# Med*Helper*

A Smart Tool to Research and Track Medications

**ANANTH DANDIBHOTLA**
**GRADE: 10**
**METHACTON HIGH SCHOOL, EAGLEVILLE, PA**

## Problem Question

Can technology improve prescription compliance?

## Hypothesis

Prescription compliance can be improved upon using smartphones  and cloud technologies.

# Materials

- Android based Smartphone (Version 4.2 JellyBean or Higher)
- Desktop or laptop with a 2.5 GHz Intel Core i7, 16GB 133 MHz DDR3 RAM
- IDE – Android Studio 1.5
- Java SE development kit 8 (JDK)
- Test Medications
- Internet Access
- Zxing library

# Procedure

- Install Android SDK for App development
- Review DailyMed website's requirements, URL specifications
- Gather a few sample medications for testing. Make sure use the same set to get repeatable results during the App development.
- Get familiar with data structures, Web services, JSON and figure out how to parse incoming data.
- Research on the web to find an api that helps scan UPC barcode and provides the data.
- Incorporate Toast messages to let user know when an action is complete.
- Add features (Search screen, Scan screen, My Meds list, Meds Taken list, Email notifications ) to achieve the goals of the project.
- Setup Amazon Web Services. Configure Apache, TomCat, MySQL
- Create Med*Helper* database in MySQL

# Motivation

$ 100 - 289 Billion/Year In Rx Non-Compliance Costs

- Failing to comply with medication prescriptions costs U.S. anywhere between $100 to $289 billion a year.
- Up to 50 % of medications aren't taken as prescribed.
- Medication noncompliance creates major headaches for patients and doctors.
- According to the study, failure to follow prescriptions causes some 125,000 deaths a year and up to 10% of all hospitalizations.

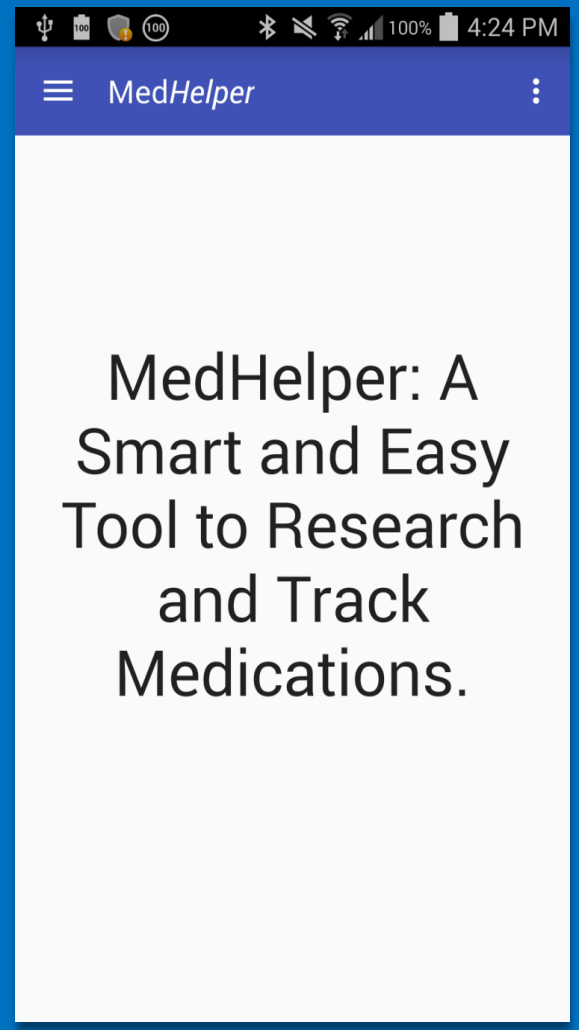
Source Notes: *Annals of Internal Medicine*

# Med*Helper* Goal

My project's goal is to develop an easy to use app by leveraging the power of smartphones.

