

# App Deconstruction

LUCA VENTURA | DVP 2  
E: [LAVENTURA@FULLSAIL.EDU](mailto:LAVENTURA@FULLSAIL.EDU)

# MAIN APP METHOD

## DATABASE STRUCTURE

---

The first thing I am going to discuss and deconstruct with this application is going to be the possible database behind it, because there can be no app without content from a database. Because this app is centered around digital books the main database table would house the books' information. This table would have the obvious book-related fields such as `book_id`, `title`, `author`, `num_pages`, etc. Diving a little deeper there could be a chapter's table that is normalized with the books table to contain all of the chapters for each book that would in turn be connected to a pages table for each pages' content. Next we have the activities and games that each book is associated with which would require two more tables that are again normalized with the books table. The first of these would be the activities table which would hold things like book-related quizzes, coloring images and more. In this table would also be a `book_id` field that links back to the books table to associate itself with a particular book (the same thing would be done for the games table). The other thing we need to think about are the app "skins", which don't actually change any of the layout or functionality, just the design aspect. This would be a simple database table that would again be normalized with the books table because everything involving the skin pertains to that specific book. Some of the SQL queries to load information from these tables might be:

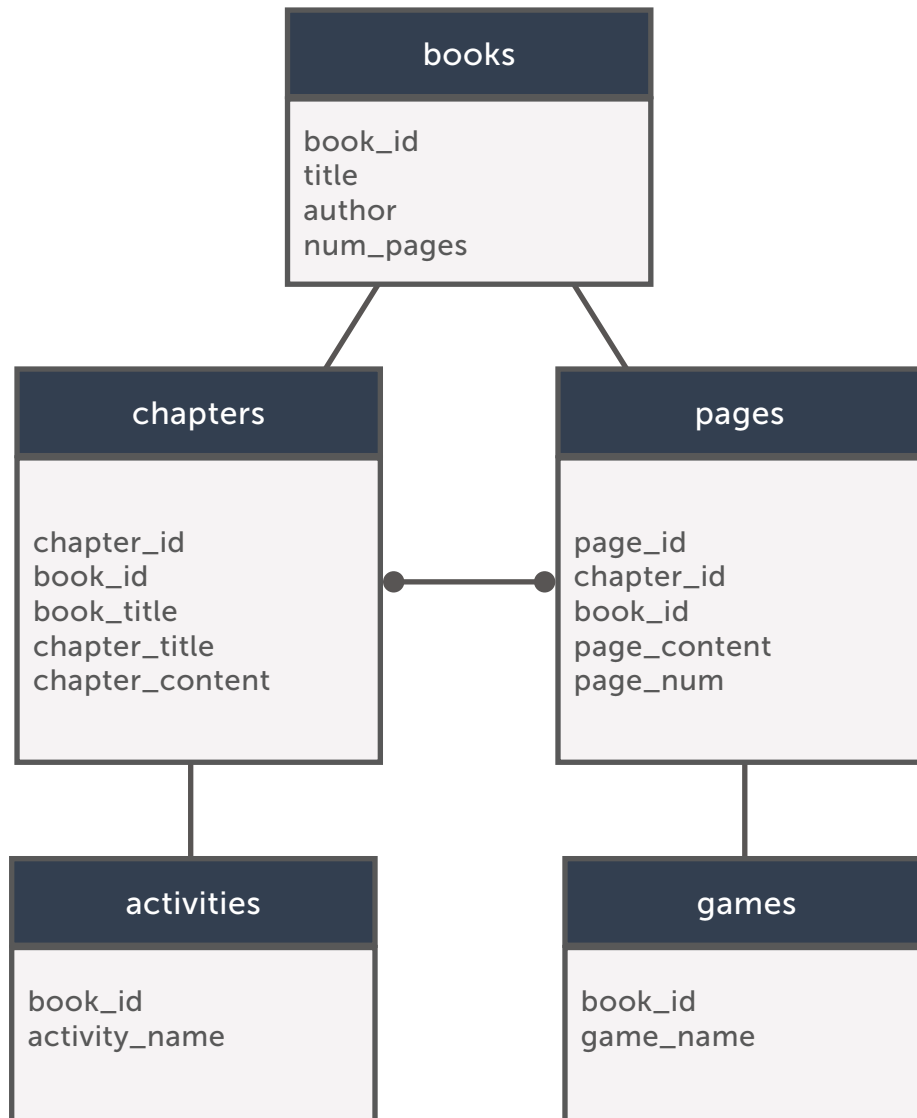
### Load Chapters/Pages:

```
select chapter_title from chapters where book_id = "1234" limit 5  
select page_content from pages where chapter_id = "2" and book_id = "1234" and  
page_id = "6"
```

