PSEUDOCODE

1. Login Users
   1. Initialize userList
   2. Constructor function LoginUsers(userList)
      1. Set userList to userList
   3. function addUser(name, password)
      1. Create a SetsList
      2. Create a new user(name, password, SetsList)
      3. Add user to userList
      4. Return the user
   4. function FindIndexOfUser(username, password)
      1. Set index to -1
      2. Repeat for every user in userList
         1. If the user at this index’s name and username are equal and their password match, set the index to this user
         2. Return the index of the user
   5. function getUserList()
      1. Return userList
   6. function getUser(x)
      1. Return the user at index
2. Login
   1. Initialize LoginUsers
   2. Constructor function Login(LoginUsers loginUsers)
   3. function LoginButtonActionPerformed()
      1. Set Username to the text inside UsernameField
      2. Set Password to the text inside PasswordField
      3. Create new StringBuilder for warnings
      4. If Username is empty
         1. Append warning to display "Username must not be empty."
      5. If Password is empty
         1. Append warning to display “Password must not be empty.”
      6. If the warnings exceed 0
         1. Open input warning pane
      7. Else
         1. Set the current user to FindIndexofUser(Username, Password)
         2. Initialize boolean for signIn and set it to currentUser is not equal to -1
         3. If signed in
            1. Create a new object for the home screen for the current user
            2. setLocationRelative to home screen
            3. Dispose the login page
            4. Set home screen visibility to true
         4. Else
            1. Open incorrect username and password pane
   4. function SignUpButtonActionPerformed()
      1. Create new object for sign up page to add an account
         1. Dispose the login page
         2. Set sign up page visibility to true
3. Sign up
   1. Initialize loginUsers
   2. Constructor function(boolean modal, LoginUsers loginUsers)
      1. Inherit parent frame and modal
      2. Set loginUsers to loginUsers
      3. SetLocationRelative to parent Frame
      4. function CreateAccountButtonActionPerformed()
      5. Set Username to the text inside UsernameField
      6. Set Password to the text inside PasswordField
      7. Set ConfirmPassword to the text inside ConfirmPasswordField
      8. Create new StringBuilder for warnings
         1. If Username is empty
            1. Append warning to display "Username must not be empty."
         2. If Password is empty
            1. Append warning to display “Password must not be empty.”
         3. If ConfirmPassword does not match Paasword
            1. Append warning to display "Passwords Must Match"
         4. If the warnings exceed 0
            1. Open input warning pane
         5. Else
            1. Add this inputted Username and Password to UserList
            2. Create Login Screen object
            3. SetLocationRelativeTo sign up page
            4. Dispose the sign up page
            5. Set the login page visibility to true
   3. function BackButtonActionPerformed()
      1. SetLocationRelativeTo sign up page
      2. Dispose the sign up page
      3. Set the login page visibility to true

1. Home
   1. Constructor function Home(User CurrentUser)
      1. Enhanced for loop adds all sets from CurrentUser into combobox
      2. Enables combo box, create set, and disables edit set, flash cards, matching game buttons
   2. Function SetSelectionBoxItemStateChanged()
      1. If selection box has a set selected, enables other buttons, if not they remain disabled
   3. Function EditSetsButtonActionPerformed()
      1. Finds the index of the set using search method under set class
      2. currentSet variable is declared by getting all sets from currentUser then getting the set at index
      3. Creates new edit sets page with currentSet and currentUser passed through
      4. Sets location relative to current page
      5. Disposes current page
      6. Sets edit set page true
   4. Function CreateSetsButtonActionPerformed()
      1. Makes new sets with blank title
      2. Creates new edit sets page with new set and currentUser passed through
      3. Sets location relative to current page
      4. Disposes current page
      5. Sets edit set page true
   5. function FlashCardsButtonActionPerformed()
      1. Finds the index of the set using search method under set class
      2. currentSet variable is declared by getting all sets from currentUser then getting the set at index
      3. Creates new flash cards page with currentSet and currentUser passed through
      4. Sets location relative to current page
      5. Disposes current page
      6. Sets flashcard page true
2. Edit Sets
   1. Initialize global variables for CardTable
   2. Constructor function(boolean modal, Sets currentSets, User CurrentUser)
      1. Set title to the set name
      2. Declare CardTable as global reference
      3. Initialize two dimensional list
      4. For every card in CardArray length
         1. Add card from CardArray to CardTable by row
   3. Function CreateSetButtonActionPerformed()
      1. If title field is empty
         1. Have a warning to display "Title cannot be blank."
      2. Else
         1. Delete current set and erases cardList
         2. For sets in the currentUser’s sets
            1. If the sets name is equal to the set’s title

Set x = the index of the set

For every card in CardArray length

Add card from set to global cardList

Remove odd set to replace with updated set

* 1. Set x = 0
  2. Set y =0
  3. For every row in CardTable
     1. New card is made and added to cardList
     2. Add 1 to x
     3. Set y = 0
  4. Creates a new set to replace the deleted one under same name with updated cardList
  5. Dispose Edit Sets page
  6. Set Home Screen visibility to true
  7. Function BackButtonActionPerformed()
     1. Dispose the Edit Sets page
     2. Set Home Screen visibility to true
  8. Function AddCardButtonActionPerformed()
     1. Add row to CardTable with terms and definitions from TermField and DefinitionField
     2. Set term field to blank
     3. Set definition field to blank
  9. Function RemoveCardButtonActionPerformed()
     1. Set i = selected row from CardTable
     2. Remove the row i

1. Sets
   1. Initialize global variables name, CardArray and StudyLaterArray
   2. Constructor function(String name, String[][] cardArray)
      1. Set set name to given name
      2. Set cardarray to passed card array
   3. Function getCardArray()
      1. Returns cardarray
   4. Function getStudyLaterArray()
      1. Return studylaterarray
   5. Function getname()
      1. Return name
   6. Function contain(String term, String def)
      1. Sets index to -1
      2. For length of studylater array
         * 1. If term and def are in a studylater columns

Set index to row they are both in

* + 1. Return index

1. Flashcard
   1. Initialize global variables
      1. currentUser
      2. Set index to 0
      3. Set showTerm to true
      4. CardArray
      5. StudylaterArray
      6. Set x to 0
      7. Value
      8. currentSet
   2. Constructor function(Sets set, User currentUser)
      1. Set currentSet to set
      2. Set currentUser to currentUser
      3. CardArray equal to set.getCardArray()
      4. StudyLaterArray equal to set.getStudyArray()
      5. Set cardbutton text to first card in CardArray
      6. Value is equal to index of card +1 multiplied by 100 then divided by size of card array for percent completion value
      7. Set progress field to Progress x/size of cardarray
      8. If card in current set is in studylater array, a star will appear, if not it’ll be blue
   3. Function scaleStar()
      1. Make image icon object with path to star image
      2. Make image object equal to imageIcon image
      3. Make Image object for scale setting its width, height to size of Star box
      4. Makes new imageicon object equal to new scaled image
      5. Sets star field as the image icon
   4. Function scaleBlue()
      1. Make image icon object with path to blank image
      2. Make image object equal to imageIcon image
      3. Make Image object for scale setting its width, height to size of Star box
      4. Makes new imageicon object equal to new scaled image
      5. Sets star field as the image icon
   5. Function BackButtonActionPreformed()
      1. Dispose the Flashcard page
      2. Set Home Screen visibility to true
   6. Function GoLeftButtonActionPerformed()
      1. If index isn’t 0
         * 1. Index minus 1
      2. Set cardbutton to cardarray at index of index
      3. If card in current set is in studylater array, a star will appear, if not it’ll be blue
      4. Value is equal to index of card +1 multiplied by 100 then divided by size of card array for percent completion value
      5. Set progress field to Progress x/size of cardarray
   7. Function GoRightButtonActionPerformed()
      1. If index isn’t size of cardarray
         * 1. Index add 1
      2. Set cardbutton to cardarray at index of index
      3. If card in current set is in studylater array, a star will appear, if not it’ll be blue
      4. Value is equal to index of card +1 multiplied by 100 then divided by size of card array for percent completion value
      5. Set progress field to Progress x/size of cardarray
   8. Function CardButtonActionPerformed()
      1. If showTerm is true
         1. Card button text is term
         2. Set showTerm false
      2. Else
         1. Cardbutton text is definition
         2. Set showTerm true
   9. ViewStudyLaterButtonActionPerformed()
      1. Make new studylater page passing current set and currentuser
   10. Dispose of current flash card page
   11. AddTermToStudyLaterButtonActionPerformed()
       1. If studylater array doesn’t contain current card
          1. Adds cards into studylater array
          2. Adds star to card
2. Study Later
   1. Initialize global variables
      1. currentUser
      2. StudyLaterArray
      3. DefaultTableModel model
      4. currentSet
   2. Constructor function StudyLater(String[][] studyLaterArray, User currentUser)
      1. Set currentUser to currentUser
      2. Set StudyLaterArray to StudyLaterArray
      3. Set StudyLaterArray to get StudyArray
      4. Declare CardTable as global reference
      5. Set rowCount to the length of StudyLaterArray
      6. For every row in rowCount
         1. Add a row fromCardList to CardTable
   3. Function RemoveCardButtonActionPerformed()
      1. Set i to selected row in StudyTable
      2. Remove i from StudyLaterArray
      3. Remove i row from StudyTable
   4. Function BackButtonActionPerformed()
      1. Creat4 new object for flashcards page with currentSet and currentUser
      2. SetLocationRelative to Study Later page
      3. Dispose Study Later page
      4. Set Flashcards page visibility to true
3. User
   1. Initialize global variables
      1. Sets
      2. Name
      3. Password
      4. Studylater
   2. Constructor User(String name, String password, LinkedList<Sets> Sets)
      1. Set name to name
      2. Set password to password
      3. Set sets to set
   3. Function removeSets(Sets set)
      1. Removes set from list of sets
   4. Function removeSets(int index)
      1. Removes set from set list given an index
   5. Function addSets(String name)
      1. Makes new card array
      2. Makes new set with name and card array
      3. Adds set to set list
      4. Return set
   6. Function addSets(String name, String[][] cardArray)
      1. Makes set with name and card array
      2. Adds set to set list
      3. Return set
   7. Function getSets()
      1. Returns sets
   8. Function FindIndexofSet(String name)
      1. Set index to -1
      2. For size of set list
         1. If set name equals name
            1. Set index to position in loop
   9. Function getName()
      1. Return name
   10. Function getPassword()
       1. Return password