

PROJECT Design Documentation

The following template provides the headings for your Design Documentation. As you edit each section make sure you remove these commentary 'blockquotes'; the lines that start with a > character and appear in the generated PDF in italics.

Team Information

- Team name: Insert Name Here (Team B)
- Team members
 - Sree Jupudy
 - Kesa Abbas
 - Jon Pofcher
 - Scott Court
 - Chris Tremblay

Executive Summary

Purpose

Allow players to play checkers with other players that are signed in. Allow them to play on a board where pieces can be easily dragged and dropped.

Glossary and Acronyms

Term	Definition
VO	Value Object

Requirements

As a player I want to be able to sign in and see other people I can play checkers against. Once I have selected an opponent I want to be able to play a game of checkers by the American Standard Rules. I want to be able to make a move by dragging a piece when it is my turn. I want to be able to resign a game at any point and have it end, or end the game by winning or losing. I want to be able to sign out at any point after signing in.

Definition of MVP

Every player will be able to sign in and then play a game. Two players must be able to play checkers by the American Standard Rules. Players must be able to choose to resign at any point which will end the game.

MVP Features

- As a player I want to make a move so that the game progresses.
- As a player I want to end the game so that a winner is declared.
- As a player I want to start a game so that I can play checkers with an opponent.
- As a player I want to sign in so I can play a game of checkers.
- As a player I want to sign out so I can stop checkers.

Roadmap of Enhancements

- Spectate, As a Player I want to spectate a game so that I can see the moves being made in real time.

- Replay, As a player, I want to be able to replay the game so that I can see what moves I used in the previous game.

Application Domain

This section describes the application domain.

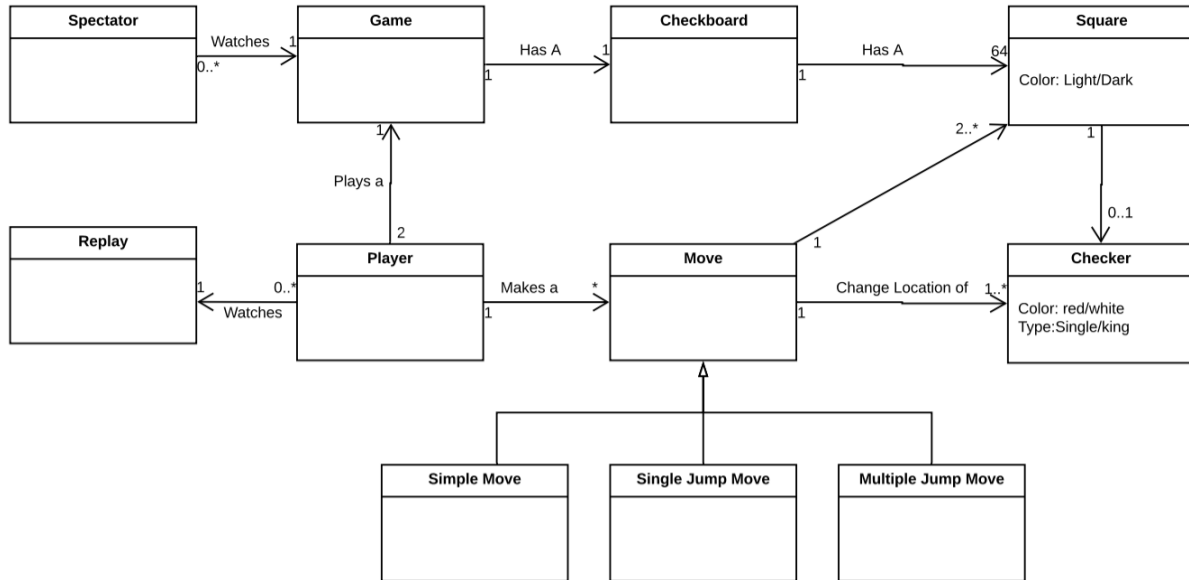


Figure 1: The WebCheckers Domain Model

A Player plays of game checkers, where they can make a move. A move changes the space the checker is at on the checkerboard. The checkerboard holds and displays the state of the game. A player plays a game of checkers against another player with a variable amount of spectators watching the game. A player can watch a replay of a game that has already been played.

Architecture and Design

This section describes the application architecture.

Summary

The following Tiers/Layers model shows a high-level view of the webapp's architecture.

As a web application, the user interacts with the system using a browser. The client-side of the UI is composed of HTML pages with some minimal CSS for styling the page. There is also some JavaScript that has been provided to the team by the architect.

The server-side tiers include the UI Tier that is composed of UI Controllers and Views. Controllers are built using the Spark framework and View are built using the FreeMarker framework. The Application and Model tiers are built using plain-old Java objects (POJOs).

