Software Requirements Specification

For Group 9 Flashcard Project

March 1, 2019

Version 1

Prepared by:

Saally Campos, Tonglin Liu, Adrian Ridder, Blake Silvernail

**Table of Contents**

Revision History.. 3

**1 Introduction.. 4**

1.1 Overview.. 4

1.2 Goals and Objectives. 4

1.3 Scope. 4

1.4 Definitions. 4

1.5 Document Conventions. 5

1.6 Assumptions. 5

**2 General Design Constraints. 5**

2.1 Product Environment 5

2.2 User Characteristics. 6

2.3 Mandated Constraints. 6

2.4 Potential System Evolution. 6

**3 Nonfunctional Requirements. 6**

3.1 Usability Requirements. 6

3.2 Performance Requirements. 7

3.3 Security Requirements. 7

3.4 Legal Requirements. 7

3.5 Other Quality Attributes. 7

3.6 Documentation and Training. 7

3.7 External Interface. 7

3.7.1 User Interface. 8

3.7.2 Software Interface. 8

**4 System Features. 8**

4.1 Required Features 8

4.1.1 Feature: Manage Deck 8

4.1.2 Feature: Study 10

4.1.3 Feature: Sharing a Deck 11

4.2 Optional Features 13

4.2.1 Feature Classroom Mode 13

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| 1 | 3/1/2019 | Group 9 | Initial Document |
|  |  |  |  |

# **1** **Introduction**

## **1.1** **Overview**

The flashcards application will be a web application that provides users with a electronic digital substitute for studying. The flashcard system will allow any user with an account to have their own sets of flashcards for studying. This user can edit, add, and delete cards and sets of cards, and share these flashcard sets with other users as well. The application will have functionalities that help the user study and learn the cards in their set.

This document provides a detailed overview of the requirements for the Flashcard web application. This document will outline the general design constraints, functional requirements, non-functional requirements, and system features for the application.

## **1.2** **Goals and Objectives**

The objective of this project is to create a digital alternative to other methods of studying, such as physical flashcards. The flashcards application is expected to:

1. Allow users to create accounts
2. Allow users with accounts to create a flashcard set
3. Allow owner of flashcard set to edit, add, or delete the cards and set.
4. Allow user to study flashcard set by randomly displaying one of the cards in the set, and having the user guess what is on the other side.
5. Keep a statistics history so the user can see their progress.
6. Allow owner of flashcard set to share with other users and give owner, copy, or use privileges to other users.
7. Have administrator users that can disable accounts and perform site maintenance.

## **1.3** **Scope**

The Flashcard application will provide users with the ability to create, edit, and study their flashcard sets digitally. Users will be able to access their Flashcard sets by logging into their account. Users will also be able to share and give different levels of access to other users.

## **1.4** **Definitions**

* **Flashcard Application** - the product being described in this document, it will fulfil the requirements specified in this document
* **Project** - the process of developing the Flashcard Application.
* **Flashcard** - a study tool consisting of two sides (frontside and backside) of textual information.
* **Deck/set** - a collection of flashcards
* **User** - the person who will be using the application. It will generally be students but it could be anyone who wants to learn or study some topic using flashcards, for example studying a foreign language.
* **Actor** - a user or other software system that is involved and receives value from a use case
* **Use Case** - a description of the interaction between the system and an actor that is goal-oriented and used to describe several scenarios that will arise throughout the use of the application.
* **Scenario** - one path of the use case
* **Developer** - the person or team that will be developing the system.
* **Stakeholder** - anyone who is interested in the project’s outcome, including customers, users, developers, testers, managers, etc.
* **Shall** - describes a mandatory requirement
* **Should** - describes a desired but optional requirement
* **May** - describes an optional requirement
* **Controls** - elements of the user interface, the elements that the users will interact with on the front end.

## **1.5** **Document Conventions**

Areas of this document that will be completed at a future date are marked TBD along with an assigned owner for the section and estimated complete date.

## **1.6** **Assumptions**

It is assumed that the user can access the machine on which the application runs via the internet.

# **2** **General Design Constraints**

## **2.1** **Product Environment**

Relational database using MySQL.

This software system will be a dynamic flash-card web application, and as such, will require interfacing with a relational database (MySQL). The database will be used to store information input by users as well as retrieve that information for presentation on the front-end webpage. Authentication data will also be stored within this database (username/password) to provide the ability to both authenticate users as well as present them with the information tied to their account.

## **2.2** **User Characteristics**

Student User: This can be any student at any grade level from elementary school to the college level. The older student are most likely proficient with technology, while younger students may need some guidance.

Teacher User: Teachers may want to create flash card sets to share with students to study. They will want to share their sets with use only privilege. Teachers that encourage technology in the classroom are usually proficient with technology.

Admin User: Admin users will have all the functionality of the above users, but will also need the ability to delete user accounts.

## **2.3** **Mandated Constraints**

The Flashcard application shall be a web application. It must also be a web solution that can be extended to mobile platforms. A relational database must be used to satisfy the requirement of data permanence.