The app will hopefully help medication users adhere to prescription compliance with the use of:

- **A My Medications List** - Allow users to add their frequently taken medications
- **Medication History** - Record when users take the medications
- **Email Notification** - Email medication usage  notifications to assigned care takers
- **Cloud Storage**– Safe and secure data storage in the Cloud



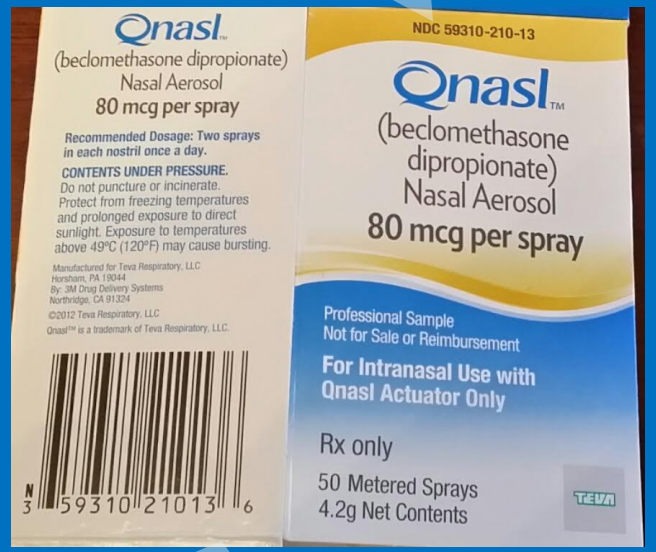
# Background Research Conducted To Understand NDC & UPC formats

A lot of research was done to understand the unique identifiers assigned to drugs.

- Researched about National Drug Code (NDC) and Universal Product Code (UPC).
- Discovered format differences within NDC.
- Implemented functionality to extract NDC  from UPC.
- Finalized a reliable source for drug data by Experimenting with FDA's (openFDA) web services and DailyMed web services (contains drug listings as submitted to the Food and Drug Administration (FDA).
- DailyMed was chosen for its reliability when retrieving data and ease of use when parsing it.

**NDC - National Drug Code**

- Assigned to all U.S. medications
- 10 Digit code with 3 segments




**UPC - Universal Product Code**

- 12 digits
- Uniquely assigned to each trade item

# Background Research Conducted To Understand Cloud Technologies and Amazon Web Services

- Researched about Amazon Web Services by reading various online documentation on Amazon.
- Researched how to isolate database server from application server and proxy server.
- Identified the configuration settings to setup the integration between proxy, app, and database server.
- Learned about ways to identify various problems, how to read log files, and searching the error code on Google.
- Researched REST services and code to retrieve data
- JSON structure was widely used by websites to send data,  the same format was used in the project

# Sample Drugs Tested Using FDA (openFDA) and DailyMed

| Drug Name | FDA's openFDA product_ndc search | DailyMed | Drug Search Method: Scan UPC | Drug Search Method: Manual NDC input |
|---|---|---|---|---|
| **MethylPREDNISolone**<br>**NDC: 0603-4593-15**<br>Control Sample | ✗ | ✓ | ✓ | ✓ |
| **SUDAFED PE**<br>**NDC: 50580-679-25** | ✗ | ✓ | ✓ | ✓ |
| **Mucinex DM**<br>**NDC: 63824-072-35** | ✗ | ✓ | | ✓ |
| **Zyrtec-D**<br>**NDC:50580-728-12** | ✗ | ✓ | | ✓ |
| **CVS COLD RELIEF**<br>**NDC: 59779-873-60** | ✗ | ✓ | | ✓ |
| **Therafluocinonide**<br>**NDC: 0093-0264-92** | ✗ | ✓ | ✓ | ✓ |
| **Tylenol**<br>**NDC:50580-404-08** | ✗ | ✓ | | ✓ |
| **Theraflu**<br>**NDC:0067-6426-06** | ✗ | ✓ | ✓ | ✓ |
| **Delsym**<br>**NDC:63824-171-63** | ✗ | ✓ | ✓ | ✓ |
| **Children's Mucinex**<br>**NDC:63824-256-12** | ✗ | ✓ | ✓ | ✓ |

# Programming Concepts Used

- This is an Android based app
- App uses an Android swipe navigation bar
- Each app screen extends a Fragment (android class)
- Each app screen has its own class and layout file
- Helper classes were created to parse URL data, to  get, parse, and store medication data
- Used DailyMed's RESTful APIs to get drug data
- Drug data was provided in JSON and XML format
- Data was parsed using Java classes
- JSONObject, JSONArray were used
- Relational Database (mySQL) is used to store data
- Tomcat 8 is run on application server.
- REST is used to retrieve data

# Technologies Used In Med*Helper*

**UI**

- User Interface is an Android based **smartphone app**
- Android development kit, Android Studio, third party scanning library.

