

Requirements and Analysis Document for **Marc**

Thomas Li, Gustav Albertsson, Alexander Sandberg, Victor
Johansson, Mathias Forsman

2018-09-24

version 4

1. Introduction

For students, there will often be situations where they need to learn definitions and vocabulary, and while important, this is a process that can often become tedious and suffer from a lack of cognitive stimulation. *Marc* aims to make studying more rewarding and convenient by providing tools to easily organize the information that they need to study, as well as several different ways of practicing, by way of different kinds of more “game-like” methods, such as quizzes or other challenges. The application also keeps track of the user’s performance while studying in order to make it easier to view progression and improvement, encouraging the user to develop more regular review sessions. By showing stats and user progression, it is also easier for the user to feel accomplishment when they achieve goals, whether set up by themselves or as existing in-application challenges.

e they

- The application addresses the difficulty and general lack of stimulation that comes from studying to remember things such as vocabulary, that primarily needs to be memorized.
- The application will provide tools for organizing groups of user-defined information and tools to promote repeated reviewing of said information.
- The main user group will probably be students, who often needs to memorize definitions and vocabulary.
- Since the application will be developed for mobile devices, it is suitable for use on the go. For example, while commuting to and from school, one could use the perhaps otherwise wasted time to do some quick studying. The application should be designed to allow for quickly starting up and closing down review sessions.

1.1 Definitions, acronyms, and abbreviations

- “app” = Shorthand for “application”
- “cards” = Refers to entities containing two sets of information. One of the sets should contain the information that the user can see while reviewing, while the other should contain the information that the user should memorize, with the help of the first set. These sets will often be referenced to as the “front” and the “back” of the card.
- “gamemode” = Different kinds of exercises in the application for practicing the information contained in the cards.

2. Requirements

2.1 User Stories

User story

Story Identifier: 0

Story Name: Memorization Training Tool

Description

As a student, I want an application where I can practice my different subjects using flashcards and other types of fun exercises such as quizzes, so that I can study more efficiently.

Confirmation

Functional

- The application should allow for doing exercises with both flashcards and any other supported exercise types (such as quizzes).
- The application should allow for adding/removing/editing decks of flashcards.
- The application should keep stats of the user's progress and exercise results.

Non-functional

Availability:

- The application should be working on a phone using a version of Android of 5.0 or above (Lollipop).
- Flashcards and stats should persist between application sessions, on the same device.

User story

Story Identifier: 1

Story Name: Usable flashcard

Description

As a student, I want a flashcard to be visible in the application, with a pre-determined question written on the front, and the answer on the back, so that I can test myself on the information on the card.

Confirmation

Functional

- The card should be well fitted towards the current screen.
- The card should have some text written on it.
- The text written on the card should fit to the size of the card by being shrunk.
- The card should be clickable, which reveals the correct answer to the question.
- The card should be marked with a Q when the question is displayed.

- The card should be marked with an A when the answer is displayed.

Non-functional

None.

User story

Story Identifier: 4

Story Name: Show user performance

Estimated time: 6 days

Description

As a student, I want to be able to tell the application if my answer to the card was correct or not on a predefined deck, and for the application to keep track of my performance, so that I am able to see the result.

Confirmation

Functional

- The deck should contain a number of predefined flashcards, with questions and answers on the front and backs.
- The flashcards should each have a “correct” and “incorrect” button below.
- Clicking the “correct” or “incorrect” button should show the next card in the deck.
- The user inputted answer (correct or incorrect) should be saved till the end of the game
- At the end of the game, the results should be displayed.

Non-functional

None.

User story

Story Identifier: 5

Story Name: Building main menu

Estimated time: 3 days

Description

As a student, I want my application to start in a main menu, so that I can navigate to playing my deck.

Confirmation

Functional

- The main menu should have a button to navigate to the deck of flashcards.

###Non-functional

- There should be a header which reads “Marc”.

User story

Story Identifier: 7

Story Name: Add/remove/edit flashcards

Estimated time: 10 days

Description

As a student, I want to be able to add/remove/edit flashcards in the existing deck, so that I can modify my deck of cards.

Confirmation

Functional

- ~~The screen should be reachable from the home screen.~~
- The text on each card should be editable.
- Each card should be removable.
- There should be a (+) button, which when clicked, adds a card with a blank question and answer.
- ~~With the back button, you should be able to go back to the home screen.~~
- Tapping a card should enable editing of the card.
 - While editing a card, there should be buttons for saving changes, and for discarding changes.
- Swiping a card to either direction should mark the card for deletion. A button prompt should appear which allows the user to confirm the removal of marked cards.

Non-functional

- There should be a screen where all cards in the deck should be displayed with both question and answer.
- Removing, editing and adding a card should force the user to do such an action to the Note associated to the card.

User story

Story Identifier: 12

Story Name: Add/remove/edit cloze flashcards

Estimated time: 10 days

Description

As a student, I want to be able to add/remove/edit cloze flashcards in the existing deck, so that I can modify my deck of cards.

