Actors:

*Guest*- anyone who is using the system without being logged in.

*Registered User*- anyone who is logged into the system with an account in the *User* table.

Guest Cases­

**Use Case Name:** **Create a New Account**

Actor: Guest

Steps:

1. The system asks the guest for a new password and unique username.
2. If the guest has provided valid input, they confirm the information and a new account is added to the database.

SQL Statements:

DELIMITER //  
CREATE PROCEDURE spNewUser  
(IN username VARCHAR(20), password VARCHAR(20), role VARCHAR(20))  
BEGIN  
 INSERT INTO user VALUES(username, password, role);  
END //

CALL spNewUser(‘sampleusername’, ‘samplepassword’, ‘basic’);

SELECT u.Username, COUNT(\*) FROM User u JOIN UserDeck ud ON u.Username = ud.UserName JOIN Deck d ON d.deckid = ud.deckid GROUP BY u.Username;

**Use Case Name: Search For a Card**

Actor: Guest

Steps:

1. The user picks search criteria from name, color, mana cost, artist name, etc.
2. The user provides input for each criteria they selected.
3. Search returns matching cards from the DB in a sorted list.

SQL Statements:

DELIMITER //  
CREATE PROCEDURE spGetAllCards  
(IN SetName CHAR(20), )  
BEGIN

SELECT c.cardname, ci.manacost, ci.power, ci.cardtype, ci.toughness, ci.cmc, s.setname,

si.releaseddate, COUNT(\*)

FROM Card c

JOIN Card\_Information ci ON ci.cardname = c.cardname

JOIN MTGSet s ON s.setname = c.setname

JOIN Set\_Information si ON si.setcode = s.setcode;

END //

CALL spGetAllCards();

SELECT c.CardName, COUNT(\*) FROM Ruling r

JOIN Card c ON c.cardname = r.cardname

WHERE c.cardname = “some card name”;

SELECT c.cardName, COUNT(\*) FROM Split\_Flip\_Card sfc

JOIN Card c ON c.multiverseid = sfc.multiverseid\_card

JOIN Card\_Information ci ON ci.cardName = c.cardName

JOIN Card\_Format cf ON cf.cardName = c.cardName

JOIN Card\_Type ct ON ct.cardName = c.cardName

JOIN Card\_Color cc ON cc.cardName = c.cardName

WHERE c.multiverseid IN (SELECT MIN(multiverseid) FROM Card GROUP BY CardName) GROUP BY CardName;

**Use Case Name: Search for a Deck**

Actor: Guest

Steps:

1. The user picks search criteria from deck name, name of the user who created the deck, and deck format name.
2. The user provides input for each criteria they selected.
3. Search returns matching decks from the DB in a sorted list.

SQL Statements:

DELIMITER //

CREATE PROCEDURE deckSearch

(IN \_username VARCHAR(20), \_decknameVARCHAR(20))  
BEGIN

SELECT d.deckname, u.username

FROM deck d

LEFT OUTER JOIN userdeck u on u.deckid = d.deckid

WHERE (d.deckname = \_deckname AND u.username = \_username);

END //

CALL spDeckCount(bob, bobsdeck);

**Use Case Name: View a Deck**

Actor: guest/registered user/admin

Steps:

1. The user requests to see a deck.
2. The system serves the detailed deck page.

SQL Statements:

DELIMITER //

CREATE PROCEDURE spGetDeck(IN \_deckid INT)  
BEGIN  
 SELECT c.multiverseid, c.CardName, si.SetName, ci.ManaCost, ci.CMC,

ci.Power, ci.Toughness, c.MultiverseID, dhc.MainboardQty, dhc.SideboardQty

FROM Card c

INNER JOIN Card\_Information ci ON ci.CardName = c.CardName

INNER JOIN Deck\_has\_Card dhc ON dhc.Card\_ID = c.ID

INNER JOIN MTGSet s ON s.SetName = c.SetName

INNER JOIN Set\_Information si ON si.setcode = s.setcode

WHERE dhc.deckid = \_deckid;

GROUP BY d.idDeck;  
END //

CALL spGetDeck();

Registered User Cases­

**Use Case Name: Create New Deck**

Actor: Registered User

Steps:

1. The system asks for a deck name (unique per user), deck format, and admins select whether the deck is a user deck or a player deck.
2. The user is brought to the deck view for their new deck.

SQL Statements:

DELIMITER //

CREATE PROCEDURE spNewDeck

(IN \_username VARCHAR(20), IN \_decknameVARCHAR(20), \_format VARCHAR(20))  
BEGIN

INSERT INTO Deck (DeckName, Format) VALUES (\_deckname, \_format);

INSERT INTO UserDeck VALUES (\_username, LAST\_INSERT\_ID(), true);

END //

CALL spNewDeck();

**Use Case Name: Add Cards to Deck**

Actor: Registered User

Steps:

1. From the deck view, registered user selects to modify their deck.
2. A dialogue is presented for the registered user to quickly search for cards within the deck’s format and add them to the deck.

SQL Statements:

DELIMITER //

CREATE PROCEDURE spUpdateQuantity

(IN \_iddeck INT, \_idcard INT, \_sideboardQty INT, \_mainboardQty INT)  
BEGIN

UPDATE Deck\_has\_Card SET SideboardQty = \_sideboardQty, SET MainboardQty = \_mainboardQty)

WHERE Card\_ID = \_idcard AND idDeck = \_iddeck;

END //

CALL spUpdateQuantity(15, 12309, 2, 3);