Final Project Report

           For this project, I decided to create an app called DrawTablet. It is a drawing app with some basic functions, such as a color changer, width changer, eraser, delete all, and a save feature. My app uses a Draw class, which extends View class, to create a canvas that allows the users to draw in. I used several other classes to accomplish my goals. I used the Path class to allow the user to create a drawing, which are essentially paths. The onTouchEvent function detects where the user has placed their cursor/finger and it draws a path depending on how the user moves their cursor. As the user is drawing, I used a Bitmap to save their drawing from the canvas to a bitmap. When the user lifts up their cursor, the onTouchEvent method detects it and saves the entire canvas to a bitmap and resets the path to allow for a new path (user drawing) to be created. I used the onSizeChanged method to create the bitmap and store the bitmap to a different canvas. I also used the Paint class to allow the user to draw (create paths) and change their color. The color and the eraser feature works similarly. The color button allows the user to enter ARGB to choose any color they want, and the eraser just sets the paint color to white. To change the width, the user can enter a width and the setR method in the drawing classes simply changes the stroke width of the paint. The delete function resets the paths and the bitmap, which has the previously stored drawing, clears the screen by painting the entire screen white. The save feature uses a combination of ContentValues, Uri, and OutputStream classes to save the bitmap. I also used a toast to notify the user when their drawing is saved.

           To create the menu items, I create a new resource directory with a menu type. Then, I created a menu.xml to create the menu items. I used the onCreateOptionsMenu method to create a menu inflater, which displays the menu in the top right-hand section of the app. Then, I used the onOptionsItemSelected method to perform different actions depending on the menu item click. Each of the menu item used an icon depending on what it does. I used a website (<https://fonts.google.com/icons?selected=Material+Icons>) to download some images to use as icons. When each of the menu items is clicked, a new alert dialog is created. The alert dialog uses an inflater to open a new layout depending on the item clicked. If the width menu item is clicked, a dialog which has a text box to enter a width the user desires, and a set width button which changes the width of the paint. If the color menu item is clicked, a dialog opens up which allows the user to enter ARGB values, each into their text box. There are two buttons in this layout. First is a test color button, which allows the user to enter ARGB values and a box at the top of the layout displays a color, according to the values the user entered. Another button is set to color, which sets the color of the paint to allows the user to draw using that color.

           The majority of my app is based on the View class. There were many challenges I faced when creating this app. I could not decide what app I wanted to create. I eventually landed on a drawing app. I wanted to create a fun app for myself to doodle during my free time. The idea of a drawing app sounded simple, but in reality, it was a lot harder to make. I had to do a lot of research and watch tutorials to create the menu, menu items, and alert dialog. The saving feature was not easy either. I had to research how Uri, ContentValues, and OutputStream worked for saving. Another issue I ran into was when changing width and color. In my first iteration, changing color and width would change the color and width of the existing drawing as well. To get around this, I used a bitmap to save everyone on the current canvas. Then used a different canvas to store everything from the bitmap. Some of the basic features like color changer, width changer, and eraser were very simple to make.