TEAM 2 – SPRINT 6 Planning Report

Database localization

We have a centralized localization table to handle translations for multiple entities. This approach is highly scalable and avoids schema changes when adding new translatable entities or languages.

Localization Table:

- translation_id (Primary Key)
- entity_type (enum) - The type of entity being translated (e.g., 'card', deck')
- country_code (enum) - e.g., 'ru', 'ja' , 'fi'
- entity_id (INT) - The unique identifier of the specific entity (e.g., card_id or deck_id)
- translated_text (TEXT) - The actual translated content
- additional_text (TEXT) - The actual translated content , second row for answers

A unique composite index will be created on (country_code, entity_type, entity_id) to ensure that only one translation exists per country for a given entity instance. This prevents duplicate translations and guarantees data integrity.

All operations will be handled via their respective DAO and Entity classes.

Example sql queue:

SELECT c.card_id, COALESCE(l.translated_text, c.front_text) AS front_text FROM cards c LEFT JOIN localization l ON l.entity_type = 'card' AND l.entity_id = c.card_id AND l.language_code = 'ru' WHERE c.card_id = 2;

TASKS:

- Adding changes to DB scheme
- Adding changes to the classes Entity, DAO classes
- Adding documentation and instructions for localization
- Bug fixing and testing
- Trello updates

Best Team 2,

Georgii Afanasev