Immersive Android Mobile Application: “Athenaeum”

Ira Etzel

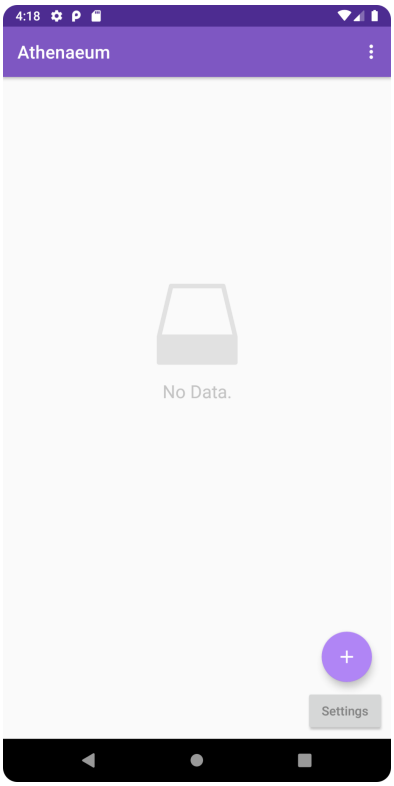
CSU Global

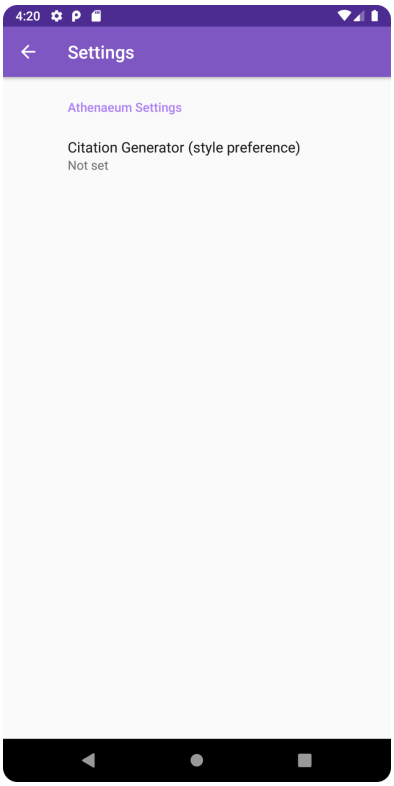
CSC470: Software Engineering

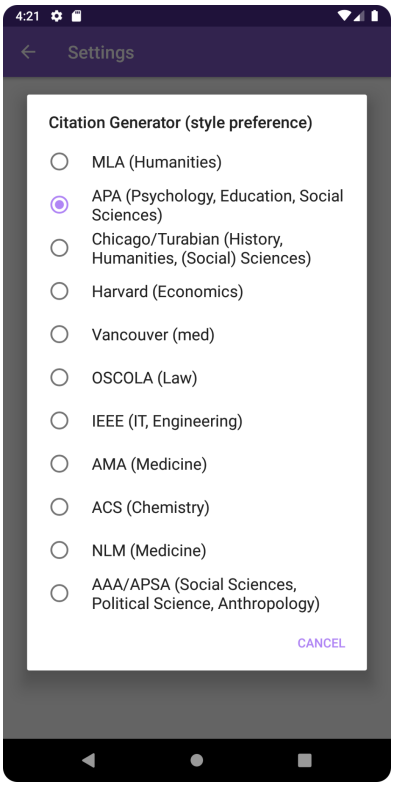
Prof. Leaston

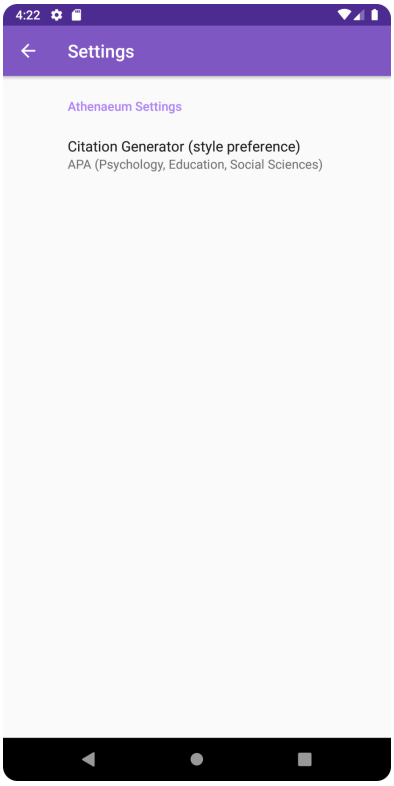
3/13/21

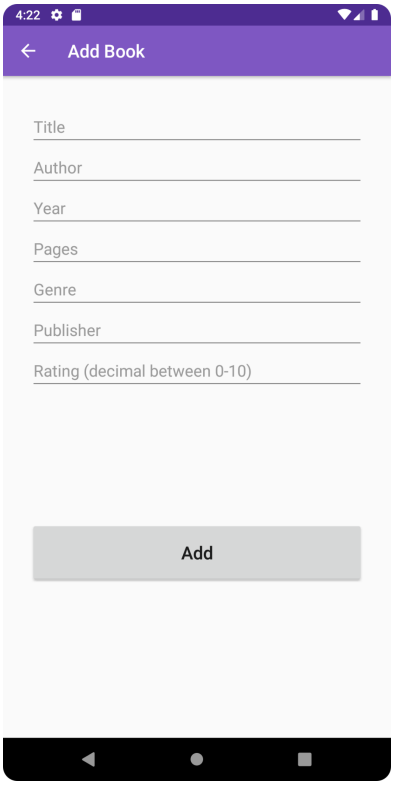
Immersive Android Mobile Application: “Athenaeum”

