Bilkent University

Department of Computer Engineering

Senior Design Project

Nino: Nino is not OCR

Final Report

Ata Deniz Aydın, Ecem İlgün, Ergün Batuhan Kaynak, Selim Fırat Yılmaz

Supervisor: Hamdi Dibeklioğlu

Jury Members: A. Ercüment Çiçek and Özcan Öztürk

Final Report
May 9, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1 Introduction

Note taking amounts to an important part of studying materials for many students, be they grade school or graduate students. In many cases, there are no lecture notes or slides readily available, or the ones that are available do not cover all of the lecture content. This requires students to take notes during the lecture.

However, in some cases, the student has to note down the words fast without truly processing their content, or write down equations without having the time to analyze and thus understand them. In the case of writing down sentences, one can achieve faster note taking with tablets/computers [1] but when the lecture notes consist of plots, graphs or mathematical equations, our market research shows that there is no hardware/software help to achieve faster note taking than writing it down [2].

In some cases, scribing all the things written on the board (even without making sense of it) proves to be quite hard, therefore many note taking strategies have arisen [3]. Many of these strategies revolve around the rule that one first takes notes of keywords, and then rewrites the notes after class [3]. However, there are two possible problems with this strategy: the student might forget the information conveyed via that particular keyword, or might miss important information and not write a keyword about it.

Currently, EverNote™, OneNote™ and Google Docs™ offer OCR (optical character recognition) features in which an image is converted to a text. However, these conversions do not cover mathematical equations or figures. As it is discussed above, many courses include some kind of figures or mathematics, therefore capturing only text with OCR might lead to loss of important information. Hence, current software are not capable of providing a complete note taking system.

We thus propose an app that will convert handwritten and printed notes into an editable format which can be exported as LaTeX or PDF. These notes will preserve the alignment of the original picture, in either landscape or portrait format. The student will be able to add his/her own notes on top of the ones extracted from the image. In this way, we hope to engineer a note taking system that would make the lecture not only more interactive for the student, but will also let the student listen to more of the lecture.

2 Overview

Nino (Nino Is Not OCR) is an Android application, targeted primarily for tablets, that can detect text, equations and figures in pictures taken of notes on paper or a whiteboard.

The user can take pictures directly within the application or load preexisting pictures from the device. After the program analyzes the picture, it may present annotations on the picture and store them so that the user can later view and search through them. The application consists of a client Android and web interface for end users, as well as a back-end server application for data storage and processing.

In the client application, the user is first presented with a login screen where the user may register or log in. After logging in the user may either add a new picture to their library, whether by camera or from the device, or view previously annotated pictures. After a picture is taken it is sent to the server for processing. The server first segments the image into regions containing text, equations, graphs or other figures, parse the text or equations (in LaTeX) contained in each region, and then process the text and equations in order to provide helpful hyperlinks to the user. For example, the program may link a concept to an online resource such as the Wikipedia page for the concept which the user would be able to open without leaving the application. The program can also process the text in the document to summarize its content and tag it with keywords in order to enable more comprehensive searches.

After processing, the server stores the annotated picture in the cloud storage reserved for the user and sends it back to the client application. The client application then displays the notes overlaid on the image in an interactive interface where the user may click on links, rearrange or modify the notes as desired. The user may also share the annotated picture either directly as an image or as a LaTeXor PDF file.

The notes the user has uploaded are collected within the personal storage space allotted to the user, so that the user can view and modify them similarly, search through them, and categorize them e.g. with respect to course title and subject so as to assist the program in improving its tagging and summarization of notes.

3 Final Architecture & Design

The large-scale architecture of the application follows the client-server model. The client side is in the form of an Android application that interacts with a single user, allowing the user to capture images and send them to the server to be converted to a note, or view or edit already created notes. The server side handles most of the data processing necessary to convert images into notes, such as text detection, recognition and analysis, as well as maintaining the data stored for each user including notes stored in the cloud and authentication information.

3.1 Client

The client is the user interface layer of Nino, where the user interacts with notes to create and edit them. The client contains the presentation and controller subsystems. The presentation subsystem contains user interface elements for the user to interact with. The controller subsystem is responsible of creating appropriate requests depending on the events initiated by the views and then communicating with the server to send the requests. The presentation is itself broken into different modules for authentication, viewing and editing notes and viewing the user profile.

Along with custom created layouts and Android activities, Nino uses an open source note taking project, Scarlet-Notes [16], to enhance UI presentation and logic. Nino's activities and logic are connected to Scarlet-Notes, and most of the functionality is changed to fit Nino's use case. The client uses additional libraries to capture and process images as well as rendering Markdown text, which is stored in notes.

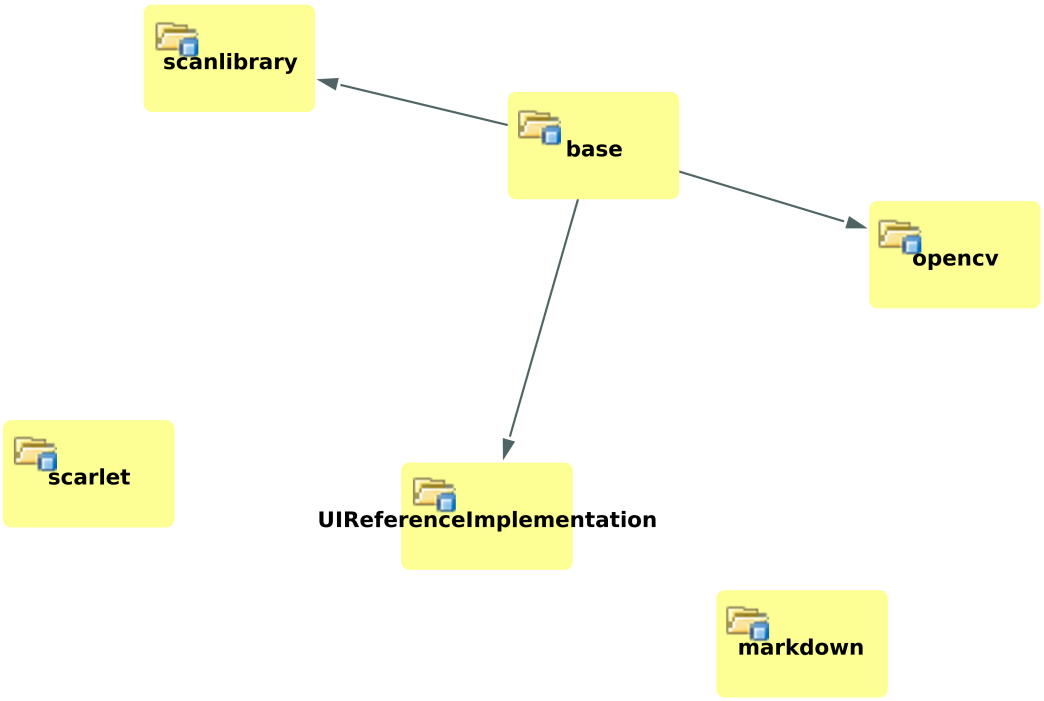

Figure 1: Overview of client and external libraries used.

