Final Assignment Course 2: Catch The Pink Flamingo

Hello!
I tried to conceptualize and represent the assignment as stated in the task description. I am not sure I got it perfectly. I think the task description is quite vague and airy. Nevertheless, I tried my best. Thank you for your time reviewing this. If you leave me a comment, it will be appreciated. Bye!

Assignment provided information

Let's imagine now that you are part of a company called, **Eglence Inc.** (If you took the Intro Course before this course, you might remember our imaginary company Eglence Inc.)

One of the products of Eglence Inc. is a highly popular mobile game called "**Catch The Pink Flamingo**". The *objective* of the game is to catch as many Pink Flamingos as possible by following the missions provided by real-time prompts in the game and cover the map provided for each level. The levels get more complicated in mission speed and map complexity as the users move from level to level.

It's a **multi-user game** where the players have to catch Pink Flamingos that randomly pop up on a gridded world map based on missions that change in real-time. For the player or team to move to the next complexity level, they need to have at least one point in every map grid cell, i.e., cover the whole world map. An example mission would be "Catch the Flamingos on land with stars on their belly" in which the player should only click on flamingos that match the mission criteria, in this case, stars and being on land. If the player tags any other flamingo on the map, he/she or his/her team gets a negative point (-1) on that map location.

After the initial **sign up**, a player (user) is asked to play the *Level 1* individually without joining any team. This is where the user gets trained as a player and starts building a game history. Level 1 is an easy entry to the game composed of only 64 (8x8) grid cells and longer, more obvious, fun missions. Upon completion of Level 1, the player gets asked if she/he wants to join any team or form a team and will continue the rest of the time as a team player even if that means the user is a 1-person team of her/his own.

At the beginning of each level, the game creates a brand new map with more cells than the level before. The complexity of the missions also increases. The missions change more frequently as the levels increase.

The players keep in touch via **chat boards** assigned to the teams and also via social media, e.g., Twitter.

There are some things to consider while designing an information system for this game:

- **Ranking of Users:** Each user will be ranked individually by the speed and accuracy of their click to completion. The rankings get tracked in real-time and can be viewed both via the mobile app and the website for the game. In addition to score, speed and accuracy based ranking, the other players can see what parts of the map the user has the most points for. The players are also categorized based on their history as "rising star", "veteran", "coach", "social butterfly" and "hot flamingo". These refer to the qualities of players in addition to the game statistics.
- **Ranking of Teams:** The teams are ranked publicly. There is a maximum of 30 members in a team and a minimum of 1 member. The players "ask" to join a team and get voted in when 80% of the team members allow. A team may choose to "recruit" if they think a player can contribute or "outvote" a player if a player is not contributing. The players are also allowed to change their teams and bring all their points along. The competition is built on "point-based economy" and it is encouraged by the game providers. When all players leave a team, the team automatically gets removed from public and archived by Eglence Inc.

- **In-game Purchases:** Users are allowed in game purchases including binoculars to spot the mission specific flamingos, special flamingos that count for more than one grid point, ice blocks to freeze a mission for 20 seconds when needed, and trading cards to transfer the extra points from some grid cells to the ones without any points.
- **Game Completion:** The game never ends, meaning that there will always be a more complicated next level. A challenge for Eglence Inc. is to keep the game interesting and engaging for players who have been around for a long time. They make use of big data analytics to make sure the veteran players are still around.

Assignment submission instructions

The goal of this assignment is to design the data structures required for creating the game. Remember, that there are three aspects of the game:

1) Designing the data required to run the game
2) Keeping track of what is going on when people are actually playing the game, and
3) Analyzing how people played past games so that we can improve the way the game is played

With this in mind, let us list what we would need.

First, the game will have **users**, i.e., the players and we need to know some details about them. This information will typically be a relational table that may look like this.

Users (userID:longInteger, userName:string, joiningDate:date, dateOfBirth:date, currentLevel:int, authenticationKey:string)

Diagrammatically, this would look like:

Users

userID: long	**UserName: string**	**joiningDate: date**	**dateOfBirth: date**	**currentLevel: int**	**AuthenticationKey: string**
100	Greg Olson	10/12/2015	4/8/1996	1	4AI2Q87DE670
101	Cathy Pearson	10/23/2015	3/17/1998	3	4J9FMG5O5F67
102	Joseph Forsyth	11/4/2015	10/19/1994	3	V73O7M82HJ9D

where **userID** is underlined because it is a primary key for this relation.

