Open Source Science Learning IOS Application

Nick Shier
nshie172@mtroyal.ca

Charlotte Curtis

Department of Mathematics and Computing
Mount Royal University

Abstract. Large technology companies like Apple and Meta are able to provide large salaries to top talent engineers. This takes top talent away from startups and small research teams in other areas of research like in the bioinformatics field, which has a larger societal impact such as curing disease. I wanted to use my computer science knowledge to create a free resource for students to learn quality higher education science content on their phones. This birthed the idea of an open source IOS application to help students learn using active recall based strategies. Active recall is a strategy used by medical students all over the world to commit mass amounts of information to long term memory. For this project I created Sci-cards, a free, open sourced IOS application to help students practice their active recall for biology, in preparation for writing the Medical College Admission Test (MCAT). This was done by content reviewing a Kaplan 2024 Biology Textbook and creating 25-35 high quality flashcards for the important concepts of 12 topics. I detail my planning process and experience using a Mac Mini with the XCode IDE, Swift programming language, GRDB ORM, and SQLite database to create this application over a 3 month period.

Keywords: Education, Science, IOS.

1 Introduction

As an undergraduate Computer Science student at Mount Royal University, the jobs that I hear my peers talk frequently about hoping to get after graduation are at big-tech companies like Amazon, Google, and Netflix. While I think delivery, search, and entertainment are bonuses to our lives, I began to wonder if these high paying tech jobs are attracting top talent to a field with a smaller societal impact than biological studies. One study found that 20% of all expert A.I researchers work at just 5 big-tech companies such as Apple and Meta. The authors suggest that large tech companies can afford to attract the best talent, while research labs and startups struggle to find qualified people [1].

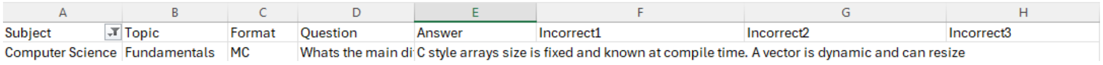
For this project I considered exploring the bioinformatics field, which has large societal impacts from advancements in tools like CRISPR-CAS9 gene editing. CRISPR is a gene editing technique discovered in 2013 that hopes to defeat viral infection and genetic mutation which cause incurable hereditary disease [2]. After deeper research, I concluded a software project in this area would be out of scope for the 3 month project timeline.

This paper discusses my planning, design, experience, the possible future additions, and the societal benefits of creating a free, open sourced IOS science flashcard app. The name of the application is Sci-cards, the reason is that for someone who is wanting to find a flashcard app for science, this name is self-explanatory. The goal of this project was to enhance students' active recall ability through repeated testing of high quality biology and computer science flashcards. Coming up with the idea for this app came from trying to study for the Medical College Admission Test (MCAT) 1 year ago, and finding a similar free desktop application called Anki. This app has a very large community of users, developers, and flashcard creators. While thinking about the improvements I could make to this app, I considered the problems I personally had. One of the main problems I found with Anki is that the user has to find, import, and quality check the flashcards from other places on the internet. This can be very time consuming and frustrating when importing cards that have errors. Another issue I had was the user interface to create your own flashcards was not intuitive to use quickly. Lastly, when I wanted to get the application on my mobile phone to enjoy learning on the go, I found the application was very expensive at $34.99 CAD.

To fix issues 1 and 2 for my application, I manually created high quality flashcards and the cards are loaded into the application upon download. To fix issue 3, I will release the application as free.

2 Development Plan

When researching possible ways to develop an IOS application I discovered I could use the Swift programming language on a Mac, or use React Native. The pros of developing an IOS application with Swift is this is the programming language created by Apple for mobile application development. I also own an Iphone which I figured could be beneficial to test on, separate from an emulator. I considered using React Native solely because your app can be designed to work on IOS and Android, but I don't have any web development experience using a language like Javascript, and I do have object oriented experience with C++. Thus, I went with the Swift language. I have briefly tried Android development with Kotlin before, and I know that it's extremely beneficial in development to have a device with a good enough CPU to update the preview fast when you are developing UI elements. I went with the Mac Mini M1 because the M1 chip has a fast 8 core CPU that works great for native applications like Xcode. Because I am manually creating and loading the flashcards into the application, the easiest solution to initially represent the flashcards was using excel (see Fig. 1).



