Software Project Management 2021/2022

Flashee

D2.1.1 – Software Requirements Specification

Authors

∉ Ângelo Paiva.
∉ Burak Tinman.
∉ Jan Frank.
∉ Álvar San Martin.
∉ Pedro Henriques.

Reviewer

∉ Ângelo Paiva.
∉ Burak Tinman.
∉ Jan Frank.
∉ Álvar San Martin.
∉ Pedro Henriques.

Approver

- « Version, approver’s name, preferably not in the author’s list »

Table of Contents

3. Functional Requirements
4. Non-functional requirements
5. Mock-ups

1. Introduction

1.1. Purpose

The purpose of this document is to serve as a guide to designers, developers and testers who are responsible for the engineering of the Flashee project. It should give the engineers all the information necessary to design, develop and test the software.

1.2. Scope

This document contains a complete description of the functionality of the Flashee project. It consists of use cases, functional requirements, and nonfunctional requirements, which, taken together, form a complete description of the software.

1.3. System overview

The system is designed as a web application and consists of flashcards which can be used to learn different topics in a fast and short manner. Flashcards can be created or imported by the user itself and then used to learn the new topics by a fast-paced quiz mode. This is achieved by flashcards which consist of two sides: a question side and a translation or answer side to check if the solution of the answer was correct. Furthermore, flashcards can be tagged by the user so they can be accessed and sorted easily.

