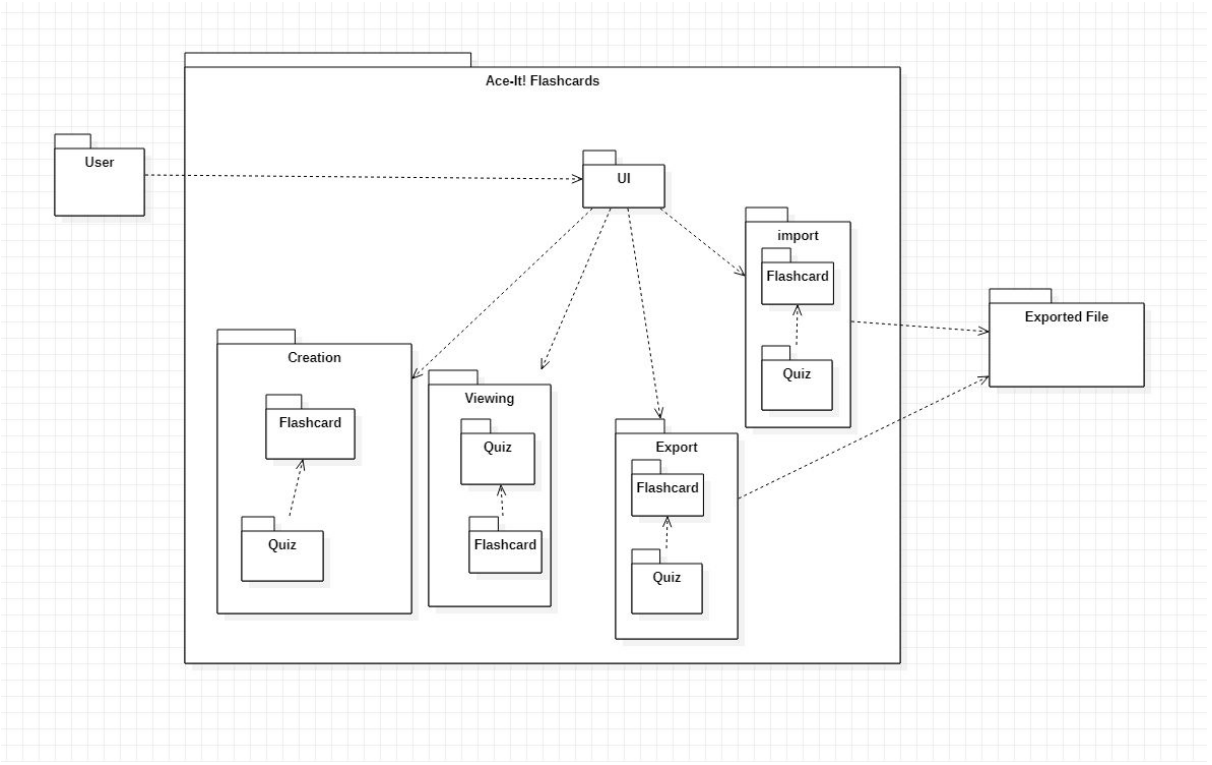
Blackwell, Dakota 300083663
Dhillon, Navdeep 300161244
Paterson, Alex 300118506
Gilbert, Grant 300153121

