CoderLab

May 15-31
Advisor - Joe Zambreno

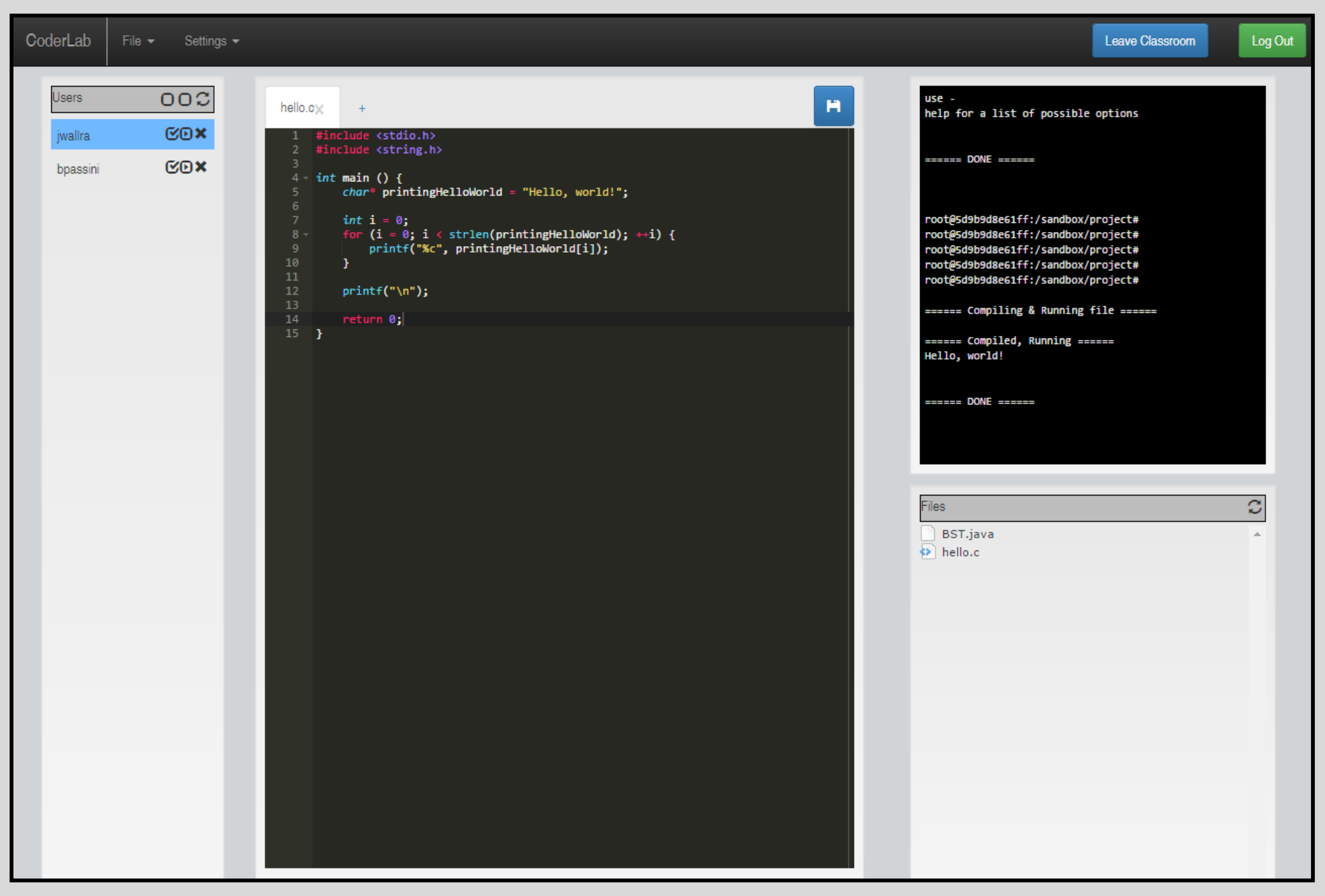
Jake Bertram Daniel Smith
Erich Kuerschner Kyle Tietz
Bryan Passini Jacob Wallraff

What is CoderLab?

