We have a Model (AskNoteModel)

* A model has decks. A deck has flashcards and a UNIQUE title. A flashcard has two sides, each containing text.
* A model has quizzes. A quiz has one deck, a user who is a tester, and a user who is a testee. Either the tester or the testee must be the current user.
* A model has a list of friends
* A user
  + A user has a username and a user ID
* Current page type
* Stack of pages
* List of notifications

Functions

Get decks

Get deck (takes in title)

Get current page

Function to go to last page

Should pop the top of the page stack and make it the current page

Get friends

Get notifications

Get quizzes

We have a view (AskNoteView)

NOTE: we must be careful to get *deep copies* of decks and flashcards so as to

not “save” things before the user clicks save (or somewhere keep a copy to revert back to if the save doesn’t happen)

* View has a header panel
  + Header panel has a home button
  + Header panel has a back button
  + Header panel has a notification icon (button?)
  + Header panel has a page title
* View has an App panel
  + An app panel can be one of these page types:
    - Home page
    - Decks
      * List of decks
    - Edit deck
      * Takes in the deck
    - quiz self
      * a deck
    - Friends list
      * List of friends
    - Quiz (first page when you click quiz)
    - Card editor
      * A flashcard
    - Choose deck
      * List of decks
    - Select quiz
      * List of quizzes
      * List of friends
    - Tester page
      * Flashcard
      * Testee’s response (a string)
    - Testee page
      * flashcard
    - Login
    - Register
    - Notifications
      * List of notifications
  + each page type shall have its own corresponding java class that extends JPanel
* A menu (the file – edit – view – account – help thing at the top of the screen)