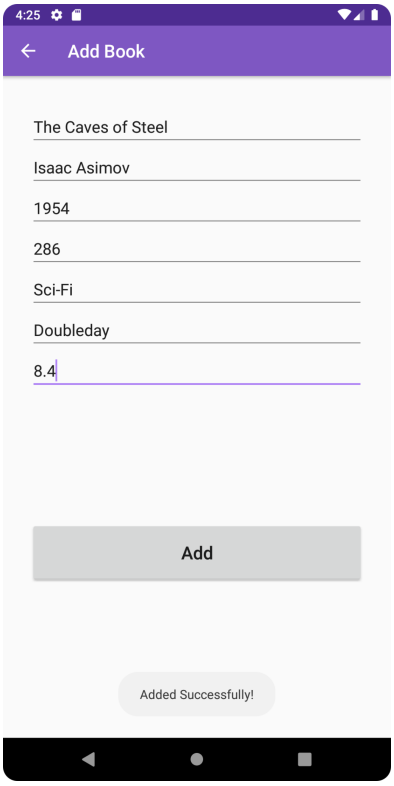


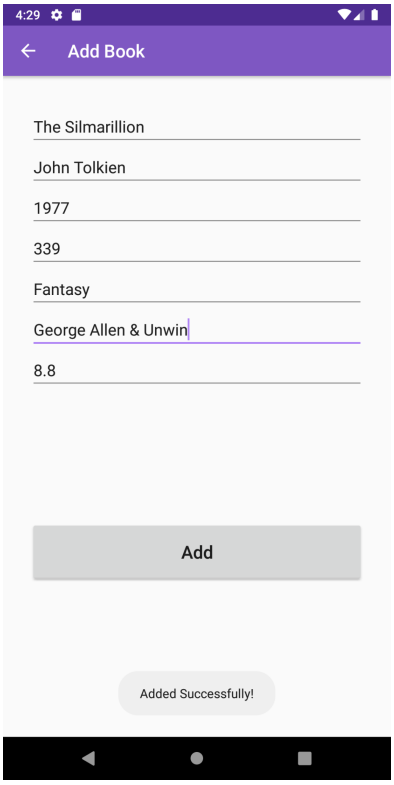


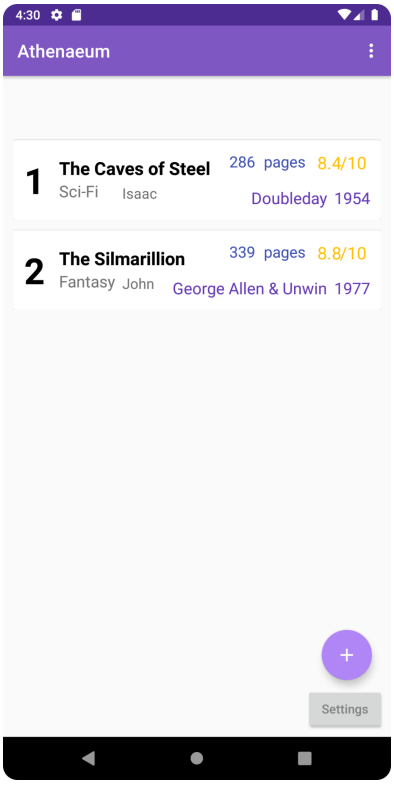


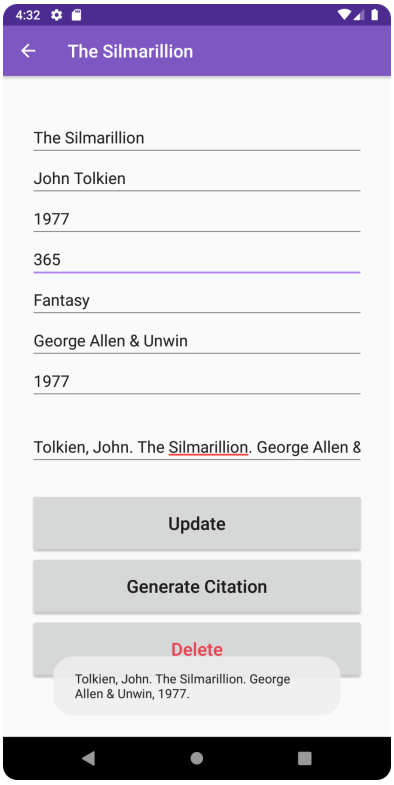


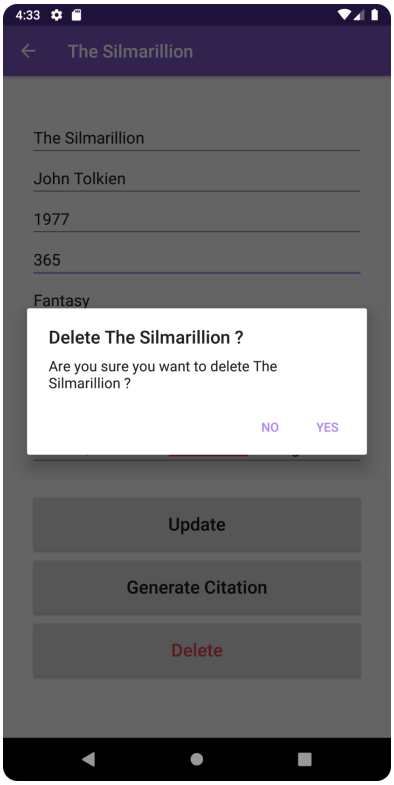