CoderLab is a browser-based collaborative coding environment

- Developed to solve problems with learning code in large classroom settings
- Designed around the student-teacher dynamic
- Integrated with ISU Single Sign-On system
- Oriented towards ISU introductory programming courses in Java and C

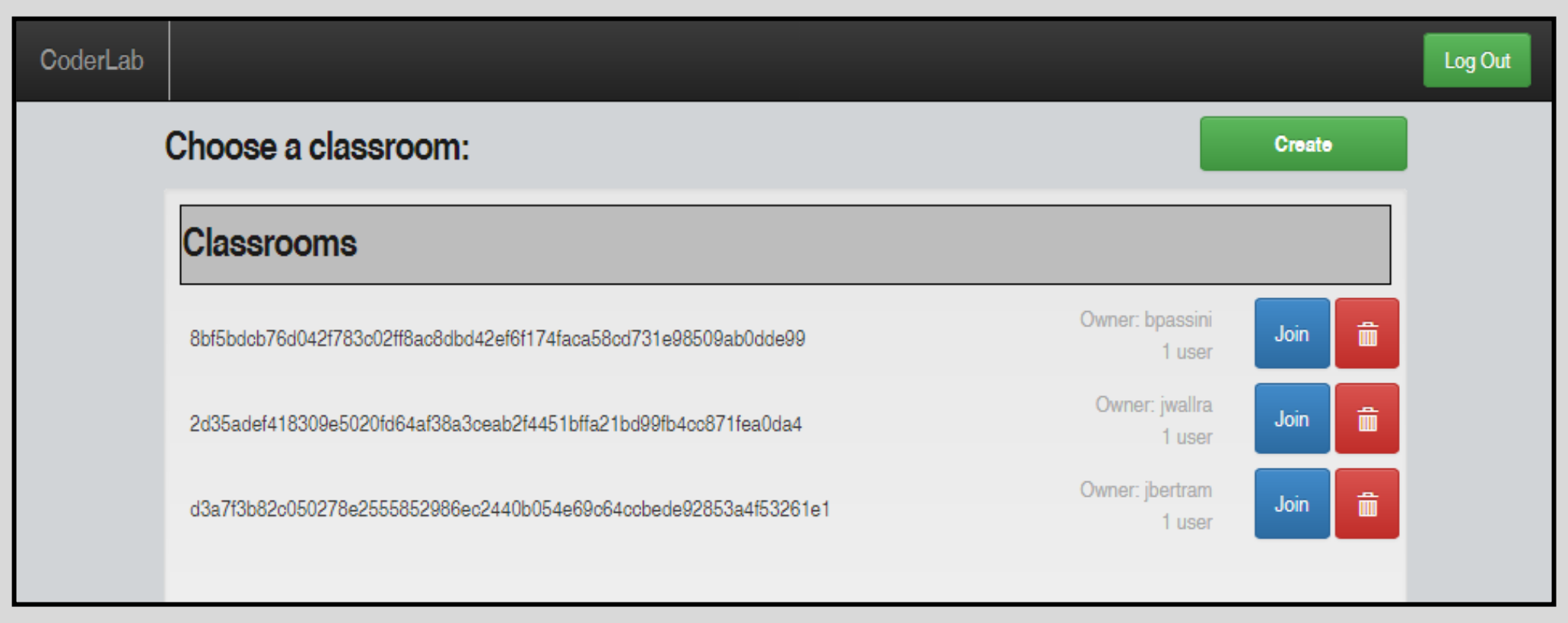
Main Development Page



```
#include <stdio.h>
#include <string.h>

int main () {
    char* printingHelloWorld = "Hello, world!";

    int i = 0;
    for (i = 0; i < strlen(printingHelloWorld); ++i) {
        printf("%c", printingHelloWorld[i]);
    }

    printf("\n");

    return 0;
}
```

```
use -
help for a list of possible options

====== DONE ======


root@5d9b9d8e61ff:/sandbox/project#
root@5d9b9d8e61ff:/sandbox/project#
root@5d9b9d8e61ff:/sandbox/project#
root@5d9b9d8e61ff:/sandbox/project#
root@5d9b9d8e61ff:/sandbox/project#

====== Compiling & Running file ======

====== Compiled, Running ======
Hello, world!


====== DONE ======
```

Classroom Selection Page



Choose a classroom:

Classrooms

8bf5bdcb76d042f783c02ff8ac8dbd42ef6f174faca58cd731e98509ab0dde99	Owner: bpassini 1 user	Join
2d35adef418309e5020fd64af38a3ceab2f4451bffa21bd99fb4cc871fea0da4	Owner: jwallra 1 user	Join
d3a7f3b82c050278e2555852986ec2440b054e69c64ccbede92853a4f53261e1	Owner: jbertram 1 user	Join

