

# A STRUCTURED NOTE-TAKING PROGRAM TO AID IN DATA RETENTION AND PROMOTE ORGANIZATION USING MOBILE DEVICES AS A MEDIUM

Mingyuan Gao<sup>1</sup>, Armando Contreras<sup>2</sup>

<sup>1</sup>Oaks Christian school, 31749 La Tienda Rd, Westlake Village, CA 91362

<sup>2</sup>Computer Science Department, California State Polytechnic University,  
Pomona, CA 91768

## **ABSTRACT**

*In recent years, note-taking applications have become a common tool due to the simple organizational features it can provide. A key aspect of note-taking applications is the ability to efficiently record and edit large amounts of data. SQL databases are well suited to this task because they provide a reliable and scalable way to manage structured data, allowing note-taking applications to more efficiently store notes, tags and metadata associated with each note. They also support search and retrieval operations, enabling users to quickly find the information they have recorded. In addition, SQL databases provide features such as data stabilization, alternation and access, which help ensure the integrity and security of user-entered data. By utilizing SQL database replication and sharing features, note-taking applications can distribute their data across multiple servers, improving performance and reliability. In our experiments, the app demonstrated its reliability and memory capabilities. With a mean load time of 3 seconds. In summary, note-taking applications benefit from using SQL databases due to their efficiency, scalability, and data management features. These advantages help users maintain integrity, find information quickly, and keep their notes secure.*

## **KEYWORDS**

*Note-Taking, Application, Flutter, SQL*

## **1. INTRODUCTION**

Memory is the part of a human controlled by the brain to store and retrieve data. It is important for people to keep track of their responsibilities and duties on a daily basis. However, nowadays we find ourselves surrounded by a variety of situations that can make it hard to manage and keep track of our tasks and responsibilities, especially when it increases in complexity and quantity. Not to mention that humans can be afflicted by illnesses that affect memory directly, such as Dementia or Alzheimer's disease [1]. Large amounts of stress lead to forgetting important information [2]. School, business, work, as well as cluttered environments contribute to the issue which in combination to the aforementioned causes can make the effect exponentially worsen if left untreated [3]. As such this introduces our main concern, ineffective note-taking. An approach that can negatively impact learning and productivity. Using physical methods, such as pen and paper, can be time-consuming and unstructured, which can complicate finding specific information when needed. Handwritten notes can also be tough to read, especially when they

were written quickly, leading to vagueness and misinterpretation of material [4]. This can be especially problematic for people who heavily count on their notes to prepare for examinations or projects. Additionally, physical notes are prone to be lost and damaged too, resulting in the loss of important data. People who misplace their notebooks or forget to bring them to class can miss out on essential information and key points. One last issue we are worried about is that traditional note-taking methods can lead to a lack of engagement during classes, especially for students who have trouble keeping up with lectures.

Note-taking is an important part of students and professional life, but presents various problems such as editability, loss/damage potential and readability. Making an app solves this by avoiding those issues and providing various benefits [14]. They are easy to organize, share, and access in comparison to traditionally handwriting notes. They can be limited by the device's technology, however, which led to the application focusing on it being quick, reliable and easy to use. However, this introduces a new problem, technology itself can be distracting for their users and in the case of apps, this is a given [18]. If given a positive approach, it can help in both a professional and classroom setting though. They are an effective way of storing data and studying it afterwards making it a strong tool to have at one's disposal [15]. For this reason, we stuck to limiting factors that encourage its educational use with a simplistic design. Note-taking in a collaborative environment has been proven to increase productivity and performance from those involved too, another driving factor in the development of this application [16]. Given all these advantages and restrictions we modeled potential improvements for the app, although it may introduce new problem variability.

Our proposal is the creation of a note-taking app since this method of capturing data offers various advantages over those with the traditional pen and paper. It is an ideal choice for professionals, students, and anyone who needs to take notes on a regular basis. It provides a structured and organized platform for note-taking, allowing individuals to categorize, search, and tag notes for quick and efficient retrieval. This can help address the challenges of ineffective note-taking, such as difficulty in finding specific information and incomplete notes. Also, a large section of the world population has access to technological devices and subsequently applications [5]. This makes them ideal to take them on-the-go as well, which shifts our focus to this area of development. Simply put, Note-taking apps furnish a system for methodical and structured note-taking, alongside the capacity to search, label, and group notes [6]. Furthermore, they can synchronize across various devices, guaranteeing that such notes are continuously accessible and current. With note-taking apps, individuals can store their notes in the cloud, ensuring that their notes are always backed up and accessible in case of loss or damage to their devices. It is worth noting they can be equipped with additional features such as taking pictures, recognizing handwriting, and capturing video output too. Such tools can elevate the user experience and simplify the writing process so users can focus their attention on reading and learning. By integrating the aforementioned features, we plan to create an app ourselves that can gather more information, render notes more accessible, and deliver a more immersive note-taking experience.