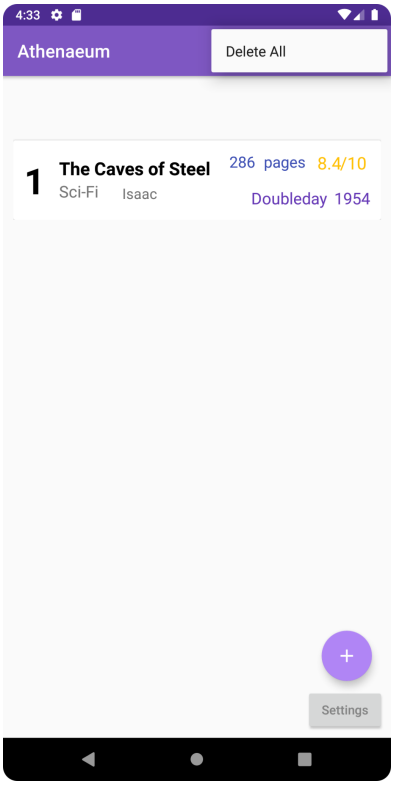


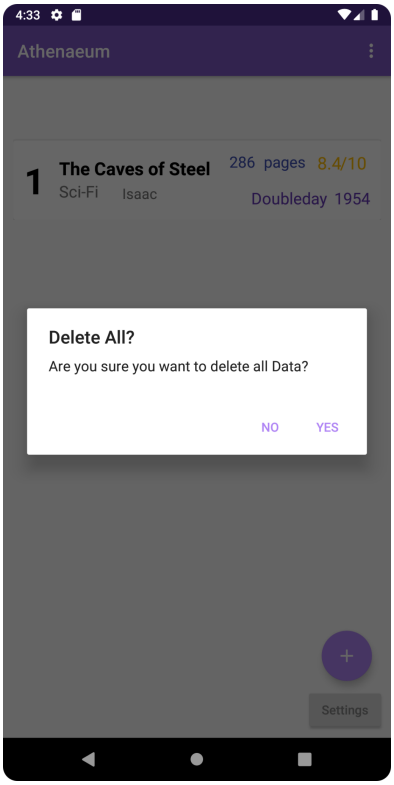

4:29
Add Book
The Silmarillion
John Tolkien
1977
339
Fantasy
George Allen & Unwin
8.8
Add
Added Successfully!


