

Domain Models

Transcript:

What can we do for you?

What I need is something like a platform that allows users to play chess online. Anyone could register to this platform, for example, by email address, which would be unique for that user. To register, the person should provide a nickname (also unique, maybe public???) and a password.

What can a user do on the platform?

Hmmm. User could create a new match (so user can play it). Since user can't play by herself, user should be able to invite another user to join the match. Perhaps user could send more than one invitation, something like "first come, first served", so the first user accepting the invitation will be the one joining the match. Is that possible?? I guess a user also needs to be able to reject an invitation, so it would be nice if the user who sent it received a notification.

It would be cool if a user could be part of multiple games simultaneously, though maybe user would want to quit any game at any time. I think a user would also want to be able to unregister.

The platform also needs to record the history of matches played by a user. Info like players, start and end dates and times, and results would be useful, you know, to know who won or lost or if there was a tie. I guess info about abandoned games should also be recorded. All this info would be part of the user profile, which can only be viewed by registered users.

The gameplay, well—chess has rules that must be followed during a match. Besides that, of course, a game can't start until enough players have joined, and I'm guessing that after a match starts, no other player should be able to join.

Who starts the match?

If I'm not wrong, that should be specified in the game's rules. Otherwise, the user who created the match would be the one making the first move. The system should be able to determine whose turn it is... according to the rules, right? A player can only make moves when it's her turn... allowed moves, that is... the rules.

What else?

Oh right. The state of the matches should be saved in some way, so the user can play whenever user wants. I guess that users won't be playing the whole time, so for

example, a **user** would make a **move** whenever it is **her turn** and **log out**, and after a while, **user** would **come back** and **check** if the other **player** made a move, then it's **her turn** again. **Asynchronous matches**, I think that describes it. The **system** needs to know when a **game** is **over** and should **inform** the **players** who **won** or **lost**, all according to the **rules**.

Nouns:

- *User: Cyrus*
- *Match: Cyrus*
- *Platform: Anthony*
- *Invitation: Anthony*
- *State: Casey*
- *History: Casey*
- *Rules: Ethan*
- *Game: Ethan*
- *Turn: Enzo*
- *Dates & Times: Enzo*
- *Players: Enzo*

Verbs:

- *Create a New match(Match, Game)*
- *Invite another User(Match, Invitation)*
- *Provide a nickname(User)*
- *Reject an invitation(User)*
- *Register(User)*
- *Unregister(User)*
- *Saved(Game)*
- *Received a notification(User, Invitation)*
- *Play whenever(Game)*
- *Be Saved(Game, State, Matches)*
- *Log Out(User)*
- *Determine whose turn(Game, Rules, Turn)*
- *Quit Game at any time(User, Game)*
- *Play whenever user wants(User, Game, State)*
- *Record(Platform, History)*
- *Recorded(Game, State)*
- *Inform results(User)*
- *Display info about abandoned games(Game, State, History, User)*