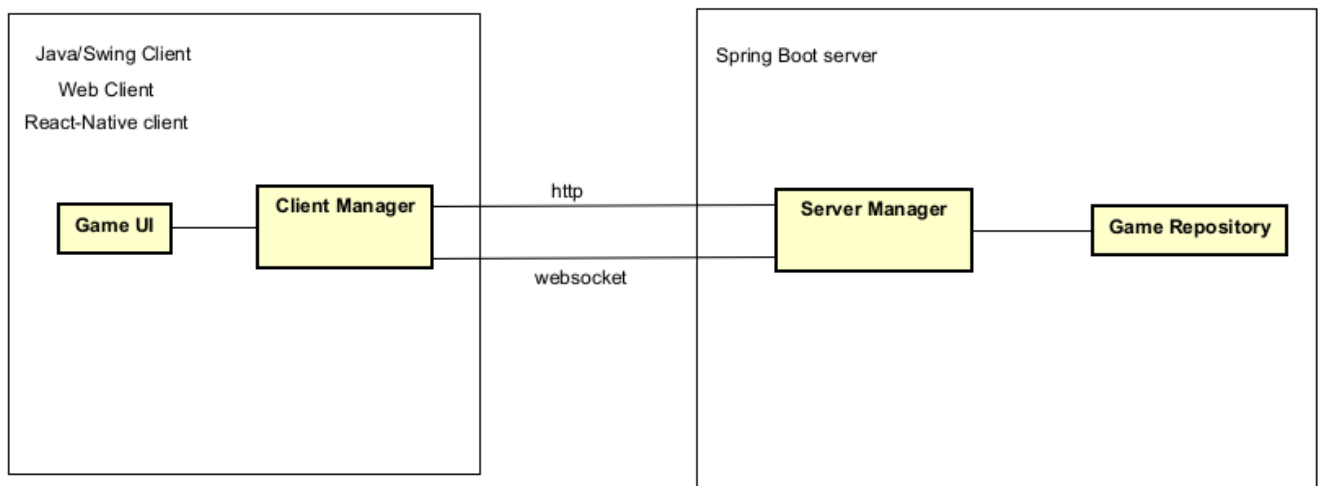


Project Plan

Project Description

- The goal of this project is to build a GameSuite online application which allows players to play a suite of board games such as checkers, chess, Shogi, and Go. This project will serve as a portfolio project to demonstrate the skills I have learned in earning my degree to potential employers. It will demonstrate my knowledge of software design principles, various languages of differing paradigms such as object-oriented and service-oriented (java, javascript, etc.), knowledge of front-end and back-end frameworks (Spring Boot, React-Native, Flutter, etc.), understanding of networking principles such as the MVC architecture, as well as many other important skills. So far I have finished the checkers game and implemented the client-server interaction as well as a Java/Swing client, a web client, and a React-Native mobile client.