Basic Information.txt

An instructor creates a classroom, which is an isolated development environment

—Students can view and join active classrooms

—Each classroom has its own file system and shared documents

—Students can see code the instructor or other students are writing in real time

CoderLab provides a complete coding environment

—Users are able to create and edit multiple files

—The Ace code editor provides syntax highlighting

—A shared terminal is used to compile files and execute commands

Classroom administrators can grant or revoke resource permissions

—Everyone has read-only access and can see changes being made to files and entries in the shell

—Individuals can be granted write permissions

—Editor and shell permissions can be set separately

—Limits access to those approved by the instructor

More applications of CoderLab:

—Working on a project with group members

—Asking students to complete examples in class

—Learning how to use basic Linux commands

—Keeping personal work in a separate environment

Technologies

```
root@CoderLab:/myProject/Technologies# ls
```

Docker
Manages container environments for classrooms

ShareJS 0.7
Enables collaborative code editor

TermJS
Renders the shell in the browser

PTYJS
Connects shell to Socket I/O streams

Socket I/O
Handles web socket streaming for shell

BrowserChannel
Allows web socket message passing (browser sync)

NodeJS 10.x
Generic server platform

Bootstrap 3.3
Web app styling and menu dialogs

JQuery FileTree
Displays list of files in the browser

Operating Environment

- A virtual machine, running **RedHat Enterprise Linux (RHEL)** was provided by Iowa State
- Iowa State also provided **SSL certificates** and a **domain name**
- **Apache reverse proxy** for directing web requests to the web services
- **Docker** is used for providing isolated Linux environments to end-users
- **NodeJS** powers these web services
- The client simply needs a modern browser

Requirements

Functional

CoderLab shall:

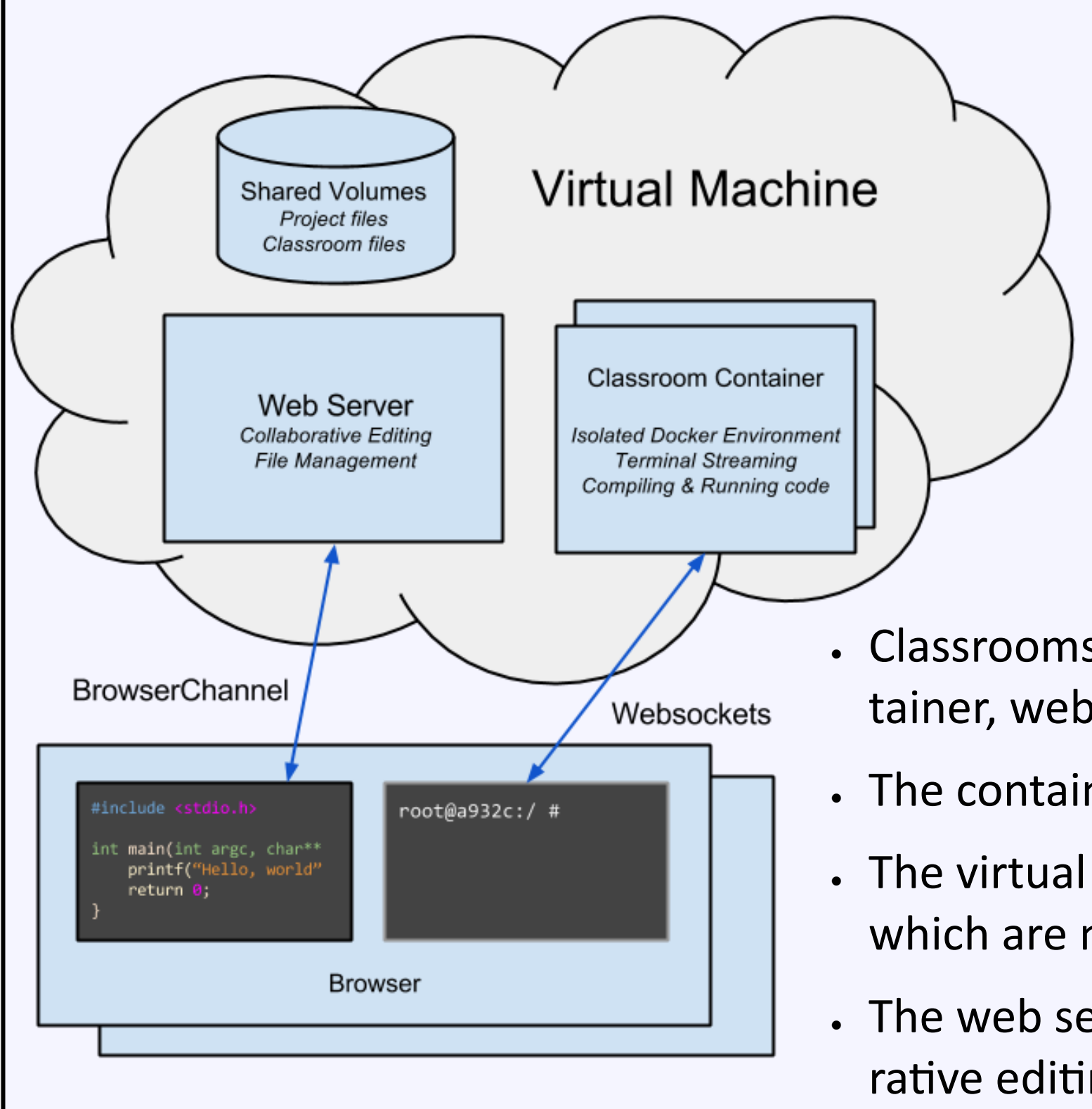
- Require users to log on using ISU Single Sign-On system
- Display a list of all available classrooms to join
- Allow users to create, join, leave and stop classrooms
- Display a list of users, a collaborative coding environment, an interactive UNIX shell and a list of files belonging to the classroom
- Allow users to create and download files
- Allow each user to have their own set of permissions
- Only allow users with edit permissions to make changes to files in the collaborative code editor
- Only allow users with shell permissions to interact with the UNIX shell
- Only allow users with admin permissions to change the permissions of other users

