

1000 Words

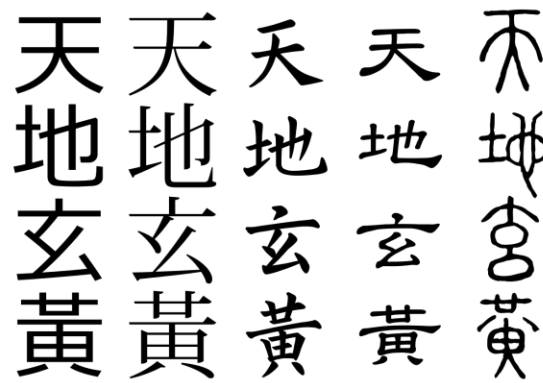
by

Elsie Adu & Esther Adu & Sergio Andres & Corey Della-Pietra

An app for visual learners and for those who enjoy the simple things in life. The Android application “A Thousand Words” allows a user to visualize a memo, draw it, explain it with audio capture, and save it. This application is an app for users who find it necessary to visualize and understand memos or tasks, to create artwork reflecting their world, and for those who want to teach themselves through art. As creators of this application, we hope to please visual learners, children, artists, and anyone who is captivated by their surroundings and thoughts and simply wish to trace the fine linings of them and attach those linings with a few words.

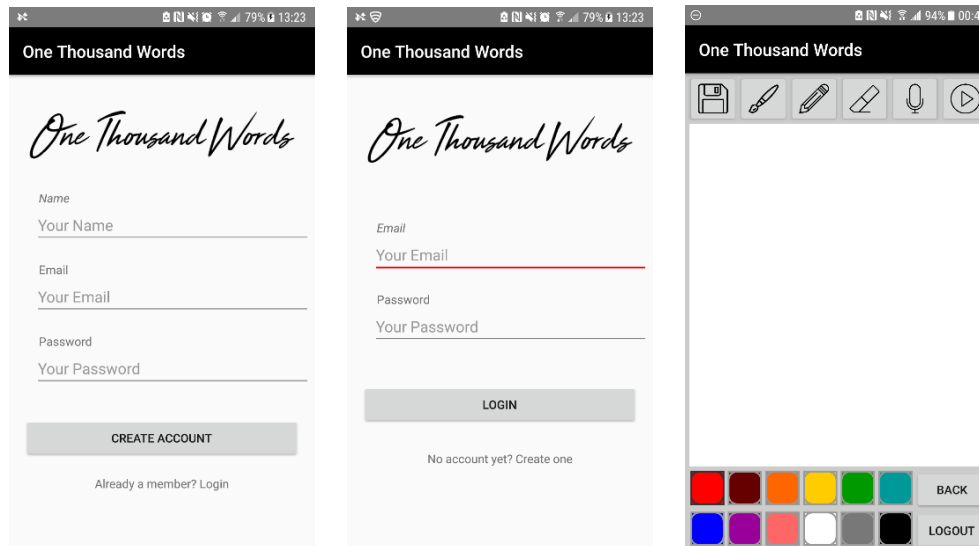
The two related apps that rival this app are the “Samsung Android Memos” application and “Adobe Illustrator Draw and Photoshop”. The “Samsung Android Memos” includes a checkbox task, image, drawing, and an audio capture. It is useful for organizing and creating memos. However, it is difficult to save as all ideas are mainly locked in the app and requires a user to send their memos through email or add it to their laptop. So if a user were to lose their phone, then all their memos are lost forever. The “Adobe Illustrator Draw and Photoshop” is an app made specifically for artists. It allows a user to pick different pen options, has a wide range of colors and allows a user to save their work to their device or an Adobe Creative Cloud. However, it lacks an audio capture since its focus is on art.

Benjamin Franklin once said, “Tell me and I forget. Teach me, and I’ll remember. Involve me and I learn.” The main target for this application is for those who see themselves as visual learners. Those who wish to attach a few words with the work in which they have created. This could range from anyone and any age. It is created for the poets, who want to say a few words, but create some art alongside. It is for the mother, who wishes to create a learning tool for their children, in which she can attach words with pictures in order to make it a little easier for her child. It is for the student, who has a hard time understanding the concept of a lecture, but desires to learn and is aware that some words and a picture can help. It can even be for the older citizens of the world, who just enjoy drawing and would like to leave some note alongside it. It is for the dreamers, the thinkers, the pupils, the artists, the writers, the adolescents, the grown-ups, the teachers, or those who simply love to hear the sound of their voice and create some art as well.



Source: Wikipedia.

The idea of the app came about while observing the characters of different Asian writings. Through the desire of wanting a visual aid as well as assure oneself that pronunciation of the wordings were correct, a lightbulb went through and thus, a visual helper was needed. After sitting down as a group, we began going through a list of ideas that would be needed in order to make this an amazing app. We needed a database to store the user's information, a canvas so the user may have a blank slate to create whatever they want, and a voice recorder to complete the idea. We first began by creating the user interface so we knew which buttons we have to use and how to make it simple enough for the user. Then we went on to create a database, so we at least would have the user's information stored and they may have their own personalized experience. Afterwards, the main goal of the application: voice recording and a sketch canvas along with paint to allow the user to show off their personality was finalized. This solidified the main objective of the software.



