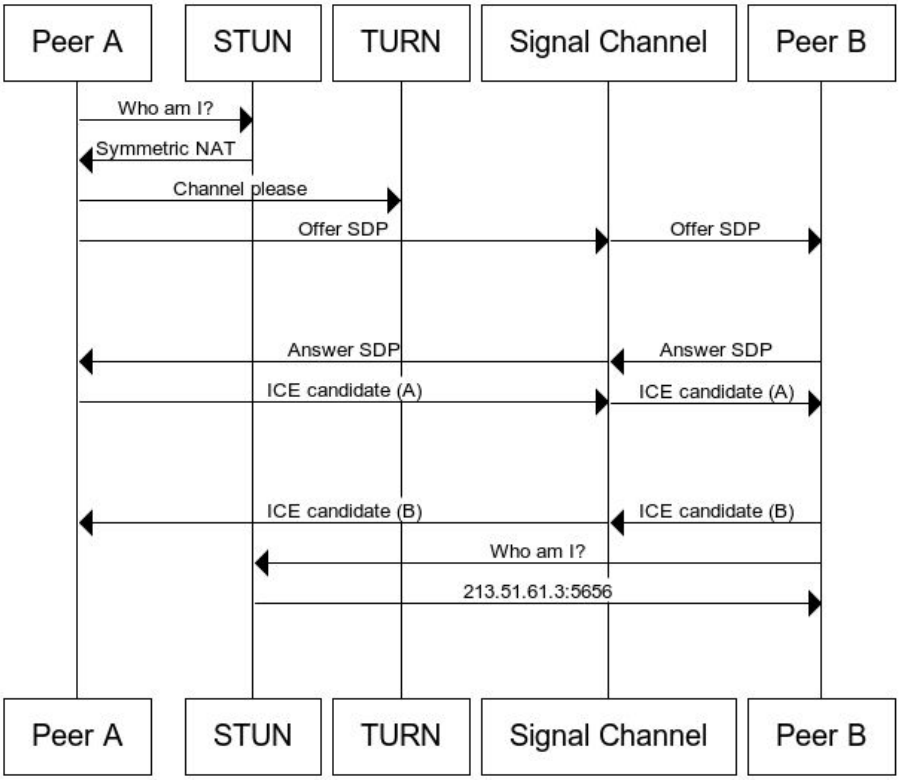
- Then user1 should see user2's video

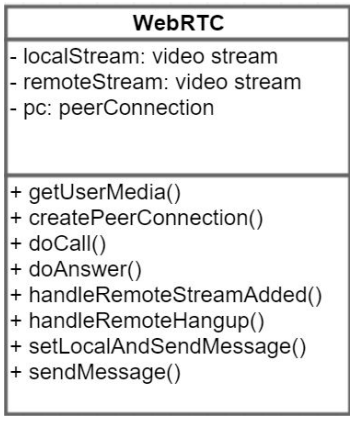
System Models (Design)

User Interface:

