How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: vukoye

MemoDict

Description

MemoDict helps you to create and study flashcards on any Android device. You can create different types of flashcards, you can add image to the question, or to the answer, also multiple answer card can be created.
After reviewing the answer app will ask you rate your knowledge. Based on this answer app will arrange when is right time to ask you again. It integrates scientifically proven smart repetition algorithm which optimizes order of flashcards in the set. If one flashcard is learned it will be shown to the user later than the unfamiliar one. This method of learning is more fun, addictive and much faster and efficient than traditional learning methods.

You can create as many decks as you need., also you can select deck to be private or public. Also you can search and study public decks. All data is stored on google cloud servers and all data can be accessed based on your credentials.

Intended User

Primary intended users are students, but app can be used by anybody how want to memorize something or to increase their vocabulary.

Features

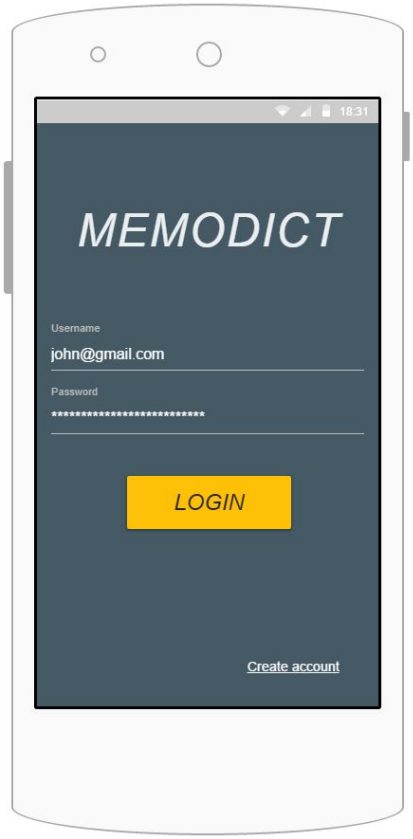
List the main features of your app. For example:

- Adding editing and deleting decks and flash cards
- Studying flashcard
- Decks can be public and private
- Saving them to the firebase server
- User database is also on firebase server.
- Image can be added to the card.
- Multiple answers flashcard is supported
- SM2 (SuperMemo 2) optimized algorithm will be used for displaying cards to the user
- After flashcard repetition time expires, push notification will be displayed to the user

User Interface Mocks

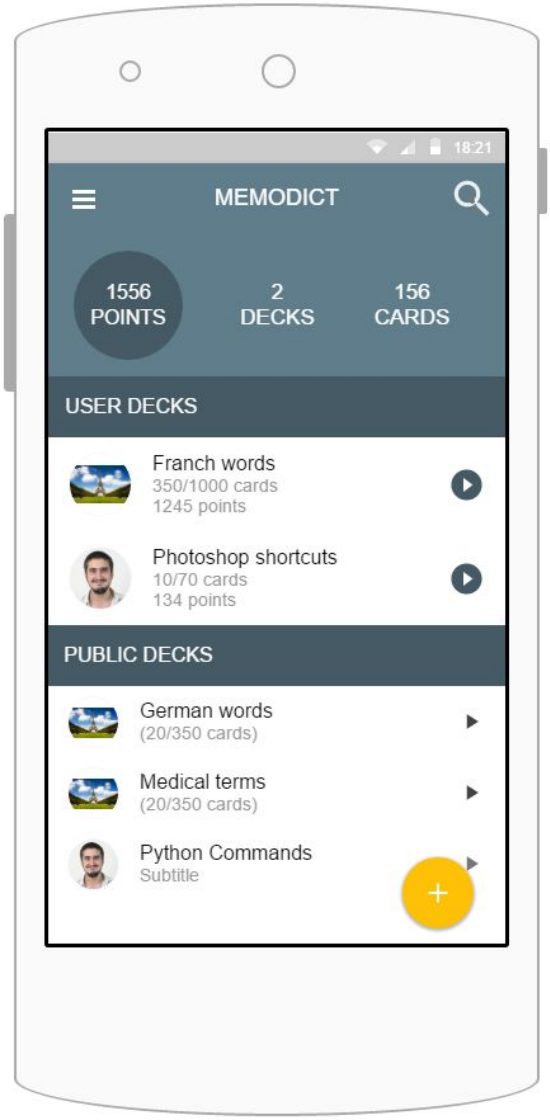
These are not final mockups. Will be refined while developing. Also logo will be used instead MEMODICT name.

