

---

# T08: Devcellence



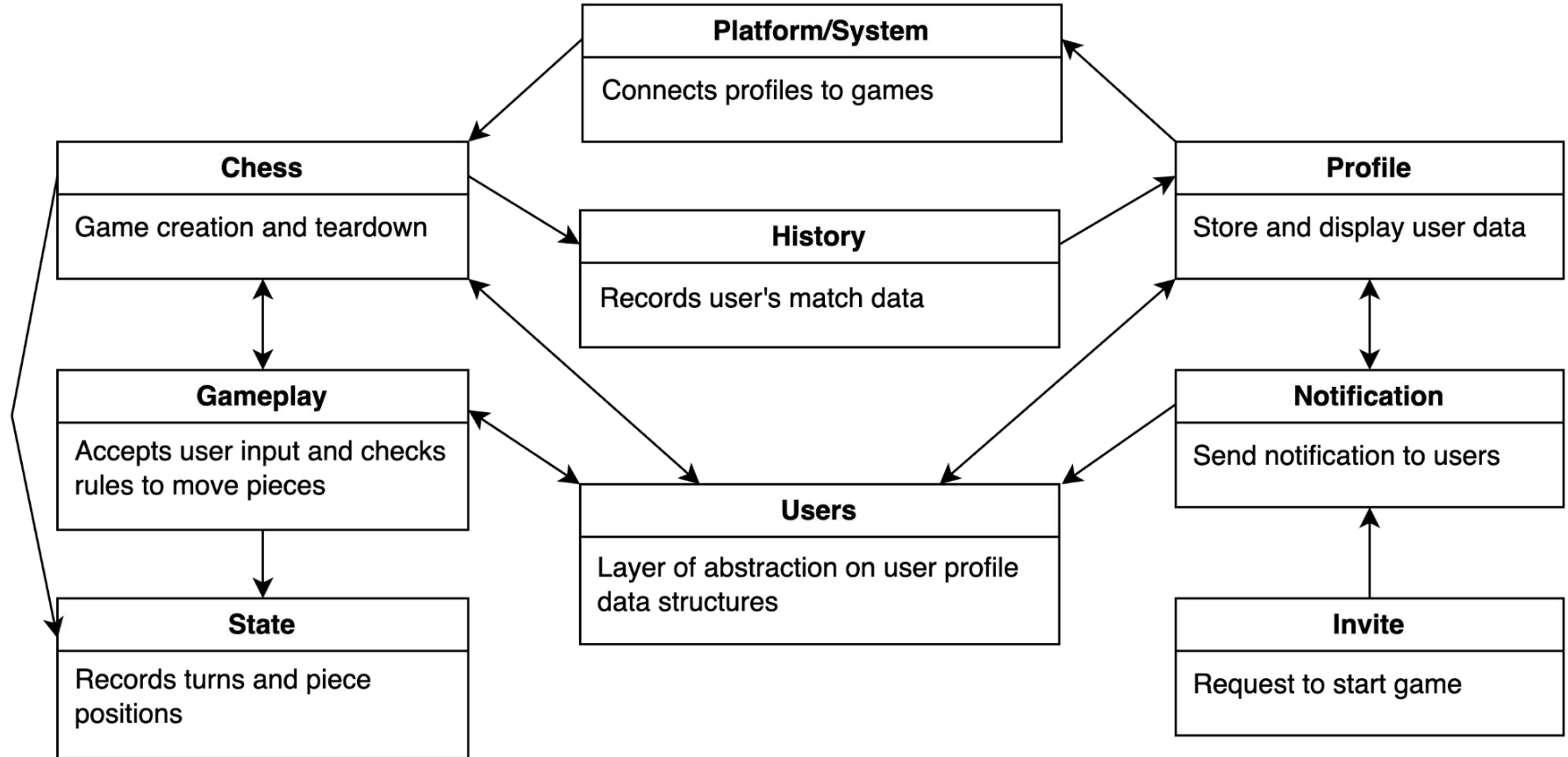
Charli Dolan, Ashly Hervey, Cooper Larson, Tyler Short, and Brexton Colburn

---

# Requirements (User Stories)

- **User stories** based on discussion of transcript
  - 10 total **stories**
  - 33 **tasks** (tentatively)
  - 3 **epics**
    - User Management
    - Game Management
    - Game Playing
- **Game Start Player Requirements-** As a user, I want ensure that a game can't start until enough players have joined
    - "...a game can't start until enough players have joined..."
  - **First Move-** As a user, I want to make the first move for any match I send an invitation for
    - "...Otherwise, the user who created the match would be the one making the first move..."
  - **Abandonment Penalty-** As a user, I want a competitive environment where users are penalized for abandoning games
    - "...info about abandoned games should also be recorded."
  - **Save Matches-** As a user, I want to be able to save my matches
    - "The state of the matches should be saved in some way..."
  - **Multiple Matches-** As a user, I want to be part of multiple matches at the same time
    - "It would be cool if a user could be part of multiple games simultaneously."
  - **Registration-** As a user, I want to register for the chess platform so that I can save progress
    - "Anyone could register to this platform..."
  - **Public Profile-** As a user, I want a profile that others can view with my match history
    - "The platform needs to record the history of matches played by a user..."
  - **Play Chess-** As a user, I want to play a chess game
    - "What I need is something like a platform that allows users to play chess online."
  - **Invite Players-** As a user, I want to send an invite to another player to join a match
    - "Since she can't play by herself, she should be able to invite another user to join the match..."
  - **Reject Invitation-** As a user, I want to be able to reject a match invitation
    - "I guess a user also needs to be able to reject an invitation..."

# CRC Cards Model



# Agile Development Approach

- **Tools Used**

- Scrum
- MS Teams
- GitHub Projects
- Google Drive
- Draw.IO

- **Key Artifacts**

- User stories
- Domain Model / CRC cards
- UML Diagram
- Growing backlog in GitHub Projects
- Project wiki
- Team collaboration & communication

- **Our approach to working as a distributed team:**

- **Communication:** We schedule at least one extended team call each week to ensure everyone, including online students, stay connected.
- **Accountability:** We operate with a shared goal and clearly defined expectations for each other. If you get stuck, ask for help.
- **Trust:** We rely on each other to complete tasks, when a team member says they will complete a task, it gets done.

# Conclusion

- **Main User Story**

- Focus on gameplay:
  - Play Chess- As a user, I want to play a chess game

- **We used**

- An agile development process via:
  - User stories, CRC cards, a UML Diagram, GitHub, and more

- **We Value**

- Communication, Accountability, and Trust

# Questions?