Confirmation

Functional

- The text on each card should be editable.
- Each card should be removable.
- There should be a (+) button, which when clicked, adds a card with a blank text input.
- The syntax for adding a cloze should be something unusual, which is not easy to write without intent.
- Tapping a card should enable editing of the card.
 - While editing a card, there should be buttons for saving changes, and for discarding changes.
- Swiping a card to either direction should mark the card for deletion. A button prompt should appear which allows the user to confirm the removal of marked cards.

Non-functional

- There should be a screen where all cards in the deck should be displayed with the current text written on them.
- Removing, editing and adding a card should force the user to do such an action to the Note associated to the card.

User story

Story Identifier: 8

Story Name: Add/remove deck

Description

As a student, I want to create/remove a deck of flashcards, so that I am able to keep collections of different types/groups of flashcards.

Confirmation

Functional

- Swiping a deck to either direction should mark the deck for deletion. A button prompt should appear which allows the user to confirm the removal of marked decks.
- A (+) symbol should be visible when you're in the "Decks" screen.
 - After clicking the (+) symbol, you should be presented with a pop-up which allows you to enter the name of the deck.
- Two buttons should exist for the deck creation pop-up, a confirm button and a discard button.

- Clicking the confirm button should present you with an empty deck, with the title visible on top.
- Clicking the discard button should present you with the “Decks” screen.

###Non-functional

User story

Story Identifier: 13

Story Name: Side menu

Description

As a student, I want to be able to navigate the application by using a side menu, so that I can navigate the application more quickly.

Confirmation

Functional

- The side menu should be reached when the toolbars’ menu button is clicked, or when the user swipes from the edge.
- The side menu should hold all main menu buttons.
- The side menu should hold a “Settings” button.
- The side menu should hold a “Logout” button.
- The side menu should be closed by clicking outside the menu, or by swiping the menu back to the left.

Non-functional

- The icons for the menu entries should match the corresponding entries in the home menu (if applicable)

User story

Story Identifier: 11

Story Name: Users system

Description

As a student I want the application to have a system for using multiple (local) users, so that I can share a device with others such as family etc. and still have our decks separate for each other.

Confirmation

Functional

- If no user is logged in the app should start at a screen where the user can choose/create a user.
- When the profile has been selected, the main menu should appear.
- The picked user should persist between sessions; the app should automatically log the previous user in at application start, unless the user explicitly logs out from the side menu.
- The user should be able to delete their profile from within the settings menu, *after* logging in.
- The user should be able to logout using the side menu.
- The user profiles should be persistent between sessions.

Non-functional

- Decks and games shall no longer be associated with “MemorizationTrainingTool”, but rather by different users. Users should instead be associated with ”MemorizationTrainingTool”.

User story

Story Identifier: 9

Story Name: Persistent storage of decks

Description

As a student I want my decks/cards to persist between sessions so that I can easily practice my flashcards without needed to rewrite them everytime.

Confirmation

###Functional

- The Decks are saved between restarts of the app
- The Notes within a deck should be saved between restarts of the app

###Non-functional

User story

Story Identifier: 10

Story Name: Saving statistics

Description

As a student I want my stats to be saved, and saved between sessions so that I know how im performing on my different decks.

Confirmation

Functional

- Each deck should contain stats concerning the user's performance on the deck, as well as specific stats for different game modes.
- The following stats should be tracked:
 - Average percentage of correct answers, one value per game mode.
 - The user's personal best (highest achieved number of correct answers), one value per game mode.

Non-functional

User story

Story Identifier: 2

Story Name: Animated flipping of flashcards

Description

As a student, I want a good looking animation for when I flip a flashcard, so that I the application better resembles real life use of flashcards.

Confirmation

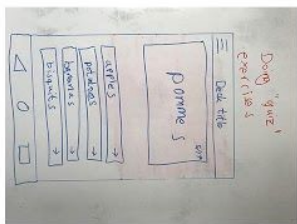
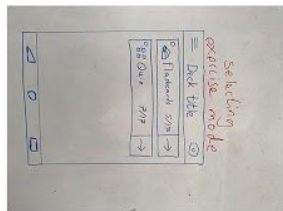
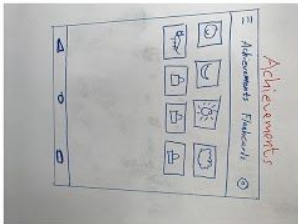
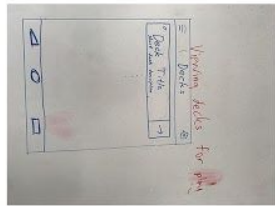
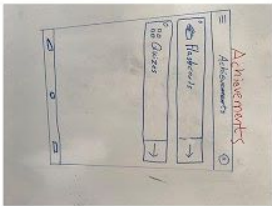
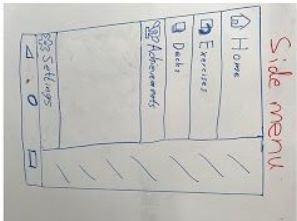
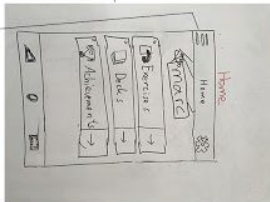
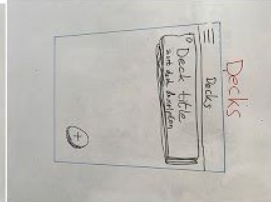
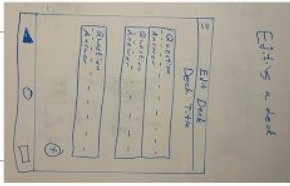
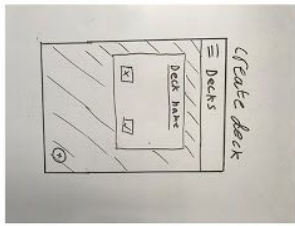
Functional