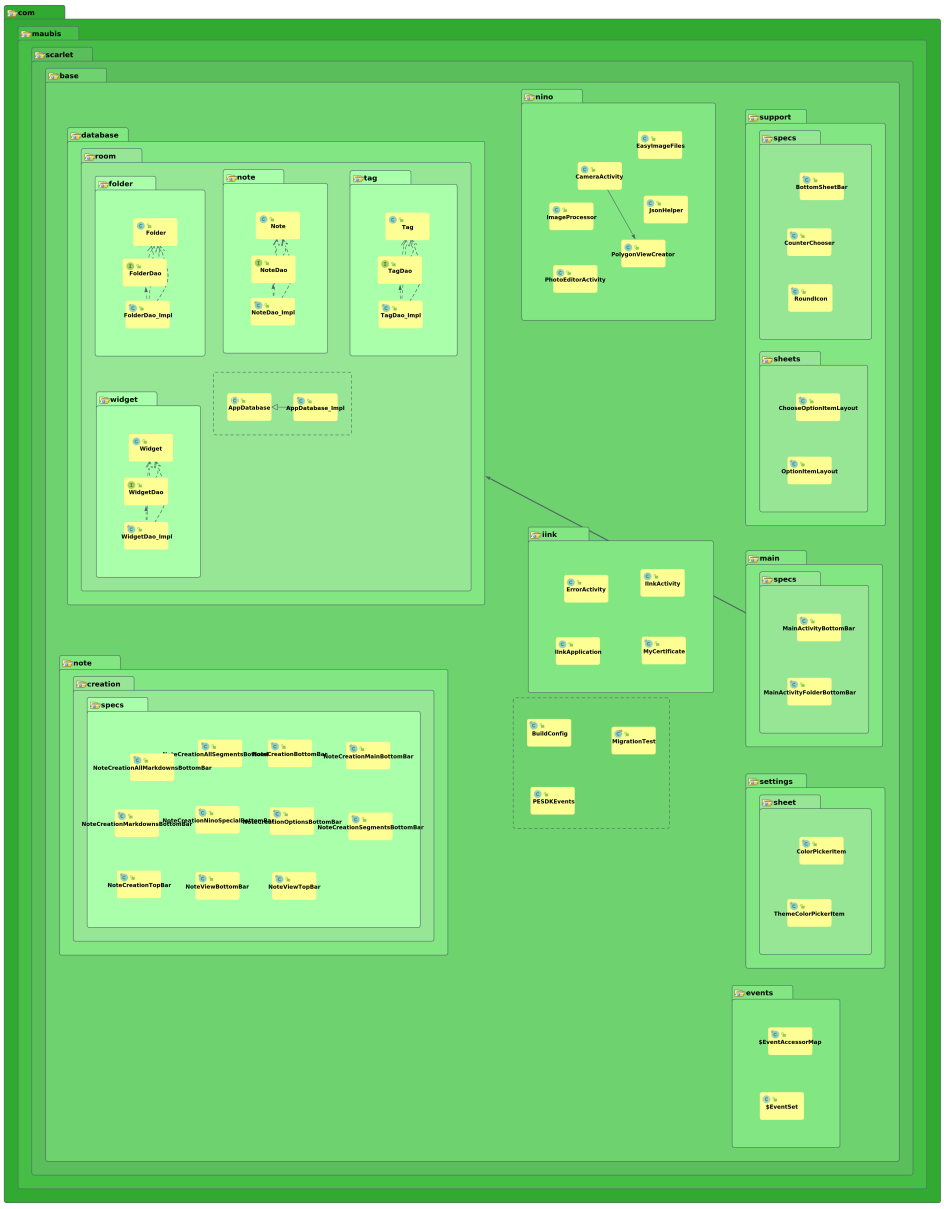

Figure 2: The partial* hierarchy of modules inside the client system. *: No automation to generate UML diagrams of Kotlin classes exists at this point, only java classes are shown.

3.2 Server

An image and the required operations on that image arrive as a request to the server. The server then processes the image with the given modules and creates an output. The output is then sent back to the client as a response. In addition, the server also handles authentication and updates to user information database. The server is separated into the Logic and Data layers. The request handling is also inside the server side of the project, and is maintained by the REST API and Django [6].

The logic layer governs the database operations and the note processing pipeline. User management also maintains a list of active users, handles invocations to log in, log out or create a new account, and authorizes accesses or modifications to the storage of each user. Lastly, note processing receives an image, processes it and converts it into a note, or receives and processes existing notes either for the purpose of training the models used or annotating notes again after editing.

The data layer is where the models and other persistent information are kept. Django is also used to manage the database to store for each user the storage space of notes reserved for them, as well as authentication data and shared data used in note processing and other configuration data.

When an image is received in a POST request to create a note, the server first saves the image and registers it in the database, and then sends it to the ABBYY FineReader OCR API [17], from which it receives lists of bounding boxes for lines and paragraphs of text as well as for images, stored in a dictionary data structure. The bounding boxes are further sent to the Mathpix API [18] to be further distinguished into equations, plots, figures and tables, and then the images detected are sent to the Google Cloud Vision API [19] in order to label them and suggest similar images. Other POST requests receive a JSON object from the client storing the digitized and possibly edited version of the note, in order to analyze the text in the note to recognize keywords and generate possible test questions from it, or export the note into LaTeX or PDF format.

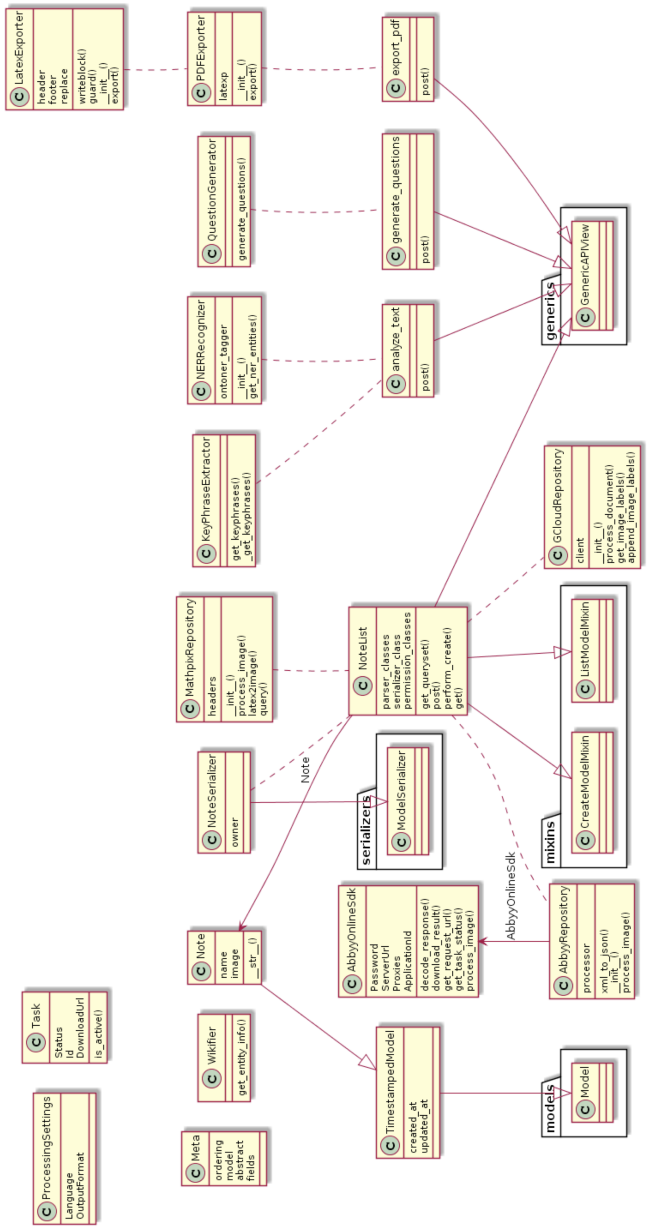

Figure 3: Server UML Diagram.