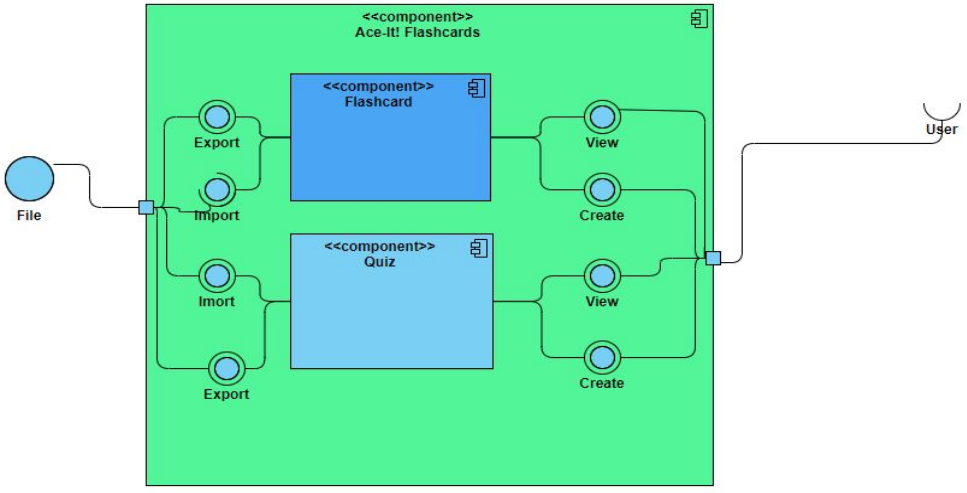
Group 2 Deliverable 3

System Architectural Model

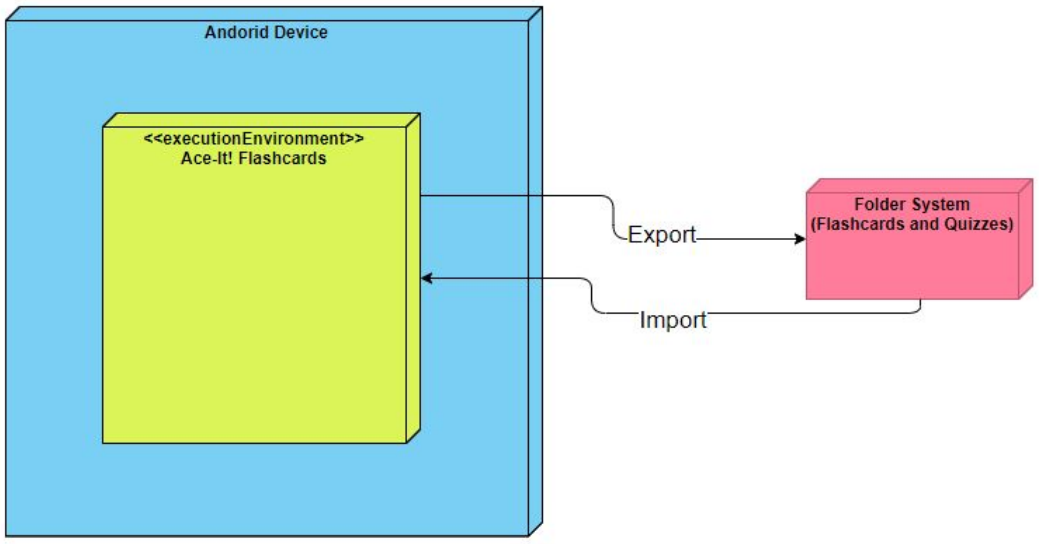
Package Diagram



Component Diagram



Deployment Diagram



How the system supports the functional and quality attributes

Functional Properties

We were able to let user input enter their own text and save the text into files. Our app lets the user create flashcards by clicking the create flashcard button.

Upon opening a flashcard the user is presented with the question and may tap the card to reveal the answer. If a user is stuck on a question they have an option to reveal a hint that can guide them to an answer instead of revealing the answer outright.

Quizzes may be created from a list of previously created flashcards and given a name. Opening a quiz presents the user with the corresponding list of flashcards.

Tags can be created and then added to flashcards as a means of easily identifying cards at a glance. Multiple tags may be added to one card.

Quality Attributes

Avoiding bloat and complicated processing helps to keep the app lightweight and in return keeps resource usage down and increases battery life.

To achieve an acceptable response time and fluid app usage, we decided to go with a minimalistic approach. Keeping views simple increases performance when loading large numbers of flashcards.

To facilitate reusability the team incorporated standard MD5 hashing of the object data when saving so the files could be easily dehashed if needed. The app works on any android phone running KitKat 4.4 or later, which will support devices all the way back from 2013.