Details of the components within these tiers are supplied below.

Overview of User Interface

This section describes the web interface flow; this is how the user views and interacts with the WebCheckers application.

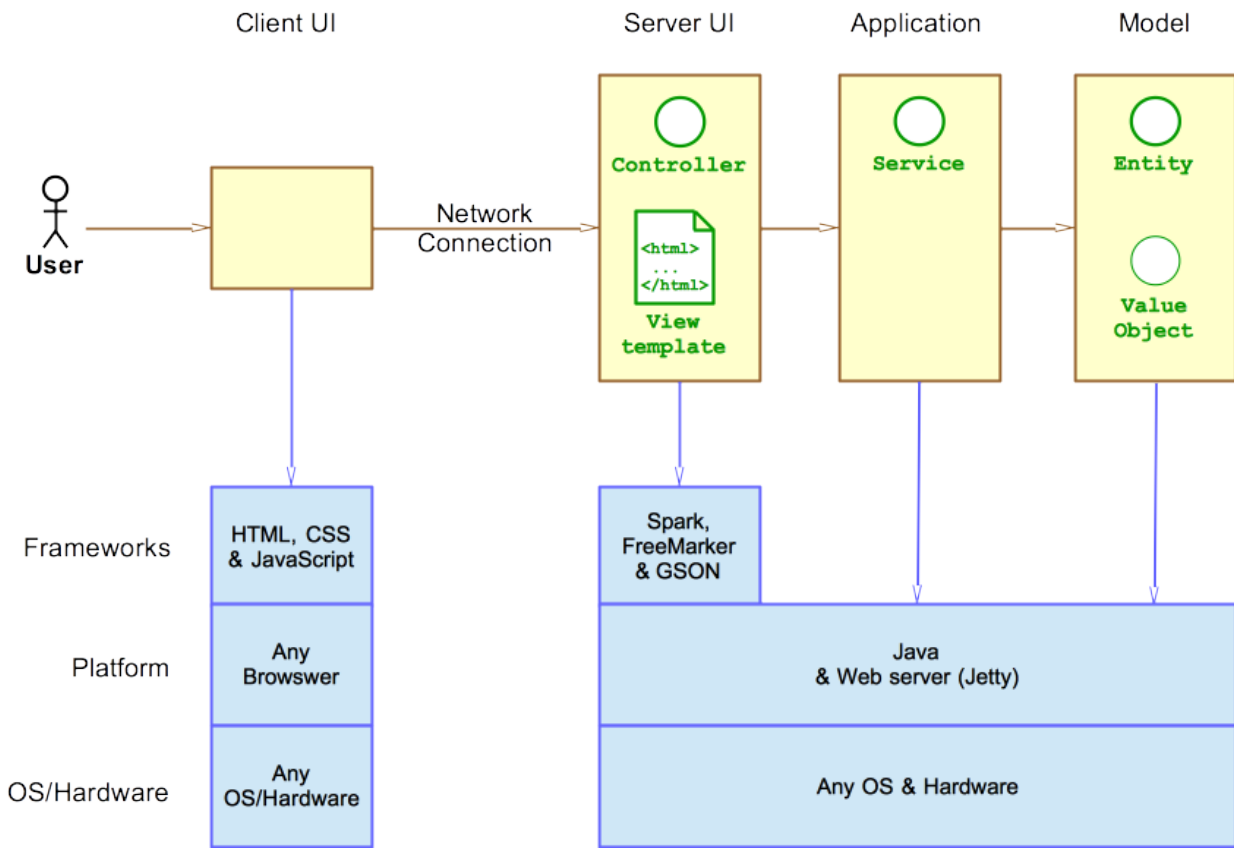


Figure 2: The Tiers & Layers of the Architecture

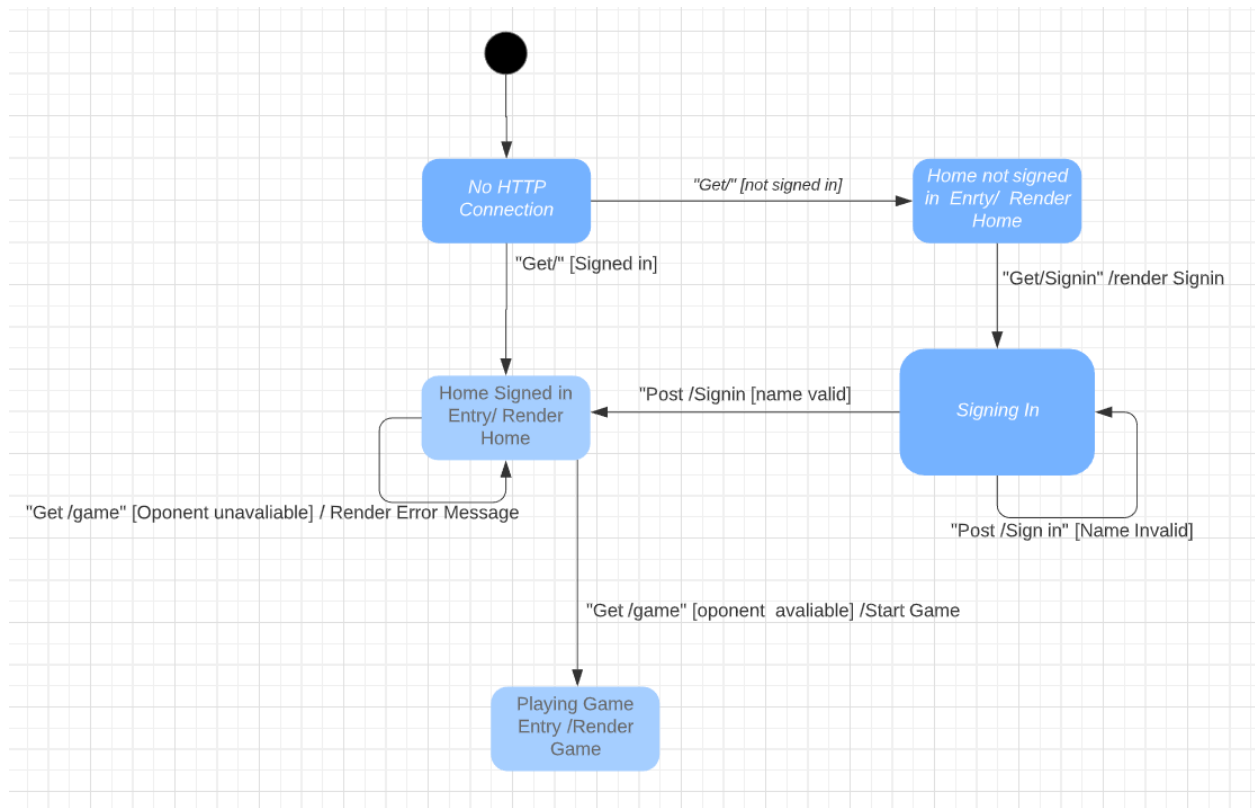


Figure 3: The WebCheckers Web Interface Statechart

When a player first connects to the WebCheckers site they will be met with home page and an option to sign in. Clicking the sign in button will take them to the sign in page where they can enter a name, and sign in with it. If the name is not acceptable it will redirect them back to the sign in page. If the name is accepted it will send them back to the home page where they can see other players that are signed in. Upon selecting another players name, it will send them to a game page where they can play their game of checkers.

UI Tier

Provide a summary of the Server-side UI tier of your architecture. Describe the types of components in the tier and describe their responsibilities. This should be a narrative description, i.e. it has a flow or “story line” that the reader can follow.

At appropriate places as part of this narrative provide one or more static models (UML class structure or object diagrams) with some details such as critical attributes and methods.

You must also provide any dynamic models, such as statechart and sequence diagrams, as is relevant to a particular aspect of the design that you are describing. For example, in WebCheckers you might create a sequence diagram of the `POST /validateMove` HTTP request processing or you might show a statechart diagram if the Game component uses a state machine to manage the game.

