Ranji

Documentation

Leif T. Sundkvist - 21 August 2017

Introduction

Target audience

The intended target audience for this application are people with an interest of and will to learn the Japanese language, more specifically reading and memorising kanji. The general theme of the application is light colours with a strong touch of cuteness, a theme often associated with and seen among younger people with an interest in Japan.

App description

Are you planning a trip to Japan with a sense of unease that you won't be able to read all those store signs or restaurant menus? Learning and memorising Kanji can be tedious at times. Ranji is an application for you who wishes to study Kanji easily and more effectively whilst still having fun doing it. Ranji offers you the following features:

- Search for any kanji character and view information about it.
- Create and manage *study lists*, consisting of Kanji that you want to memorise or keep track of.
- Play a mini-game for memorising Kanji within study lists. You'll remember them all in no time!

Good luck with your studying, and do your best!

Grade estimation

I'm aiming for a G for this assignment. The reasons for this was because I underestimated my own project and thought it would take less time to implement all the features I wanted. I would have liked to add several features such as a adding a view for displaying the strokes comprising a kanji or adding new kanji from the kanji list activity.

Ethics and safety

This application really has no concerns for ethics or safety as it is not related to any social networking nor data-gathering whatsoever, is offline and only has one aspect regarding storing data which is the images that the user chooses to attach to each kanji.

However, the camera must not be used outside of the application and only when the user actually intends to take a picture. Furthermore, it is up to the user not to take any pictures that would infringe on other peoples' or organisations' rights

Usage

Main menu

- The user may search for kanji by specifying either a word in English, such as *Time*, or a character, such as 時.
- The user may engage in a quick play, but this feature has not been implemented due to lack of time; see the Kanji List section for this functionality (Could be added in the future, not essential).
- The user may click the "Study lists" button to see created study lists.
- By clicking the "About" button, some information about the application will be displayed. (Could be added in the future, not essential).

Figure 1. The main menu.

Managing study lists

Within the study list view, the user may
- Create or remove study lists
- Click a study list to manage its contents

Within a study list, the user may
- Create or remove kanji from the study list
- Click a kanji to view its properties

Viewing Kanji

When the user selects a kanji within a study list, he/she may
- View more information about the current kanji, such as the onyomi, kunyomi, grade and SKIP code.
- Add or change a picture related to the kanji. Associating kanji characters with pictures or real-life objects makes them easier to memorise.
- Add the kanji to a study list.

Figure 2. Viewing more details about a kanji.

Playing the mini-game

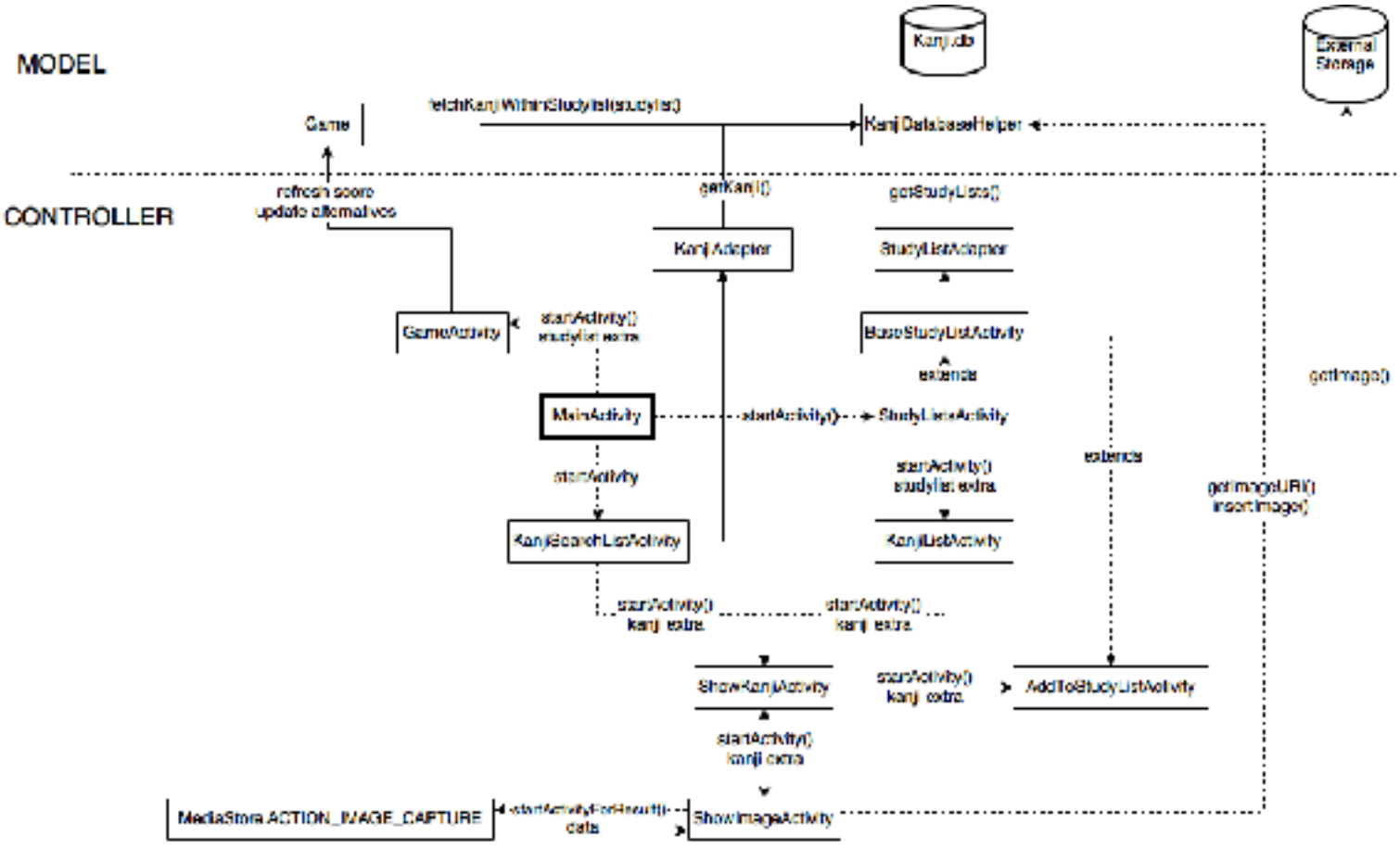
The user may play a mini-game using all the kanji within a study list. The rules of this mini-game are as follows:

- Each round, the user must guess what the kanji displayed within the box means in English.
- For each correct answer, the points increases. If the user chooses to skip the question, another one is generated.
- Because there are four alternative buttons, there should be at least four kanji available in the study list, or there will be multiples of the same kanji shown in the buttons. I chose to go with at least five kanjis, because that's a more magic number.

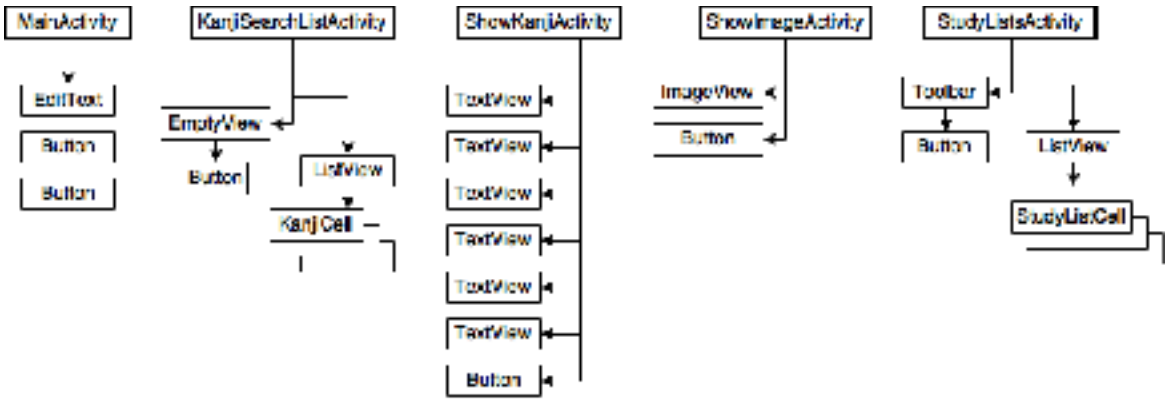
Figure 3. Playing the mini-game.

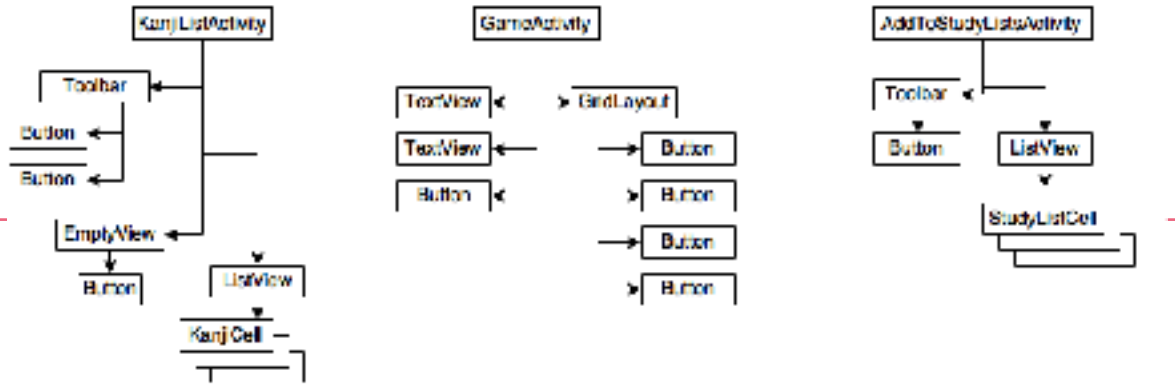
Application Structure (updated)

I chose to use two images showing the model-controller and the view aspects separately, in order to conserve space and reduce a lot of arrows pointing to the left and right.