Status Report

During the development of the Ace It! Flashcards app, our team was able to implement most of the functionality we had originally planned. During the planning phase many ideas were proposed for various different apps, but keeping in mind the time restrictions and the varying experience levels within the team we were able to narrow down our focus to an app that provided diverse utility with only a few core components.

The Ace It! Flashcards app provides the user with the functionality to create flashcards with custom questions and answers. Once a flashcard is created it is then exported as a serialized file to the app context folder and later imported when loading the app so users can save flashcards. During the creation of a flashcard users can include tags to organize flashcards as well as hints for each question. The option to create flashcards with selectable multiple choice answers was something that was dropped from the final build of the project. After some discussion the team had decided that multiple choice answers were not a crucial aspect of the flashcard experience and elected to drop this functionality to afford more time on other parts of development.

Quizzes can be created by selecting specific flashcards from a list of all saved flashcards. Once a quiz is created, it is shown in a list of all quizzes with their names. Selecting a quiz will show the corresponding flashcards.

Tags can be created and then added to flashcards during creation or editing for further organization. They show at the bottom of each flashcard in the list. Upon tapping a tag in the tag list, it will display all cards associated with that tag.

The other feature we decided to abandon was the importing and exporting of flashcards. This was originally intended as a way for users to share cards with one another, like a teacher sharing cards with students. After testing this we discovered that our implementation was too cumbersome to be of use; navigation a file structure on a mobile device doesn't feel natural. We did, however, discover that these files could be used as a way of maintaining persistence of flashcards, quizzes, and tags between uses.

The app has been submitted to the Google Play Store and is pending review.

Demo Summary

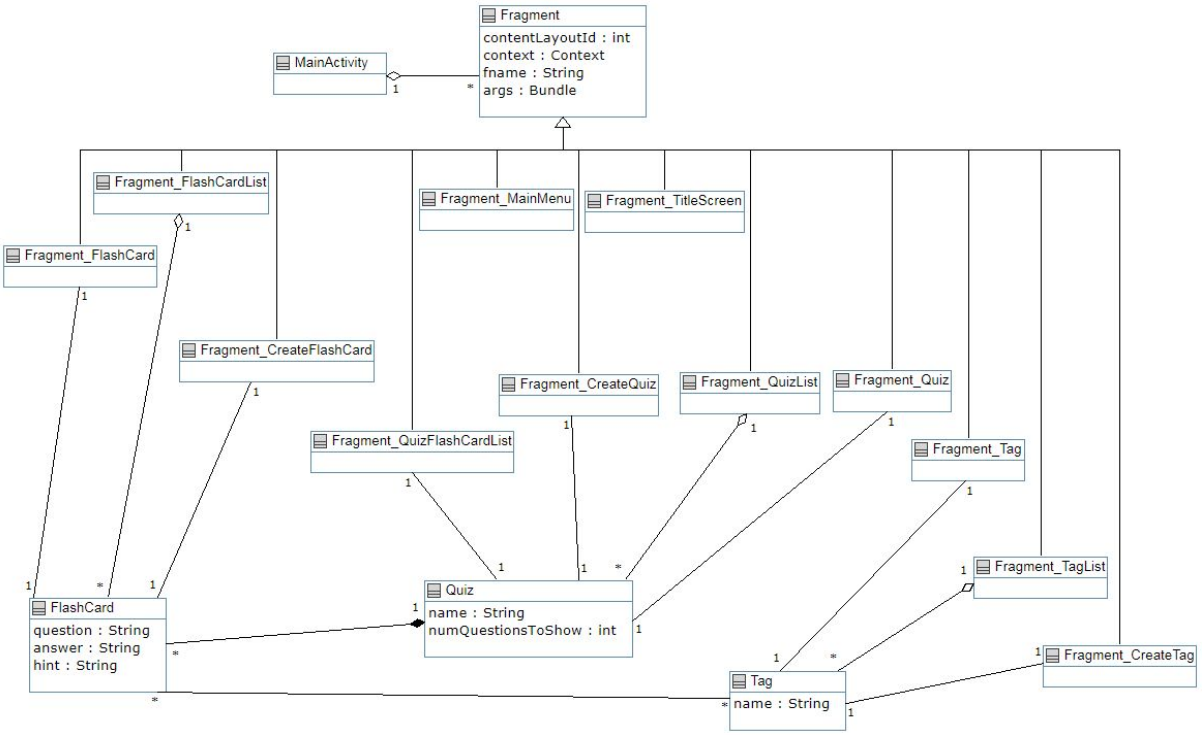
We'll showcase creation of a flashcard, creation of a quiz from existing flashcards, editing and deletion of flashcards and quizzes, usage of hints and tags, self-testing with flashcards and quizzes, and file import/export functionality.