**Security**

- Uses **Google's Sign-in** technology to validate user
- Google authenticates the user and provides user credentials

**Data Source**

- **DailyMed's REST APIs** are used to retrieve drug name and details in **real-time**
- Data is secured during transit due to https. Get/Post tested using google chrome extension

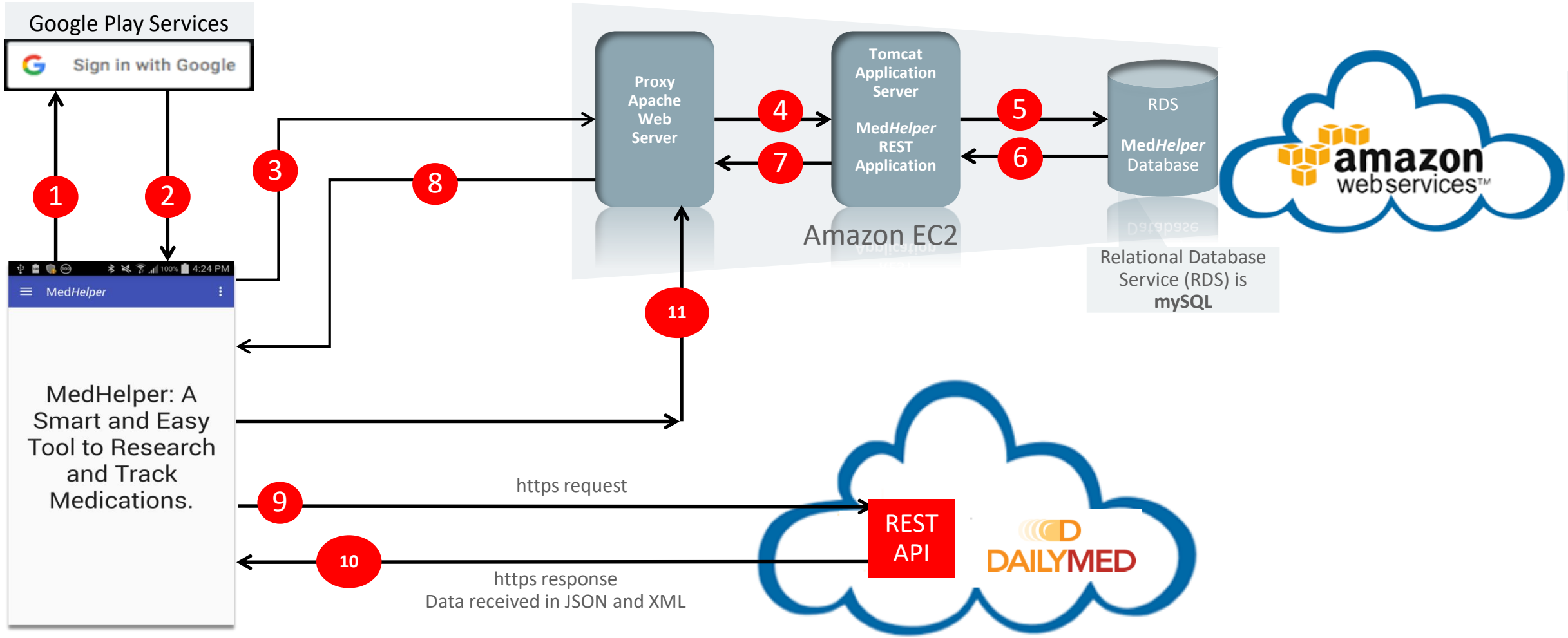
**Cloud**

- Leverages **Amazon cloud** capabilities to store and retrieve data
- Amazon EC-2 instance running **Linux** is used. An **Apache server** acts as a proxy. The instance runs **Tomcat** to handle the application side.

**Database**

- Amazon RDS is used, specifically a **mySQL** database. This is where the data is stored.