The features in the app include a register/login page, a drawing view along with an optional audio capture. The register/login page has a user create and login with their unique email and password, using EditText, where firebase database allows saving to the user's account once they click the register or login button.

The main feature of our app is the ability to draw and record in one frame. This allows the user to create these memos composed of a drawing and a recording saved as one. Creating our main Drawing feature took two files, a DrawingFragment and a DrawingView. The former is where all the buttons, and the main drawing interface is instantiated. The colors shown to the user are created by using ImageButtons with each color defined in the XML file. Our app features 12 different color options where each ImageButton uses an OnClickListener that connects to the DrawingView. The DrawingView is where all of the android Graphic objects are instantiated like Path, Paint, and Canvas. When an ImageButton of a new color is selected, the colors hex code is parsed by the android.Graphics Color and the Paint object is set to that color. This is found in the setColor method in the DrawingView. The user is allowed to change between multiple brush sizes and eraser sizes, both work similarly so the code is also very similar. When the user clicks the Brush or Eraser ImageButtons, they have the options to click between a range of four sizes. The ImageButton the user clicks determines which size to load on the setBrushSize function from the DrawingView. The brush sizes are defined by us in an XML

file called `dimens.xml`. Here, the brush sizes we want are defined. We chose 10dp, 20dp, 30dp, and 40dp, as the smallest, small, medium, and large sizes brushes. These values are then converted to the dimensions of pixels they correspond to. The `TypedValue` class allows the user to return a float that corresponds with the amount of pixels needed as the brush size. The `Paint` object's `setStrokeWidth` method is then called with the new brush size. An important part to do next is to set the transfer-mode to `PorterDuff.Clear` or null, depending on if the brush is for erasing or painting. These are the basic actions in the `DrawingFragment`. The `DrawingView` is where all the components to draw something are initiated. To draw something, there are four basic components: A `Bitmap` to hold the pixels, a `Canvas` to host the draw cells, a primitive (we used a path), and a `Paint` object that describes the color and style. The actual drawing method is the `onTouchEvent` in the `DrawingView`, this method works when the user first presses down on the screen to draw. After, the `MotionEvent.ACTION_DOWN` is called, setting up the beginning of the path. After the user moves their finger, the `lineTo` method in the `Path` class is called, which stores the (x, y) point where the path moved too. Once the user lifts their finger, the `MotionEvent.ACTION_UP` is activated and the `Canvas` takes the `Path` created, the `Paint` currently being used and calls its `drawPath` method. This is the main method that implements our Drawing feature.

When a user chooses to create a new drawing or go back, an alert dialog asks them if they really want to discard their changes and if they do, their drawing is discarded. If the dialog is cancelled, they can click on the save button, where their artwork would be downloaded to their gallery located in the phone. Before the user can save, permission is needed for the app to access their gallery. If denied, the user would be unable to save until allowing the application to do so. Otherwise, their artwork will be placed in their phone to be looked upon later. A user can also click the microphone button to record audio and then choose the playback button to listen to their audio. Just like saving to a gallery, the application also needs permission to record audio. If the user grants this permission, then they may record as much or as long as they please. If not, the user would be unable to use this feature.

A feature that was never implemented was the use of a ListView, in which would ultimately display previous works the user created, where a list of image would appear and have the individual be able to edit or just simply view their artwork. This part of the application never got finished mainly due to errors that were never fixed or were cleared up too late. One error included the use of Firebase's storage, where the bug was uncommon, taking up actual coding time. There were also different ways that was done in order to display a ListView, which included accessing the user's phone and displaying it through the app, but it would not have worked well if the user were to go into the application through a different phone.

Since the app was at first surrounding the idea of languages, a future development that was tossed around was having a character tracer where the user can learn words in other languages, say Chinese, by tracing the character presented to them. This feature was not implemented because of the necessity to know other languages and there would have to be a package of data holding these characters for the user to play with.

"A Thousand Words" is an app that lets people play with their imagination or simply with the thoughts that can't be written down. It gives users the option to create whatever they please because a white canvas and the sentences that form in your head is sometimes all you really need. We allow our users to do all the talking and also paint a picture that could be worth more than a thousand words.

Works Cited

Butler, Michael. <http://smallbusiness.chron.com/backup-android-memos-28376.html>. n.d. Website. 28 July 2017.

Smith, Sue. <https://code.tutsplus.com/series/create-a-drawing-app-on-android--cms-704>. 19 April 2013. 21 July 2017.

TutorialsPoint. https://www.tutorialspoint.com/android/android_audio_capture.htm. n.d. 21 July 2017.

“Add Firebase to Your Android Project | Firebase.” n.d. *Google*, Google, n.d., 21 July 2017
firebase.google.com/docs/android/setup.

“Benjamin Franklin Quotes.” BrainyQuote, Xplore,
www.brainyquote.com/quotes/quotes/b/benjaminfr383997.html