We conducted experiments on section 4 to test and find the limitations of the application. First, we studied the limitations of Flutter's display features and libraries to see if it could fulfill the basic requirements for users to understand the options available on their screen. Various currently popular smartphone devices were fed a set of notes to investigate how many notes they could handle. Experiment 1 demonstrated that the app should be hard set to limit the quantity of notes to 280 given this was the lowest result. Since a main goal is to make it reliable and usable to as many people as possible, we would decrease the maximum to permit leeway on the variability smartphones may have from one another. In Experiment 2, we aimed to find file load times in seconds in pursuit of the maximum amount of data the application could load in a reasonable time frame. The results, as we expected, were exponential. We searched for the point in which it

would not exceed a certain amount of wait time, which ended up being 3 seconds maximum as it seemed a good maximum quantity given the nature of this app as we inferred from the investigation of Li, S. & Chen, C.H. [17].

## **2. CHALLENGES**

In order to build the project, a few challenges have been identified as follows.

### **2.1. Accessibility**

Accessibility is an important requirement for the app to reach as many people as possible. One of the major hurdles is for it to be available for both iOS and Android systems. We consider building it with the Android Studio IDE since it possesses all the tools necessary to make it. We will likely require 2 different languages for each operating system, with the wide array available which includes but is not limited to Python, Java, C/C++ and Swift. Since it would have to be built potentially twice using this method, it might be better to learn and use a cross-platform friendly language or framework which can make it available in both platforms and code it once. The problem with this would be the learning process which can take as much or more time than creating them in 2 different languages. Options given this selection would be Xamarin, Flutter, react Native and Angular.

### **2.2. What Method to Utilize**

We can take various routes on storing users notes data including but not limited to: JSON, Pickle files, Firebase, and SQL. We have to consider what method to utilize and what would be the most efficient for the target audience. It needs to have the capacity for storing large amounts of information, be secure to keep data safe, and be a reliable method to access the data. An online database such as Firebase can store the information online and would require an authentication method, such as username and password [7]. On the other hand a local file such as JSON or even SQL can save it on the user's device [8]. We will develop the integral components, test multiple of these methods and decide what would be the best option once the basic display is set up.

### **2.3. GUI**

An intuitive, easy to use screen display or GUI is important for user to app interactions. We need to keep it simple so display settings and organization would be a priority focus for this issue. The app would have to be separated into multiple files representing objects and screens. We need to consider what classes we want to separate information into in order to store and display it in an organized fashion. Looking into the library repertoire available for Flutter should offer various options for this as well as expand the options GUI element can display and build upon an expansion of tools for features that can potentially simplify the user experience [9]. Doing all this whilst making the app lightweight in memory usage and load-time efficient will be a slow but important task.

## **3. SOLUTION**

The app uses the Flutter framework which is specialized to build natively compiled programs. We used it for its wide library accessibility, expandability, and is programmed in the Dart language. When the app is run, it spawns note card holders and extracts data from a locally stored SQL file, which is empty upon the first use. The SQL file stores and updates data that has been respectively added or modified by the user. Information is separated by note and represented on

the screen separately via GUI represented as note box objects. Said objects display the title and date of creation. By default, there are no notes, instead a message shows up prompting users to create new notes by selecting the button on the lower right corner of the screen. When clicked, a new screen is displayed which includes two input sections prompting users to write the title and data they would like to include. The app also displays a save button on the lower right section of the screen. Saved notes are stored on the SQL file and updated to be shown as a note box object. Selecting any of the available box objects allows users to edit or delete the note represented by a pencil and trash can icons respectively in the top right corner of the screen. Edited notes prompts the same screen as if adding a new note, filled with the currently saved data. Deleting a note clears it from the SQL file and takes it away from the note box object view.