- When the text of the flashcard changes by tapping the card, the change should be animated by fading the text in/out

Non-functional

None.

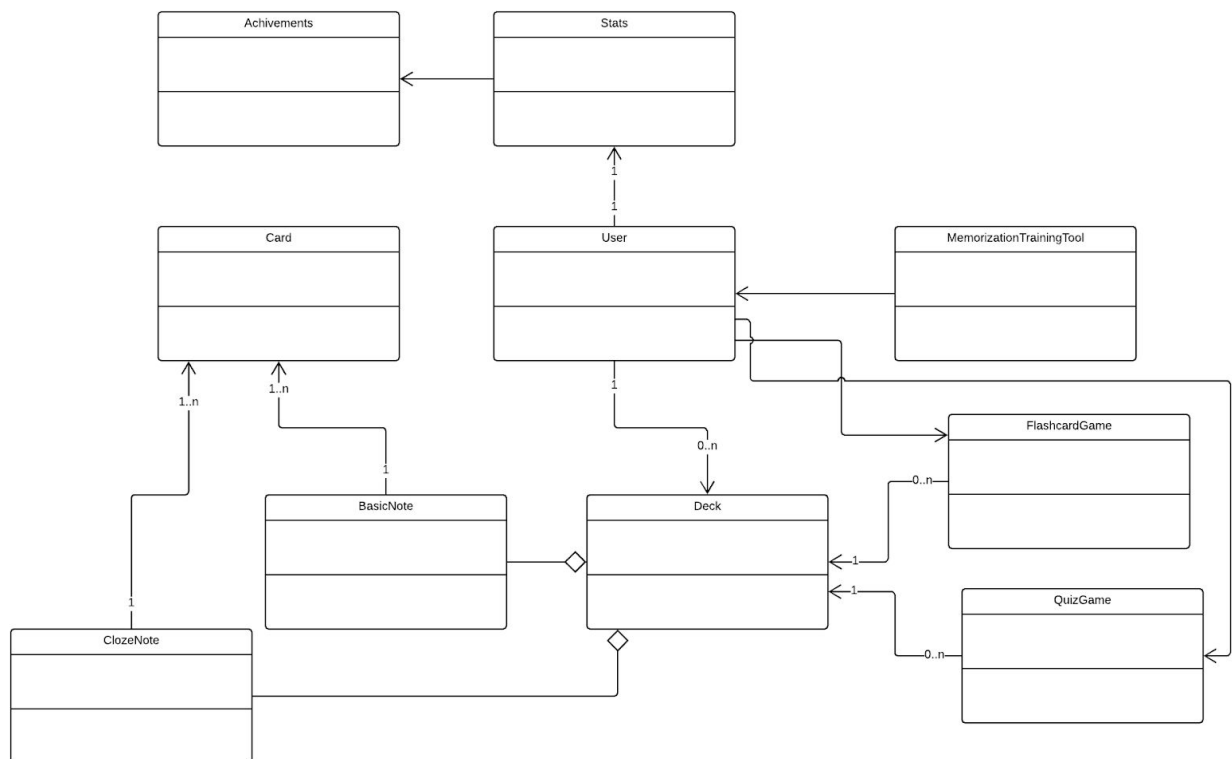
2.2 User interface



when finished

when finished

3. Domain model



3.1 Class responsibilities

The Deck class has the responsibility of holding cards that can be used in a exercise. It also holds user statistics for the specific deck.

The Card class holds the information for a flashcard, which includes text for the the front and the back side of the flashcard.

The Stats is an abstract class which holds basic statistics information.

The QuizStats class contains statistics that are specific for the Quiz gamemode.

The FlashcardStats class contains statistics that are specific for the Flashcard gamemode.

The Game class is an abstract class that holds some basic methods that a GameMode has.

The QuizGame class contains rules needed for the Quiz gamemode, using this class one should be able to play a game of Quiz on a deck. The idea then that the player wont be able to flip the card to see right answer but instead get four alternatives where one is right answer and the other 3 is randomized answers from another questions.

The FlashcardGame class contains rules needed for the Flashcard gamemode. If you use this class, these rules should be applied to the game.

The MemorizationTrainingTool class has the responsibility of holding references to all decks and the types of gamemodes that could be played.

4. References