View





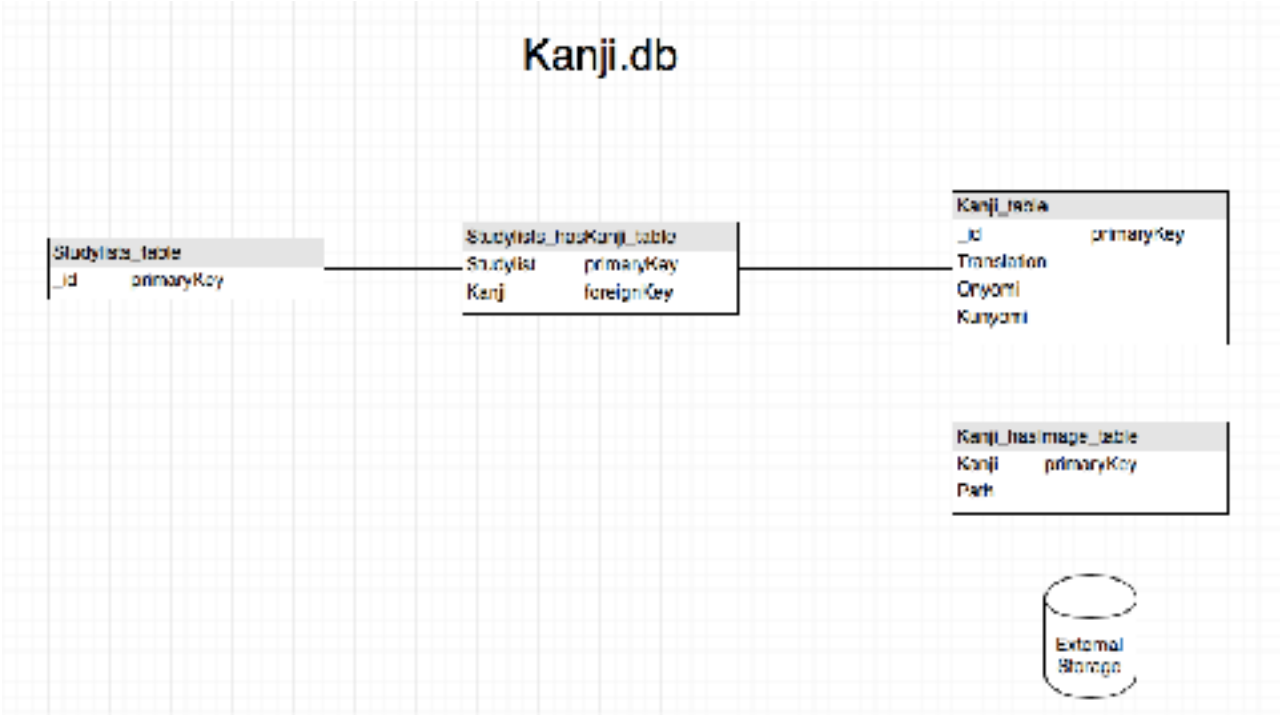
Database structure

The following image shows the structure of the database used within the application.

A Kanji item within the Kanji_table consists of a character (also an id), a translation in English and a kunyomi and onyomi pronounciation.

A studylist within the Studylists_table consists of a name (also an id), and may have several kanjis as per the relation Studylists_hasKanji_table.

A Kanji may have one image, and the External Storage path to that image is stored in the Kanji_hasImage_table.



Notes

All uses of images and *Pusheen* icons within the applications will be replaced with my own icons in further releases of this application, and if it would be released on Google play.

Resubmission notes

- Added information about the structure of the application, aside from the structure of the database as in the previous hand-in.
- Fixed the compilation error where a method needed an extra argument.
- All views now support rotation.
- Added support within the views so that data is properly restored should the app be destroyed upon rotation or by other means.
  - The dialogs was the most difficult part to fix, because they wouldn't persist on destroy. I fixed them by using custom DialogFragments which persisted upon orientation change.
- Game
  - Fixed the error where the user got a correct answer repeatedly by just choosing the top-left alternative within the game activity. (Caused by the alternatives not being shuffled, so that the first alternative button was always the correct one.)
- Fixed the errors that caused stack trace. This error was caused by an incorrect SQLite insertion where I attempted to insert five columns into a table consisting of only two. This was caused by a variable *cv* that should instead have been named *cv2*.
- Additional fixes
  - Added functionality so that only either wester letters or japanese Kanji may be entered for search. The user may thus not search for special symbols or characters, etc. For this I used a regex matching any type of letter from any language.