If a dynamic model, such as a statechart describes a feature that is not mostly in this tier and cuts across multiple tiers, you can consider placing the narrative description of that feature in a separate section for describing significant features. Place this after you describe the design of the three tiers.

Application Tier

Provide a summary of the Application tier of your architecture. This section will follow the same instructions that are given for the UI Tier above.

Model Tier

Provide a summary of the Application tier of your architecture. This section will follow the same instructions that are given for the UI Tier above.

Design Improvements

Discuss design improvements that you would make if the project were to continue. These improvement should be based on your direct analysis of where there are problems in the code base which could be addressed with design changes, and describe those suggested design improvements. After completion of the Code metrics exercise, you will also discuss the resulting metric measurements. Indicate the hot spots the metrics identified in your code base, and your suggested design improvements to address those hot spots.

Testing

This section will provide information about the testing performed and the results of the testing.

Acceptance Testing

Report on the number of user stories that have passed all their acceptance criteria tests, the number that have some acceptance criteria tests failing, and the number of user stories that have not had any testing yet. Highlight the issues found during acceptance testing and if there are any concerns.

Unit Testing and Code Coverage

Discuss your unit testing strategy. Report on the code coverage achieved from unit testing of the code base. Discuss the team's coverage targets, why you selected those values, and how well your code coverage met your targets. If there are any anomalies, discuss those.