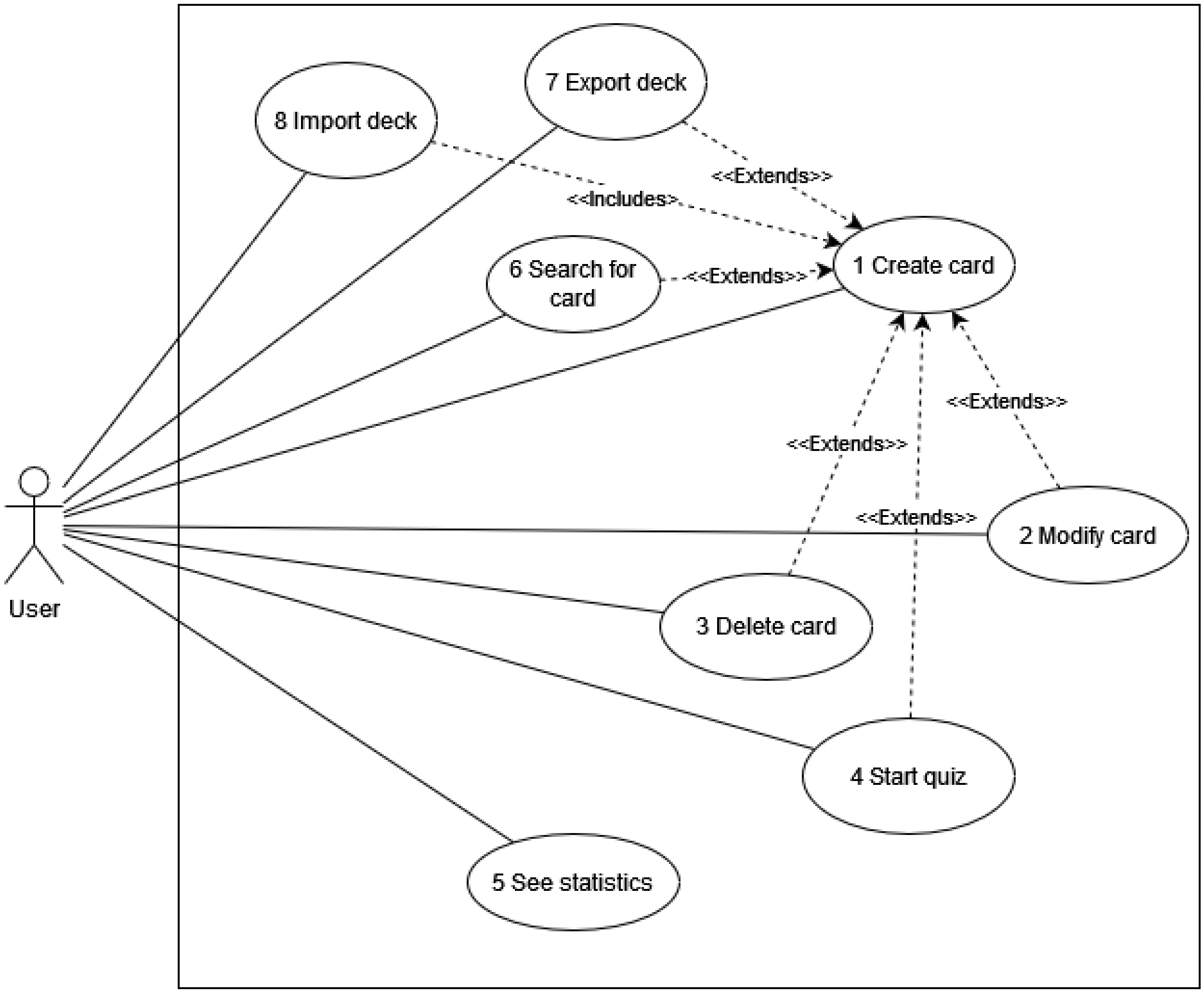
1.4. References

Any references to other documents should be included here. These may include other documents in the organization, work products, articles, and anything else that is relevant to understanding the SRS. If there is an organizational intranet, this section often includes URLs of referenced documents.

Software Development Plan: understand the plan of the software development
Quality Assurance Plan: features, metrics (so we can better understand some of the tests)
Quality Assessment Report:

2. Use Cases

The "Use Cases" section contains each of the use cases.
Divide into subsections as needed.
Present **Use Case Diagrams**.



8 Import deck
7 Export deck
<<Extends>>
<<Includes>
6 Search for card
<<Extends>>
1 Create card
<<Extends>>
<<Extends>>
<<Extends>>
2 Modify card
User
3 Delete card
4 Start quiz
5 See statistics

Name	**UC-1: Create card**
Summary	A user creates a card in his deck.
Rationale	The most important functionality of the program, the user creates a card from a scaffold with all the fields.
Users	Normal user.
Preconditions	-
Basic course of events	1. The user clicks in “Create new card” 2. A scaffold of a card is displayed with the fields 3. The user fills the fields and add tags 4. The user confirms the data is correct and the card is created
Alternative paths	1. If the action is triggered from an import deck function this function adds the new cards to the deck. 2. During the step 3 if any field is empty, the confirm button will not be available.
Postconditions	The card was saved correctly in the local storage as a JSON string.

36

Name	**UC-2: Modify card**
Summary	A user modifies the content of a card
Rationale	We assume a user can make mistakes while creating a card or wants to make changes after the initial creation, so a way of editing card is added.
Users	Normal user.
Preconditions	At least a card exists on the deck.
Basic course of events	1. The user clicks in the modification button of the card. 2. The fields will appear as modifiable. 3. The user changes the values. 4. The user confirms the changes.
Alternative paths	1. During the step 3 if any field is empty, the confirm button will not be available.
Postconditions	The card was saved correctly in the local storage as a JSON string.

37

Name	**UC-3: Delete card**
Summary	A user deletes a card he no longer needs.
Rationale	Since we can store an infinite number of cards the user has to be able to choose which cards, he wants to keep and which he no longer needs.
Users	Normal user.
Preconditions	At least a card exists on the deck.
Basic course of events	1. The user clicks on the delete button of a card. 2. A warning is displayed. 3. The user confirms the deletion.
Alternative paths	
Postconditions	The card was removed completely from the storage.

38
39

Name	**UC-4: Start quiz**
Summary	A random quiz is created for the user.
Rationale	With all the cards the user will be able to create random quiz to test his knowledge about an issue.
Users	Normal user.
Preconditions	At least a card exists on the deck.
Basic course of events	1. The user clicks on the "Quiz button" 2. The programs show options about the quiz, tags, number of questions etc. 3. The user chooses the options. 4. During the quiz, the user answers the questions from flashcards. 5. When the quiz finishes the user can see his evaluation. 6. The result is saved.
Alternative paths	2. During the step 3 if any field is empty, the confirm button will not be available.
Postconditions	The user saw the evaluation.

