Database requirements

Milestone 1: Database

- Requirement analysis:

- °User Information needs to be stored & handled.
- °There needs to be an extensive library with card objects for deckbuilding.
- °There needs to be a table that stores overall deck object information for proper allocation of user created content.
- °There needs to be shared table that stores information from the library and deck information for exact identification.
- °Access Control needs to be implemented in all layers.
- °Column data-types must be restriced to the bare-minimum for added security as well as performance.

- Planning:

°table: archiveuser: →

1. Needs a unique identifier, that is ideally generated automatically, for guaranteed identification.
2. Needs a user name for data privacy reasons.
3. Needs an email-adress and password for access control.
4. Email-adress needs to be unique in table and cannot be null, to avoid data theft.
5. Password needs to be hashed before being stored. (backend)

°table: cards: →

1. Primary Key card_id needs to be implemented manually, due to the advantage of an allready existing set-id for correct data storage.
2. Several columns will need implementing, that can't be null: card_id, card_name, card_type. These are absolutely required for identification and filter purposes.
3. Several columns will accept null values, since some of the objects to be represented, will simply not have these values, whilest others might: super_type, sub_type, mana_value, cmc.

°table: decklists: →

1. The primary key needs to be an automatically generated value, that has to be unique.
2. A foreign key user_id is needed for identification and access control.
3. It needs a deck_name for easier identification, when using archive functionalities.
4. It should have a timestamp for proper documentation.

°table: cards_decklists: →

1. Is a shared table and needs a Unique constraint on four coulmns for proper storage (card_id, deck_id, side_board, maybe_board). This ensures, that only one table needs to be implemented rather than multiple. This should increase performance.
2. The following columns cannot be null: card_id, deck_id, side_board, maybe_board.
3. A column for quantity is needed to track the exact amounts of Objects someone wants in their deckbuild.
5. A column for format as a filter option should exist.
4. card_id and deck_id are foreign keys. This should prevent the data table from storing just any object.

<The implementation and the chosen databse contents (for testing purposes) can be found in the deckbuilder_archive_xampp_MySQL_database_contents.pdf file.>