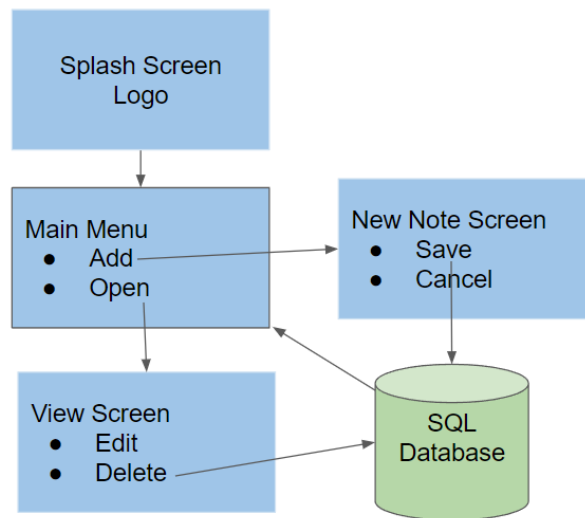


Figure 1. Overview of the solution

An important component in the application is the SQL local database. In the app, data written as notes is stored within the device as a file that is quick to read and write from. This allows for safe and easy data transfer that's private to the user. The system relies on using SQL commands to organize the data and control it as coded to withhold only the data type(s) enabled for it.

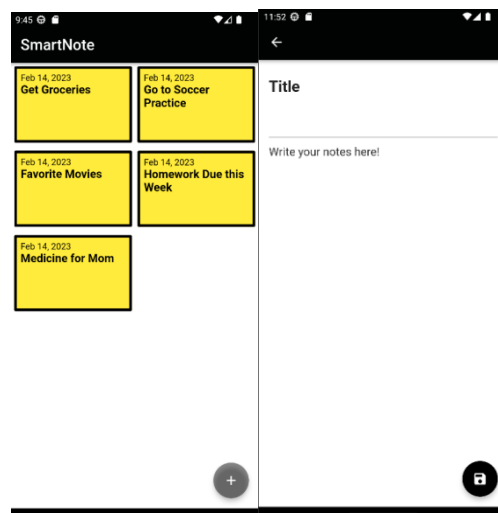


Figure 2. Screenshot of Main page and Title page

```

@override
Widget build(BuildContext context) => Scaffold(
  appBar: AppBar(
    title: const Text(
      'SmartNote',
      style: TextStyle(fontSize: 24),
    ), // Text
  ), // AppBar
  body: Center(
    child: loading ? CircularProgressIndicator(): notes.isEmpty
      ? const Text(
        'Add Some Notes Here, it is Empty...',
        style: TextStyle(color: Colors.black54, fontSize: 24),
      ): buildNotes(), // Text
  ), // Center
  floatingActionButton: FloatingActionButton(
    backgroundColor: Colors.black45,
    child: const Icon(Icons.add),
    onPressed: () async {
      await Navigator.of(context).push(
        MaterialPageRoute(builder: (context) => AddEditNotePage()),
      );
      reloadNotes();
    },
  ), // FloatingActionButton
); // Scaffold

Widget buildNotes() => StaggeredGridView.countBuilder(
  padding: EdgeInsets.all(2),
  itemCount: notes.length,
  staggeredTileBuilder: (index) => StaggeredTile.fit(2),
  crossAxisCount: 4,
  mainAxisSpacing: 5,
  crossAxisSpacing: 1,
  itemBuilder: (context, idx) {
    final note = notes[idx],

```

Figure 3. Screenshot of code 1

The app demonstrates the note display section which loads the data from the local SQL database and the note page with all related details to each individual note. Note display automatically runs when the program is initialized, serving as a main menu. The build widget displays a message prompting users to add new notes if none are found, otherwise it calls on the buildNote widget. It also displays an “add new note” button at the bottom. buildNotes reads the SQL file information and separates it into equally-sized boxes representing each note created by the user. Each note displays the creation date and title, they are then linked to a touch-based gesture detector which when selected leads to the note display screen of the app containing said specific data. In the note display screen users can use the modification features available which is similar to the add note screen to make it intuitive for users.

Another component found in the app is the data editing feature which dives into the local SQL files database installed. This focuses on the edit and delete sections which access the data by blocks and separate sections. In particular, editing retrieves the data from the currently selected note and displays it in an “add note” screen with the current information on it as displayed on the image below. The memory location retrieved from the SQL database is kept track of in a local variable so it can be utilized for the next step. Once the user finishes making changes to their note’s title and / or reminder, if the user exits the editing screen by selecting the back arrow, the notes are updated in the memory location of the SQL database and the user is returned to the notes page which we refer to as the main menu.