4 Algorithms & Methods Used

4.1 Text recognition

For the implementation of server-side text recognition, we had surveyed and tested various readily available OCR libraries as well as trainable neural network models, which however only worked on lines of text and required a prior segmentation algorithm. The ABBYY FineReader SDK [17] was the only library we found that simultaneously segmented a document image into lines of text and boxes containing non-textual figures. The Google Cloud Vision API [19] could also detect text, equations and images in documents. However, the API only returned bounding boxes for each individual character or word, and needed to be processed further to be brought into a format usable for further layers of processing.

On the Im2LaTeX-100K dataset intended for OpenAI's im2latex (image to LaTeX) task [20], which provided a collection of PDF documents of articles in theoretical physics along with their LaTeX source code, ABBYY was able to recognize lines of text almost perfectly including some punctuation, while lines containing mathematical expressions were misread as containing symbols such as currency signs or paragraph breaks. However, standalone lines of equations or other figures were nevertheless mostly detected as lines of text or boxes containing images. On 20 pages selected, 10 of which were predominantly textual and the other 10 containing equations or other figures, the edit distance between the original text and the output given by ABBYY yielded an average error rate of 6% for pure text and 43% for documents containing mathematical expressions. The average response time of the ABBYY SDK, for a document image of resolution 300 pixels per inch, was approximately 17 seconds on the server.

4.2 Mathematical expression recognition

For the task of offline equation recognition, we have likewise only encountered the Mathpix API [18] as a standalone product for recognizing the mathematical expression given in a predetermined region of text, and while there were models for recognizing mathematical characters and combining them into expressions using stochastic grammars, we could not find a method to detect regions of mathematical expressions from a document image. Hence, the bounding boxes returned by ABBYY are used to identify possible regions of

equations, and both the lines of text and the regions of images identified by ABBYY are sent individually to the Mathpix API to identify possible expressions ingrained in the text or standalone equations, where lines identified by ABBYY as containing only text are skipped for efficiency. The API is also able to recognize text in a line as well as detecting whether an image given is a plot, a geometrical figure or a table, but in some cases the API may only recognize the equation in the image while returning a low confidence value. To avoid this, results of confidence less than 80% are omitted and the text given by ABBYY is used instead. For clear, high-resolution document images, the API is able to improve on the performance of ABBYY greatly on documents containing equations, while some standalone equations were not detected at all by ABBYY hence could not be recognized by Mathpix. It was however not possible to quantify the improvement of accuracy using edit distance since the LaTeX or PDF generated by the program is of a different format than the files given in the dataset, which was not the case for plain text. After processing by the ABBYY SDK, processing each box returned by ABBYY through Mathpix takes approximately 13 seconds on average, when requests are sent to Mathpix in parallel with a pool of 10 requests.

5 Impact of Engineering Solutions

5.1 Global Impact

Nino is aimed to be a note taking aid with its main target users as students. Note taking is crucial for courses if electronic resources provided by the faculty does not cover all course material. The notes students take often form the basis of their study material.

A quick search from Youtube yields that video reviews of traditional note taking methodologies are more popular than video reviews of note taking apps [14]. It may indicate that note writing is used more than digital note taking systems. Our group assumes that a student would favor the easier and more accurate solution, and current note taking apps with OCR lacks just that. Current apps which offer OCR as a feature are designed to recognize only texts. Our key observation is A lecture students would like to take notes of might involve some form of information other than pure text. This is especially true for STEM students, since their courses involve mathematics or figures in one way or another. Nino aims to be first in its market to solve this problem.

Main innovation of Nino is exploring aforementioned area in 'digital note taking' market that is not yet explored by any famous note taking app such as Evernote. We believe that a note taking app that is easy to use and accurate has the chance to succeed writing notes.

If our system is developed further, it will offer a third option in taking notes besides writing and typing them. Since it is less time consuming then both of the other options, it may revolutionize the way students take notes.

It can also lead digital note taking market to incorporate recognizing information other than text into their OCR. In its current state, OCR feature is seen as a cool prototype feature to add in a note taking app. however it is not actively used by people due to its insufficiency in recognizing many information forms. Current note taking apps are not developing their automatic note taking features. Nino may lead the way in growing these features to be more than prototypes.

5.2 Social Impact

Trying to scribe every piece of information provided in a lecture may cause students to not pay attention into listening that lecture. Therefore, oftentimes students have to choose between note taking during the lecture then revising after the lecture, or paying full attention during the lecture then trying to find study material in another way.

Nino provides students with an effortless way to keep notes of a lecture. All they need to do is taking photos of lecture notes or the board, and Nino will convert them into an editable digital format. Therefore, they now have a third option of paying full attention into the lecture and letting Nino do the note taking part.

Providing a new and easier way to gather information can impact the way people obtain and process knowledge beyond our grasp. Just as the invention of writing provided an easy way to obtain information and dominated the way we now handle any information, automated note taking may also revolutionize learning in students. Obviously the idea to automate note taking does not belong to us. However, our application aims to be the first to provide a system that shows a note taking app can do more than recognizing texts and misrecognizing anything else resembling texts. Nino aims to be the first note taking app to provide a system that can be actually used in lectures.

Using Nino can also affect the way a lecture is handled. It is expected trivialize the

time it takes to take notes, therefore a lecturer will not need to wait for students to finish writing. Since Nino provides notes in editable format, students can take notes on their lecture notes such as summarizing the material. This allows them have time to process the information during the lecture, rather than relying on revising the notes when they get home. Our app also aims to increase interaction between a lecturer and the students, since the students will not be constantly busy with writing.

Furthermore, there are many side use cases of Nino. If further developed based on its accuracy on handwriting, it can also help educators digitize and distribute their previously handwritten material. Researchers can also use Nino to digitize and take notes on a conference presentation.

As described above, there are many social and especially educational situations that Nino offers a new approach, impacting the path these situations take.

5.3 Environmental Impact

Just as e readers provide an environmentally friendly alternative to printed books, Nino can provide an environmentally friendly alternative to using paper.

6 Related Contemporary Issues

This section explains some of the issues we have faced during development and the course of action we have taken towards solving them.