Name	**UC-5: See statistics**
Summary	The user can see his personal statistics.
Rationale	Since we store the evaluation after each quiz, the user should be able to see his statistics to check his progress.
Users	Normal user.
Preconditions	The user did at least one quiz.
Basic course of events	1. The user clicks on “Check statistics” 2. The program shows the statistics of the user like previous results
Alternative paths	3. During step 2 if any field is empty the program will instead show the user some screen saying “No quizzes done yet, check later”
Postconditions	The user saw their personal statistics.

Name	**UC-6: Search for card**
Summary	The user can search for a specific card.
Rationale	For editing or checking up a card, the user must be able to find the specific card he is searching for. The user will be able to make simple searches with the search bar included, displaying the resulting cards by creation date.
Users	Normal user.
Preconditions	At least one card exists in the memory.
Basic course of events	1. The user clicks on “Search card” button 2. The user writes question and / or tag of desired card 3. The user obtains the desired card
Alternative paths	4. If the specific card is not available, the user will see a message on the screen (“Card not available)
Postconditions	The user obtains the desired card for viewing and / or editing

42

Name	**UC-7: Export deck**
Summary	The user exports a group of cards (a deck).
Rationale	Since people can create cards, people will be able to share groups of cards so other people can use them without having to create them themselves. When you export your cards, you export all of them
Users	Normal user.
Preconditions	At least a card exists on the deck.
Basic course of events	1. The user clicks on the “Import/Export” button 2. The user chooses the “export” option 3. The user chooses the cards he wants to export 4. The user presses the “Export” button 5. The program confirms the deck was exported
Alternative paths	1. In step 4, if no cards are chosen, the “Export” button is greyed out
Postconditions	The cards are exported into the specified directory into a single JSON file.

43

Name	**UC-8: Import deck**
Summary	The user imports a group of cards (a deck)
Rationale	The user does not want to create cards again when using the application on previous sessions. The user can import a previously exported deck so he can continue with his progress. The program will display the cards that are going to be imported with the total number and the tags imported. If there is any problem importing the file nothing will be displayed.
Users	Normal user.
Preconditions	The import file must be a JSON file
Basic course of events	1. User clicks on “Import/Export” Button 2. The user chooses the “Import” option

	3. A file manager will open so the user can navigate to the specific JSON file 4. The user chooses JSON file and clicks on “OK” button 5. Application confirms imported deck
Alternative paths	2. If the file is not a JSON file or the JSON file is not in the expected form, an error will be shown instead, and no cards will be imported.
Postconditions	The deck is imported to the application.

44

3. Functional Requirements

Name	**FR-1: Non-case sensitive search**
Summary	The search feature must not have case sensitivity.
Rationale	This will aid users in finding the topic they want to study without worrying if the text is in uppercase, lowercase or a mix of both, thus allowing them to efficiently learn new things. https://www.merriam-webster.com/dictionary/search
Requirements	When a user invokes the search function, the software will automatically do a non-case sensitive search. The search will match any text that has the same letters as the search term, even if the case is different. If the search term matches the original text of the card, then it should lead the user to where that card/topic is located, independent of case sensitivity. The app will display the cards in the home screen, ordered by creation date.
References	UC-6: Search, All the mockups.

45

Name	**FR-2: Statistics**
Summary	The statistics the user sees must be actively updated and detailed.
Rationale	Being capable of viewing the statistics will motivate the users to learn even more or even compete with other users.
Requirements	These statistics must be accessible: • Quantity of cards in each deck. • Total number of cards in each topic. • Total number of topics. • Quantity of both successful and unsuccessful tries on each Topic.
References	UC-5: See Statistics, MK-5: Statistics

46

Name	**FR-3: Create card**
Summary	The user must be able to create cards.
Rationale	This is the basis of our entire application, without it there is no Flashee.
Requirements	The creation of a card must have: • Ability to assign them to a certain topic and/or deck; • Description (optional); • An answer/translation.
References	UC-1: Create card, MK-2: card creation

47

Name	**FR-4: Modify Card**
Summary	The user must be able to alter existing cards.
Rationale	In case the user is unhappy with how a card turned out, or it has a typo, or any problem whatsoever that might leave the user unsatisfied, we want to allow the user to edit the cards to fit them how they wish.
Requirements	The Modifying of a card must have: • Ability to reassign them to a different topic and/or deck; • Modify the Description; • Alter the answer of a certain card.
References	UC-2: Modify card, MK-2: card creation