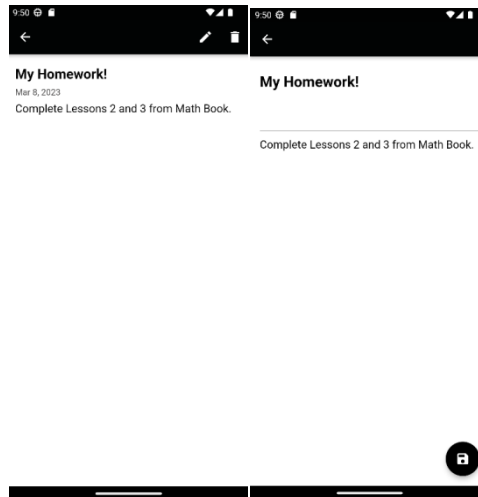


Figure 4. Screenshot of my homework page

```

override
Widget build(BuildContext context) => Scaffold(
  appBar: AppBar(
    actions: [buildButton()],
  ), // AppBar
  body: Form(
    key: _formKey,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.start,
      children: [
        NoteFormWidget(
          number: number,
          title: title,
          description: description,
          // onChangedTextColor: (textColor) => setState(() => this.textColor = textColor.value),
          onChangedNumber: (number) => setState(() => this.number = number),
          onChangedTitle: (title) => setState(() => this.title = title),
          onChangedDescription: (description) =>
            setState(() => this.description = description),
        ), // NoteFormWidget
      ],
    ), // Column
  ), // Form
  floatingActionButton: FloatingActionButton(
    backgroundColor: Colors.black,
    child: Icon(Icons.save),
    onPressed: () async {
      addOrUpdateNote();
    },
  ),
);

```

Figure 5. Screenshot of code 2

The other component, the delete feature, follows the same steps as those for the edit note section by keeping track of the selected notes data memory location via a variable. However, this option clears the data section once it is selected. The database information is re-organized (defragmented) to make efficient use of space. Afterwards, the user is then returned to the main menu.

One additional component included is a math keyboard capable of displaying mathematical notation which is usually unavailable in the default keyboard of many smart phones. This was possible by including the math\_keyboard dart based library. Said import contains symbols for, but isn't limited to: logarithms, sine, cosine, tangent and square root. This component can be accessed from the note editing and adding screens. Users can select the line found between the title and writing sections. It provides several benefits to students and professionals who need to work with mathematical equations and formulas.

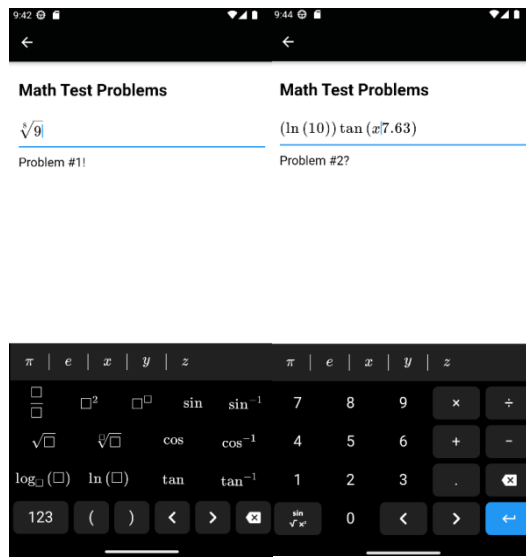


Figure 6. Screenshot of math test problems

```

Widget buildDescription() => TextFormField(
  maxLines: 20,
  initialValue: description,
  style: TextStyle(color: Colors.black, fontSize: 18),
  decoration: InputDecoration(
    border: InputBorder.none,
    hintText: "Write your notes here!",
    hintStyle: TextStyle(color: Color(0xff2c2a2a)),
  ), // InputDecoration
  validator: (title) => title != null && title.isEmpty
    ? "You must have a description!"
    : null,
  onChanged: onChangedDescription,
); // TextFormField

Widget mathWriter() => MathField(
  // No parameters are required.
  keyboardType: MathKeyboardType.expression, // Specify the keyboard type (expression or number only).
  variables: const ['x', 'y', 'z'], // Specify the variables the user can use (only in expression mode).
  decoration: const InputDecoration(), // Decorate the input field using the familiar InputDecoration.
  onChanged: (String value) {}, // Respond to changes in the input field.
  onSubmitted: (String value) {}, // Respond to the user submitting their input.
  autofocus: true, // Enable or disable autofocus of the input field.
); // MathField

