Model Documentation

Models Proposed:

- **Deck: Here we will save the data related to a deck, which is composed by three camps related to the kind of deck and its identifier .**
- **Card: This model stores all the information to identify a card, and extra information that will be used to filter and search by the fields explained in the following document, sorting it by different criterias.**
- **CardsInDeck: This informations is used to know which cards are contained in a deck, this way we can have all the information of deck and card separated and identify the containing cards in a simple way.**

Models fields:

<u>**Deck:**</u>
- **name: editable charField that contains the name of the deck.**
- **playerClass: this charField shows the player class of the deck choosing an option between the shown in the models file.**
- **deckString: this string is an identifier of the cards composing the deck this way two different decks can be the same and can be shared and builded by this string.**

<u>**Card:**</u>
- **CardID: this field identifies the card.**
- **DeckStringID: this field is a representation of the card when the string of the deck is builded.**
- **name: an string that contains the name of the card.**
- **cardSet: identifier of the set in the card collections of the card, this identifier, as many of the card model, will be used to filter cards in a search.**
- **type: information about the card workness, and the nature of it, as spell, creature and more, as mentioned this field will be used for filter.**
- **rarity: kind of rarity of the card, used for filter.**
- **cost: the mana cost of the card, used for filter.**
- **attack: the attack of the card, used for filter.**
- **health: int field representing the health of the card, used for filter.**
- **text: description of the card text including the effects of the card.**
- **race: rtace of the card, used for filter.**
- **playerClass: the player class of the card, it allows you to put a card in a deck, or not, if the class is not the same, it may be used for filter too.**
- **img: an url of the image of the card, we will use it to print and show the card.**

CardsInDeck:
- **deck: an identifier of the deck.**
- **card: identifier of the card.**
- **numOfCards: a number of cards contained in a deck it must be 1 or 2.**

Deck

0..*

Deck
+ name: String
+ playerClass: String
+ deckString: String

has

0..*

Managed by Django   1

User

0..*

consults

0..*

CardsInDeck
+ deck: int
+ card: String
+ numberOfCards: int

0..*

Card
+ cardID: String
+ DeckStringID: String
+ name: String
+ cardSet: String
+ type: String
+ rarity: String
+ cost: int
+ attack: int
+ health: int
+ text: String
+ race: String
+ playerClass: String
+ img: String