AnkiDroid's GSoC 2024 Ideas List

>> Direct Link to Ideas List <<

AnkiDroid is a free Android flashcard app which makes remembering things easy[1] via use of spaced repetition. Because it's a lot more efficient than traditional study methods, you can either greatly decrease your time spent studying, or greatly increase the amount you learn.
Anyone who needs to remember things in their daily life can benefit from AnkiDroid. Since it is content-agnostic and supports images, audio, videos and scientific markup (via LaTeX), the possibilities are endless.
For example:

- Learning a language
- Studying for medical and law exams
- Memorizing people's names and faces
- Brushing up on geography
- Mastering long poems
- Even practicing guitar chords!

We are proud to have 4.8 stars on the Play Store, and an installed audience of 2.7 million (as of February 2024). We have an active community of developers, with many new contributors joining every year and even new localizations. Many of the contributors discovered Android programming with AnkiDroid and we are always happy to help people contribute, independently of Google Summer of Code.

Community

You can reach us on the Anki Discord (#ankidroid-gsoc & #ankidroid-dev) or our mailing list.

We primarily use Discord for real time communication, and GitHub issues for context on issues which would be useful for future contributors.
Our GSoC Mentors/Org Admins are as follows:

● Arthur Milchior	https://github.com/Arthur-Milchior
● David Allison	https://github.com/david-allison
● Manikant	https://github.com/krmanik
● Shridhar Goel	https://github.com/ShridharGoel

Mike Hardy (https://github.com/mikehardy) is a maintainer and Org Admin

Our GitHub repository is: https://github.com/ankidroid/Anki-Android/ and our Wiki contains a lot of useful information, including our Getting Started page.

[1] The first paragraph comes from anki's website.

Ideas List

In this document, you'll find a list of projects that AnkiDroid offers for GSoC in 2024.

If you want to propose something which is not in the ideas list, kindly discuss it with a mentor. Please see [Your Project Here] for details.

Table of Contents

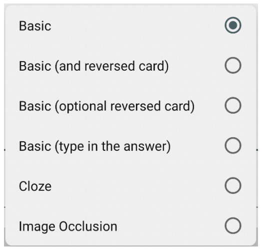
Note Editor: Note Type Preview (175 hours)

Problem

AnkiDroid is a flashcard app with a complex HTML/field-based templating engine. We currently have difficulties explaining a number of concepts to new users, both while onboarding, and for intermediate users:

- The unexpected fact that the user adds 'notes' to the app, not 'cards'
- One note can generate multiple cards
- Various 'Note Types' have unique properties
- A user can create or download additional note types

Currently, the user is provided with a text-based selection screen:



[✓ Basic]
Basic (and reversed card)
Basic (optional reversed card)
Basic (type in the answer)
Cloze
Image Occlusion

Expected Outcomes

In order to resolve the above issues, we want to modify this screen to provide a preview of each note type available in the system, showing

- The number of cards which will be produced when the note type is used
- A visual preview of how each of the cards will look
    - Each card has a separate HTML template, so the designs may vary
- Taking into account some special features:
    - A note type may request that the user types in the answer
    - Cloze deletions: 1 input produces 1…n cards
    - Image occlusion: 1 input
- The ability to open our Card Template Editor

The screen should allow a user to open up our Note Type Management screen and our manual. We should aim for the screen to prefer graphical elements over text

Language:

EITHER:

- Kotlin & XML
    - If the screen is Android-specific
- Svelte (Typescript + HTML)
    - If the screen is to be integrated into all Anki clients

Difficulty: Medium

Mentor(s):

- David Allison

Note Editor: Instant Add Note Editor (175 hours)

Problem

We want to make it as painless as possible to add new notes to AnkiDroid.

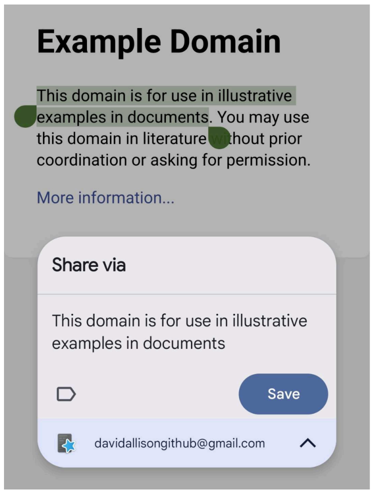
It currently takes 4 or 5 taps from Chrome to add a note with a cloze deletion

By producing an ‘Instant Add’ editor, we can allow a user to create a cloze deletion with a tap, compared to a long press, followed by a tap. This is a significant time-saver when a user has many notes to add, making the app more pleasant to use

Expected Outcomes

A user is able to select text in Android applications and select “Anki Card”. This will open an ‘instant add’ editor which allows the user to tap words or phrases in the text and turn them into cloze deletions.