```

Figure 7. Screenshot of code 3

Some pros to having math annotations in a note-taking app is improved accuracy, reducing the possibility of error that can happen with handwritten calculations. It can categorize and tag their notes, making it easier to locate specific formulas or equations when needed. This increased organization can save users time and improve their workflow. One more benefit is that it can eliminate the need for other specialized software for mathematical equations. It can enable users to create interactive learning experiences. With a note-taking app that supports math annotations, users can create and share their notes with ease. As such, math annotations can be a valuable tool for anyone who needs to work with mathematical equations and formulas.

## 4. EXPERIMENT

### 4.1. Experiment 1

A concern we had for the app is displaying data correctly when users add too many notes onto their screen. Given different applications have different dimensions, it might enlarge the icons or display more notes.

We will conduct the experiment by trying the application on different devices. To simplify the process we will download a set of different emulators for Android and iOS phones. We will take pictures and take notes of whether or not all notes are displayed in a 2-column view. We hope to visualize the errors that can potentially arise from having too many notes (20 in this case) and notes with 1000+ characters each. Characters will be composed exclusively of those found in the ASCII table. We will make an SQL file to feed it into each device we test it for. We decided to try it on 6 diversely sized Android and 3 of the more recent iOS devices respectively. Given iOS releases new phones less frequently and various Android devices have similar screen sizes, we believe this would be an acceptable threshold which we can use to adequately determine accuracy [10].

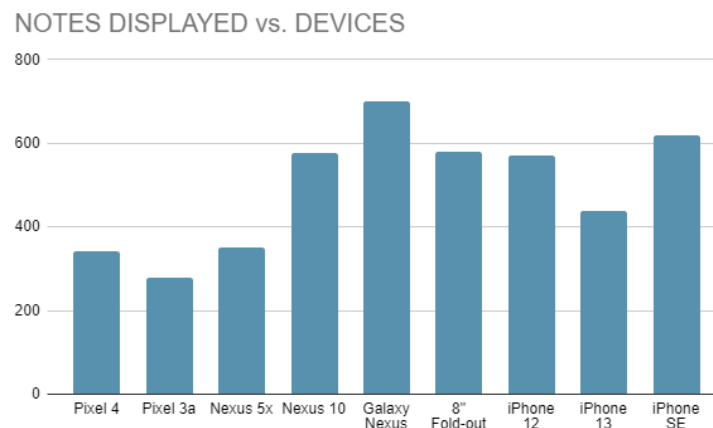


Figure 8. Notes displayed vs. devices

Figure 8 displays the number of notes multiple smartphone devices compare the amount of notes they can display given their dimension and local memory. The blue columns represent the amount of notes each smartphone can represent. The lowest amount was linked to the Pixel 3a for a total of 280, which is to be expected as it is one of the older models of smartphones from the given selection. The highest one was the Galaxy Nexus, which given the screen size and increased technological and processing capabilities demonstrate its superior quality overall [11]. The median for this data was 572 for the iPhone 12 model. The mean of the notes displayed was 495.56, which is close to the limit we wished to aim for, but will reduce to have it below the minimum of 280. We believe the current results are a mixture of local memory and data processing capabilities. Fortunately this sets a precedent for the limits of such systems and can help improve the app development in the case of future updates.

## 4.2. Experiment 2

Another potential concern with the app is the limit behind locally stored data for devices which could have performance or space issues. It could crash the program, make it inaccessible or lose data potentially as well. Experiment 2 targets this issue by feeding SQL files with filler data separated onto note files of increasing character quantities. Similarly to Experiment 1, characters will be composed exclusively of those found in the ASCII table. The files individually contain data into notes of different sizes as follows: file 1: 1 note with 1000 characters, file 2: 2 notes with 1000/2000 characters, file 3: 3 note with 1000/2000/3000 characters ... file 100 with 100 notes 1000/2000/.../99000,100000 characters. We will feed the files onto a single Emulator for the Android. We decided to run it on the Pixel 3A (API 33 x86\_64) with a heap size of 256 and ram size of 1536 because it has similar statistics to various basic Android phones. The data will



be fed to it using the files mentioned above one by one. We will register load times for all files or up until the emulator presents a problem such as crashing. We took the load time by adding a second-counting chronometer via code that starts when the program opens the SQL file and finishes when it attempts to display the first note. The Load Time columns are rounded down to whole numbers in seconds.