# Med*Helper* Data Flow Diagram/Flow Chart

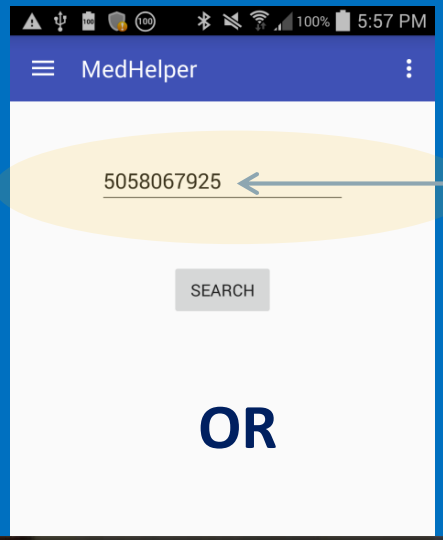


# Sample Pseudo-Code

```
Input Medication{

        User inputs NDC  code in the textbox;
        User scans a barcode;

        getNDCfromUPC{
        //user scans a barcode
          Use Zxing library to get text from UPC barcode;
          Perform barcode to ndc conversion();
        }

        // Call the DailyMed's API to get SPL_ID

        Call DailyMed{
                        Use the NDC code to call DailyMed Api

        "https://dailymed.nlm.nih.gov/dailymed/services/v1/ndc/"+text+"/spls.xml"
                        Asynchronously Get Data;
                        This returns a string of JSON Data;
                        DailyMed returns drug name and spl_id;

                        //Call DailyMed's API to get drug details
                        Use the SPL_ID to call DailyMed's API;

        https://dailymed.nlm.nih.gov/dailymed/services/v2/spls/"+s+"/packaging.json""
                        DailyMed returns drug details;
                        Asynchronously Get Data;
                        This returns a string of JSON Data;
                        DailyMed returns drug name and spl_id;
                        ParseDailyMedData();
        }

}
```

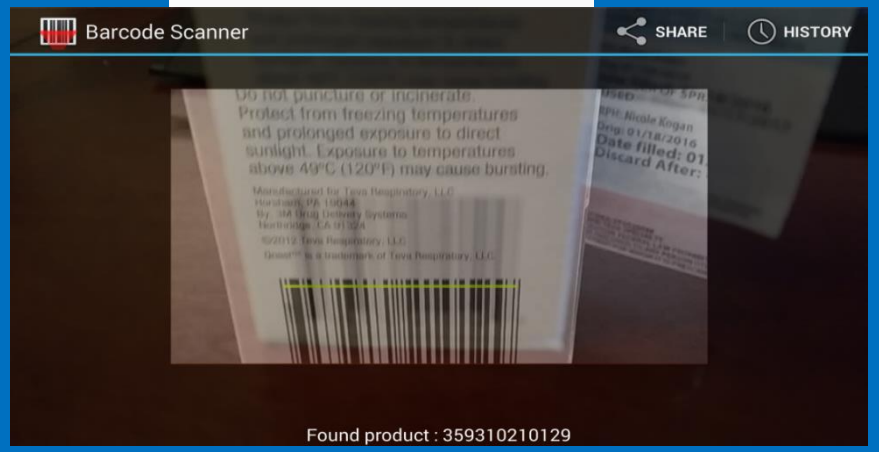
```
ParseDailyMedData (){

                //Create java JSON object. Run through the data and store what
is needed.
                // Code to parse data
                try {
                                JSONObject jsonObject = new
JSONObject(packagingData);
                                JSONObject dataObject =
jsonObject.getJSONObject("data");
                JSONArray productArray = dataObject.getJSONArray("products");
          JSONObject productObject = productArray.getJSONObject(0);
          ……..
          Medication.searchList.add(new Medication(name,s,ndc));
       } catch (JSONException e) {
          e.printStackTrace();
       }


}
```