Screen 1 - Login Activity



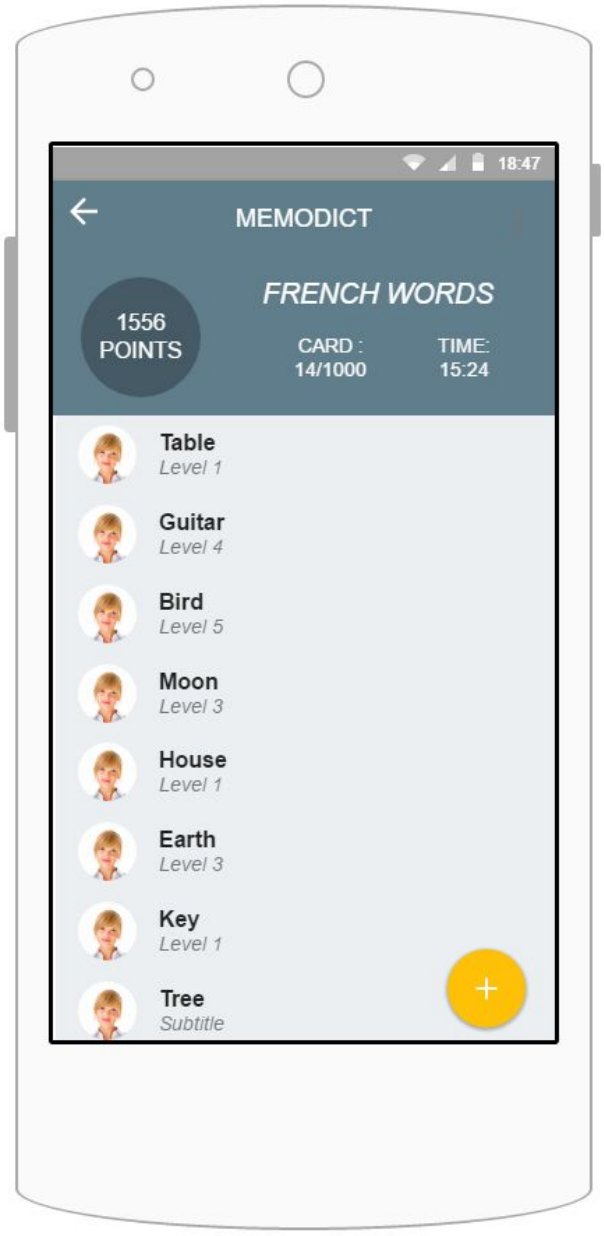
Standard login screen, user is able to create account if no account is provided

Screen 2 - Main Screen



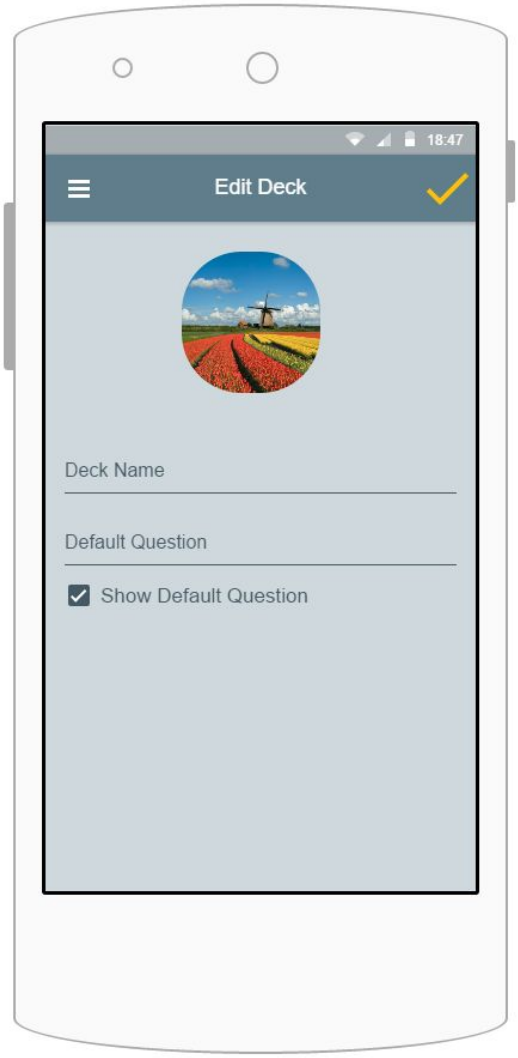
MainActivity screen, displays user statistics (points will be represented in circular progress bar). All user and some public decks are shown.