### Activates/Games:

```
select activity_name from activities where book_id = "1234"  
select game_name from games where book_id = "1234"
```

# DATABASE UML



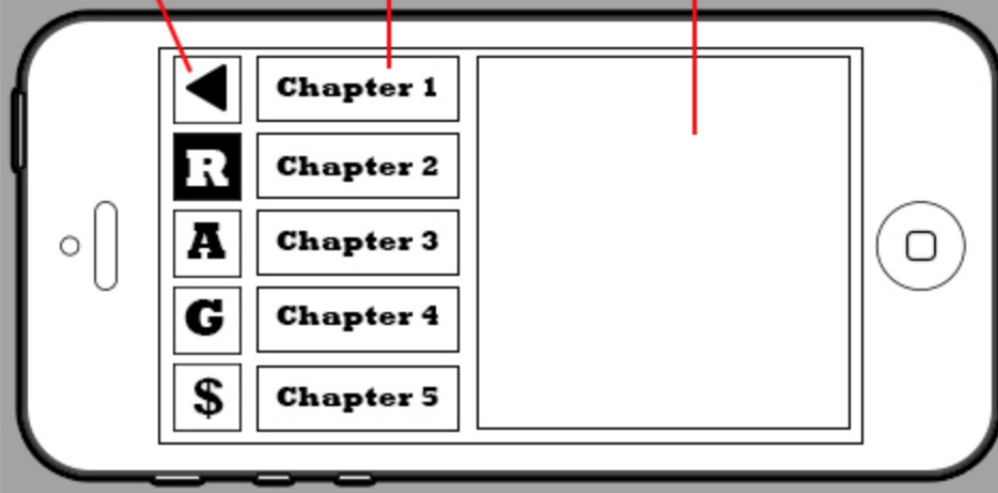
# MAIN APP METHOD CONTINUED

## PSUEDO-CODE

- Back Button (To Main Menu)
- Read Book Button (Opens Chapters)
- Activities Button (Opens Coloring, Quizzes, Etc)
- Game Button (Opens Book Related Games)
- Buy Books Button (Takes User To Buy Books Page)

Chapter Content, Activity Content  
Game Content, Etc, Load Here

Chapters, Coloring Images,  
Quizzes, Etc, Load Here



Book Class (name, chapters, skin, games, activities, bought);  
Chapter Class (name, book, content);  
Games Class (name, book, game);

Activity Class (name, book, quiz, coloring\_images)

```
List<Book> bookList();
List<Chapters>
bookChapts();
List<Games>
bookGames();
List<Activity> bookActv();
```

Buy books: New page which loads all unowned books (book class has "bought" bool).

```
for (int i = 0; i < 5; i++) {
Console.WriteLine(this.Book.Chapters.Name) }
```

```
LoadBooks() {
foreach (string s in myBooks) {
Console.WriteLine(b.Book.Name) }
```

```
LoadActivity() { load activity where book_id =
"this.Book.book_id" }
```

```
LoadGame() { load game where book_id =
"this.Book.book_id" }
```

```
LoadContent() { load chapter/game/activity
based on user selection from list of options.
foreach (object in list) { write object name }
```

CLOSURE

# REFERENCES

---

No references used in this assignment.

LUCA VENTURA

E: [LAVENTURA@FULLSAIL.EDU](mailto:LAVENTURA@FULLSAIL.EDU)