# Remind

By:
Keith Abdulla

June 8th, 2016

# Table Of Contents

# Introduction

### Welcome to Remind

Smartphones are increasingly getting more distracting and it is taking a toll on our productivity. Joe Kraus a partner at Google Ventures said that “in the pre-smartphone era we accessed the internet roughly five times per day, in longer chunks. Today, with smartphones, we’re accessing it 27 times a day” [9]. However, when having constant conversations through text messages or other social media platforms, we tend to access our phones even more. In order to see our phone consumption usage, there are several time managers that track the applications we’re on the foreground. A foreground app is when the phone’s screen is on and the application’s view is currently on the screen. Although, there are some downfalls in other time managers such as a lack of assurance in security, their user interface is outdated, or some only update app usages every 3 to 10 minutes. Understanding the pitfalls, I wanted to create an application that offers what other time managers don’t, which is why I built Remind. With a beautiful design and simple interface, Remind allows users to see an up-to-date time of their app usages. Along with the ability to set helpful reminders on specific applications in hopes to be less distracted on the phone. Therefore, Remind users are those who are trying to become more aware about their app usage. Those who are mindful about their time will set reminders on certain app usage, and in result becoming more productive and mindful about their time.

### Remind Features

In order to give Remind users a satisfying experience, there are additional features present. By using Material Design principles, I create a welcoming experience to their app usages and provide an intuitive interface to start interacting quickly with Remind. [3] In addition to a mobile version, there is also a reactive tablet version to offer those who have a tablet an even more user friendly experience. Then the main reason for Remind is to make you become less distracted, in which is why there is also a widget available in order to glance at reminders easily, without opening up Remind each time. Lastly, not only can the users get useful notifications when they go over a certain time of an application, they can also share their results to others about how they’ve only been on Instagram for only 5 minutes. Or in my case, share to my friends how much I’ve been working on this app Remind!

# Creating Remind

### Why build Remind?

Rather than reading about distractions and how it's hurting our productivity, I want to give the users a first hand experience of how they spend time on their phone. I wanted to build an application that allowed users to easily comprehend their app usage statistics and interact with helpful reminders in hopes to make users become more mindful of their time.

The real problem when it comes to our phones is not that it's just distracting, but we've become passive on how much time we waste on our phones. On average, interruption from work takes about 23 minutes to get back on task [5]. Therefore, knowing that statistic and that our phones is one of the easiest entry points to distract us, then how would we get anything completed? Realistically, attaining focus from a distraction on a phone compared to an another form of distraction can happen much more quickly. Such as responding to a quick text message might distract us for a minute rather than 23 minutes. Therefore, the important thing to note is not only the duration, but the number of time our phones distracts us. If it took us 3 minutes to become focused back on the task, but our phone interrupts us 60 times in a day, then summing that up would be a total of 3 hours of unproductiveness.

Even though we know our phones are a problem to our productivity; we don't do anything about it, because in our perspective we are getting some tasks completed throughout the day. However, what if the users could actually see their own statistics on their productivity, rather than reading about someone else's statistics in an article? What if the users want to use their statistics to their advantage and start training their focus better? What if users wanted to set mindful reminders to keep theme in line with their goals? To answer these questions you can use Remind as a solution.

# UI Design

In this section, I present the application design step by step in order to inform the readers on how to not only use Remind, but also talk about why I decided to integrate certain features. In the process, I will talk about some of the design implications and what were the tradeoffs in deciding which way to move forward with.

