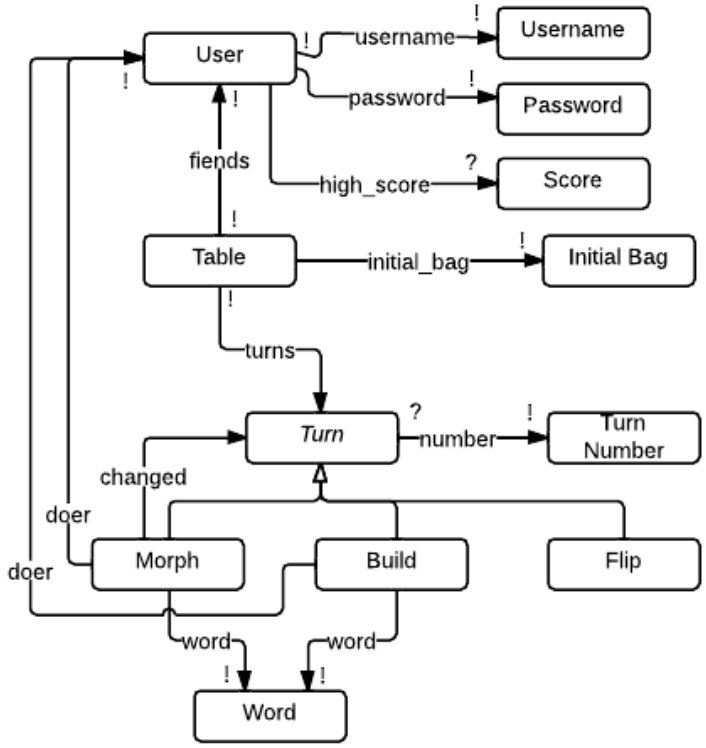
Teamwork Plan

1. **Stakeholders:** People who like word games.
2. **Resources:**
    a. Computational: We'll need a server to run the game. We'll use Heroku.
    b. Cost: none
    c. Time: 6.170 has due dates, and team members have other classes and things to do.
3. **Tasks:**
    a. Design:
        i. Purpose and goals: Damien
        ii. Context diagram: Leon
        iii. Key concepts: Leon
        iv. Data model: Ben
        v. Feature descriptions: Damien/Leon
        vi. Security: Damien
        vii. User interface: Bethany
        viii. Design challenges: All
    b. Implementation (main; secondary)
        i. Database schema (Ben)
            1. Not very much effort. Consists of creating a skeleton of each Rails model class, so that data fields and associations are laid out.
        ii. UI (Bethany; Leon)
            1. Very involved, on both the planning/design and coding fronts. Includes making images of game and lobby components, building views in HTML and JS, and styling with CSS.
        iii. Game model (All)
            1. One of the most involved parts. Includes the logic of the game rules, and all computations for creating and modifying game state.
        iv. Signup and accounts (Ben; Bethany)
            1. Not much effort, mostly built into Rails, just need to make views for it.
        v. Networking (Damien; Ben)
            1. Could be a significant effort. Will need to either get websockets working, or determine some other websocket-like medium (e.g. long pull), or change the rules of the game to eliminate time-dependence.

Minimum Viable Product

One-player games only. Just the gameplay component (no lobby).

1. **Feature descriptions**
    a. **Play Anagrams by yourself.** Just sign up, and you can play Anagrams in single-player mode!
    b. **Scores and Match History.** Keep track of how you're improving at the game.
2. **Issues postponed:**
    a. Multi-player games.
    b. Lobby, including "online status" and the process of challenging another user / accepting challenges within the lobby.
    c. Ranking system.
    d. Viewing and sharing replays.
3. **Reduced object model diagram**



Extra constraints:
In the relation changed: Morph -> Turn, the target Turn cannot be a Flip.

Notes:
Same as the notes for the full object model.