Screen 3 - Deck viewer



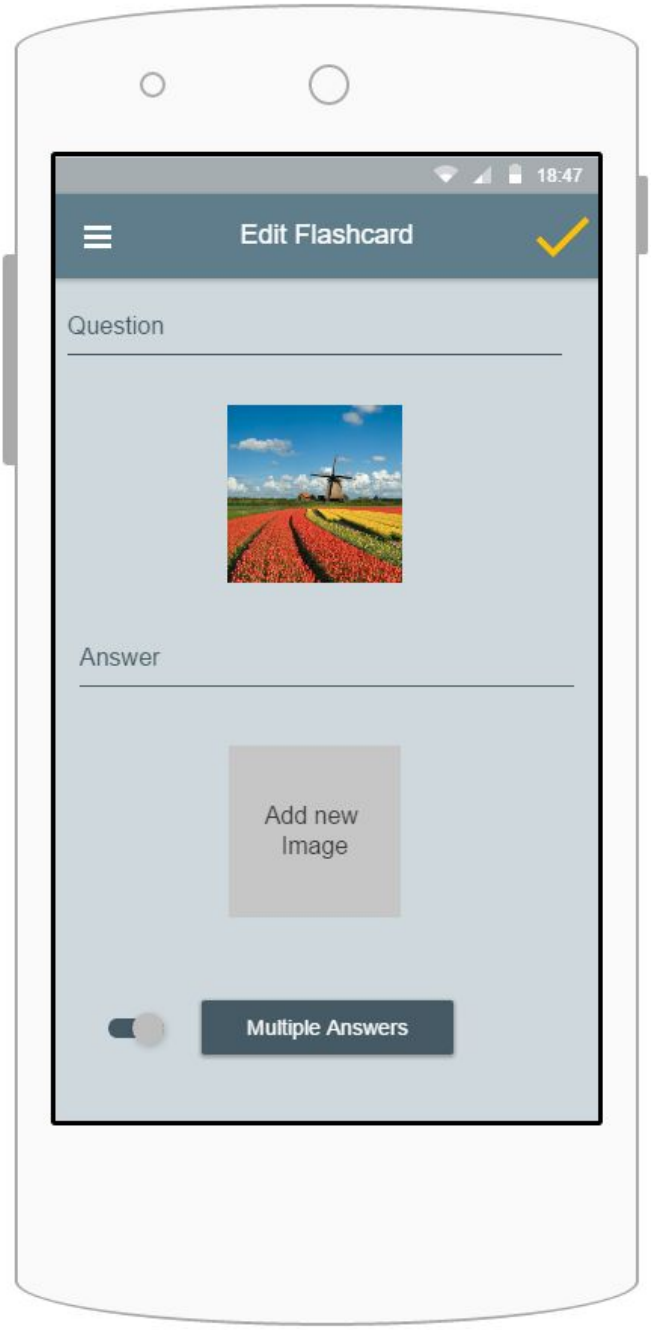
Screen for displaying list of flashcard for certain Deck, simple statistic for particular deck is shown.
User can add or edit flashcard .