Non-Functional

CoderLab shall:

- Be reliable and fault tolerant
- Be responsive and scalable
- Be easy to use
- Have a professional look and feel

Real-Time Collaboration

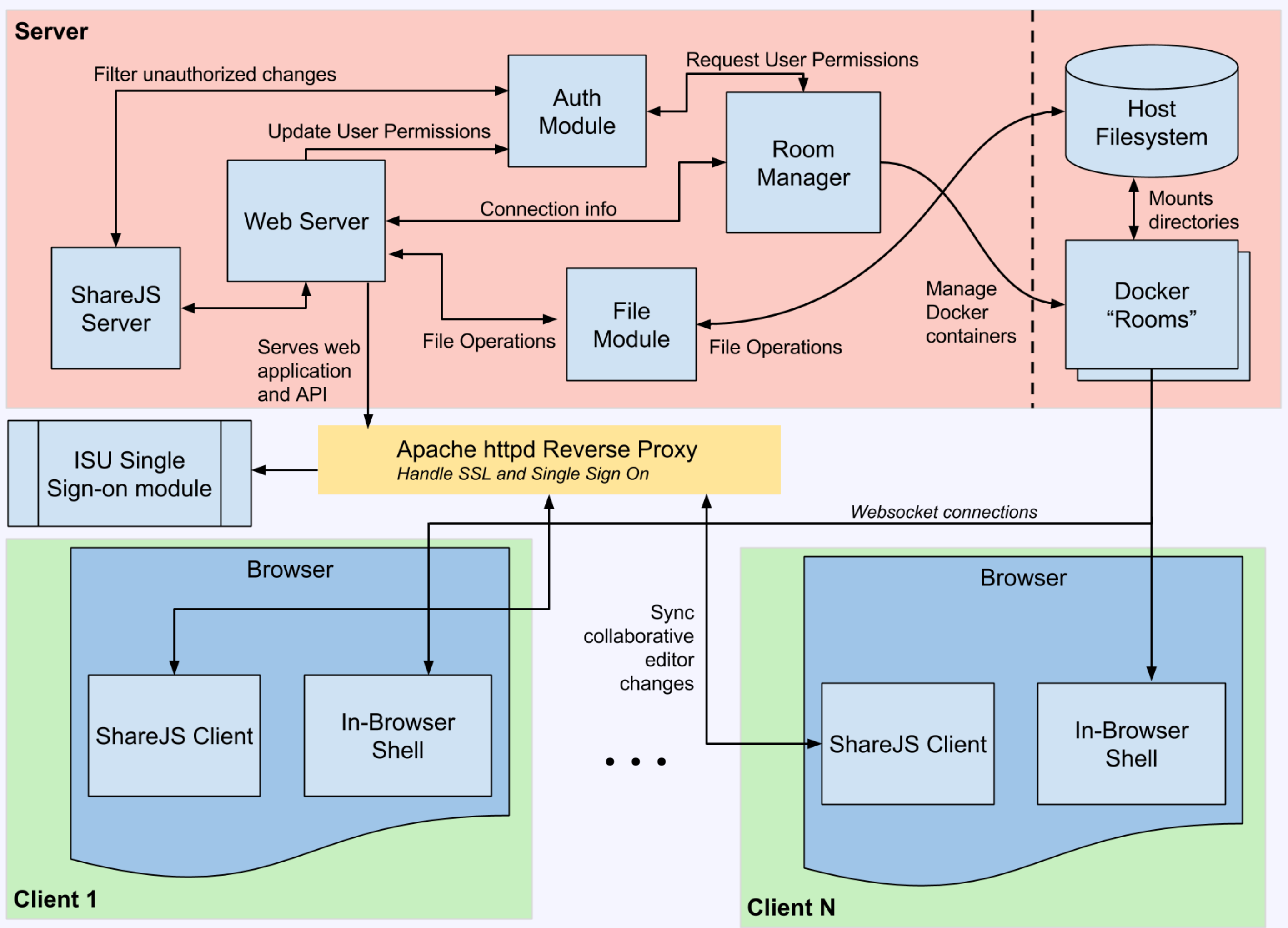


```
#include <stdio.h>

int main(int argc, char**
    printf("Hello, world"
    return 0;
}
```

```
root@a932c:/ #
```