## **2.4** **Potential System Evolution**

The Flashcards project may evolve later on into a mobile android or iOS application, so that users can access their Flashcards sets from anywhere. We want to create an overall study tool for students so the application may evolve to include various other study tools, such as creating a study plan for a specific flashcard set, and reminders to study the Flashcards as part of the study plan.

# **3** **Nonfunctional Requirements**

## **3.1** **Usability Requirements**

The system shall be user friendly to both young and older users. First time users should be able to create a new set of flashcards in under 5 minutes. The system’s usability will also be measured by a survey given to all first time users after using the application for at least 3 hours, the survey should conclude that 90% of users are able to create and edit set of cards, study the cards, and share sets of cards with other users without reading the user manual.

## **3.2** **Performance Requirements**

The application should open up in less than 5 seconds. The system should be able to support 30 users at once. No operation can take more than 5 seconds, 95% of operations can take no more than 3 seconds.

## **3.3** **Security Requirements**

Normal users will have account security through a secure login to their account and flashcard sets. After 10 failed attempts to login the user’s account will be locked for 10 minutes. If the user forgets their login information, username or password, they can have this information reset for their accounts through a verified email. Users will have owner rights to all of their flashcard sets, other users who are not owners of the set will not be able to edit the set. Users can assign owner, copy, or use privileges to other users. Administrator accounts can disable accounts and performs site maintenance as needed through a secure login to their admin account.

## **3.4** **Legal Requirements**

FERPA (Federal Education Rights and Privacy) should be taken into account in the case where teachers may want to use this to give students a flashcard set for any potentially graded work. Since the system will keep a history of a users score (how well they scored in a specific flashcard set), the system shall ensure that these scores are only visible to the appropriate users (the teacher and student user only)

## **3.5** **Other Quality Attributes**

The system must be a web solution that can be extended to mobile platforms.

## **3.6** **Documentation and Training**

Users should be able to use the Flashcard application without documentation or training. However, all stakeholders will be provided with a user guide and system documentation.

## **3.7** **External Interface**

### 

### **3.7.1** **User Interface**

The application’s interface should not interfere with the user’s learning experience. It should therefore be minimal and conducive to focused attention on the subject matter, not the interface.

Permissible media elements within the interface itself should be text or an image.

The interface should be intuitive. Therefore all tasks should be able to be completed without prior training. 90% of users should be able to create a new card deck in under 1 minute. 85% of users should be able to add a new card to a deck in under 1 minute. 95% of users should be able to view and flip a card in under 1 minute. 90% of users should be able to select a different deck of cards in 2 minutes.

The interface should convey both a professional and fun attitude. It should be professional enough that it could be used in a work setting for training purposes, but the colors and layout should be vivid to encourage feelings of excitement using the application.

### **3.7.2** **Software Interface**

TBD - Blake Silvernail 3/8/2019

# **4** **System Features**

## **4.1** **Required Features:**

### **4.1.1** **Feature: Manage Decks**

**Description:** The user will be able to create custom decks of flashcards. Priority: High

**Cost**: medium

**Risk**: low

**Value**: high

**Use Case: Create new deck**

**Actors**: All users

**Basic Path**

1. User clicks “Create new deck” icon
2. System displays new window with text box for title
3. User enters new deck’s title
4. User clicks “Save” icon to save the deck
5. System saves the new deck
6. System closes window

**Use Case: Add new card to deck**

**Actors**: All users

**Basic Path**

1. User clicks “Add card” icon
2. System displays new window with text box for front and back side of card
3. User inputs card’s front side contents
4. User inputs card’s back side contents
5. User clicks “Save” icon
6. System saves card to deck
7. System closes window

**Use Case: Edit cards in deck**

**Actors**: All users

**Basic Path**

1. User goes to settings
2. User goes to deck management
3. User selects correct deck
4. System shows user list of cards
5. User selects card
6. User clicks “Edit card” icon
7. System displays new window containing card’s information with editable text boxes for front and back side
8. User makes edits
9. User clicks “Save” icon
10. System saves card
11. System closes card window

**Use Case: Delete cards from deck**

**Actors**: All users

**Basic Path**

1. User selects settings
2. User goes to deck management
3. User selects deck
4. System displays new window showing list of deck’s contents
5. User selects card
6. System displays card’s information
7. User clicks “Delete” icon
8. System displays new window asking for confirmation
9. User selects confirm
10. System closes confirmation window and card window
11. System deletes the card

**Use Case: Delete deck**

**Actors**: All users

**Basic Path**

1. User goes to settings
2. User goes to deck management
3. User selects deck
4. System displays deck contents
5. User clicks “Delete” icon
6. System displays dialogue window asking for confirmation
7. User confirms choice
8. System closes confirmation window
9. System deletes deck contents as well

### **4.1.2** **Feature: Study**

**Description**: Users will be able to study any deck of cards that they have created or own.

Priority: High

Cost: medium

Risk: low

Value: high

**Use Case: Create Random Pool (Quiz) of Cards**

Actors: Student/Teacher

Basic Path

1. User selects an initial category of cards.
2. System displays cards in collection.
3. User clicks a button to generate a flash card quiz.
4. System generates a random subset of cards from the selected category.