## 3.1. Present Page

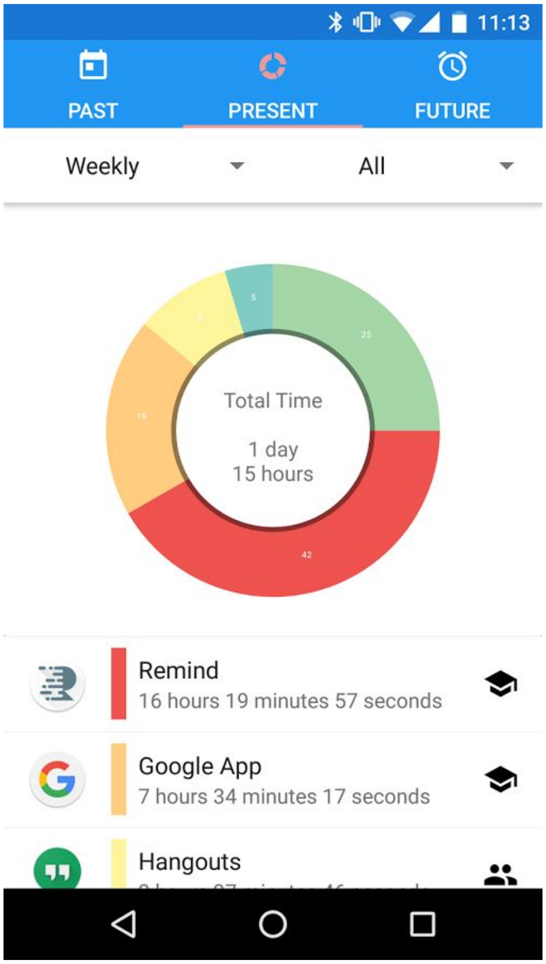


Figure 1. Present Page Mockup of Remind

The present page is the main entry point into the app and this is where the users become aware of how well they are doing on their phones in respect to productivity. This is called the default page, because this page is always set to show on the app's first launch. One of things that the user will notice on this page is the ability to sort through their app usage based on the current day, the beginning of the week, the month, or the year. Additionally, on first boot of the app and after accepting permissions, I give the user a quick action using an element in Android called a "Snackbar" [10], to show to the users they can tag and share elements. The reasoning behind tagging, is that not all applications are distracting. For instance, me building Android Applications is productive and this can be seen in Fig.1 on how I separate apart the different elements using certain icons. In addition to tagging, this unlocks the ability for the user to sort through the categories and get a better view on how they are doing with their time. I believe by making small actions such as tagging elements, helps the users critically think about not only categorizing the apps, but make it a habit for themselves to come back and keep checking their results. From this, I hope the user becomes more mindful of their time, which in turn will lead to better productivity. Another design decisions was integrating the ability to share results. By giving others the ability to share their results, it'll help Remind in attaining more users, by mentioning Remind in the share response. Lastly, I found that a lot of people didn't want to see their statistics, because they knew they are on their phone too much. However, the homescreen is designed to attract the user into the app and look less intimidating. By using white space, spacial elements, bold colors, and an interactive pie chart; I allow the user to become more comfortable with the setting. Users shouldn't be scared of their results, but use it as a tool to do better if they want to become more productive.

### 3.2. Future Page

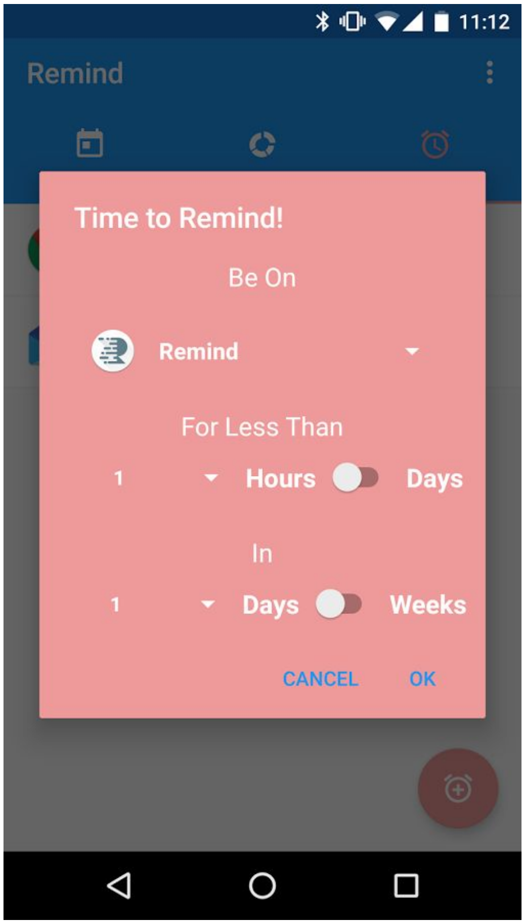
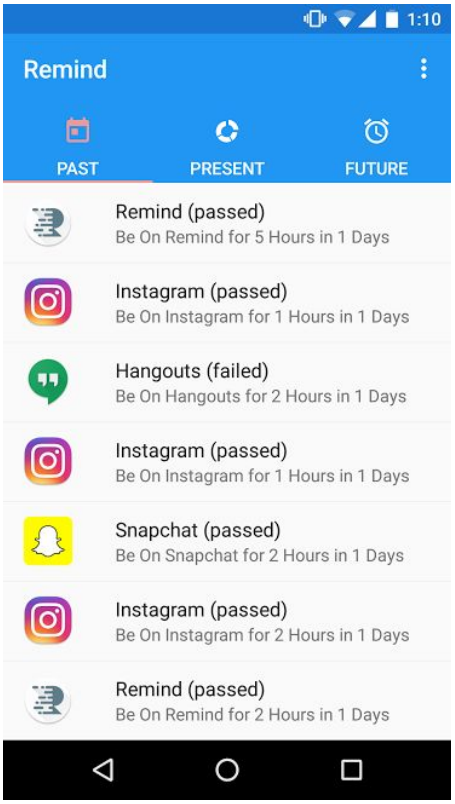