- Classrooms consist of a Docker container, web server, and user files
- The container hosts the shell
- The virtual machine stores files which are mounted to a container
- The web server contains the collaborative editing features
- BrowserChannel streams user edits

Functional Diagram



Web Server - NodeJS server that handles main communication with client browser. Acts as a middle-man between the client and other modules

ShareJS Server - Manages live collaborative documents, passing file updates to all client code editors in real time

Auth module - Maintains a list of client connections and retrieves user permissions from the Room Manager. Filters unauthorized edits to files and changes to user permissions

Docker Classrooms - A Docker image configured to act as an isolated Linux environment for connected users. The image contains the web service that provides a collaborative terminal session and the ability to compile and run code

Room manager - Manages creation, deletion, startup, and shutdown of Docker classrooms. Contains classroom information, including the Docker container status, who created the classroom, and permissions of users currently in the classroom

File module - Communicates with the host filesystem to create, load, and delete files in the classroom. Handles incoming client-side edits and syncs them with the filesystem

Host filesystem - Contains project files specific to each classroom. Classrooms mount these directories, allowing the files to sync between host and classroom

ISU Single Sign-on - Apache httpd module that interacts with ISU Single Sign-On to ensure that users are authenticated

Apache httpd Proxy - The Web Server sits behind a reverse proxy, which handles HTTPS and the Single Sign-On integration

ShareJS client - Keeps files in the code editor synchronized across edits from multiple clients

In-browser shell - Communicates with a shell process on the classroom container to allow command execution and code compilation

Testing

The test space for CoderLab was large because it scales to the number of users, so testing was often done in a situational, 'as needed' environment. Many reasonable test cases were examined and fail-safes were built in for several harmful scenarios such as the shell crashing or failure to fetch user permissions. At the end of each week outstanding bugs were assigned to team members to test and resolve.

Integration Testing

- Tested modules in isolation first
- Integration testing used when combining features
- Regression testing after new features added

Usability Testing

- Tested with as many group members as possible
- Often revealed backend bugs
- Brought CoderLab to COMS 228 recitation for feedback

Security Testing

- Ensure each user has correct permissions
- Grant or revoke permissions for multiple users
- ISU single sign-on

Project Management

JIRA

- Issue tracking tool used for code features and bug tracking
- Organized development into weekly Sprints

GitLab

- Code hosted on ECpE GitLab instance
- Provided revisioning, branches, and a web interface

Team meetings

- Monday meetings to review weekly Sprints
- Development sessions on the weekend

IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering