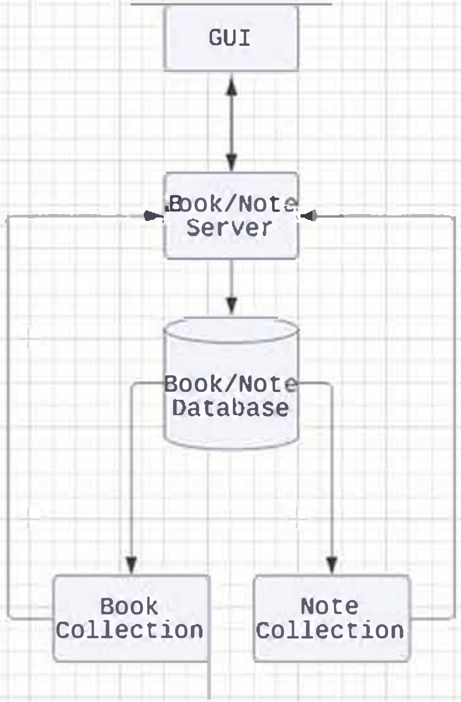
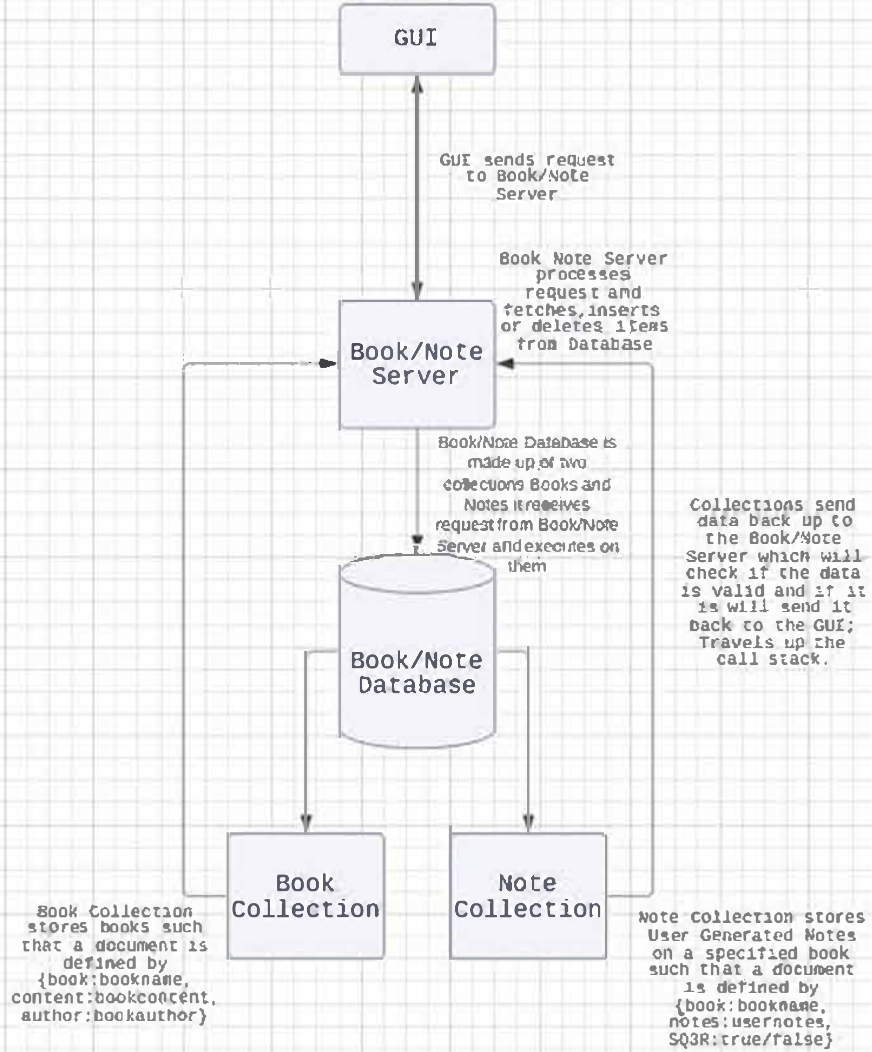
Active Reading Assistant Architecture (Static Model)



Active Reading Assistant Behvioural Model



GUI sends request to Book/Note Server

Book Note Server processes request and fetches,inserts or deletes items from Database

Book/Note Database is made up of two collections Books and Notes it receives request from Book/Note Server and executes on them

Collections send data back up to the Book/Note Server which will check if the data is valid and if it is will send it back to the GUI; Travels up the call stack.

Book Collection stores books such that a document is defined by {book:bookname, content:bookcontent, author:bookauthor}

Note Collection stores User Generated Notes on a specified book such that a document is defined by {book:bookname, notes:usernotes, SQ3R:true/false}

GUI

Buttons == Functions + User Input

Buttons:
Finish -> Save to DB-> go to Book Selection
Save -> Save to DB
Delete -> Delete from DB after warning
Hide -> Hides Notes Window
SQ3R ->Toggles SQ3R text

Notes (Window) == UI/User Input

Notes (Window):
Resizeable Scrollable Text
Scrollable line numbers

Content(Window) == Read Only, Book Text

Content (Window):
Resizeable Scrollable Text -> Content of Book

Book(Object) == Organised Content/Data

Book:
Content == Text
Author == Text
Title == Text

Notes(Note Object) == Organised User Generated Content/Data
Notes (Note Object):
Content == Text
Book Title == Text
SQ3R =True/False

NOTE/BOOK SERVER

```
Main Functions:

Insert_Notes:POST
->(notes) ->void
# takes in the notes that need to
be inserted into the Note
Collection

Insert_Books:POST
->(book) ->void
# takes in the book that needs to
be inserted into the Book
Collection

Update_Notes:PUT
->(id,notes) ->void
# takes in the id of a note and
Overwrite the notes at id with
object 'notes'

Delete_Notes:DELETE
->(id) ->void
# takes in the id of a note and
removes it from the Notes
Collection

Oelete_Books:DELETE
->(id) ->void
# takes in the id of a book and
removes it from the Book
Collection

Delete_Library:DELETE
->() ->void
# takes in no args and drops the
Book Collection (Library)

Get_Notes:GET
->(id) -> Document = dict{}
# Retrieves the notes at specified
id from Note Collection

Get_Books:GET
->(id) ->Document = dict{}
# Retrieves the Book at specifled
id from Book Collection

Get_Library:GET
->() ->array[] of Document =
dict{}
# takes in no args and returns all
documents in Book Collection as an
array

Helper Functions:

connect: connects to mdb server
Used by all, URI= localhost:27017
for non deployment purposes

note_exists: check if notes exist
for a certain book
Used by Update_Notes

Delete_note-library:DELETE
#DO NOT USE ONLY USED IN TESTING
```