We also know that users would form **teams**. However, these teams would possibly grow and shrink with time. But it is reasonable to assume that one user will be a part of only one team at any time, and unless we know that the user joined another team more recently, the user belongs to the team she joined last. We can also model teams as relations, which might look like this:

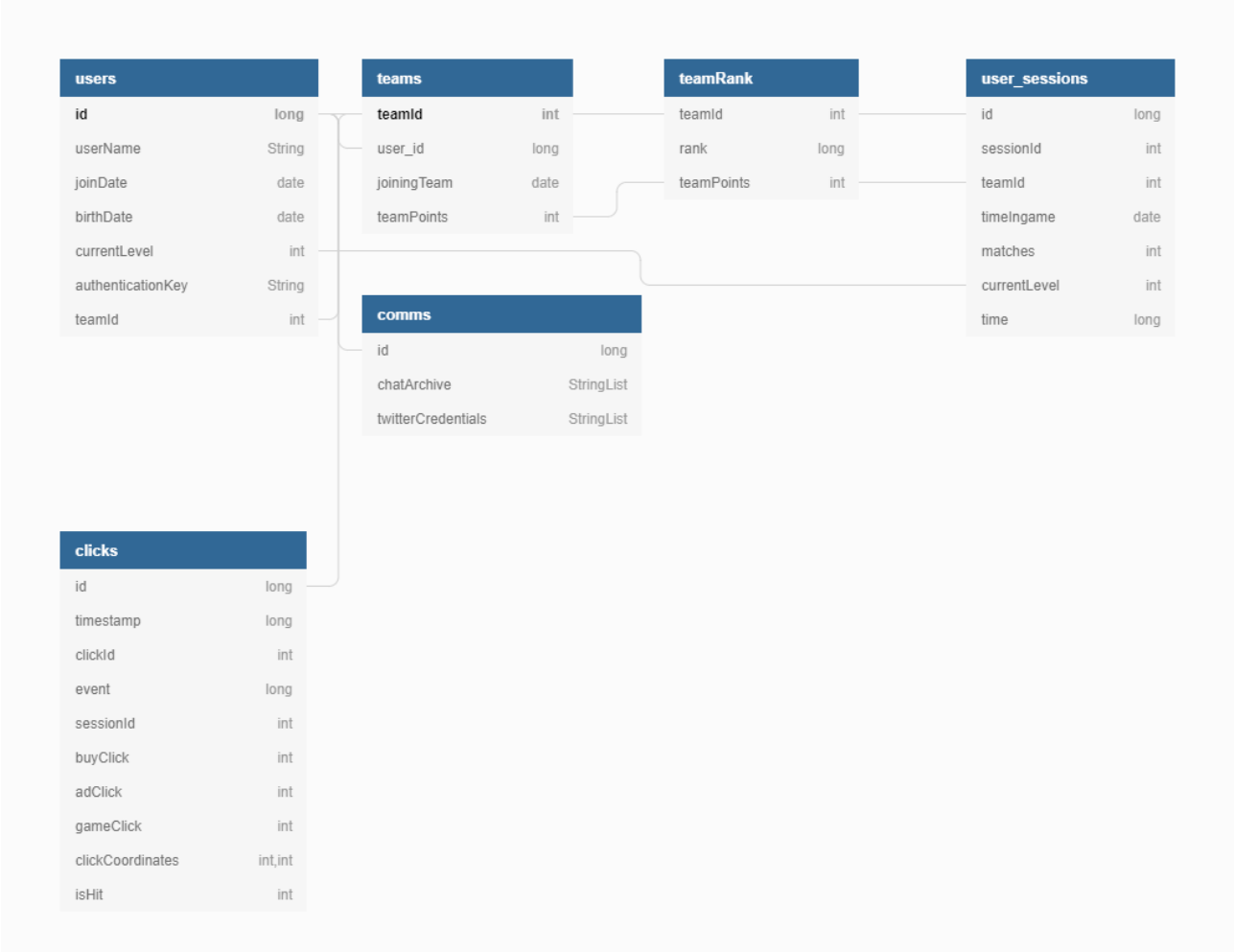
Teams

teamID: long	userID: long	joiningDate: date
10000	100	10/15/2015
10100	101	10/24/2015
10100	102	11/4/2015

With this table we should be able to determine the members of a team at any point in time, and track how a user has changed teams over time. Remember that a level 1 user will not appear in this table.