**Use Case: Answer Verification**

Actors: Student

Basic Path

1. System displays the question associated with a card.
2. System prompts user to type in the answer.
3. User enters answer into a text box.
4. User clicks “Submit” button to initiate the answer verification.
5. System compares data entered by user with data stored in the card.
6. If user data matches card data, system will display a “Correct” message on screen.
7. If user data does not match card data, system will display an “Incorrect” message on screen.

**Use Case: Track User History**

Actors: Student

Basic Path

1. System will store the number of questions answered within each set of cards as well as the number of correct answers.
2. When a user correctly answers a flash card, both the overall count and correct count will be incremented.
3. When a users answers a flash card incorrectly, only the overall count will be implemented.
4. Users can click a “View History” button within a card set to display an overall “grade” for a particular card set.
5. Users can click a “Clear History” button within a card set to clear previous history.

**Use Case: Change Selected Card Set**

Actors: Student/Teacher

Basic Path

1. System will display a list of card sets owned by user.
2. User clicks on an available card set.
3. System will display selected card set.

### **4.1.3** **Feature Sharing a Deck**

Description: Users will be able to share their decks of cards with other users and assign owner, copy, or use privileges to other users.

Priority: High

Cost: Medium

Risk: Medium

Value: High

**Use Case: Owner Privileges**

Actors: User (Sender), User (receiver)

Basic Path

1. Once sender is in the settings for the desired deck, they will click they share icon.
2. The system will prompt them to select which users they want to share with.
3. The sender will type in the email address of the receiver.
4. The system will prompt the sender to decide which privilege the receiver will have (owner/copy/use)
5. The sender will select owner privilege.
6. The system will automatically send an email to the receiver to notify them that a new deck was shared with them.
7. After clicking the link in the email, the deck will be added to the receiver's account where they can view, use, and edit the deck as if it was their own.

**Use Case: *Copy privilege***

Actors: User (Sender), User (receiver)

Basic Path

1. Once sender is in the settings for the desired deck, they will click they share icon.
2. The system will prompt them to select which users they want to share with.
3. The sender will type in the email address of the receiver.
4. The system will prompt the sender to decide which privilege the receiver will have (owner/copy/use)
5. The sender will select Copy privilege.
6. The system will automatically send an email to the receiver to notify them that a new deck was shared with them.
7. After clicking the link in the email, the deck will be added to the receiver's account where they can view and use the deck.
8. If the receiver wants to make any changes to the deck, they will need to click the copy icon.
9. The system will copy the deck to a new deck in the receiver’s account, which the receiver can now edit as their own.

**Use Case: *Use privilege***

Actors: User (Sender), User (receiver)

Basic Path

1. Once sender is in the settings for the desired deck, they will click they share icon.
2. The system will prompt them to select which users they want to share with.
3. The sender will type in the email address of the receiver.
4. The system will prompt the sender to decide which privilege the receiver will have (owner/copy/use)
5. The sender will select Use privilege.
6. The system will automatically send an email to the receiver to notify them that a new deck was shared with them.
7. After clicking the link in the email, the deck will be added to the receiver's account where they can view and use the deck. The receiver will not be able to make any changes to the deck or copy the deck as their own.

## **4.2** **Optional Features:**

### **4.2.1** **Feature Classroom Mode**

**Description:** Users can also be teachers, the classroom mode will allow teachers to add students to their class where they can share cards to students, monitor students progress, and how well they are doing, and even give quizzes.

Priority: Low

Cost: High

Risk: Medium

Value: Medium

**Use Case: Create classroom**

Actors: User (Teacher), User (Student)

Basic Path

1. Teacher will be logged into their personal account, and click the create classroom icon
2. Teacher can add student users by entering their email address
3. After entering the email address, the system will automatically send students an email with a link that will prompt students to create an account and enroll them in the course automatically
4. Teacher can use feature 4.1.1 to manage any deck in their personal account and share anyone to the classroom, using feature 4.1.3.
5. The system will share the deck, with use privilege only, to all the students enrolled in the class.
6. Students will be able to log into their accounts and view the study cards assigned by teacher.

**Use Case: Manage Classroom**

Actors: User(Teacher), User(Student)

Basic Path

1. The teacher can create assignments (or quizzes) for students in the classroom by clicking “Create Assignment” icon.
2. The system will prompt the teacher to select the deck of cards they want to be included in the assignment, which specific cards they want the system to quiz students over, and the types of questions they want in the assignment (True false, multiple choice, fill in the blank).
3. The system will create an assignment using the selected cards, of multiple choice, true false, and fill in the blank questions.
4. The teacher can set a due date for the assignment.
5. The system will automatically email students to notify them of a new assignment.
6. The system will allow students to log in and do the assignment until the due date is reached, then the system will lock the assignment.
7. The system will record the students scores and make it visible to the teacher.