Screen 4 - Deck Editor



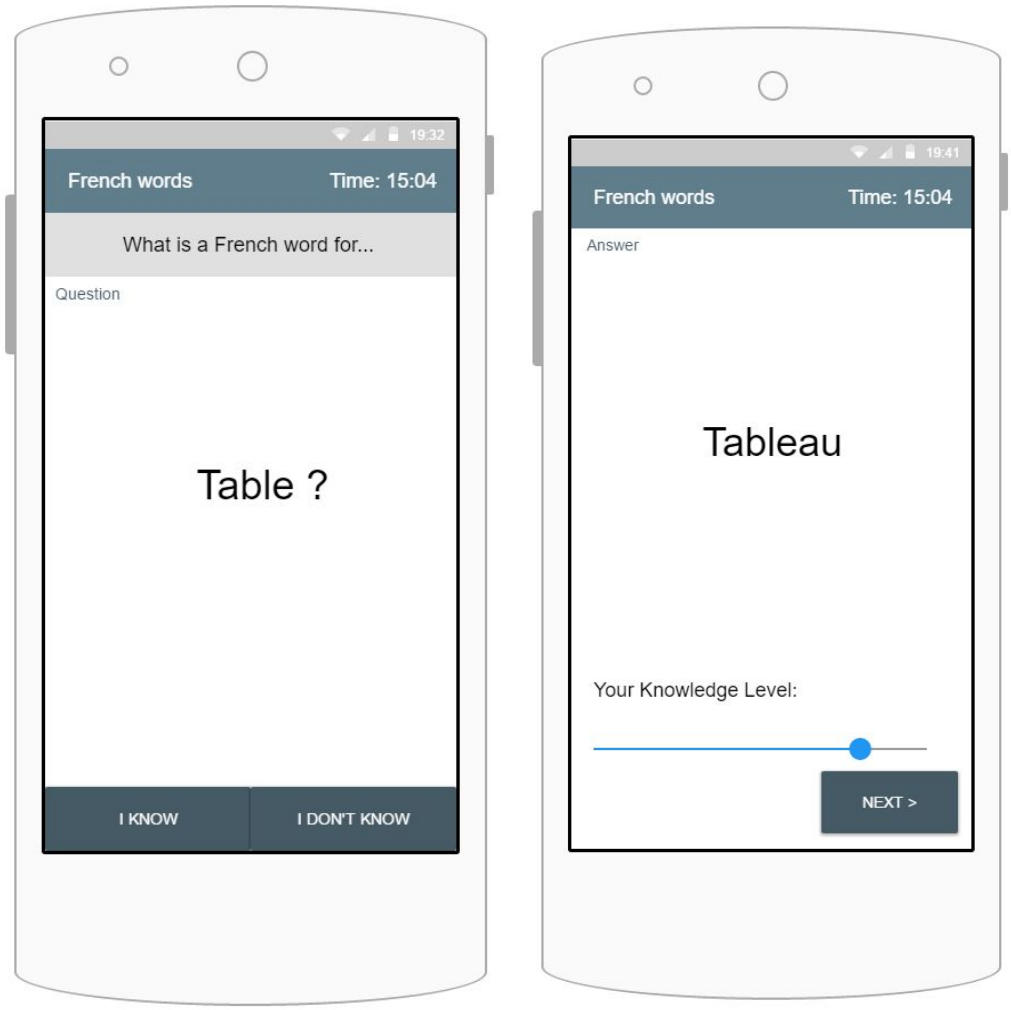
Simple deck editor. Deck image can be added. Deck default question is useful for learning words, than default question is same for all words. For example: “What is the French equivalent for English word:”

Screen 5 - Flashcard editor



Simple flashcard editor. User is able to write question and answer for the flashcard, also can add image for both question and answer. If multiple answer question is select, user can insert multiple answers, or if no answer is provided random answers will be added.

Screen 6 - Flashcard player



Flashcard player, on first part of the screen question is shown, also user is asked if he knows the answer. If user select I Don't Know, answer will be shown immediately, but if I Know is selected, multiple answers or knowledge level control will be presented. Based on answer, flashcard repetition time will be calculated with SM2 algorithm.

Widget

Simple widget which display the number of cards that should be displayed to the user

Key Considerations

How will your app handle data persistence?

Flashcards, decks and user statistics will be will be stored in Firebase server, some settings will be stored in local db or shared preferences will be used

Describe any corner cases in the UX.

If user does not provide level of knowledge after seeing the answer but close the app, default level of knowledge will be accepted.

Describe any libraries you'll be using and share your reasoning for including them.

- Picasso to handle the loading and caching of images.
- Firebase for user authentication and data persistence
- ButterKnife for views binding

Describe how you will implement Google Play Services.

I am planning to use couple of Firebase Services

- Firebase authentication: for user authentication
- Firebase realtime database for storing flashcards and user data
- Firebase analytics: for analysis of user's behavior.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Create Android project
- Setup Firebase and Picasso library.

- Create git project and configure it.

Task 2: Implement Users Login

- Build basic layout for Login Activity
- Implement Firebase Authentication library

Task 3:Working with Decks

- Create MainActivity layout, add RecyclerView and Adapter for the Decks
- Build Deck model in Firebase DB
- Create Edit/Add Deck Activity
- Implement Search Deck Mechanism
- LoaderManager will be used for displaying list of decks

Task 4: Working with FlashCards

- Create Deck Activity layout, RecyclerView and Adapter for the FlashCards
- Build FlashCard model in Firebase DB
- Create Edit/Add Flashcard Activity
- LoaderManager will be used for displaying list of flashcards

Task 5: SuperMemo 2 Algorithm

- Implement SuperMemo 2 Algorithm for calculation of card repetition

Task 6: FlashCards Player

- Create Player Activity for displaying flashcards and accepting user inputs
- After user sees the answer ask for level of knowledge for the given flashcard.

Task 7: Implementing the Widget

- Create Widget which displays number of cards that should be visible to the user.

Task 8: Implement Push server and Firebase Message Receiver service

- Implement simple push server which stores time of next repetition, user id and message id, this server will send FCM message to the users device
- FCM message will be received with FirebaseMessagingService, which will display notification to the user and update the widget.
- After each user's answer from the FlashCard Player screen Push server will be updated through *IntentService* which will sent data to the user.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"