4:30
Athenaeum
1 The Caves of Steel 286 pages 8.4/10
Sci-Fi Isaac Doubleday 1954
2 The Silmarillion 339 pages 8.8/10
Fantasy John George Allen & Unwin 1977
Settings


4:32
The Silmarillion
The Silmarillion
John Tolkien
1977
365
Fantasy
George Allen & Unwin
1977
Tolkien, John. The Silmarillion. George Allen &
Update
Generate Citation
Delete
Tolkien, John. The Silmarillion. George Allen & Unwin, 1977.


4:33
The Silmarillion
The Silmarillion
John Tolkien
1977
365
Fantasy
Delete The Silmarillion ?
Are you sure you want to delete The Silmarillion ?
NO
YES
Update
Generate Citation
Delete


4:33
Athenaeum
Delete All
1 The Caves of Steel 286 pages 8.4/10
Sci-Fi Isaac Doubleday 1954
Settings


4:33
Athenaeum
1 The Caves of Steel 286 pages 8.4/10
Sci-Fi Isaac Doubleday 1954
Delete All?
Are you sure you want to delete all Data?
NO
YES
Settings

Application Features

In the “Atheneum” app, a user may create/read/update/delete book data (title, author, pages, year, publisher, genre, rating), and select a shared preference to determine a particular style of citation to generate after a “Generate Citation” button is tapped. The following options for citation style are included in the app:

Style	Basic Format for Citing a Book
MLA (humanities)	Last, First. *Title.* Publisher, year.
APA (psych, edu, social-sci)	Last, First initial. (Year). *Title*. Publisher.
Chicago/Turabian (history, humanities, social-sci, sci)	Last, First. *Title*. n.p.: Publisher, Year.
Harvard (economics)	Last, First initial Year, *Title,* Publisher, n.p.
Vancouver (med)	Last First initial. Title. [place unknown]: Publisher; Year.
OSCOLA (law)	First Last, *Title* (Publisher | year).
IEEE (IT, engineering)	First initial Last, *Title*. [place unknown]: Publisher, year.
AMA (med)	Last First initial. *Title*. Publisher; year.
ACS (chem)	Last, First initial. *Title*; Publisher, year.
NLM (ned)	Last First Initial. Title. [place unknown]: Publisher; Year.
AAA/APSA (soc-sci, poli-sci, anthropology)	Last, First. Year. *Title*. [place unknown]: Publisher.

Required Technologies Analysis

Choice of IDE

Initially, some options to consider for an IDE were VS Code, Processing for Android, IntelliJ, Xamarin, and Android Studio. The latter turned out to be the most reasonable choice, as all other IDEs had at least 1 major restriction to avoid.

VS Code. A text-editor (rather than an actual IDE); restrained development plugins.

Processing for Android. Restricted to API 29+; app has no need for advanced graphics.

IntelliJ. Less refined for Android Development; slightly more clutter due to .idea files.

Xamarin. Limited to .NET; in an analysis of several programming languages, Pereira, R. et. al. (2017) indicates that C# & F# (.NET) perform significantly poorer than Java:

Table 4. Normalized global results for Energy, Time, and Memory

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Table 5. Pareto optimal sets for different combination of objectives.