For this assignment, we will present incomplete specifications for a few data structures and your task will be to complete them.

Proposed database structure



Table/data file	Fields/data type	Description
users	id: *long*	Primary Key, incremental. Digits uniquely identifying each user.
	userName: *String*	User nickname
	joinDate: *date*	Date the user joined the platform
	birthDate: *date*	Age of the player

	currentLevel: *int*	Player laver in game
	authenticationKey: *String*	Player password
	team: *int*	Secondary key, incremental. Team eventually joined by the player when level 2 is reached
teams	teamId: *int*	Secondary Key, incremental Digits uniquely identifying a team
	user_id: *long*	Users' ids member of a specific team
	joiningTeam: *date*	Date a user joined a team
	teamPoints: *int*	Level of the team
teamRank	teamId: *int*	Secondary Key, incremental Digits uniquely identifying a team
	rank: *long*	Rank of the team
	teamPoints: *int*	Level of the team
user_sessions	id: *long*	User Primary Key
	sessionId: *int*	Tertiary key, incremental Uniquely identify users' sessions
	teamId: *int*	Secondary Key, current team identifier
	timeIngame: *date*	User time spent gaming
	Matches: *int*	Number of matches
	currentLevel: *int*	Current level
Table clicks	Timestamp: *long*	Unique time markers
	Event: *long*	Relevant event generated by click
	sessionId: *int*	Game session
	buyClick: *int*	Kind of buying event generated by click
	adClick: *int*	Kind of advertising event generated by click.
	gameClick: int	Kind of game event generated by click.
	clickCoordinates: *"int,int"*	In game click coordinates
	isHit: *int*	Is a flamingo hit?
comms	Id: *long*	Primary Key. User identifier
	chatArchive: *StringList*	Chat archive
	twitterCredentials: *StringList*	Social media credentials

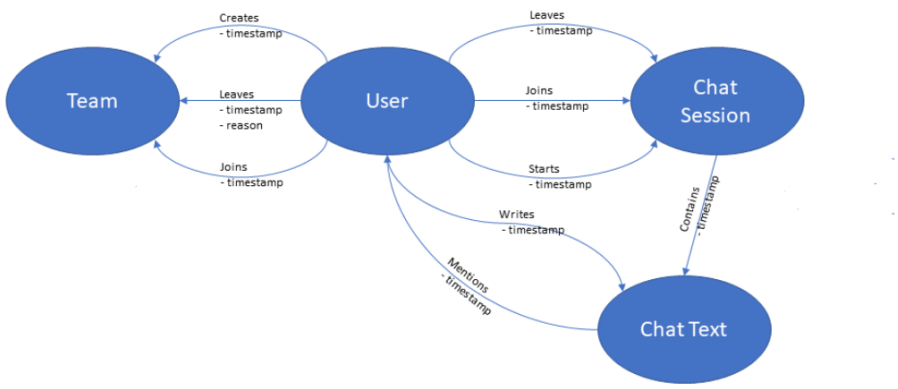
COMMS extended, social media and chats graph

This should conceptualize how I thought of the "Communication data", with focus on data management.

1. Which teams are having more conversations?
   That should be a simple counter. It is not specified if conversations means "contacts", number of words/token exchanged, I assume we are talking about "contact", hence a counter would suffice.

2. Do users chat more (or less) before they leave a team?
   Count chats for session_id.
3. What are the dominant terms (words) used in a chat session within a specific time period?
   Token analysis should be performed, given a text cleaned from stepwords and confounding terms, very common but also very uninformative.
4. Which users are most active in a specific chat session?
   Still, the problem should be accounted with periodical token counting.
5. How many chat sessions is a user participating in at the same time?

To address this question, every chat session should have an uniquely identifying key.



Tree properties

Flamingo_subtype_id should be extended with a list of properties. These could be fixed, or the result of combinations.

E.g.

flamingo_color (red, white, pink…)
flamingo_pattern (monocolor, bi-color…)
flamingo_frame (stars, squares, triangles…)

Since the game is conceptualized to be “infinite” in its progression, a combinatory approach appears to be preferable.