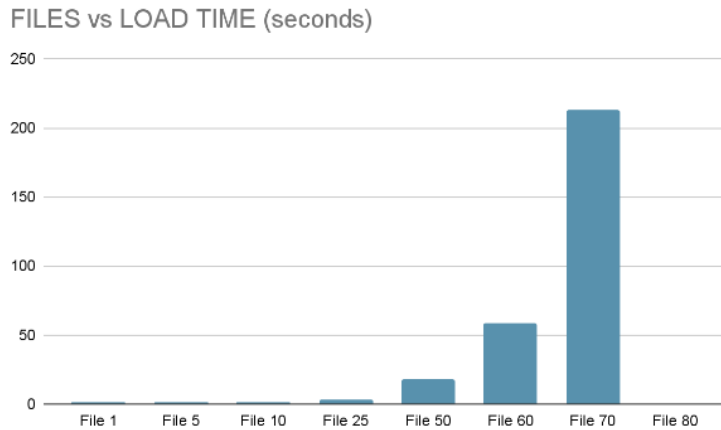


Figure 9. Files vs. Load time (seconds)

Figure 9 demonstrated expected results unlike Graph 1. Since the input of files and characters is exponential, the graph was likely to demonstrate a similar pattern. The blue columns represent the load time for the Pixel 3A smart phone before giving control to the user. The median of the data was 3 suggesting that using the app for general purpose should provide a respectable loading time and as such make the app efficient for fast and conventional use [12]. On the other hand, the mean of 42.3 was skewed mostly due to the drastic quantity and change given between files 25 and 50 being loaded into memory. Non-ideal yet surprising to know the app is capable of handling such a large dataset thanks to the SQL information handling. It should hold up to the general population's patience [13].

## 5. RELATED WORK

Researchers Aryobarzan, Christian, Jean and Steffen found that tablets are the main source of students' scholarly organization, especially when it comes to note-taking [14]. They argue it is a much better approach to that of traditional pen and paper note-taking since said digital devices also allow them to use semantic searches. This refers to automatic word-search, editability, and textual format (such as color, font and size). It refers to other devices as well, but does not go in depth on smart-phones in particular as our app does. Their popularity and portability make them an excellent medium of communication and tools, namely apps such as note-takers [14]. Albeit all other electronics are reliable options, smartphones present more benefits in a fast paced scenario such as a school or meeting.

A development and research team of Dave et al., center their attention on digital technology within a classroom setting and how to incorporate it to its environment in a positive way [15]. Their main approach is focused on note-making to take effective notes for studying newly learnt material. As such, they took their findings to make their own application taking the data collected into consideration. They built an app with various functions and tools that can be useful and interactive for users, although they do not take into account the system limits. Simplicity is a strong tool and one our app strived for. Failing to test edge cases can render the program hindering the user experience and hence conflicting with one of the apps main goals.

Researchers Dominik, Regina, Laura and Michael investigate collaborative note taking and the impact it leaves on educators teaching methods [16]. They discuss how notes are not only limited to text, but recordings and images as well to encourage a more versatile experience for individuals and maximize technological use since many devices possess cameras as well. Sharing is an important feature to aid in co-study as well. Our app aims for simplicity, speed and reliability. Introducing more data formats can fill-up memory or may depend on an online connection [16]. Our app does not require nor utilize the aforementioned for those same reasons.

## 6. CONCLUSIONS

This mobile application finds itself with a couple of limitations given the design goals. It prioritizes data security, efficiency and reliability. It provides security by making the data local to the smartphone device. Databases are commonly accessed through an online source to communicate and retrieve such data which can be in danger of being intercepted by a third party. It is efficient as the SQL declarative language can query as well as organize the information stored very quickly. In addition to the prior two points mentioned, it is reliable because, as our experiments suggest, it can effectively manage data in a way the user has complete control. It is easy to retrieve said data from it too since it's locally stored. Given design and time constraints the app didn't get some additional features which could have made it better. Namely saving images, recordings or audio. We would have had to implement additional storage methods for such data. We would have also liked to include visual options such as: fonts, colors, highlighting and underlining.