6.1 Language Range of the Product

The product is able to recognize languages that use Latin alphabet, the most commonly used writing system [15]. We have decided not to include other alphabets due to time constraints.

6.2 Internet Access

Continuous internet is not essential to view existing notes, however, it is a must to digitize new notes. We have briefly considered moving some of the computation towards client, however, this would be a burden on the Android device. Furthermore, some of the tools we used themselves rely on internet to send our requests. After taking these into consideration, we have decided that requiring internet access for digitizing notes was the only possible choice and keeping the computation in server side was the the best course of action.

6.3 Server Response Time

The more separate texts, figures, mathematical expressions a photo has, the more time it will take server to handle the request to recognize different segments. Therefore, we have chosen to parallelize some parts of the request to decrease server response time. We have also decided to stick with the core features to decrease the response time.

6.4 Authorship Issues

Since digitizing notes without the authors consent might cause authorship issues, any user who uses Nino is assumed to have accepted responsibility over the photographs they take.

6.5 Data Privacy Issues

We have decided to provide users with login so that they can save their data or reach it from another device. We are also not collecting any unnecessary data for the sake of users privacy. Our app needs to reach the devices camera to take photos and to reach devices folders to retrieve any past photo. However, we do not reach the camera or folders unless user specifically takes this action. We also do not use these data in any other way than digitizing the specific photo they choose to provide us with.

7 Tools, Technologies & Resources Used

- GitHub was used for version control. The client side mobile application was developed in Android Studio.
- For server, user and database management, the REST API and the Django framework in Python are used. [TODO server site]
- Image processing is handled using the OpenCV library in the client, and PIL (Python Image Library) in the server.
- Markdown is the format used to store and process editable text for the client and the server.
- The user interface for editing notes is built on top of the open-source Scarlet note-taking app on Android [16].
- The ABBYY FineReader SDK [17] is used to identify and recognize regions of text and images, and the Mathpix API [18] is used to further distinguish them into equations, figures, plots or tables.
- To summarize images and suggest similar images, the Google Cloud Vision API [19] is used.
- LaTeX files are generated from boxes of text and equations using the Textpos package for absolute positioning of text, and the command-line tools pdflatex and latexmk to convert them to PDF files.

8 Possible Future Development

The app is currently in a prototype state due to time and staff constraints. Therefore if it grows into become a professional endeavour, we will have more time to put into development of the project and increasing its accuracy as well. Assuming the project is in the hands of a larger company involving a research and development team, the research effort may be also given to making ML/DL models that are more suitable to the app's nature.

Other than that, reinforcement learning and crowdsourcing can be adapted in order to increase accuracy with a different approach.

9 Glossary

Note: a handwritten picture containing regions of text, equations, plots and figures.

Sketch: any digitizable type of data apart from text and equations.

Segment: a maximal rectangular region in a note consisting of a single type of data (text, equations or sketches), enclosed by a bounding box.

Category: a directory containing notes of the same context (e.g. course topic).

Client: the mobile or web application interacting with the user.

Server: the system communicating with each client and processing images sent from them.

Segmentation: partitioning a given image into different segments.

Detection: identifying segments of a particular type (e.g. text segments) without necessarily recognizing the data stored in the segment.

Recognition: identifying the content in a particular segment (e.g. recognizing the text written in a given text region).

Annotation: improving and adding to the content stored in each segment, e.g. identifying keywords in text and adding hyperlinks to keywords.

10 Appendix

10.1 User Manual

In this appendix, a user manual is presented for a better understanding of Nino application.

10.1.1 Default Home Page

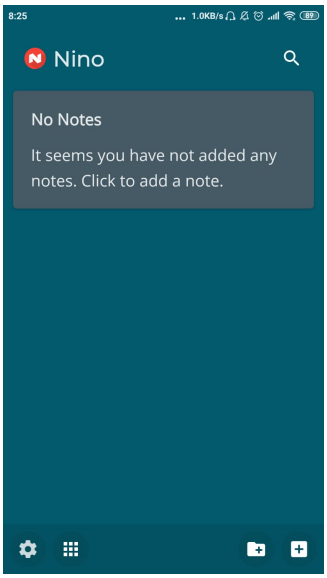

Figure 4: Default Home Page

Any user without login is first redirected into this homepage. A user without login can still use Nino the same way as a saved user would, but will not be able to back up their files or retain their files from another device.

If users click on the rectangle which encircles the paragraph starting with "No Notes", they will proceed to the "Edit Note Page"s first Figure.

If users click on the "Magnifying Glass Icon" on the top right of the page to search through Notes and Notebooks they will proceed to "Search Page" to enter their query.

Furthermore, users can click on the buttons on the button for many more actions. If

users click on the "Gear" icon on the bottom left of the page, they will proceed to "Options and Settings Slide Bar". If they click on the "9 squares" icon on the bottom left of the page, they will proceed to "Main Actions Pop Up".

There are two more interactive icons on the bottom right of the page. If the users click on "Folder" icon with a plus sign, they will be prompted to enter a name and color to create a new Folder. If the users click on "Rectangle" icon with a plus sign, they will proceed to "Edit Note Page".

10.1.2 Home Page with an Option to Sign In

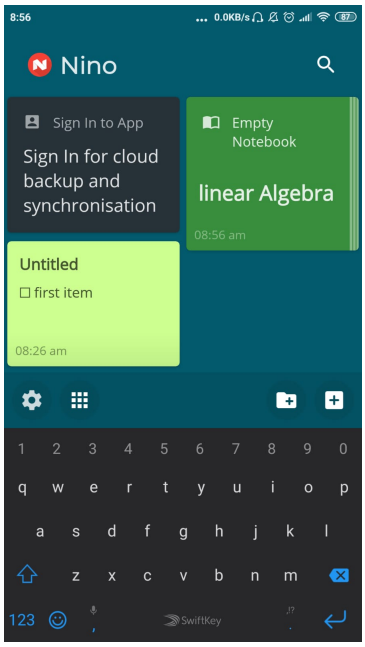

Figure 5: Home Page with an Option to Sign In (1)

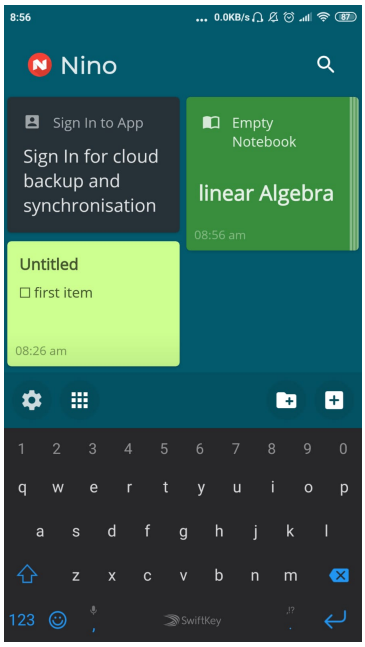

Figure 6: Home Page with an Option to Sign In (2)

After users have added at least a Note or a Notebook, the home page changes into Figure 7, where they are prompted with an option to Sign In and save their data through the top-left rectangle in the screen. This home page provides all the interaction provided with the Default "Home Page" and three more: signing in, editing a note, viewing a notebook.