# Input Search & Scan Methods




NDC code entered

**OR**




1. Send NDC
(Make Web Service calls)

DAILYMED

1. Parse UPC to NDC
2. Send NDC to DailyMed
3. (Make 2 Web Service calls)

DAILYMED

# App Makes Two Web Service Calls To DailyMed

1. First call is made to **get SPL* Id**

2. Make a second call to get **drug information** using SPL Id

```
URL url = new URL(params[0]);
urlConnection = (HttpURLConnection) url.openConnection();
urlConnection.setRequestMethod("GET");
urlConnection.connect();
InputStream inputStream = urlConnection.getInputStream();
```

## First Call is made

**To get SPL Id of the drug**

```
GetRawData theRawData = new GetRawData("https://dailymed.nlm.nih.gov/dailymed/services/v1/ndc/"+text+"/spls.xml");
```

*SPL – Structured Product Labeling

## Second Call Is Made

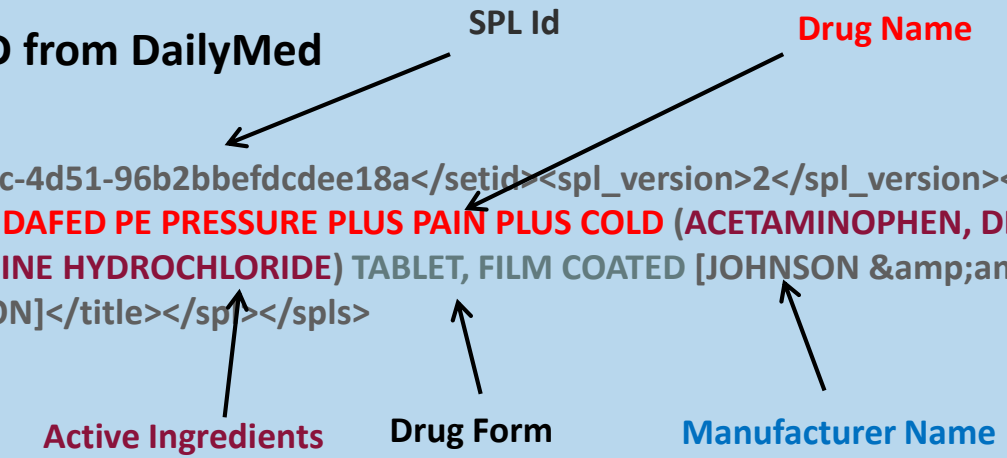
```
theRawData.reset();
theRawData = new GetRawData("https://dailymed.nlm.nih.gov/dailymed/services/v2/spls/"+s+"/packaging.json");
String packagingData = "";
try {
        packagingData = theRawData.execute();
} catch (ExecutionException e) {
        e.printStackTrace();
} catch (InterruptedException e) {
        e.printStackTrace();
}
```

**Get the details of the drug using the SPL Id**

**Code to handle if data could not be retrieved**

# Daily*Med's* Response

**First Call – Retrieving SPL ID from DailyMed**




<spls><spl><setid>cad07288-08bc-4d51-96b2bbefdcdee18a</setid><spl_version>2</spl_version><published_date>October 13, 2015</published_date><title>SUDAFED PE PRESSURE PLUS PAIN PLUS COLD (ACETAMINOPHEN, DEXTROMETHORPHAN HYDROBROMIDE, GUAIFENESIN, AND PHENYLEPHRINE HYDROCHLORIDE) TABLET, FILM COATED [JOHNSON & JOHNSON CONSUMER INC., MCNEIL CONSUMER HEALTHCARE DIVISION]</title></spl></spls>

**Second Call – Daily Med using SPL ID to get Drug Details**

returns {"data":{"spl_version":2,"products":[{"packaging":[{"ndc":"50580-679-25","package_descriptions":["2 in 1 CARTON","12 in 1 BLISTER PACK"]}],"active_ingredients":[{"strength":"325 mg","name":"Acetaminophen"},{"strength":"10 mg","name":"Dextromethorphan Hydrobromide"},{"strength":"100 mg","name":"Guaifenesin"},{"strength":"5 mg","name":"Phenylephrine Hydrochloride"}],"product_name_generic":"Acetaminophen, Dextromethorphan Hydrobromide, Guaifenesin, and Phenylephrine Hydrochloride","product_name":"Sudafed PE Pressure Plus Pain Plus Cold","product_code":"50580-679"}],"published_date":"Oct 13, 2015","title":"SUDAFED PE PRESSURE PLUS PAIN PLUS COLD (ACETAMINOPHEN, DEXTROMETHORPHAN HYDROBROMIDE, GUAIFENESIN, AND PHENYLEPHRINE HYDROCHLORIDE) TABLET, FILM COATED [JOHNSON & JOHNSON CONSUMER INC., MCNEIL CONSUMER HEALTHCARE DIVISION]","setid":"cad07288-08bc-4d51-96b2-bbefdcdee18a"},"metadata":{"db_published_date":"Feb 10, 2016 05:10:36PM EST","current_url":"https://dailymed.nlm.nih.gov/dailymed/services/v2/spls/cad07288-08bc-4d51-96b2-bbefdcdee18a/packaging.json"}}