Time & Memory	Energy & Time	Energy & Memory	Energy & Time & Memory
C • Pascal • Go	C	C • Pascal	C • Pascal • Go
Rust • C++ • Fortran	Rust	Rust • C++ • Fortran • Go	Rust • C++ • Fortran
Ada	C++	Ada	Ada
Java • Chapel • Lisp • Ocaml	Ada	Java • Chapel • Lisp	Java • Chapel • Lisp • Ocaml
Haskell • C#	Java	OCaml • Swift • Haskell	Swift • Haskell • C#
Swift • PHP	Pascal • Chapel	C# • PHP	Dart • F# • Racket • Hack • PHP
F# • Racket • Hack • Python	Lisp • Ocaml • Go	Dart • F# • Racket • Hack • Python	JavaScript • Ruby • Python
JavaScript • Ruby	Fortran • Haskell • C#	JavaScript • Ruby	TypeScript • Erlang
Dart • TypeScript • Erlang	Swift	TypeScript	Lua • JRuby • Perl
JRuby • Perl	Dart • F#	Erlang • Lua • Perl	
Lua	JavaScript	JRuby	
	Racket		
	TypeScript • Hack		
	PHP		
	Erlang		
	Lua • JRuby		
	Ruby		

Development Guidelines

Dependencies. Beyond com.google & Junit, AndroidX provides the core dependencies for the app:

General	androidx.appcompat:appcompat:1.1.0 androidx.constraintlayout:constraintlayout:1.1.3 androidx.navigation:navigation-fragment:2.4.1 androidx.navigation:navigation-ui:2.4.1 androidx.preference:preference:1.2.0 androidx.recyclerview:recyclerview:1.1.0 com.google.android.material:material:1.0.0
Testing	junit:junit:4.12 Androidx.test.ext:junit:1.1.1 androidx.test.espresso:espresso-core:3.2.0

Dependency Usage. The following table indicates which import statements are used in the 6 java files from com.csc475.atheneum:

	Add Activity	Custom Adapter	Main Activity	My Database Helper	Settings Activity	Update Activity
(android.app) Activity		x				

(android.content) ContentValues Context DialogInterface Intent SharedPreferences		 x x 	 x x 	 x x 	 x	 x
(android.database) Cursor sqlite.SQLiteDatabase sqlite.SQLiteOpenHelper			 x x 	 x x x		
(android.os) Build Bundle	 x	 x 	 x		 x	 x x
(android.text) Html						 x
(android.util) Log				 x		 x
(android.view) ContextMenu LayoutInflater Menu MenuInflater MenuItem View ViewGroup animation.Animation animation.AnimationUtils	 x	 x x x x	 x x x x x			 x
(android.widget) Button EditText ImageView LinearLayout TextView Toast	 x x	 x x	 x x x x			 x x x
(androidx.annotation) NonNull Nullable RequiresApi		 x x	 x 	 x 		
(androidx.appcompat)						

app.ActionBar app.AlertDialog app.AppCompatActivity			 x x		x x	x x x
(androidx.cardview) widget.CardView		 x				
(androidx.preference) PreferenceFragmentCompat PreferenceManager			 x		x	
(androidx.recyclerview) widget.LinearLayoutManager widget.RecyclerView		 x	 x x			
(java.util) ArrayList			 x			
	Add Activity	Custom Adapter	Main Activity	My Database Helper	Settings Activity	Update Activity

Power Usage Optimization. According to Chandrasekaran et. al. (2020), “unnecessary resource usage, faulty GPS behavior, background activities, advertisements, and high GPU usages are considered as the primary reasons for battery-related issues in smartphone applications” (para. 1). Other than avoiding the aforementioned factors, whenever possible, the most efficient file types (e.g. webp rather than png) are used.

Storage & I/O. SQLite functions as a simple RDBMS for the application, while Button + EditText + TextView + Toast (from the android.widget library) are employed to serve this app’s core I/O requirements.

Layout Design. 4 out of 5 of this project’s res/layout files (activity_add.xml, activity_main.xml, activity_update.xml, my_row.xml, and settings_activity.xml) are organized within a constraint layout. As Rabello (2022) states, “[s]ince it was introduced during 2018’s Google I/O, ConstraintLayout has been the state-of-the-art layout tool that can accommodate almost any design” (para. 26).

References

Chandrasekaran, C. et. al. (2020). “Energy Diagnosis of Android Applications: A Thematic Taxonomy and Survey.” *ACM Computing Surveys, 53*(6), 1–36. Retrieved from https://doi.org/10.1145/3417986.

Pereira, R. et. al. (2017). “Energy efficiency across programming languages: how do energy, time, and memory relate?” *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering*, 256–267. Retrieved from https://doi.org/10.1145/3136014.3136031.

Rabello, R. (2019). “Google I/O 2019: Unpacking the top news for Android app developers” *Arctouch*. Retrieved from https://arctouch.com/blog/android-development-news-google-io-2019/.