48

Name	**FR-5: Delete card**
Summary	The user must be able to delete cards.
Rationale	In case the user finds a card or topic that they no longer have any use for, they will be able to remove it.
Requirements	The deletion of a card or topic must have: • Ability to choose what to delete.
References	UC-3: Delete card, MK-1: Home screen

49

Name	**FR-6: Start quiz**
Summary	The user must be able to test himself by doing quizzes.
Rationale	Since our app has the intent to motivate the user’s learning

Requirements	The app should allow the user to test their knowledge based on: • The topic of the cards. • A certain deck in specific. • Choose a random topic/deck at random to simply learn something new.
References	UC-4: Start test, MK-4: Quiz

50

Name	**FR-7: Export deck**
Summary	The user must be able to export his own decks.
Rationale	It will allow users to share decks between each other, being capable of learning from one another's knowledge.
Requirements	The exportation of a deck, card, or topic must have: • What details of the topic, along with which decks the user wants to share.
References	UC-7: Export Deck, All the mock-ups

51

Name	**FR-8: Import deck**
Summary	The user must be able to import other user's decks.
Rationale	It will allow users to import decks created by others. This way they will be able to begin far more efficiently learning new topics, as they will not have to create the cards, decks, or topics themselves.
Requirements	The importation of a deck, card or topic must have: • The ability to import data from other users via files.
References	UC-8: Import deck, MK-3: import deck

52

4. Non-functional requirements

Name	**NFR-1: Usability**
Summary	Self-explanatory design and error prone application.
Rationale	The users want to have an application with an easy-to-understand design and an error prone behaviour so the user can navigate smoothly through the application without any disturbances and unpredicted outcomes.
Requirements	The loading time of the application for every screen the user sees should be at a max of 1 second. Furthermore, user errors while using the application should be at a max of

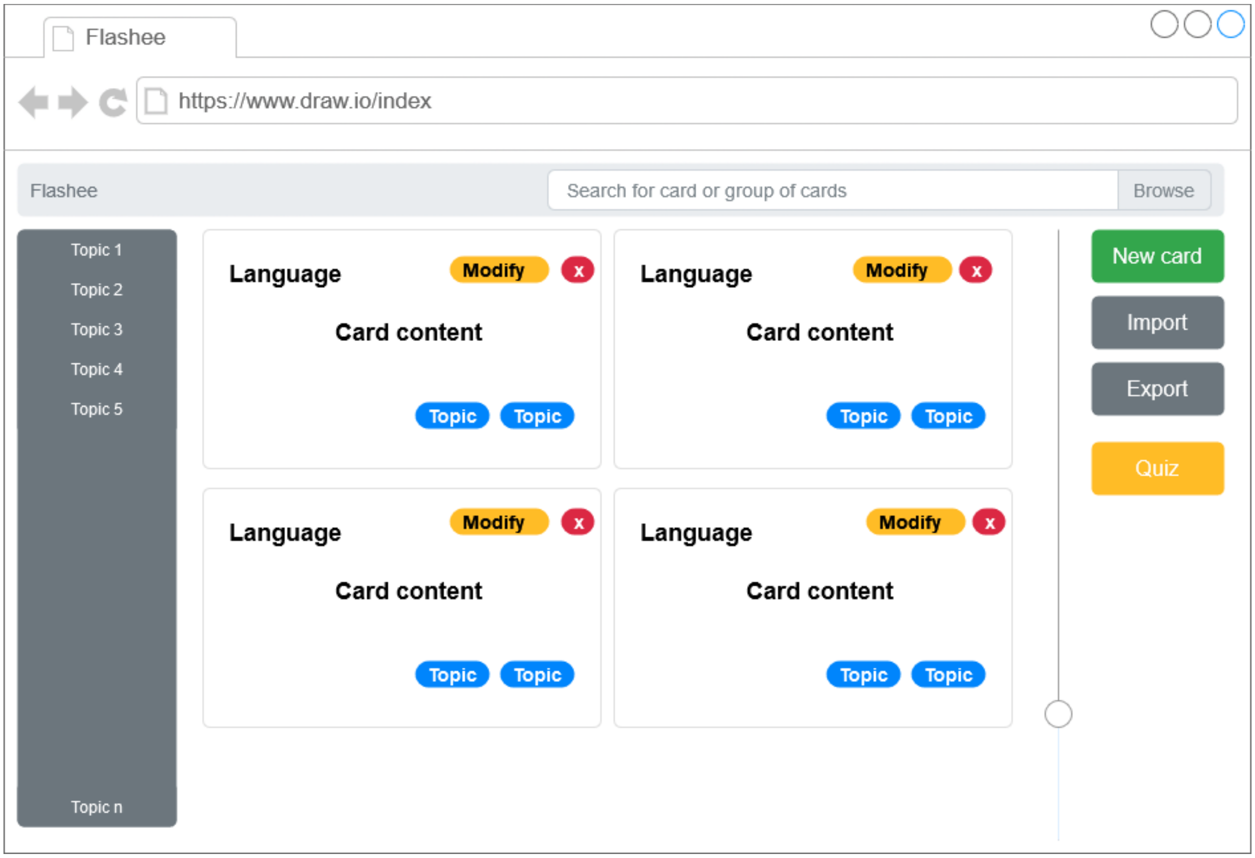
	3%. Lastly, the number of times the final application fails because of internal errors should be the 0.1%.
References	Quality Assessment Report, section 1