Figure 2. Future Page Mockup of Remind

This page is all about setting reminders and this is a secondary page, because in order to make educated reminders, the user must know how they are doing first which is the reason for the Present page. In this page, I've implemented a Floating Action Button which states to the user that it's the main action on the page, which is adding reminders! Once they hit the FAB, they'll get a dialogue that shows the screen in Figure 2. One of the design decisions here is what apps to present to the user in the 'Be On' dropdown. One thought is that I don't want to show every single app installed on the phone, because it makes the drop down menu too cluttered and it's going to show apps the user rarely uses. The way I decided to go with is only present applications that the user has used in a month's span, because I'm confident the users are using those apps and it's the right amount of information needed. Whereas if I did it for the whole year, then I'd have the problem again of the menu becoming too cluttered and if I did it for just a week, there may not be enough data to present. Another design decision was to put the icons next to the app titles, because a lot of us interact with the icons to open them and not the title. I felt that by implementing it this way, it provides comfortability in the app and the ease of choosing applications. Once the user sets a certain application to be on for less than a certain time in a certain time range, they'll notice the adapter is update in real time. Additionally, if the user has the widget on their homescreen then this will also be updated in real time. The reason for the emphasis on real time update, is to give quick assurance to the user that their action was satisfied. Lastly, in the free product flavor of this application, every other clicks to add a reminder will show a short duration ad banner. It has currently been designed to show in the bottom of the screen, centered; where it doesn't destruct any functionality for the users.

### 3.3. Past Page



Remind (passed)
Be On Remind for 5 Hours in 1 Days

Instagram (passed)
Be On Instagram for 1 Hours in 1 Days

Hangouts (failed)
Be On Hangouts for 2 Hours in 1 Days

Instagram (passed)
Be On Instagram for 1 Hours in 1 Days

Snapchat (passed)
Be On Snapchat for 2 Hours in 1 Days

Instagram (passed)
Be On Instagram for 2 Hours in 1 Days

Remind (passed)
Be On Remind for 2 Hours in 1 Days

Figure 3. Past Page Mockup of Remind

The past page is mainly a storage device of all the past reminders the user had set. This page has it's importance, because this is how the user starts becoming mindful of what they set for their goals. Many of us set unrealistic goals, because we expect our best performance, but "It is better to set goals that raise your average performance and make it more consistent [4]." Therefore, this page is very simple and it is only here to present information. Also, failure isn't a bad thing and we may need it to succeed, which is why I did not build a delete functionality here, because it makes the user stay true to their results.

## 3.4. Tablet Version

I built a tablet version to give my users the fullest and most satisfying experience. When on a tablet, the user shouldn't receive the same interface as mobile, because there is more to show in order to fill the white spaces. By creating the tablet version, the user will notice in Figure 4 how Remind integrates the present page to be the larger view on the right, whereas the present and future page are together on the left.