Current Design and Architecture

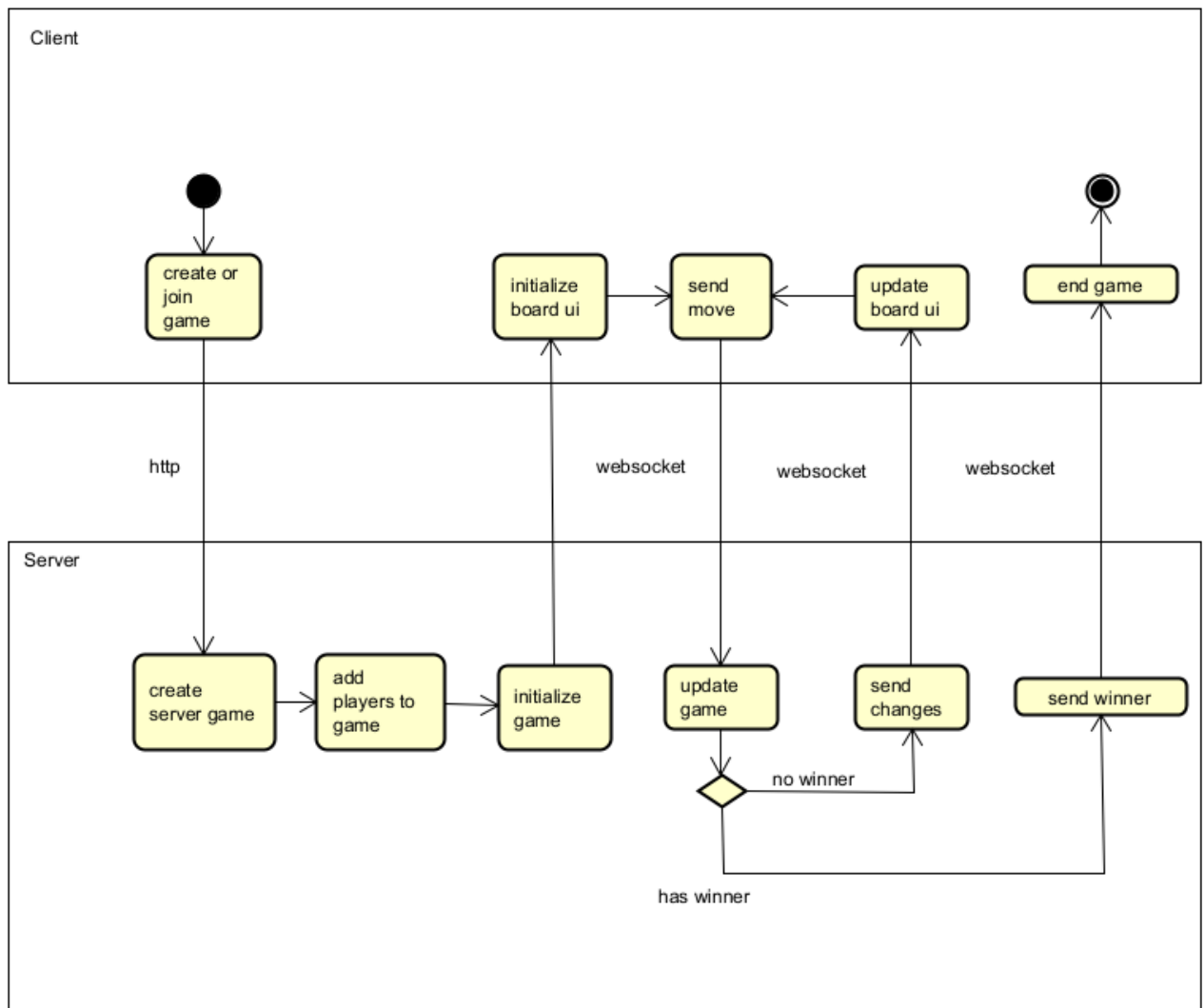


This follows a Model-View-Controller architecture. It consists of 4 main parts:

1. Server Repo: This contains the actual ongoing game state and gives the move updates to the core game logic to update the state.
2. Server Manager: This is what manages the client interactions for creating, joining, and sending moves to the game.
3. Client Manager: This is what handles server requests and sending ui inputs to the server.
4. Game UI: This is the client view of the game.

The games are created and joined through http requests and web sockets are set up so that the client can receive game updates and send moves over a consistent connection. The web socket approach is necessary for all players to receive updates from the server when a client move is made. Keeping both Rest and web socket endpoints allows for both the desktop and web clients to interact with the server in the same way. This allows for further expanding the application to a more fully featured website.

This is the high-level client-server interaction:



You can see more detailed, lower-level views of the structure and behavior in the Documents folder.

Current Project Goals

- Add user authentication and user accounts to the system using Spring Boots AOP framework.
- Upgrade from http to https.
- Add server error handling for dropped connections, wrong player moving, unauthenticated access attempts, etc.
- Upgrade/fix concurrency issues for game creation/joining and allow for multiple simultaneous games.
- Restructure architecture and improve overall encapsulation and separation of concerns.
- Add configuration files and config classes to remove hardcoded dependencies and better manage setup
- Add config and view manager to client ui
- Build C# desktop client
- Build Flutter mobile client
- Add new games such as Chess, Shogi, Go with plugin-based system
- Add user account features and database for storing accounts and game history
- Fix mobile ui issues
- Create new game creation/join game menus with a list of active joinable games

Development Process

I will be taking an iterative approach, following the agile/scrum methodology. I will adhere to these principles as I move forward in the project:

- Sprints will be 2 weeks long
- 5-10 tasks per sprint will be completed
- All functions and classes should be unit tested to validate requirements have been met
- Integration and system testing should be done before the end of each sprint
- At the end of each sprint there should be a working, usable version to push to main
- Tasks will be organized and tracked through a Taiga board
- Tasks should have clearly defined requirements with acceptance criteria
- Finished tasks should not be called complete until a pull request is made with a detailed description of any and all changes made and their purpose

Current Project Schedule

Sprint 1: 9/2-9/14

Goals:

- Fix server error handling, concurrency issues and upgrade to https
- Fix UI usability issues: unresponsiveness in mobile UI, create/join game process
- Build Flutter mobile version

Sprint 2: 9/15-9/28

Goals:

- Improve overall design and architecture and add configuration files and utilities
- Add proper user authentication and set up basic user account system
- Implement basic database features

- Implement chess and plugin architecture for adding additional games

Sprint 3: 9/29-10/12

Goals:

- Improve look and feel of all UI's (mobile, web, desktop) and improve overall usability
- Build C# desktop client
- Implement additional games: Shogi, Go