Link to Youtube video: https://www.youtube.com/watch?v=8legeo0CzNw

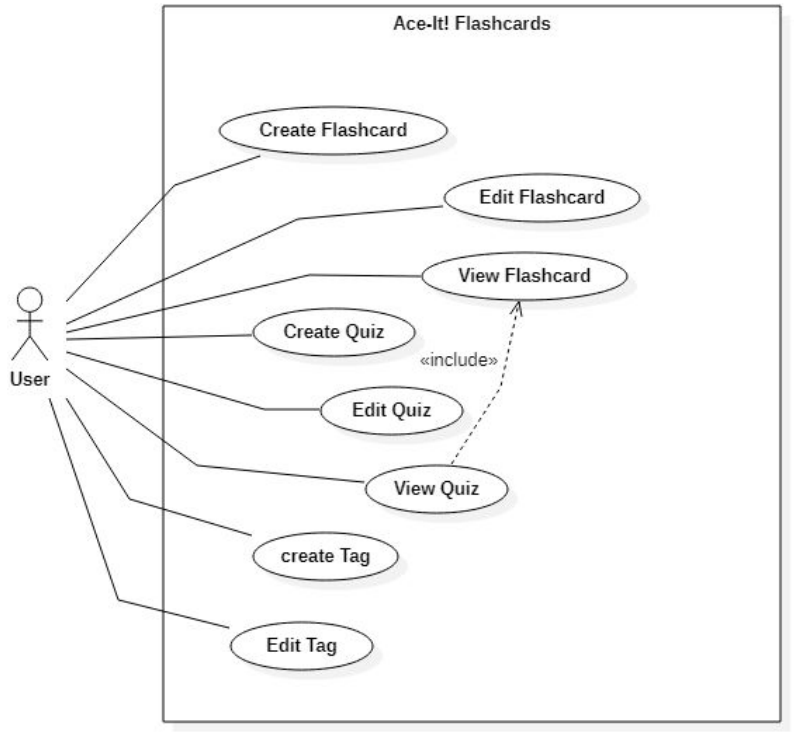
Member contributions:

Dakota: UI Design and Implementation, Fragment/XML/Navigation editing, GitLab management, Testing, Various bug fixes
Grant: Quiz class, Class Diagram, Use Case Descriptions, Javadoc, System architectural discussion, Import/Export Methods, Updated Class diagram, Testing
Alex: UI, Import/Export methods, System architectural models, Testing, Quiz implementation, youtube video, Final report,
Navdeep: Mock design, Javadoc(flashcard), contributed to Use Case Descriptions,FlashCard development, helped layout design, editable textbox & save, hint button, System architectural discussion, support for functional and quality attributes, Testing

Updated Class Diagram



Use case diagram



Participation Journal:

Alex: System architectural models, Updated Class diagram, youtube video, status report
Grant: System architectural discussion, Import/Export Methods, Updated Class diagram
Dakota: UI (Fragments / XML / Navigation work), Android Integration, Git Support, Testing and bug fixes, APK signing.
Navdeep: System architectural discussion, support for functional and quality attributes, UI, Flashcard