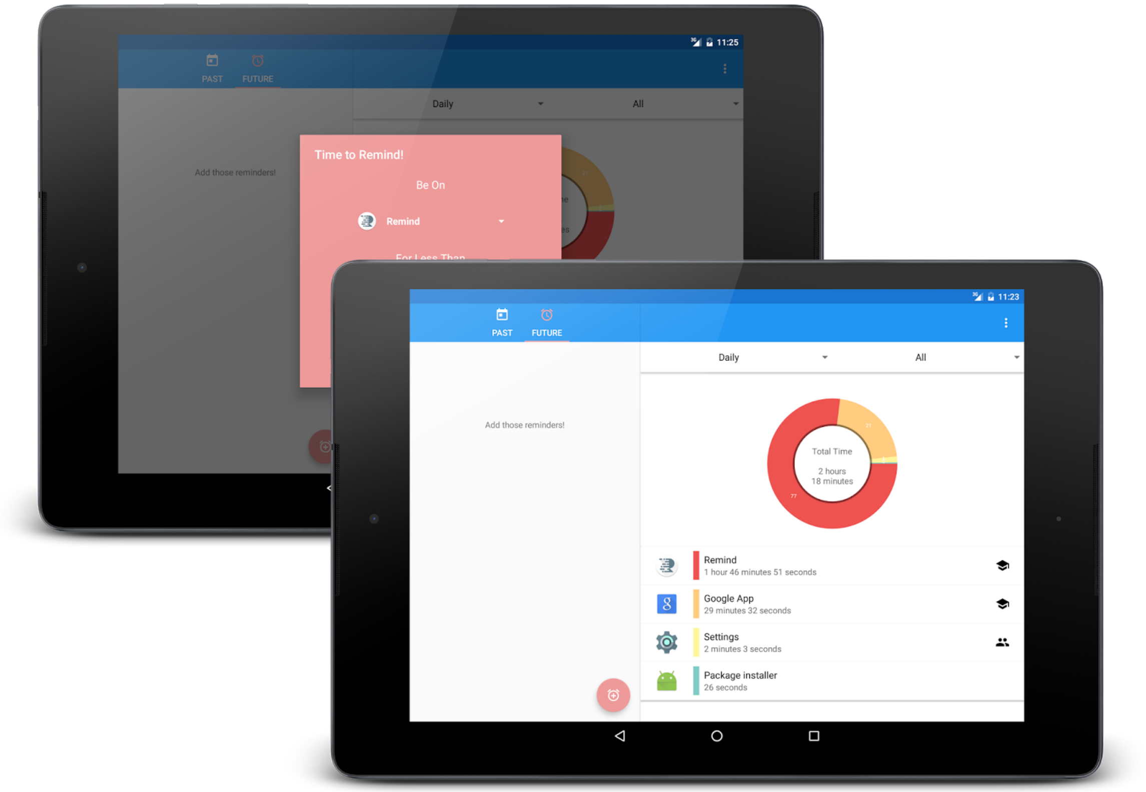


Figure 4. Tablet Mockups of Remind

# Key Considerations

When building Remind there were several things that I needed to make sure I accomplished in order to give the users a satisfying experience with Remind. To do this, I will talk about why I am requiring a low level app usage permission. How I will handle their data and assure them it is secure. Describe what the corner cases are in user experience to support events when the user might think it is a bug. Lastly, be able to give credit for the libraries used in developing Remind and why I had to use them.

## 4.1. Whoa, Using App Usage Permissions?

Having the app usage permission is the one of most important consideration to Remind, because the app is dependent on this permission and it deals with a lot of confidential data from the user. To give assurance to the users, they can see that I am not asking for network permissions, which subtracts all it’s vulnerabilities from the equation. Additionally, all usage data is aggregated by a call to a library when the application is run, which means I’m not externally storing data to the phone. This gives assurance to the user that their data is secure, because when they close the app the data is gone too.

## 4.2. Handling Data Persistence

For the present page, I don’t need to store any data on the phone, because the Usage Stats Manager generates this data based on the real time requests. However, I store the goals that are added into the application from the future page into a SQLite table. From this, I create a content provider to give my application and widget the accessibility it needs to fetch, add, update, and delete the stored data in the database. Lastly, I store tags that are tagged by the user in the present page using system preferences. I’ve used a system preferences, because I’m not saving the usage results and I don’t need the overhead of a database to store simple key and value pairs for certain items.

### 4.3. Corner Cases in User Experience

One of the corner cases that is handled is when the user scrolls down in the present page and then swipes to either the present or future page with no items. Before, the user would just see a blank screen, because they would also be swiped to the same X location in the other fragment. Therefore, if the user does fall into this scenario, the Toolbar will animate downwards with an additional empty state view with a small description for an action. The other corner case is when the user has long clicked items in a fragment and swipe to another fragment. If this were to occur, the state is not preserved and would have to tag their items again.