Name	**NFR-2: Portability**
Summary	Usage across several modern browsers.
Rationale	The users want to use the app regardless of their preferred browsers and devices they use.
Requirements	The users must be able to use the application by using either the Firefox, Chrome (or Chromium based) or Safari browser.
References	Quality Assessment Report, section 1

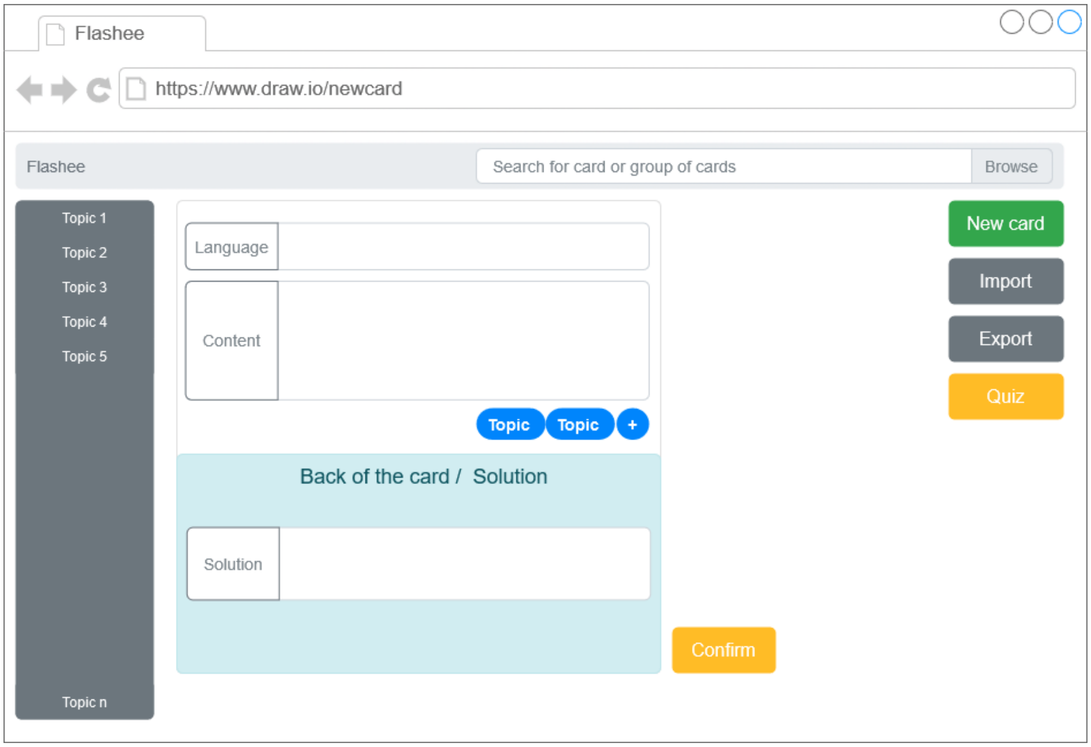
Name	**NFR-3: Performance**
Summary	Fast application.
Rationale	The users want to have an application with fast working functions to perform their actions smoothly and without interruptions.
Requirements	The complexity of the function used should not exceed $O(n^2)$
References	Quality Assessment Report, section 1