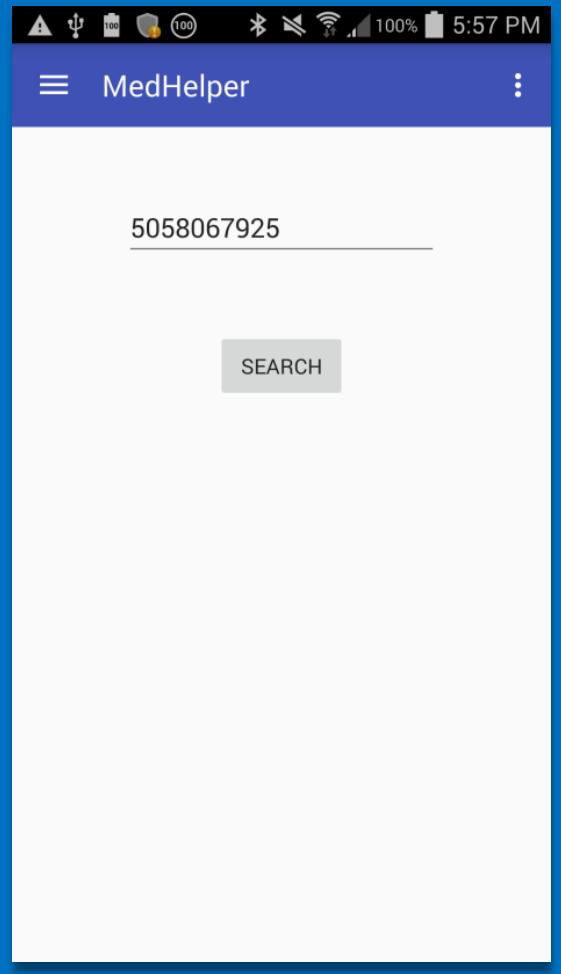
# Formatted JSON Data

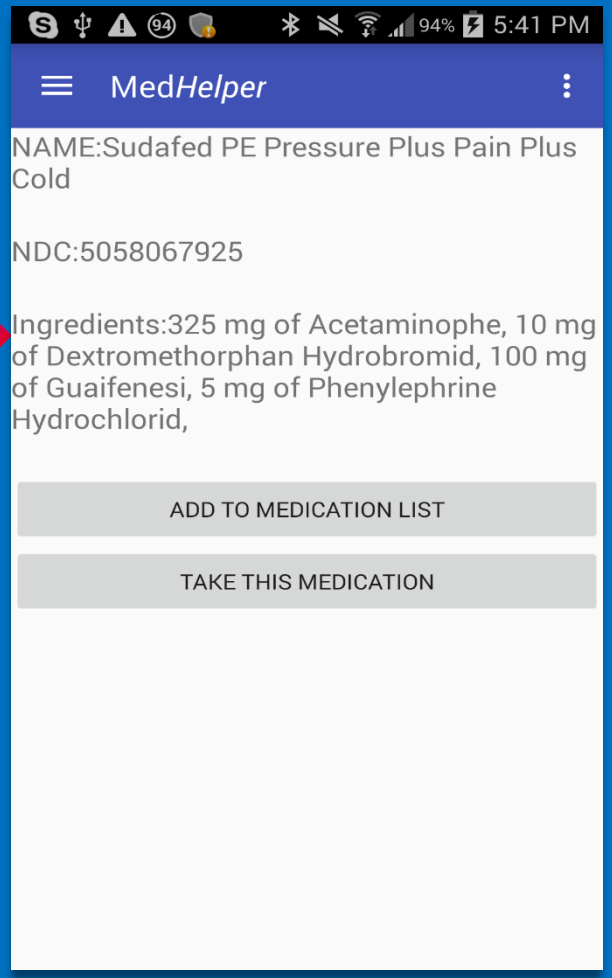

```
{
  "data":{
    "spl_version":2,
    "products":[
      {
        "packaging":[
          {
            "ndc":"50580-679-25",
            "package_descriptions":[
              "2 in 1 CARTON",
              "12 in 1 BLISTER PACK"
            ]
          }
        ],
        "active_ingredients":[
          {
            "strength":"325 mg",
            "name":"Acetaminophen"
          },
          {
            "strength":"10 mg",
            "name":"Dextromethorphan Hydrobromide"
          },
          {
            "strength":"100 mg",
            "name":"Guaifenesin"
          },
          {
            "strength":"5 mg",
            "name":"Phenylephrine Hydrochloride"
          }
        ],
        "product_name_generic":"Acetaminophen, Dextromethorphan Hydrobromide, Guaifenesin, and Phenylephrine Hydr
ochloride",
        "product_name":"Sudafed PE Pressure Plus Pain Plus Cold",
        "product_code":"50580-679"
      }
    ],
    "published_date":"Oct 13, 2015",
    "title":"SUDAFED PE PRESSURE PLUS PAIN PLUS COLD (ACETAMINOPHEN, DEXTROMETHORPHAN HYDROBROMIDE, GUAIF
ENESIN, AND PHENYLEPHRINE HYDROCHLORIDE) TABLET, FILM COATED [JOHNSON & JOHNSON CONSUMER INC., MCNEIL CO
NSUMER HEALTHCARE DIVISION]",
    "setid":"cad07288-08bc-4d51-96b2-bbefdcdee18a"