### 4.4. Libraries Used

I found that styling the contextual actionbar in Android was very difficult and I had a lot of code to keep information saved based on long clicks. After fighting with my own customized contextual action bar, I decided to use a library called material-cab [1]. It allowed me to style with ease and decouple, but reuse much of the code internally. Lastly, I used an android chart library called MPAndroid chart [2], which is for the user to visualize and interact with their usage statistics.

## Required Tasks

This is the section where I take the main features of my app and decompose it into tangible technical tasks that I used to complete my android application. By following the steps in Appendix A, someone who has some Android experience will be able to produce Remind with a little work.

# Conclusion

## 6.1. Future Considerations

Even though Remind supports both tablet and phone version, it doesn't have the ability to sync across multiple devices. Therefore, in order to do this I'll need to add desktop support for full syncing. I would like to implement this feature in the future, because any time spent on either tablet or mobile should be tracked. In order to do this, I would have to make network permissions mandatory for the paid feature so that I can have a main data center to store the various data across platforms. Lastly, I would like to add a bit more functionality to the past page. I want to start doing more analysis based on their past results and start generating hints on what times to choose or to notify that the user should track a reminder for an app, because the user has been on it for 25 consecutive minutes.

## 6.2. Remarks

I built this product to stop procrastinating on my android apps, because it was a huge problem. Procrastination is a way to avoid undesirable tasks and interestingly, academic procrastination is 70-95 % problematic in students [6]. To others this statistic may not sound too bad, because "course materials become less boring, more interesting, and more engaging when they procrastinate" [7]. In response, there is truth to this, but by doing so it also led to "a higher workload, more stress, higher frustration, more time pressure, and effort. So interrupted work may be done faster, but at a price" [8]. Therefore depending on the context, price has it's different meaning, but they are equal in their intrinsic value. For some students, price might be not getting the best grades, but if they use Remind they'll be able to transition into a new habit of setting reminders and be aware of their app usages on a day to day basis. By doing so, it'll allow the user to become more mindful of their time and the things that they must get accomplished.

# Appendix A.

## Task 1: Project Setup

- Move every static string used in activity/fragment into strings.xml
- Create two product flavors (paid and free) using Gradle
    - Make sure to use ‘freeCompile’ on adMob library in the gradle build
    - Configure dexOptions and set multiDexEnabled true
- Move all icons and picture into res/drawables
- Add Permissions PackageUsageStats and WakeLock to the AndroidManifest.xml
- Enable RTL true under the application tag
- Define colorPrimary, colorPrimaryDark, colorAccent, textColor, and secondaryTextColor in colors.xml
- Create custom theme for the application under Styles.xml using the colors

## Task 2: Implement UI for Each Activity and Fragment

- **Build UI for RemindActivity**
    - Coordinate Layout
        - AppBarLayout
            - Toolbar
            - TabLayout
        - ViewPager
        - Floating Action Button
- Build UI for fragments inside of the ViewPager
    - **Build UI for PastFragment**
        - RecyclerView
            - Detailed Item Layout
    - **Build UI for PresentFragment**
        - Two AppCompatSpinners
        - RecyclerView
            - Detailed Item Layout
    - **Build UI for FutureFragment**
        - RecyclerView
            - Detailed Item Layout

## Task 3: Build Database and Provider

- Create a RemindNotificationItem class that will maintain all the data necessary for an added goal
    - timeCreated, type, name, choice, choiceLength, notification, notificationLength, id, usageLeft
- Create a SQLiteOpenHelper with the variables used in the class
- Create a content provider
    - Authority: com.ekeitho.remind.provider
- Add content provider permission to the AndroidManifest.xml

## Task 4: Build Remind Activity

- Create a method to check if a user has accepted the usage access permission
- Check permission after setContentView, since android's Snackbar needs a context.
    - If no permission, show indefinite Snackbar, and give user quick action to get to the settings
    - If accept permission or come back from settings menu, set up rest of views
- Set up the views
    - Bind to the toolbar, FAB, view pager, view pager adapter, fragmentStatePagerAdapter, and the tablayout
        - Page one, PastFragment
        - Page two, PresentFragment
        - Page three, NotificationFragment
- Add icons and selections changes to tablayout
- Start an IntentService
    - On launch, aggregate app usage stats for the daily, weekly, monthly, and yearly results
    - Fire an alarm manager to go off every minute that grabs data on the Notification Items, if any, and relays it to the FutureFragment and the Widget
        - If notification item go over their choiceLength, send user a notification, and append to the PastFragment