To solve said limitations for the application, we would investigate different data management libraries and methods, or compress it in order to maintain the main design goals. In the end, we would still like to focus on the user experience and the program's reliability to maintain its uniqueness in comparison to other similar applications.

## REFERENCES

- [1] Gold, Carl A., and Andrew E. Budson. "Memory loss in Alzheimer's disease: implications for development of therapeutics." *Expert review of neurotherapeutics* 8.12 (2008): 1879-1891.
- [2] Moulds, Michelle L., and Richard A. Bryant. "Directed forgetting in acute stress disorder." *Journal of Abnormal Psychology* 111.1 (2002): 175.
- [3] Dance, Jeffery W. "The Attractive Nuisance: A Model to Prevent Workplace Distractions." *Journal of Multidisciplinary Research (1947-2900)* 5.2 (2013).
- [4] Orlando, Marc. "Digital pen technology and consecutive interpreting: another dimension in notetaking training and assessment." (2010).
- [5] Smith, A. "Smartphone ownership-2013 update. Washington, DC: Pew Research Center." (2013): 1793-5482.
- [6] Reilly, Mark, and Haifeng Shen. "Shared note-taking: a smartphone-based approach to increased student engagement in lectures." *The 11th International Workshop on Collaborative Editing Systems in Conjunction with ACM Conference on Computer Supported Cooperative Work*. 2011.
- [7] Ikhsan, Fathurrahman Kurniawan, Fatimah Faurian, and Aliy Hafiz. "Rancang Bangun Aplikasi Cloud Storage Dengan Angular Dan Firebase Berbasis Android." *Expert: Jurnal Manajemen Sistem Informasi Dan Teknologi* 9.2 (2019).
- [8] Petković, Dušan. "SQL/JSON standard: properties and deficiencies." *Datenbank-Spektrum* 17.3 (2017): 277-287.
- [9] Lee, Hye-Joo, and Jong-Hoon Choe. "GUI Changes of Digital Application with Aesthetic Emphasis; Limited to the Mobile App GUI with the Characteristics of Media Art." *International Journal of Art and Culture Technology* 1.2 (2017): 19-24.

- [10] Lazareska, Lazarela, and Kire Jakimoski. "Analysis of the advantages and Disadvantages of Android and iOS Systems and Converting Applications from Android to iOS Platform and Vice Versa." *American Journal of Software Engineering and Applications* 6.5 (2017): 116-120.
- [11] Khaddage, Ferial, and Gerald Knezek. "Device independent mobile applications for teaching and learning: Challenges, barriers and limitations." *Global Learn. Association for the Advancement of Computing in Education (AACCE)*, 2011.
- [12] Milanovic, Jovica V., and I. A. Hiskens. "Effects of load dynamics on power system damping." *IEEE Transactions on Power Systems* 10.2 (1995): 1022-1028.
- [13] Willocx, Michiel, Jan Vossaert, and Vincent Naessens. "A quantitative assessment of performance in mobile app development tools." *2015 IEEE international conference on mobile services. IEEE*, 2015.
- [14] Atashpendar, Aryobarzan, et al. "Semantic and Interactive Search in an Advanced Note-Taking App for Learning Material." *Learning and Collaboration Technologies. Designing the Learner and Teacher Experience: 9th International Conference, LCT 2022, Held as Part of the 24th HCI International Conference, HCII 2022, Virtual Event, June 26–July 1, 2022, Proceedings, Part I. Cham: Springer International Publishing*, 2022.
- [15] Towey, Dave, et al. "Researching and supporting student note-taking: Building a multimedia note-taking app." *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE). IEEE*, 2015.
- [16] Petko, Dominik, et al. "Metapholio: A mobile app for supporting collaborative note taking and reflection in teacher education." *Technology, Knowledge and Learning* 24 (2019): 699-710.
- [17] Li, Shasha, and Chien-Hsiung Chen. "The effects of visual feedback designs on long wait time of mobile application user interface." *Interacting with Computers* 31.1 (2019): 1-12.
- [18] Hynes, Mike. "The smartphone: A weapon of mass distraction." *The social, cultural and environmental costs of hyper-connectivity: Sleeping through the revolution. Emerald Publishing Limited*, 2021. 71-84.