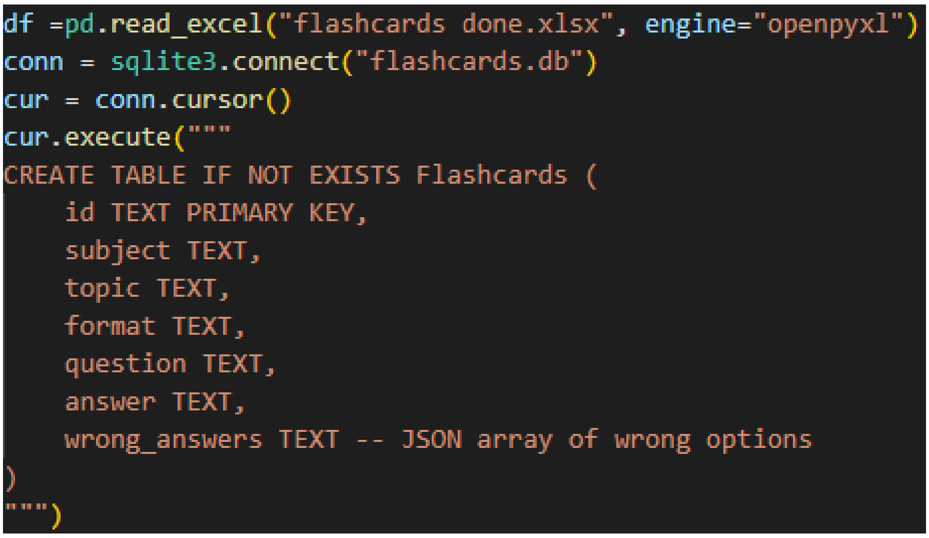
A	B	C	D	E	F	G	H
Subject	Topic	Format	Question	Answer	Incorrect1	Incorrect2	Incorrect3
Computer Science	Fundamentals	MC	Whats the main di	C style arrays size is fixed and known at compile time. A vector is dynamic and can resize			

Fig. 1. My representation for a single flashcard, stored in an Excel spreadsheet.

My hope was that it would be possible to create a struct in Swift with the same names as the excel document column headers and then import into the struct, and that this would simplify the implementation. After some research on easy to use databases with Swift, I discovered SQLite as it is known to be light weight and simple.

3 Technology Implementation

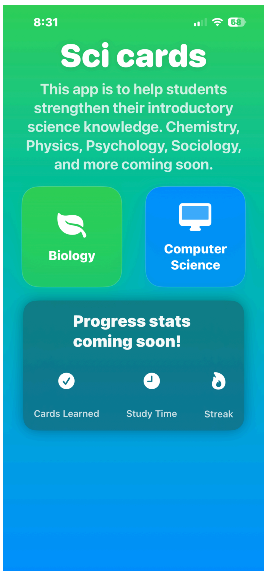
After researching how to use the pandas python library to read and extract data from an excel sheet I found blog posts from GeeksforGeeks to be very helpful [6]. I created a small python script to parse the excel document and create the SQLite .db file (see Figure 2).



```
df =pd.read_excel("flashcards done.xlsx", engine="openpyxl")
conn = sqlite3.connect("flashcards.db")
cur = conn.cursor()
cur.execute("""
CREATE TABLE IF NOT EXISTS Flashcards (
    id TEXT PRIMARY KEY,
    subject TEXT,
    topic TEXT,
    format TEXT,
    question TEXT,
    answer TEXT,
    wrong_answers TEXT -- JSON array of wrong options
)
""")
```

Fig. 2. A code snippet of the python script I created to parse an excel document of manually created flashcards, create a connection to a database called flashcards.db, and create a table inside called Flashcards, which it populates.

To start developing the application, I started with learning how to create and manipulate UI elements using Swift. I found the Swift documentation created by Apple incredibly useful [3] as it shows code examples of everything in the language. I also found my knowledge in Haskell to be useful when learning Swift as there are similar features in both languages such as the Map higher order function. I learned how to create a dynamic UI using Navigation Stack and button actions to move to other screens. I also learned how to create a visually appealing UI using ZStack and applying a Linear Gradient of different colors throughout the screen, which I found could be useful for keeping students engaged. HStack is how you arrange elements together in a horizontal row, while VStack allows you to arrange elements together in a vertical row. Next it was through combining buttons, ZStacks, HStacks, VStacks, and NavigationStacks that I learned from the Swift documentation, to create all the UI elements (see Fig. 3.).



Sci cards

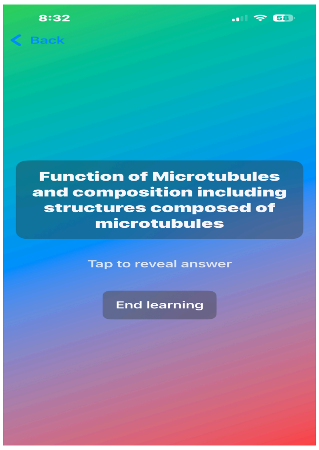
This app is to help students strengthen their introductory science knowledge. Chemistry, Physics, Psychology, Sociology, and more coming soon.

Progress stats coming soon!