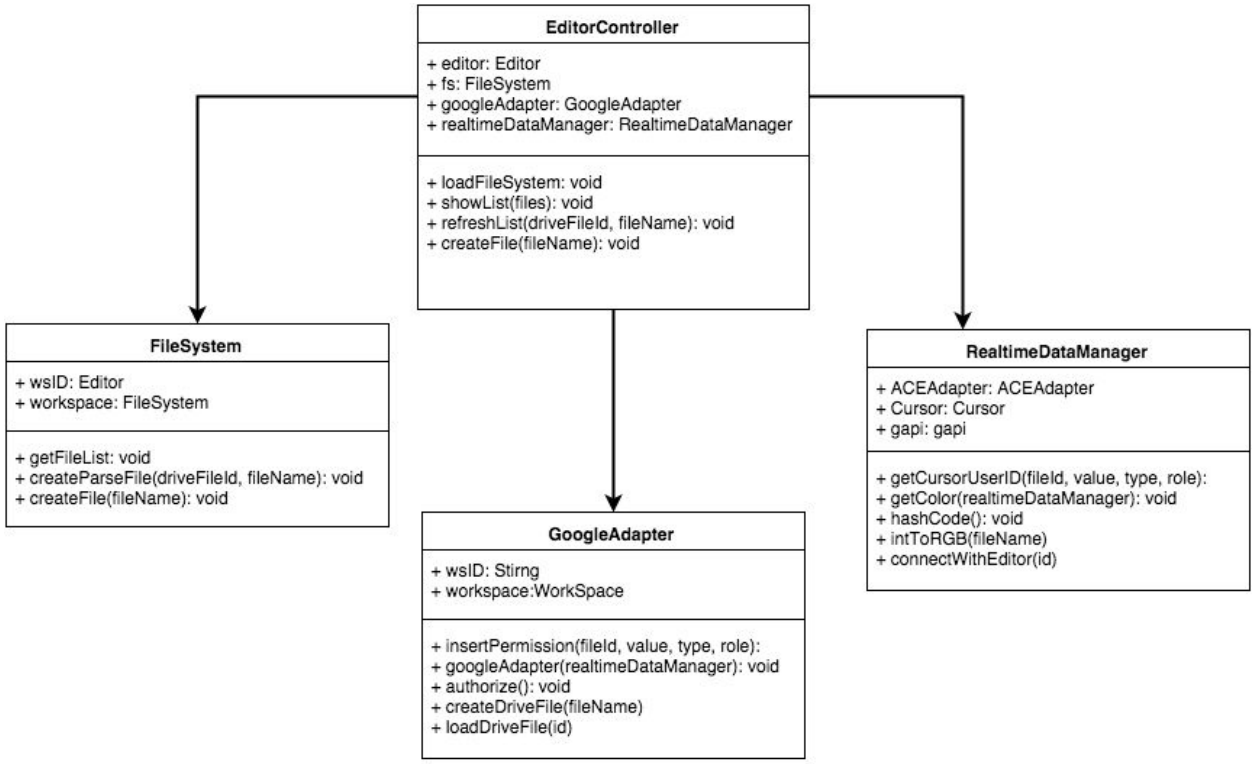


WebRTC: (from MDN)

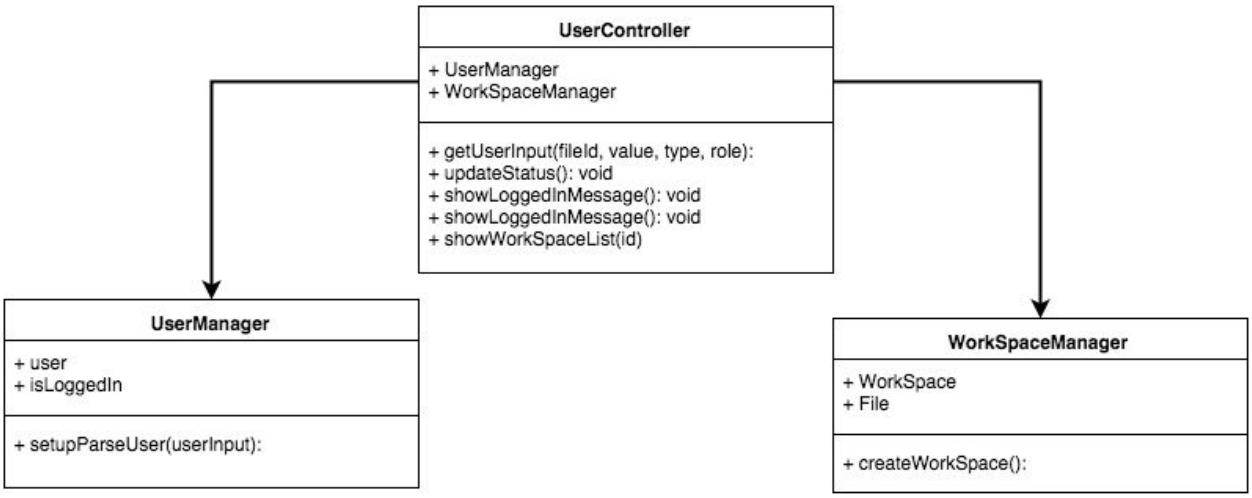




Editor Controller:



User Controller:



Appendixes:

Ace - an embeddable code editor written in JavaScript. It matches the features and performance of native editors such as Sublime, Vim and TextMate

AngularJS -

Bootstrap - HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

Cucumber - is the JavaScript test framework for Behaviour-driven development

WebRTC - a set of standards for providing real time peer to peer video chat in browser, as implemented by Chrome

Google Realtime API - a JavaScript library hosted by Google that provides collaborative objects, events, and methods for creating collaborative applications.

Jade - a terse language for writing HTML templates

JavaScript - a dynamic computer programming language

Mocha - a feature-rich JavaScript test framework running on Node.js and the browser, that allows asynchronous testing.

Node.js - a platform built on Chrome's JavaScript runtime, V8, for easily building fast, scalable network applications

Parse - a BAAS provider

Socket.io - is a JavaScript library for realtime web applications. It enables realtime, bi-directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for node.js.