  },
  "metadata":{
    "db_published_date":"Feb 10, 2016 05:10:36PM EST",
    "current_url":"https://dailymed.nlm.nih.gov/dailymed/services/v2/spls/cad07288-08bc-4d51-96b2-
bbefdcdee18a/packaging.json"
  }
}
```


# Parsing JSON Data To Show In The App

```
try {
JSONObject jsonObject = new JSONObject(packagingData);
JSONObject dataObject = jsonObject.getJSONObject("data");
JSONArray productArray = dataObject.getJSONArray("products");
JSONObject productObject = productArray.getJSONObject(0);
JSONArray packagingArray = productObject.getJSONArray("packaging");
JSONArray ingredientsArray = productObject.getJSONArray("active_ingredients");
JSONObject packagingObject = packagingArray.getJSONObject(0);
JSONObject ingredientsObject = ingredientsArray.getJSONObject(0);
name = productObject.getString("product_name");
Medication.searchList.add(new Medication(name,s,ndc));
// String ndc = packagingObject.getString("ndc");

Log.v("JSON OBJECT", jsonObject.toString());
Log.v("JSON OBJECT",productArray.toString());
Log.v("JSON OBJECT",productObject.toString());
Log.v("JSON OBJECT",packagingArray.toString());
Log.v("JSON OBJECT",ingredientsArray.toString());
Log.v("JSON OBJECT",packagingObject.toString());
Log.v("JSON OBJECT",ingredientsObject.toString());
Log.v("NAME",name);
// Log.v("NDC",ndc+"<----- NDC");
} catch (JSONException e) {
e.printStackTrace();
}
```