5. Mock-ups

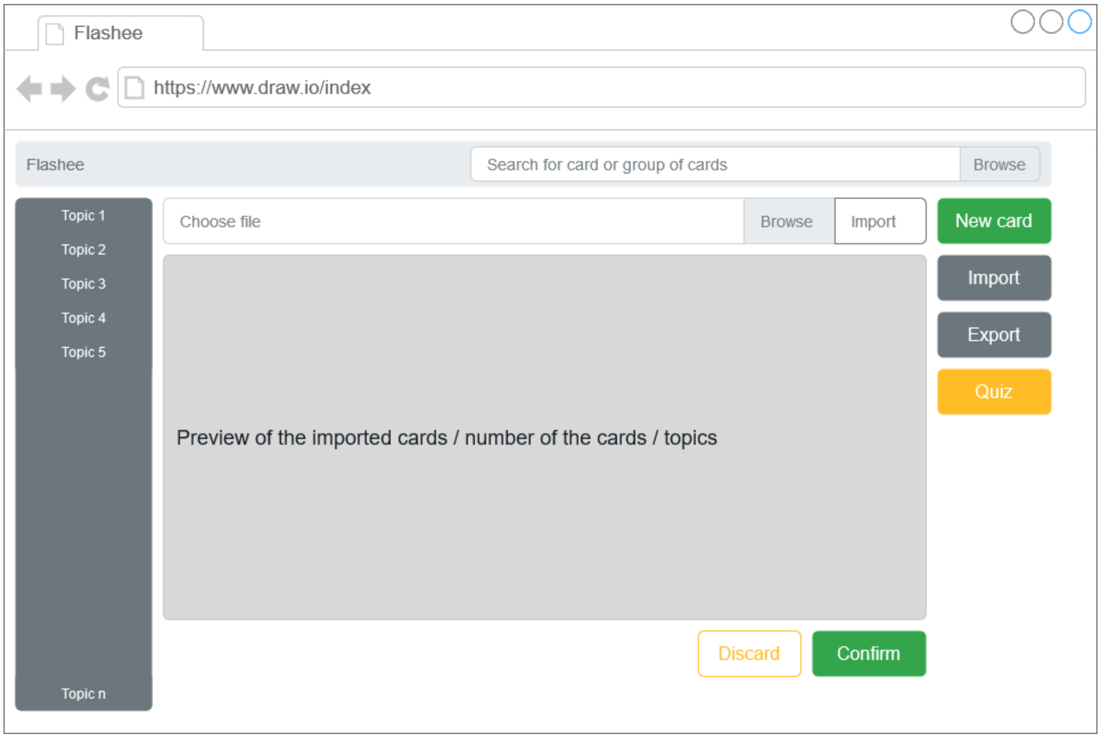
MK-1: Home screen



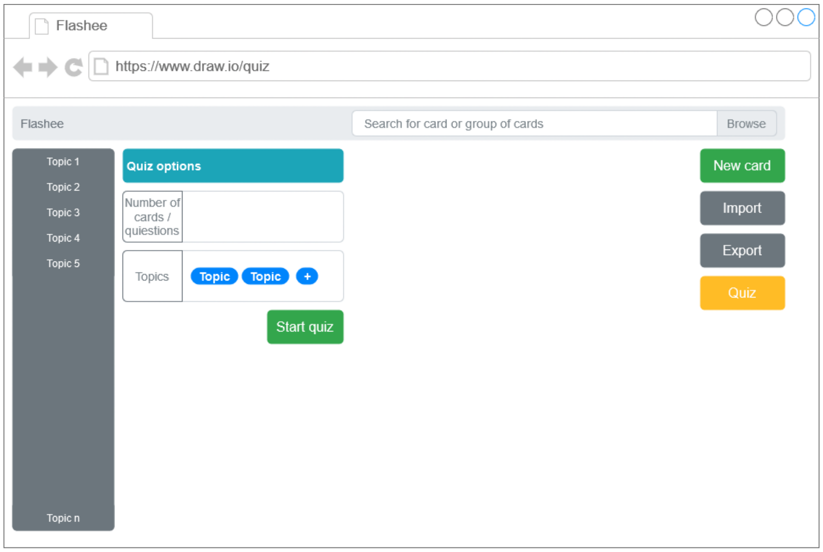
MK-2: Card creation

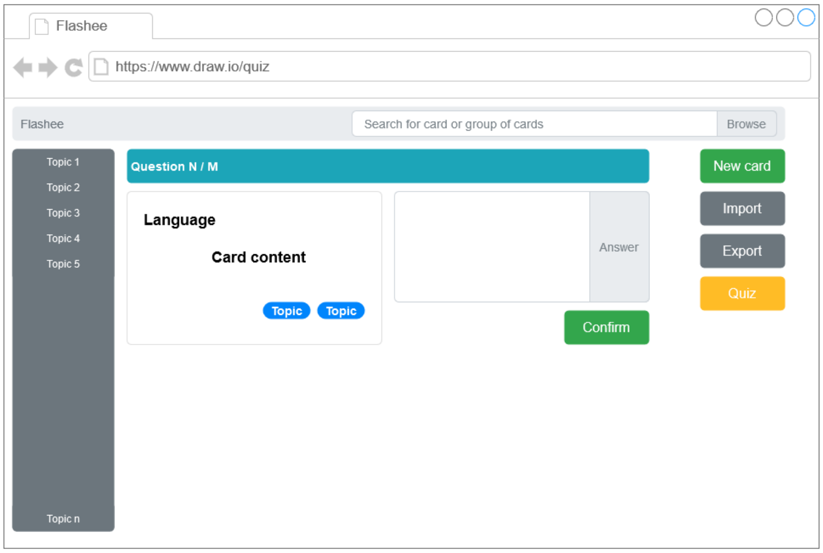


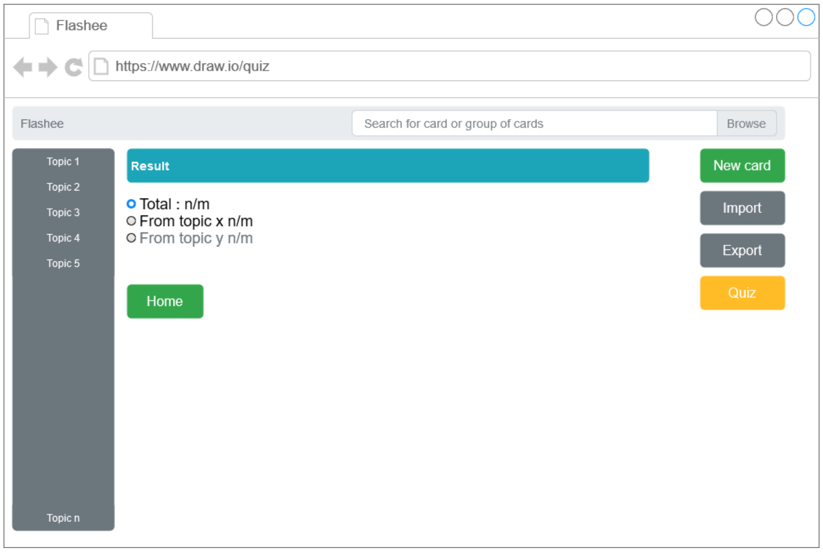
MK-3: Import deck



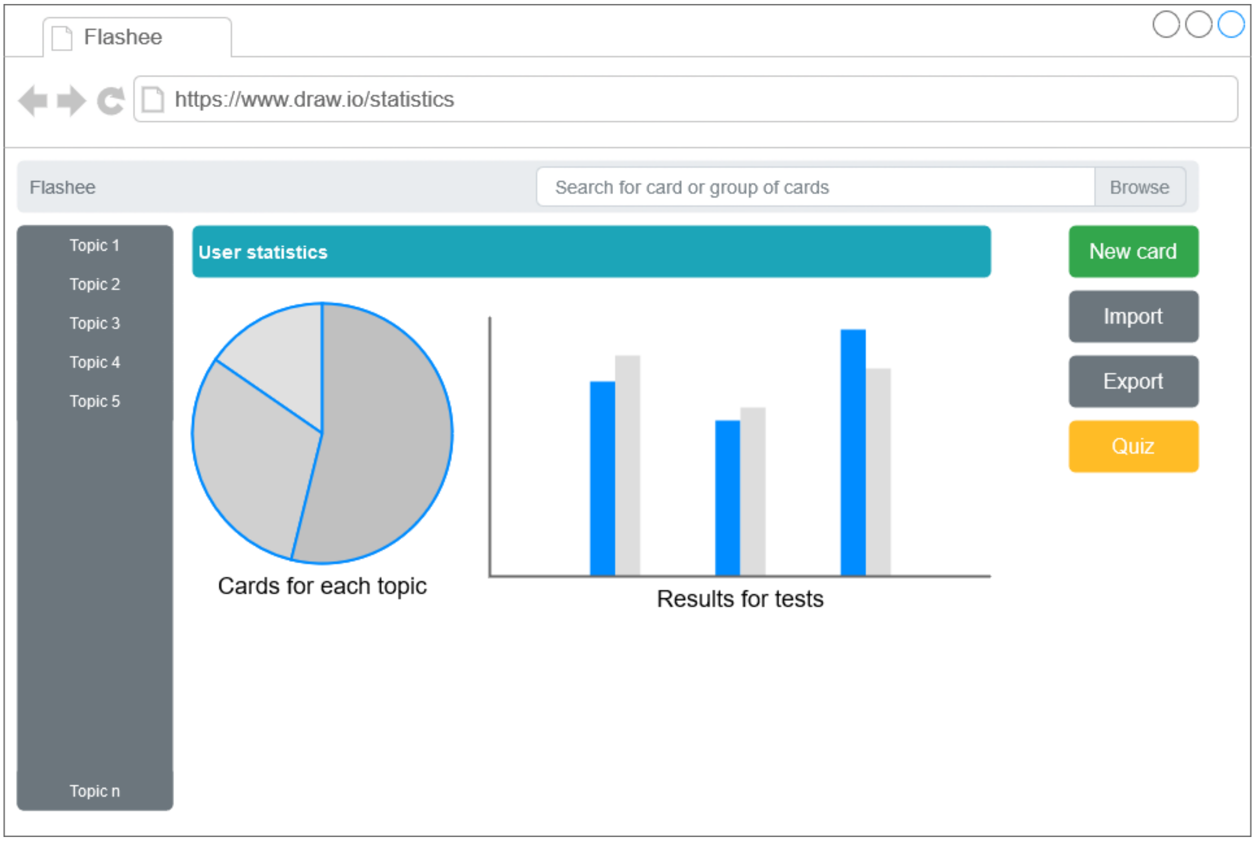
MK-4: Quiz


Flashee
https://www.draw.io/quiz
Flashee
Search for card or group of cards
Browse
Topic 1
Topic 2
Topic 3
Topic 4
Topic 5
Topic n
Quiz options
Number of cards / quiestions
Topics
Topic
Topic
+
Start quiz
New card
Import
Export
Quiz


Flashee
https://www.draw.io/quiz
Flashee
Search for card or group of cards
Browse
Topic 1
Topic 2
Topic 3
Topic 4
Topic 5
Topic n
Question N / M
Language
Card content
Topic
Topic
Answer
Confirm
New card
Import
Export
Quiz


Flashee
https://www.draw.io/quiz
Flashee
Search for card or group of cards
Browse
Topic 1
Topic 2
Topic 3
Topic 4
Topic 5
Topic n
Result
○ Total : n/m
○ From topic x n/m
○ From topic y n/m
Home
New card
Import
Export
Quiz

MK-5: Statistics


Flashee
https://www.draw.io/statistics
Flashee
Search for card or group of cards
Browse
Topic 1
Topic 2
Topic 3
Topic 4
Topic 5
Topic n
User statistics
Cards for each topic
Results for tests
New card
Import
Export
Quiz