Sports Wagering Database Proposal

Team members

- Christian Wettre
- Mahima Chandan
- Dale Pippert
- Amrutha Ravi
- Madeline Rys

GitHub URLs

Sports wagering project site

https://github.com/madelinerys/CS546-Final-Project

Sports wagering project proposal

https://github.com/madelinerys/CS546-Final-Project/blob/main/doc/ProjectProposalSportsWagering.pdf

Sports wagering database proposal (this document)

https://github.com/madelinerys/CS546-Final-Project/blob/main/doc/DatabaseProposalSportsWagering.pdf

Games collection

Each document describes an NFL game, capturing only those few aspects of a game that are relevant to the system, such as its start date and time, and final score. An NFL season is 17 weeks with 14 games per week, so this collection for a full season would have 17 * 14 = 238 documents in it. Each week, another 14 games are added to the table. Alternatively, it could be seeded with all 238 games at once. Either way an automated job is responsible for populating this collection.

Games schema

Field	Type	Description
_id	ObjectId	
away	String	designator for away (visiting) team
home	String	designator for home team
week	Integer	Week of the season, 1-17. NFL weeks start on Tuesday and end on Monday. As a frame of reference, 10/21/2020 is in Week 7. If you want to know, for example, what is the _current_ _week_ right now_ as a frame of reference you can go to https://www.espn.com/nfl/schedule and it should default to bring up the current week.
start	Date	Game start date and time GMT. The system will not allow bets to be placed on any game where **start** is >= current date and time. Upcoming games must be inserted into this table at least one week prior to *start*, so that bettors have a chance to place their wagers on the game.
ascore	Integer	Away team final score. Null up until when the game has finished and the system has processed the final score feed.
hscore	Integer	Home team final score. Null up until when the game has finished and the system has processed the final score feed.

Games example document

```
{
  _id: "5f8578c116e3409b5b276d50",
  away: "TEX",
  home: "TIT",
  week: 6,
  date: "2020-10-18T17:00:00.000Z"
  ascore: 36,
  hscore: 42
}
```

Games notes

1. GMT is four hours ahead of EDT, and five hours ahead of EST. For example, 1:00 PM EDT is 5:00 PM GMT.

2. Designators for teams defined in Teams collection as Teams.abbrv.

Teams collection

A seeded reference collection to store static information for all 32 NFL teams. This collection has exactly 32 documents in it, one per team.

Teams schema

Field	Type	Description
_id	ObjectId	
abbrv	String	Three-letter unique business key to enhance readability.
fran	String	Franchise name, typically locale/city/state of team.
nn	String	Team nickname

Teams example document

```
{
  _id: "5f85808c9dad05d358aae011",
  abbrev: "TIT",
  fran: "Tennessee",
  nn: "Titans"
}
```

Lines collection

Stores betting lines for each game. The system populates this collection on a week-by-week, day-by-day basis. It is updated with new lines every day or two via a background job.

Lines schema

Field	Type	Description
_id	ObjectId	
gameid	ObjectId	_id from Games collection
ltype	String	AML|HML|ASP|HSP|OV|UN (see notes)
num	Integer	The number, the meaning of which depends on ltype; may be positive, negative, or 0.
date	Date	mm/dd/yyyy

Lines example document

```
{
  _id: "5f85808c9dad05d358aae00a",
  gameid: "5f8578c116e3409b5b276d50",
  ltype: "ASP",
  num: -2,
  date: "10/16/2020"
}
```

Lines notes

1. ltype valid values are:
    1. **AML** Away team money line.
    2. **HML** Home team money line.
    3. **ASP** Away team spread.
    4. **HSP** Home team spread.
    5. **OV** Over points.
    6. **UN** Under points.
2. There can be many documents for any given gameid and ltype.
3. For any given gameid and ltype, the current line for that gameid and ltype is the line with the most recent date.
4. For any given gameid and ltype, bets are always placed against the line having the most recent date, i.e., the current line.

Bets collection

Records and stores bets. Each document is a bet from a bettor aka user. The user interface is used to enter bets.

Bets schema

Field	Type	Description
_id	ObjectId	
bettorid	ObjectId	_id of the bettor from Bettors collection
lineid	ObjectId	_id of the line from Lines collection
amount	Number	Dollar amount of the bet
pays	Number	Dollars this bet pays, or null if still live
collects	Number	Total dollars this bet collects should bettor win; this is equal to amount + pays
paid	Number	Dollar amount this bet paid, or null if bet is still live; may be zero indicating this bet has resolved and was a loss for the bettor
enter	Date	Time and Date of the bet
end	Date	Time and Date the bet resolved, or null if the bet is still live

Bets example document

```
{
  _id: "3e85908c9dad05d2589ae104",
  bettorid: "4e9441Cc9dad05d268aff018",
  lineid: "8015617daebe1773199ec12C",
  amount: 50,
  paid: null,
  enter: "10/20/2020",
  end: null
}
```

Bettors collection

These are users aka bettors that have signed up. Possibly (time permitting) seeded with 1000 bettors for demo purposes.

Bettors schema

Field	Type	Description
_id	ObjectId	
username	String	
pwd	String	md5 hashed password
balance	Number	Dollar balance in account

Bettors example document

```
{
  _id: "3e85908c9dad05d2589ae104",
  username: "foghorn5",
  pwd: "d4ec6fe6cec7f63630376c0af7212a52",
  balance: 250.00
}
```

1. Surprisingly, this collection will be seeded with exactly the same 1000 people as are present in an earlier people.json lab. They all like to bet!