This will be similar to the ‘share to’ dialog which Google Keep provides:



Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

More information...

Share via

This domain is for use in illustrative examples in documents

Google Keep’s “Share to” Dialog

This dialog should load quickly, and be minimal in design
The dialog should allow loading of AnkiDroid’s heavyweight ‘add note’ screen

Language: Kotlin, XML

Difficulty: Medium

Mentor(s):

- David Allison

- Shridhar Goel
- Manikant

Public Comments on Proposals

The primary justification for the feature is to reduce the amount of effort to create a note. Fogg Behavior Model - Read the “Elements of Simplicity (Ability)” section of the linked paper.

Aim to frame the motivation for the feature proposal in terms of reducing some of the 6 factors of “ability”

—-

I feel the major point missing is what the UI for 'tapping a word' would look like

That's what's going to take the most time here, both in the design, and in the implementation phase

You need to consider how to make it obvious that a word is clickable

You need to consider how to display 'c1', 'c2' etc...

Is adding hints going to be in, or out of scope? `{{c1::London::City}}`

How is [SAME_NUMBER, INCREMENT_NUMBER] displayed in the UI?

Tablet & Chromebook UI (175/350 hours)

Problem

AnkiDroid was initially designed for Android mobile phones. Over the years, Android has come to tablets and Chromebooks, but our UI has continued to be designed around the mobile phone.

We currently have ~10% of our users on Tablets or Chromebooks, and we want to improve their user experience using the app, both with the aim to improve the user experience for our existing users, and increasing the number of users who can effectively use our app on larger devices

Expected Outcomes

This primarily depends on your proposal. At the very least, we would like to modify the main ‘Deck Picker’ screen to have one menu, rather than two.

Any screen in the app is open for your suggestions. In particular, adding a live preview/editor to the Card Browser and Card Template Editor

Language: Kotlin, XML

Difficulty: Medium

Mentor(s):

- David Allison
- Arthur Milchior

Additional Widgets (175/350 hours)

Problem

Widgets were introduced to AnkiDroid in 2010. These provide significant benefit to our power users, but we have not taken advantage of improvements to either the app or the OS

Expected Outcomes

Android 12 Widget-based functionality is evaluated and integrated with the widgets when appropriate

At least one additional widget is created or modified, based off the ‘deck picker’: showing the total number of due cards, broken down into categories (new/learn/review)

We expect the GSoC proposal to include additional widgets which will be useful to our users

Language: Kotlin, XML, UI & UX

Difficulty: Medium

Mentor(s):

- David Allison
- Shridhar Goel

JavaScript Addons Support (175 hours)

Problem

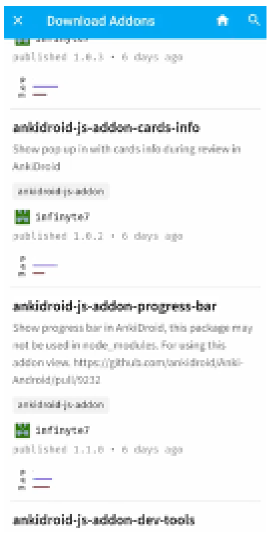
AnkiDroid uses Android webview for reviewing Cards. The default templates lead users to more complex card templates. The deck developer uploads their decks to Ankiweb and users can use the decks templates. But to use the same templates and JavaScript across all card templates it needs to modify each card template manually in AnkiDroid. Let's say user want a custom progress bar which can be implemented using HTML/CS/JS in card templates but it is not available to all decks because each deck has their own card templates. Also, we intended to extend note editor functionality by implementing addons for note editor also. Also, in future we may use a ported Anki desktop note editor, so we must design the addons in such a way that we easily integrate into Anki desktop pages.

Expected Outcomes

To solve this problem, we need to implement JavaScript addons support in AnkiDroid. The addons should have the following characteristics.