Fig. 3. The main screen the user sees when loading the application, created with a combination of ZStacks, Linear Gradients, NavigationStacks and buttons, VStacks, and HStacks elements.

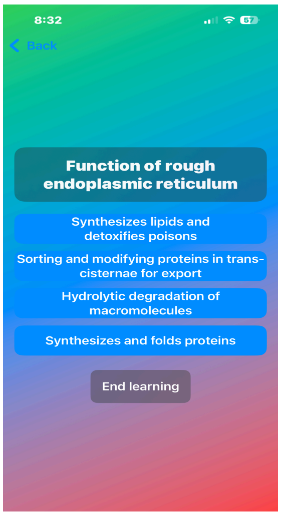
4 Content Review and Flashcard Creation

After deciding how I would represent the cards while I create them, and how they would be stored in the database for the app to query, it was time to start creating flashcards. I decided that I would create a comprehensive set of biology flashcards that would cover the must know MCAT biology content. To do this I used an old E-book textbook that I had from Kaplan, and started picking out all the most valuable information from each section. Each question is either a big concept, a definition, or topics that appear in practice questions, and knowledge checks. To set the app apart from Anki, I decided that I would create self check questions (see Fig. 4) and also multiple choice style questions below (see Fig. 5).



Function of Microtubules and composition including structures composed of microtubules

Fig. 4. The Sci-cards UI representation of the Self Check question style where users will remember the answer to the question, reveal the answer, and self-report how they did.



Function of rough endoplasmic reticulum

- Synthesizes lipids and detoxifies poisons
- Sorting and modifying proteins in trans-cisternae for export
- Hydrolytic degradation of macromolecules
- Synthesizes and folds proteins

Fig. 5. The Sci-cards UI representation of the Multiple Choice question style which aims to keep users engaged by allowing less knowledgeable users to use techniques like process of elimination.

5 Database Technologies

Once the flashcards were created, I researched how to connect the flashcards inside the database to a flashcard struct. inside the Sci-cards application. In another project I did not use an Object-Relational Mapping technology and found it cumbersome to query the SQLite database using C++. Thus, for this project I decided to try the GRDB Swift to SQLite ORM. This is a SQLite toolkit to connect Swift to SQLite to make development much easier. Lots of documentation exists for how to use GRDB, on the github repo page it shows you how to define the struct and the keywords needed to make the record fetchable from the database [4], which I needed in this project.

6 Technology Learning Takeaway

This project had a very large learning curve as I had never used Swift, Xcode, Mac, GRDB, SQLite, and I wasn't very familiar with mobile or web development strategies. I found Swift to be a strange programming language as if it's doing the work of multiple languages in one. For example, Swift is the language used to define UI containers, style, and interact with the database to perform backend operations. Thus you end up with a UI block that has multiple containers with backend code and styling, which can get messy to debug and even just strange to look at. Overall it was a positive experience stepping out of my technological comfort zone to create something in a field that I was truly passionate about, that I believe has strong societal impacts.

7 Societal Benefits of This Project

7.1 Contribution to Sustainable Development Goals

In 2015 at the United Nations meeting, 17 countries including Canada committed to the 2030 Sustainable Development Goals. Among these goals is Goal 4, the commitment to "Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all" [7]. Furthermore, goal 4.3 specifically refers to access to higher level education. This application can help English speaking students from all over the world to study their higher education science content. Free education particularly in under-funded and impoverished areas can have a dramatic impact on education outcomes, study findings from [8] found that free education can improve equity by up to 30% and increase student enrollment and attendance by 15% (see Fig. 6).

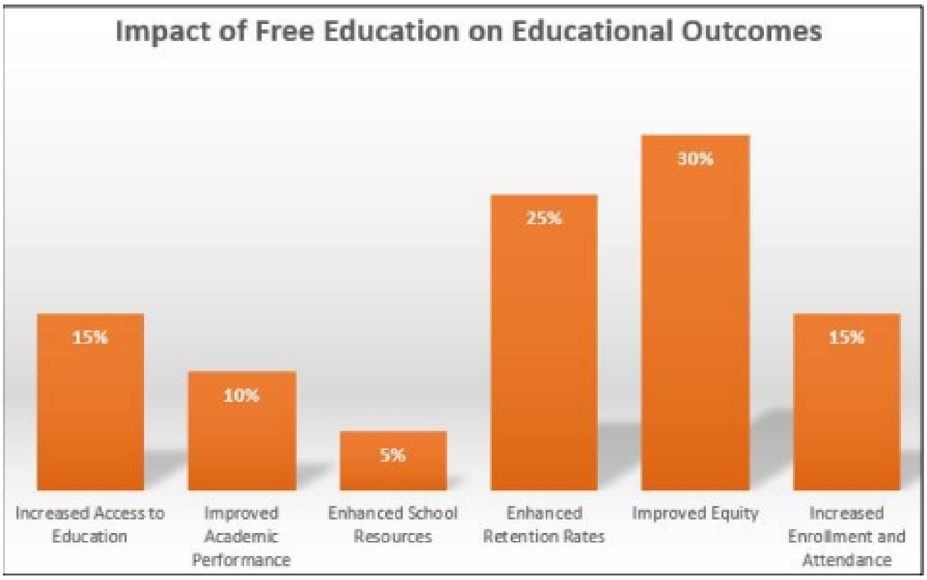


Fig. 6. A study found that free education can improve equity in education in our society by up to 30% and increase enrollment and attendance by 15%.