## Task 5: Build Present Fragment

- Hide FAB button

- UsageStats is the object that Android uses to package together app usage statistics, but has more information than need so create a Remind Stat class
    - Class must be Parceable for rotation changes and variables are: packageName, durationTime, appTitle, and isClicked
- Build Recycler View Adapter that will take two types
    - Chart Type Viewholder
    - Remind Type Viewholder
- Set up date Spinner that will allow user to sort through daily to yearly views. Set up sort Spinner, which will filter out distraction/productive/all applications.
    - Add listeners to each spinners to notify the adapter on change
- Register a Broadcast Receiver that accepts intent of ‘data-available’ which is sent from the service, when the Async Task for the service is complete
- Based on the data received from the service and which positions the two spinners are in, update the Recyclerview with the pie data and all of the remind stats
    - The pie data won’t have a legend, but a color code next to each detailed item, which will minimize space and increase usability

## Task 6: Build Future Fragment

- Show add notification FAB on this page
    - On click, show dialogue to add a notification item
        - Call Content Provider insert when user taps the dialogue ‘ok’ action
- Set up a loader manager
    - Content Provider query for items are in progress
    - Refresh adapter and recyclerview on update, which the logic for this update is in the Content Provider

## Task 7: Build Past Fragment

- Hide FAB button
- Set up a loader manager
    - Content Provider query for items no longer in progress
    - Refresh adapter and recyclerview on update

## Task 8: Build Widget

- Create a widget layout with a title bar and a list view

- Create an AppWidgetProvider that uses the layout in a remote view
    - Have click pending intent that launches the RemindActivity
- Create a RemoteViewFactory that hosts the details of the view of each item in the list
- Create a Service that launches the RemoteViewFactory, which is called when the service has gotten new data from the provider (each minute)
- Add widget service and widget provider permission to the manifest

## Task 9: Incorporate Long Click Ability

- With using the material-cab library
    - Each page will have it's own list of what has been clicked
        - On first long click, show the toolbar
        - On subsequent long click/normal click, append to list
        - On empty list, hide toolbar

## Task 10: Build Tablet UI

- Build Tablet UI
    - Past and Future fragment in a viewpager, which is the master view
    - Present fragment is the detailed view
- Edit RemindActivity to test dual pane,
    - If so, use Fragment Manager to add a Fragment to the Present Fragment's Layout.

## Task 11: Integrate AdMob and Introduce Product Flavors

- Create paid and free product flavors with help using Android Studio's build variants
- When user clicks the fab button on the free application, show admob banner on both tablet and mobile layouts
    - Need to add network permissions to the manifest for AdMob

# Bibliography

1 - "Afollestad/material-cab." GitHub. Afollestad. Web. https://github.com/afollestad/material-cab

2 - "PhilJay/MPAndroidChart." GitHub. PhilJay, n.d. Web. https://github.com/PhilJay/MPAndroidChart

3 - Google. "Introduction - Material Design - Google Design Guidelines." Google Design Guidelines. Web. https://www.google.com/design/spec/material-design/introduction.html

4 - "Goal Setting." Goal Setting | Counseling Center. N.p., n.d. Web. https://www.unh.edu/counseling-center/article/goal-setting

5 - Pattison, Kermit. "Worker, Interrupted: The Cost of Task Switching." Fast Company. N.p., n.d. Web. http://www.fastcompany.com/944128/worker-interrupted-cost-task-switching

6 - Gargari, Rahim Badri, Hossein Sabouri, and Fatemeh Norzad. "Academic Procrastination: The Relationship Between Causal Attribution Styles". Web. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3939975/

7 - Cao, L. Differences in procrastination and motivation between undergraduate and graduate students. Web. http://files.eric.ed.gov/fulltext/EJ978906.pdf

8 - Mark, Gloria. The Cost of Interrupted Work: More Speed and Stress https://www.ics.uci.edu/~gmark/chi08-mark.pdf

9 - Kraus, Joe. "We're Creating a Culture of Distraction." Web. https://sakai.unc.edu/access/content/user/vschoenb/Public%20Library/Decision-making/joekraus.com_were-20141231214903.URL

10 - Google. "Android's Snackbar." Snackbar | Android Developer. Web. https://developer.android.com/reference/android/support/design/widget/Snackbar.html