- A menu in sidebar added to have access to all addons, option to download, install, view and uninstall.
- All addons must be published by addons developer to npm, and use npm to show list of addons in AnkiDroid
- Addons should show which type of access it needs, which can be checked using metadata or manifest.json and allow that request access in JS addons, which can be done by wrapping it in a method.
  Window.VMsomeaddon = (anki) { users addons code here }
  If a user wants to access anki.getCardId and if it does not exist in manifest.json then the anki parameters of executed code should prevent it.
- Two modes of addons – Reviewer and Note editor.
- Should be run in their own context so it should not conflict with other addons.
- Options to blacklist malicious addons

Following mock up images.



published 1.0.3 • 6 days ago

ankidroid-js-addon-cards-info

Show pop up in with cards info during review in AnkiDroid

ankidroid-js-addon

infinyte7

published 1.0.2 • 6 days ago

ankidroid-js-addon-progress-bar

Show progress bar in AnkiDroid, this package may not be used in node_modules. For using this addon view. https://github.com/ankidroid/Anki-Android/pull/5232

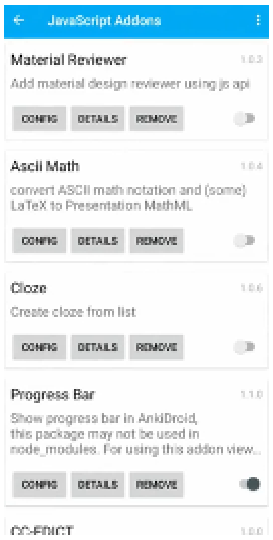
ankidroid-js-addon

infinyte7

published 1.1.0 • 6 days ago

ankidroid-js-addon-dev-tools

Fig 1.



Material Reviewer 1.0.3

Add material design reviewer using js api

Ascii Math 1.0.4

convert ASCII math notation and (some) LaTeX to Presentation MathML

Cloze 1.0.6

Create cloze from list

Progress Bar 1.1.0

Show progress bar in AnkiDroid, this package may not be used in node_modules. For using this addon view...

CC-EDICT 1.0.0

Fig 2.

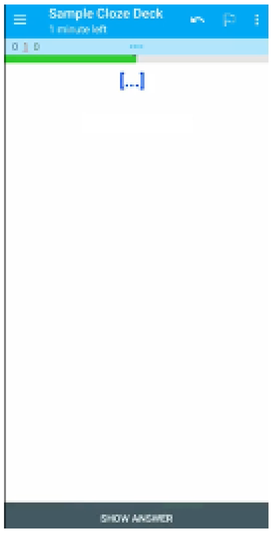
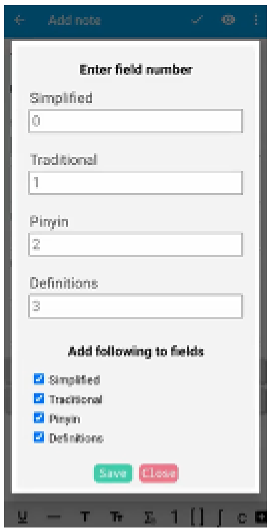


Fig 3.



Enter field number

Simplified: 0

Traditional: 1

Pinyin: 2

Definitions: 3

Add following to fields

- [x] Simplified
- [x] Traditional
- [x] Pinyin
- [x] Definitions

Fig 4.

Fig 1. Options for user to explore js addons published under ankidroid-js-addons keyword
Fig 2. Options to user for view config, details and enable/disable and remove addons
Fig 3. Progress JS addons available to all decks
Fig 4. Note editor addons which add note to AnkiDroid, the addon uses HTML and rendered in AnkiDroid Note editor and shown when a button in toolbar clicked

Language

Kotlin & XML: For exposing only required Note editor API and reuse reviewer JS API

HTML/JavaScript: For designing ankidroid js addons and usages in AnkiDroid

Difficulty: Medium

Mentors

- Manikant
- Arthur Milchior

[Your Project Here] (175/350 hours)

We welcome you to propose your own projects, but **please get in touch with [our Mentors] before writing your project proposal.**

When you contact our mentors, you should provide a brief project description for consideration (similar to the high-level descriptions in this document).

Your project description should contain:

- Why the project matters to AnkiDroid
- What you expect to do by the end of the coding period
- Expected difficulty level
- Expected time (175, 350 hours, or maybe less than that if it is combined with another project)
    - Please note: this is NOT eligible for a 90 hour project
- Technologies: We use mostly Kotlin, but some JS/CSS/Rust/Svelte may be involved for some projects

We will not accept a project proposal which is not on the ideas list unless you have discussed it and found a mentor.