7.2 Active Recall Improves Student Performance

According to a blog post by 3 doctors, the amount of medical information vastly grows each year, thus doctors must learn much more information than the year before[9]. This post compared the findings of many research papers on effective learning strategies and found that between studying once, recalling once, repeatedly recalling, and repeated spaced recalling, the most effective learning strategy by far was repeated spaced recalling. The students who used repeated spaced recall had a 75% recall ability on the final recall, while students who studied once had a less than 5% ability. A different study found that students performed better after they used a digital tool like Kahoot, showing that even digital education strategies have an improvement [10]

8 Future Additions

Currently Sci-cards can be used as a minimum viable product to help biology students get better at actively recalling fundamental Biology concepts such as in Cell Biology. For Sci-cards to be fully ready for the app store, the things I would add is at least 2 more subjects, particularly chemistry and physics as these are also large topics on the MCAT. Next, I would add an algorithm to perform spaced-repetition as this has been shown to be the most effective learning strategy by far compared to all other strategies tested [9]. This algorithm will run while the user is learning and be updating the review dates for cards based on how well the user is able to recall the answer. This allows for the review date for cards the user frequently gets right to be further away than cards the user frequently gets wrong. Currently, if the user already has a card committed to long term memory they will see that card at the same rate as another card they have not learned. Lastly, the next feature implementation I would like to add is the ability for users to create their own flashcards and add images into these cards with a very intuitive user interface.

9 Conclusion

Overall, this project was a success. The most successful factor about this project is in its benefits to society. When more subjects are added, this app can serve as a free tool for undergraduate students in North America studying for the MCAT. By providing this tool as free we are improving equity and student engagement in education and also contributing to Canada's commitment to the 2030 sustainable development goals of providing free higher education. Sci-cards the application I created will use active recall learning strategies which have been shown across multiple studies to improve test scores, and has been a tool used by medical students to commit a vast amount of information to long term memory. Stepping out of my technological comfort zone to create something in a field I was passionate about was an interesting experience. I felt frustrated when things with the code didn't work but always managed to fix it, because I wanted to see the project I was motivated to create come to life.

References

1. Manera, Andrea Competing for Inventors: Market Concentration and the Misallocation of Innovative Talent paper, https://github.com/andmanera/Job-Market/blob/main/draft/draft.pdf, last accessed 2025/12/20.
2. Sharipov, R.A., Omarov, M.A., Mulyukov, A.R., Dybova, A.I., Vyaseleva, E.T., Kayumova, N.B., Saitgalina, A.S., Ententeev, K.R., Iagafarov, I.R., Kuserbaev, I.V.: Benefits of Using the CRISPR/Cas9 System for the Correction of Genetic Mutations. Molecular Genetics Microbiology and Virology. 41(3) (2023).
3. Swift Documentation, https://docs.swift.org/swift-book/documentation/the-swift-programming-language/guidedtour/, last accessed 2025/12/17.
4. GRDB Github Repository README, https://github.com/groue/GRDB.swift, last accessed 2025/12/17.
5. Benefits of Active Recall, https://www.lecturio.com/inst/pulse/retrieval-based-learning-strategies-in-medical-education/, last accessed 2025/12/17.
6. Convert Excel To Json With Python, https://www.geeksforgeeks.org/python/convert-excel-to-json-with-python/, last accessed 2025/12/17.
7. United Nations Goal 4, https://sdgs.un.org/goals/goal4, last accessed 2025/12/17.
8. Thelma, C., Mpolomoka, D., Gilbert, M., Mulenga, D., Sain, Z.: Free Education vs Quality Education: A systematic analysis. World Journal of Advanced Research and Reviews. 23(01), 2934-2946 (2024).
9. Active Recall and Retrieval Practice in Medical Education, https://www.lecturio.com/inst/pulse/retrieval-based-learning-strategies-in-medical-education/, last accessed 2025/12/17.
10. Kalogiannakis, M., Papadakis, S., Zourmpakis, A.: Gamification in Science Education. A Systematic Review of the Literature. Education Sciences. 11(1) 1-36 (2021).