Stage 1: Deliverable

User: Fantasy hockey fans

Purpose: (a) Check fantasy hockey statistics - how a player is doing on a given day, cumulatively, and statistics (b) Play fantasy hockey

Function: What functions does your software provide to the user that help him achieve his goals?

a. Access statistics on a desired player (query / search the database)
b. Check leading players in a desired category (i.e. goals, assists, etc)
c. Check performance of players on a team by team basis
d. Draft teams/create fantasy leagues

How to use:

a. Database/query: input a Conference/player name to the command line. If list, how to sort.
b. Fantasy Draft: Manually type into command line
c. Fantasy League:
    i. Add, drop, trade players using the command line.
    ii. Back-end sets up matchups.
    iii. Computes standings by fantasy points.

Stage 2: Program Design

Requirements:
Create a fantasy league which includes a daily-updating database of players/conference/team player statistics, a fantasy draft functionality, and a fantasy league operation functionionilty (matchups, trades, add/drop, etc.).

1) Database
Storage class instance variables include:

a. Hashtable hashing players to slot of array by NHL team. Each slot contains a trie object of all players with the sort key being the player's last name. Value is either a player or a hashset of players.
b. Trie containing each player in the league

c. Two hashsets containing team in a conference (optional class, only implemented
d. Hashmap of string with team name to index of its slot in hashtable.

- Query whole league for a name: Search variable (c) (league-wide trie)
- Query conference: for each team in a conference, add all players in the team to a ___ and then sort
- Query specific team: Get index slot of team in hashtable from the hashmap and then pull/query the trie stored under that team's slot in the hashtable.

2) Draft

a. HashSet of team objects
b. New instances of 'Team' object: includes trie of players on team, instance methods (see #3) for league transactions, and team statistics (fantasy points per day, cumulative fantasy points)
c. Hashtable hashing players to slot of array by Fantasy Team (post draft). Each slot contains a team object Value is either a player or a hashset of players.
d. Trie containing all free agents / available players. This begins the draft as identical to the league-wide trie (variable (c) in the database), and as players are drafted, they are removed from this trie and added to a trie containing all busy / signed players.
e. Trie containing all busy / signed players - see (b)

<u>Player</u>: interface has three implementing classes:

A. Skater - Forward & Defensemen
    a. Goals
    b. Assists
    c. Shots on goal
    d. +/-
    e. Blocks
    f. Power play goals/assists
    g. Short handed goals/assists
    h. S/O goals
    i. Team
B. Goalie

a. Wins
b. Goals Against
c. Saves
d. Shutouts
e. Team

3) Fantasy league

1. TeamManager: Manages/computes standings, provides methods for trades, add/drop, etc.
2. Standings - SortedSet of Teams in TeamManager: Each team implements comparable and has variable keeping track of fantasy points, which is the base for compareTo.
3. Matchups - hashmap in teamManager. There is a separate hashmap containing the teams that each team already played.