If they click on the "Sign In" rectangle, the users proceed to "Sign In Page". If they choose to click on a Note or a Notebook instead, they will proceed to "View Note Page" or "View Notebook Page" respectively.

10.1.3 Search Page

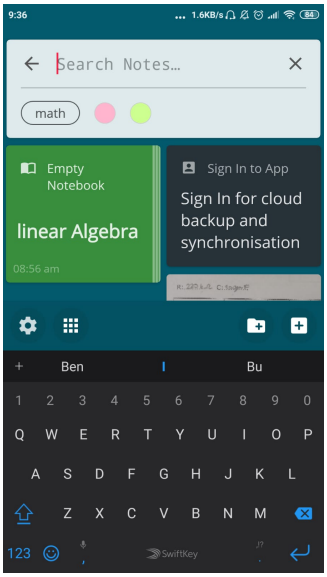

Figure 7: Search Page

Users enter query after clicking and typing on "Search Notes.." bar. Then the page will display any Note with associated Title, Notebook or Tag matching with the query. Clicking any results will result in "View Note Page".

10.1.4 Options and Settings Slide Bar

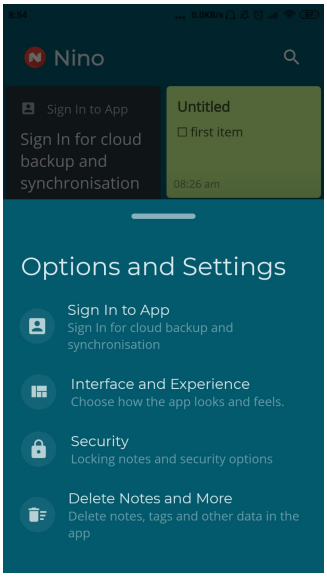

Figure 8: Options and Settings Slide Bar

Users are directed to this pop up after they click on the "Gear" icon.

Here they can click on any of the four buttons to continue.

They may choose to click "Sign in to App", which proceeds them to "Sign In Page".

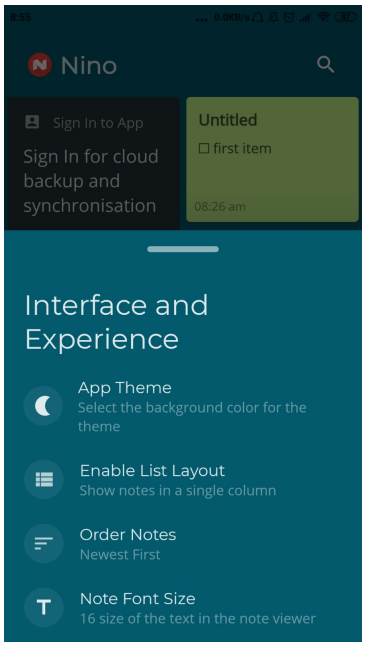

Figure 9: Interface and Experience Slide Bar

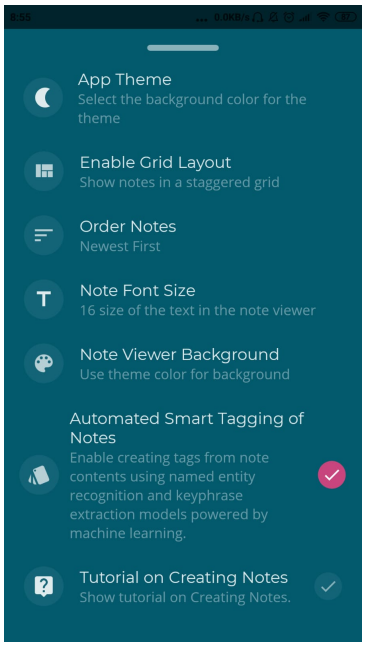

Figure 10: Interface and Experience Slide Bar (full view)

If users click on "Interface and Experience" button, "Interface and Experience Slide Bar" shows up. Here, the user can choose to alter many settings:

App Theme Selects background color of the app

Enable List/Grid Layout Shows notes as grid or a single column list

Order Notes Allows user to sort notes by either most recently modified, newest first, oldest first or alphabetical

Note Viewer Background Toggles use of theme or note color for background

Automated Smart Tagging of Notes Enables creating tags from note contents using Named Entity Recognition and Keyphrase Extraction

Tutorial on Creating Notes Enables/disables tutorial.

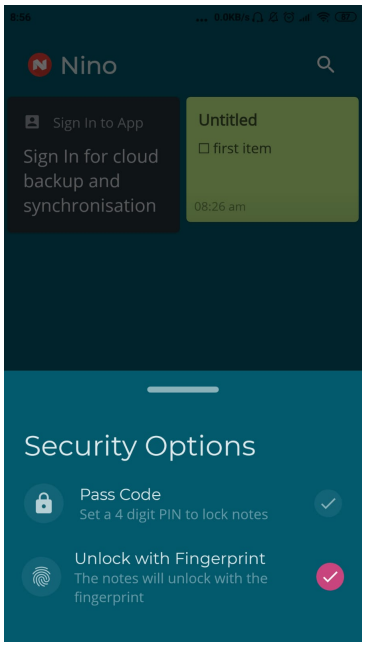

Figure 11: Security Options Slide Bar

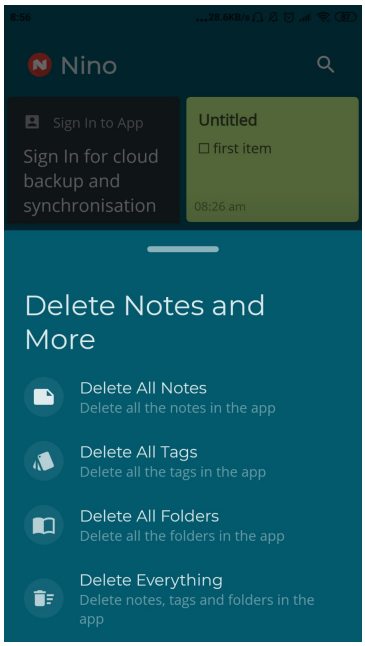

Figure 12: Delete Notes and More Slide Bar

If users click on "Security" button, "Security Options Slide Bar" shows up. Similar to "Interface and Experience Slide Bar", users may choose one of the two options:

Pass Code Set 4 digit PIN to lock notes

Unlock with Fingerprint Allows notes to unlock with fingerprint

Users may also click on "Delete Notes and More" button, which prompts "Delete Notes and More Slide Bar". Similar to previous two instances, users may choose one of the following options:

Delete All Notes Deletes all notes in the app

Delete All Tags Deletes all tags in the app

Delete All Folders Deletes all folders in the app

Delete Everything Deletes all of notes, tags and folders

10.1.5 Sign In Page

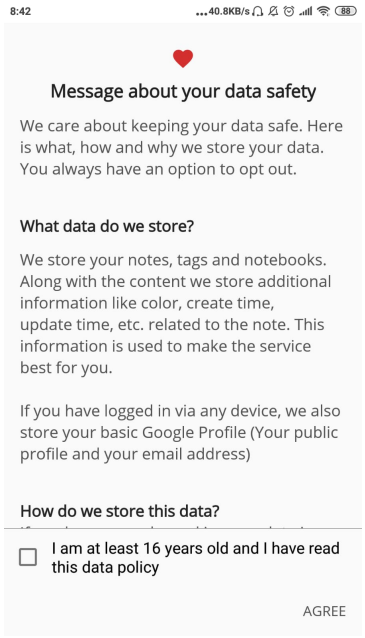

Figure 13: First page of user consent

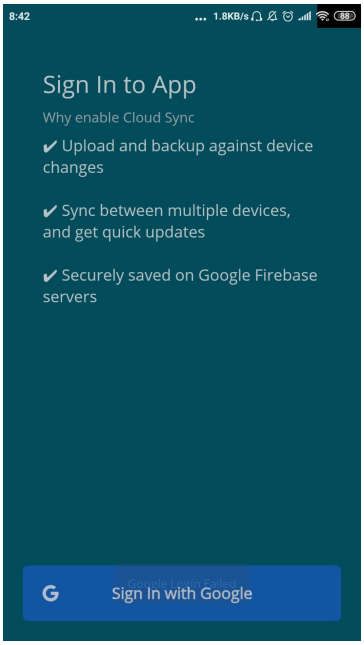

Figure 14: Second Page of Linking to Google Account

Users reach this page after requesting a sign in from either the respective rectangle on "Home Page with an Option to Sign In" or from clicking "Sign in to App" button on "Options and Settings Slide Bar".

The user first has to sign an data safety agreement that is provided by Google. Then they are prompted with another page where they can click on "Sign in with Google" to link their Google account. Google then provides a pop up asking which Google account to link to, same as any other app which uses Google authentication. After users finish signing in, they return back to the page that proceeded them to "Sign in Page".

10.1.6 Main Actions Pop Up

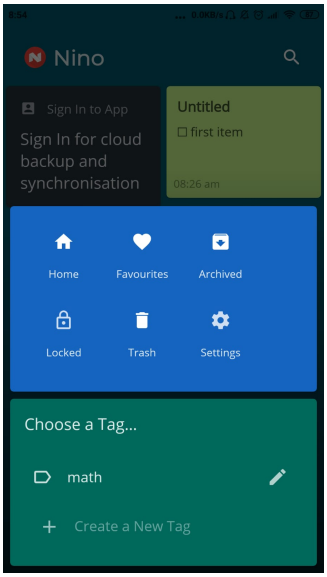

Figure 15: Main Actions Pop Up

After users are prompted with this pop up, they may choose one of the following options:

Home Returns to "Default Home Page" or "Home Page with an Option to Sign In" depending on whether they have made a note or not.

Favourites Displays Notes marked as favorite, if any.

Archived Displays archived Notes, if any.

Locked Displays locked Notes, if any.

Trash Displays Notes moved to Trash, if any.

Settings Redirects to "Options and Setting Slide Bar"

Users may also choose one of the existing tags under "Choose a Tag..." text, which will filter out all other the Notes without the specified tag.

10.1.7 View Note Page

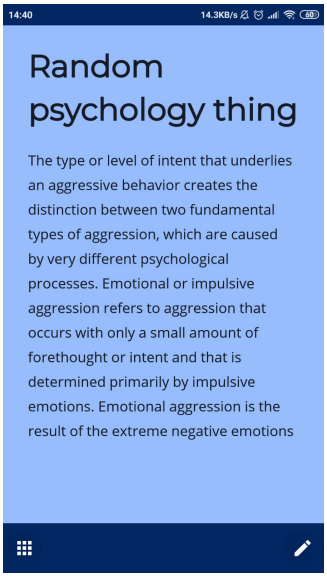

Figure 16: View Note Page

Users reach this page after they click on an existing Note. They can then scroll through the Note or edit the Note by clicking "Edit" button on the bottom right of the page, or they can move forward to "Note Options Page" by clicking "9 squares button" at the bottom right of the page.

10.1.8 Edit Note Page

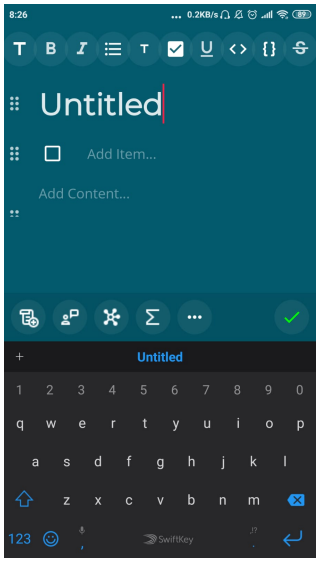

Figure 17: Edit Note First Page

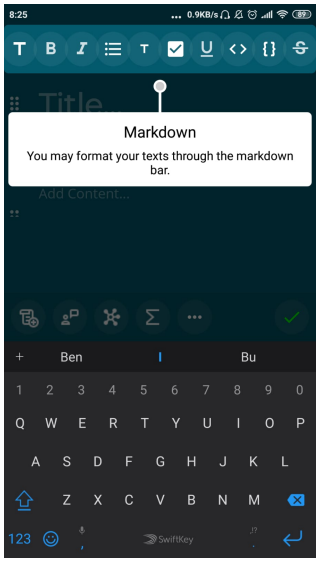

Figure 18: Edit Note Walk-through (1)

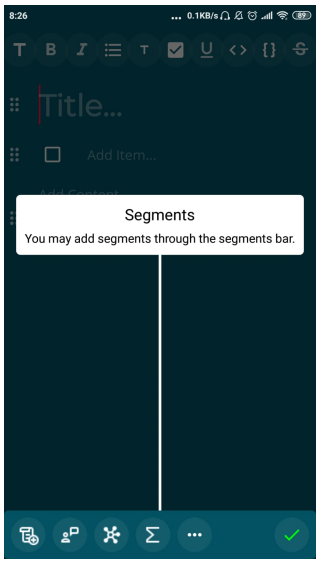

Figure 19: Edit Note Walk-through (2)

Users are prompted to edit Notes in many different ways, therefore we have provided a quick walk-through for beginners. As in the 18 users may format texts through the bar above, and as in 19 they may add new segments through the bar below.

The markdown bar provides user with many different ways to stylize their Notes including bold, italic, hashtags, lists, under-strikes and so on.

The segments bar provides user the ability to add different segments, both through camera and writing on the app. Camera will act as OCR which supports mathematical expressions, and writing on the screen will help users in adding other media that OCR does not support. With the writing on screen users can digitize almost anything including but not limited to diagrams, texts, equations, trees, plots, code etc. Some demonstrations of these are provided below:

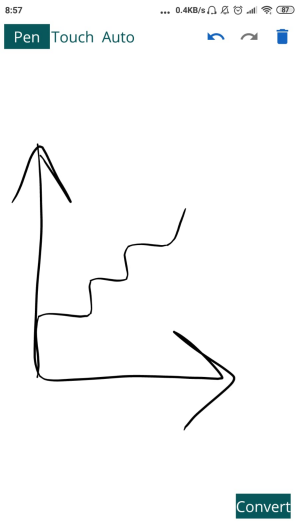

Figure 20: Hand-drawn plot

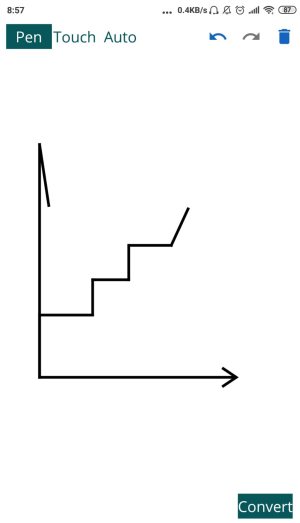

Figure 21: Digitized Plot

Figure 22: Hand-drawn Equation

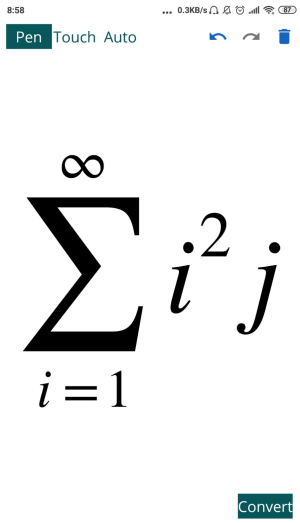

Figure 23: Digitized Equation

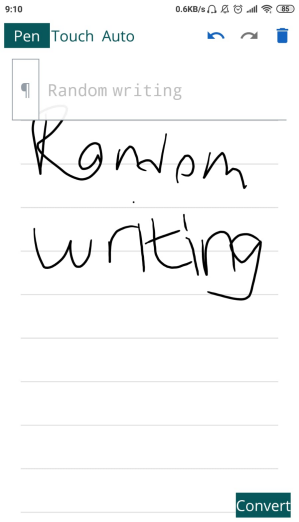

Figure 24: Hand-drawn Text Recognized

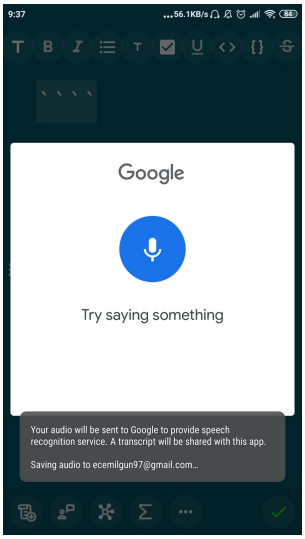

Figure 25: Google speech recognition used in converting spoken language into text

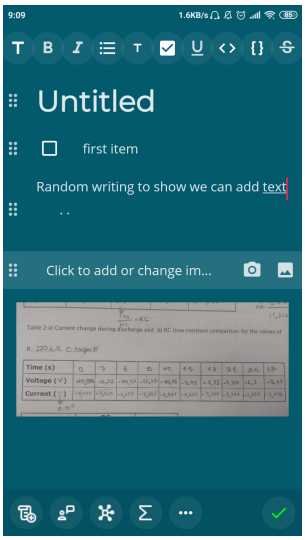

Figure 26: After adding an image via Camera app installed in each contemporary Android device

10.1.9 Note Options Pop Up

Users can reach this pop up either by clicking and holding a specific Note for a few seconds or by clicking on the "9 squares" icon on the bottom left of the "View Note Page".

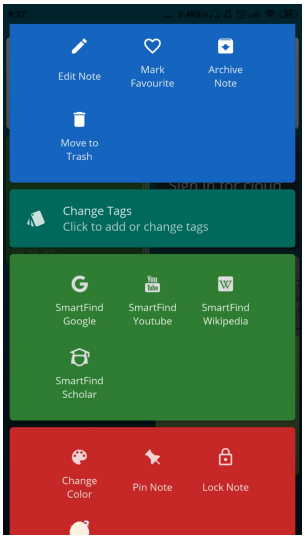

Figure 27: Note Options Pop Up

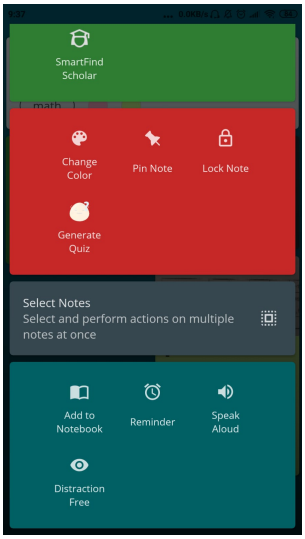

Figure 28: Note Options Pop Up (scrolled down)

After users are prompted with this pop up, similar to other pop ups, they may choose one of the following options:

Edit Note Redirects into "Edit Note Page"

Mark (Not) Favourite Toggles Note's 'Favourite' state.

Archive Note Archives the note such that it will show up when filtered as 'Archived'.

Move to Trash Moves Note to Trash.

Click to add or change tags adds tags or changes preexisting tags.

SmartFind Google/YouTube/Wikipedia/Scholar Redirects to a search of the Note's tags as query input on Google/YouTube/Wikipedia/Google Scholar.

Change Color Changes color assigned to the Note.

Pin Note Pins Note to the top of the list.

Lock Note Locks Note so that it can be accessed after unlocking via preferred security method.

Generate Quiz Generates a quiz based on the summarization and tags of the Note (This feature is still a prototype).

Add to Notebook Adds Note to a Notebook

Reminder Sets a reminder alarm for the Note.

Speak Aloud Converts the texts into sound and speaks it through AI.

Distraction Free Offers a relatively clutter free environment to view the Note.

Demonstrations of SmartFind buttons are provided below:

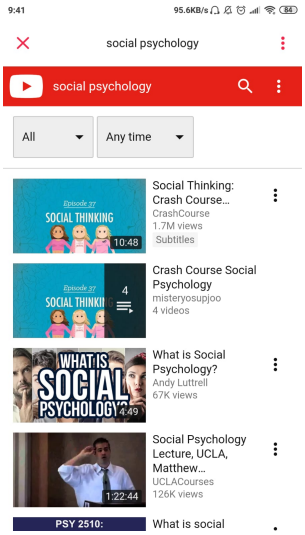

Figure 29: Result of clicking on "SmartFind YouTube" button for a specific Note and tag

Figure 30: Result of clicking on "SmartFind Wikipedia" button for a specific Note and tag

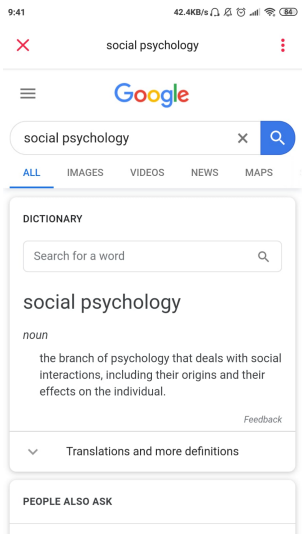

Figure 31: Result of clicking on "SmartFind Google" button for a specific Note and tag

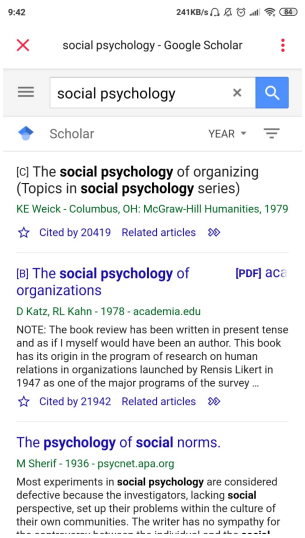

Figure 32: Result of clicking on "SmartFind Scholar" button for a specific Note and tag

10.1.10 Add Notebook Pop Up

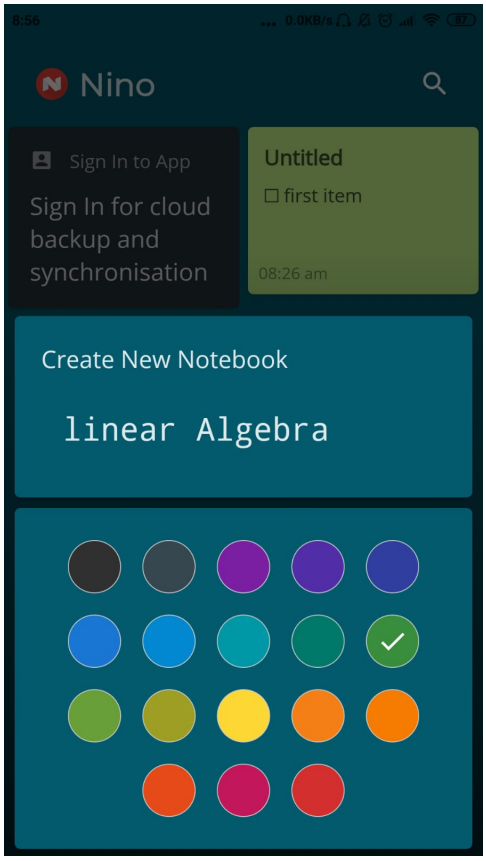

Figure 33: Add Notebook Pop Up

This pop up shows after user clicks on "Folder" icon with a plus sign or chooses to "Create New Notebook" after clicking on "Add to Notebook" button provided in "Note Options Pop Up". In either way the user has to give a name and a color to the Notebook.

10.1.11 View Notebook Page

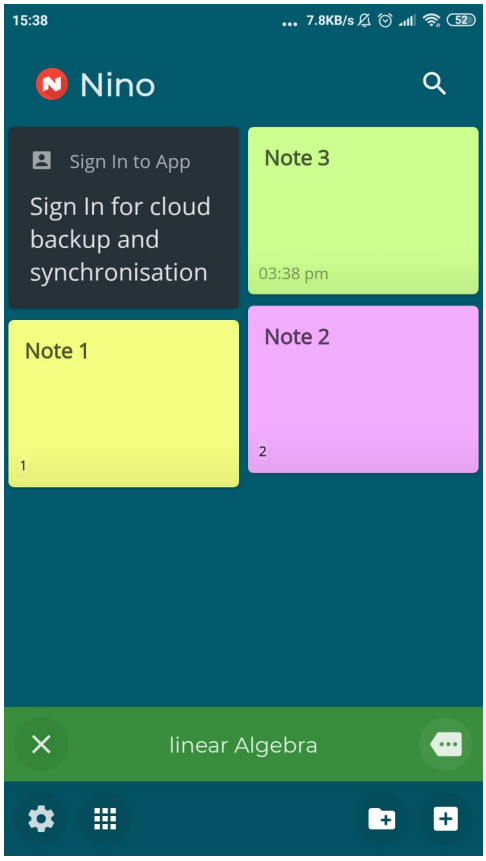

Figure 34: View Notebook Page

Interacting with this page is quite similar to "Home Page with an Option to Sign In". The only difference is that this page is filtered to have Notes that belong to one Notebook.

References

[1] "3 Easy Steps to Digitize Your Study Notes". [Online]. https://www.developgoodhabits.com/digitize-study-notes/. [Accessed Feb 18, 2018].

[2] "How to digitize your handwritten notes". [Online]. https://www.popsci.com/digitize-handwritten-notes. [Accessed Feb 18, 2018].

[3] "How to Take Study Notes: 5 Effective Note Taking Methods". [Online]. https://www.oxfordlearning.com/5-effective-note-taking-methods/. [Accessed Feb 18, 2018].

[4] "The LaTeX Project". [Online]. https://www.latex-project.org/. [Accessed Feb 18, 2018].

[5] "PDF 2.0". [Online]. https://www.iso.org/standard/63534.html. [Accessed Feb 18, 2018].

[6] "The Web framework for perfectionists with deadlines — Django". [Online]. https://www.djangoproject.com/. [Accessed Feb 18, 2018].

[7] "Ubuntu 18.10 (Cosmic Cuttlefish)". [Online]. http://releases.ubuntu.com/18.10/. [Accessed Feb 18, 2018].

[8] "SQLite Home Page". [Online]. https://www.sqlite.org/. [Accessed Feb 18, 2018].

[9] "OAuth 2". [Online]. https://oauth.net/2/. [Accessed Feb 18, 2018].

[10] "Unified Modeling Language". [Online]. http://www.uml.org/. [Accessed Feb 18, 2018]

[11] "IEEE Citation Guidelines". [Online]. https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf. [Accessed Feb 18, 2018].

[12] "AutoCompleteTextView". [Online]. https://developer.android.com/reference/android/widget/AutoCompleteTextView. [Accessed Feb 18, 2018]

[13] "LoaderManager.LoaderCallbacks". [Online].
`https://developer.android.com/reference/android/app/LoaderManager.LoaderCallbacks`.
[Accessed Feb 18, 2018]

[14] "Youtube Search for 'Note Taking' Query". [Online].
`https://www.youtube.com/results?search`$_{q}uery =$
$note + taking.[AccessedMay9, 2019]$.

[15] "Encyclopedia Britannica: Latin Alphabet". [Online].
`https://www.britannica.com/topic/Latin-alphabet`. [Accessed May 9, 2019].

[16] "Scarlet-Notes". [Online]. `https://github.com/BijoySingh/Scarlet-Notes`.
[Accessed May 9, 2019].

[17] "PDF Software with Text Recognition - ABBYY FineReader 14". [Online].
`https://www.abbyy.com/en-eu/finereader/`. [Accessed May 9, 2019].

[18] "Mathpix OCR". [Online]. `https://mathpix.com/ocr`. [Accessed May 9, 2019].

[19] "Vision AI — Derive Image Insights via ML — Google Cloud". [Online].
`https://cloud.google.com/vision/`. [Accessed May 9, 2019].

[20] "Miffyli/im2latex-dataset: Python tools for creating suitable dataset for OpenAI's
im2latex task". [Online]. `https://www.github.com/Miffyli/im2latex-dataset`.
[Accessed May 9, 2019].