# Display Search/Scan Result

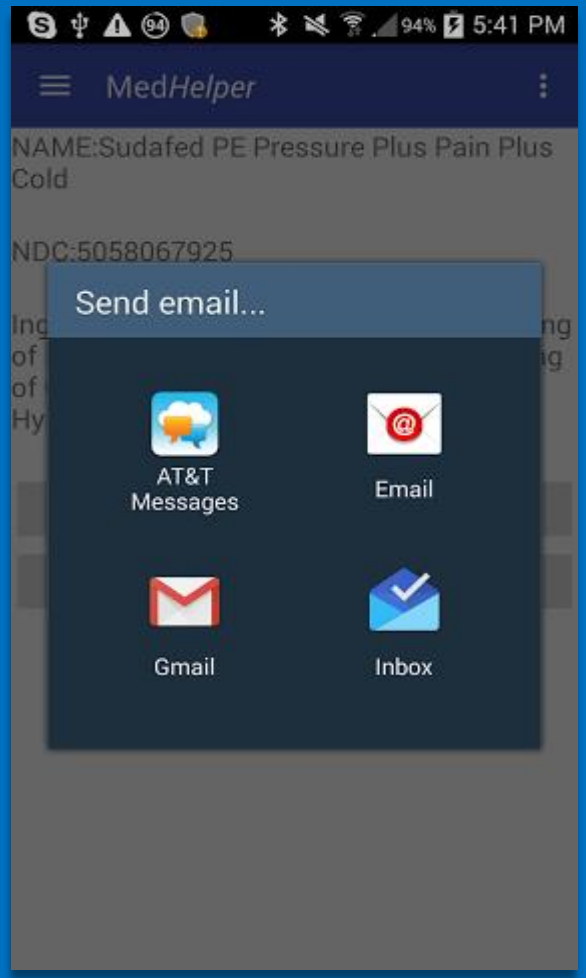


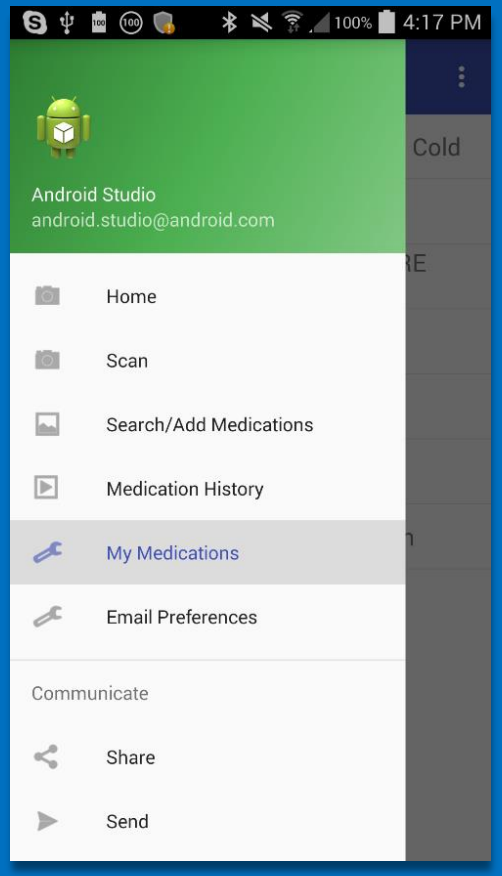

Parsed data

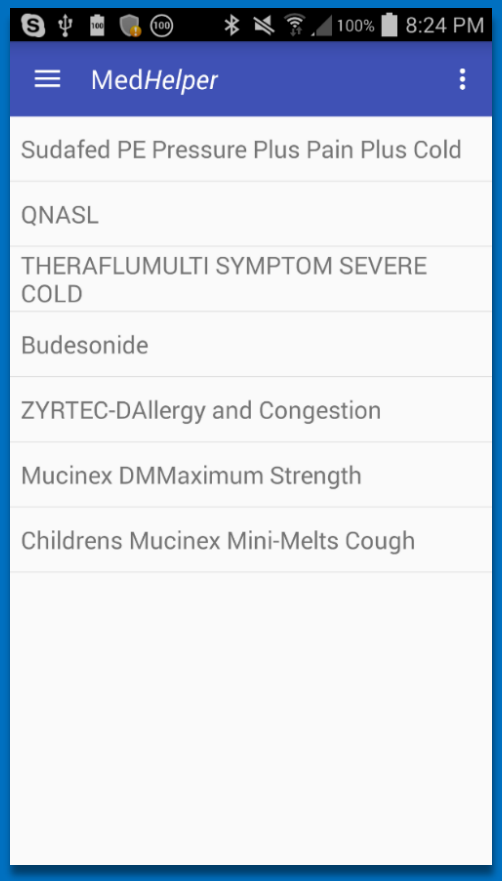
Email option

User options

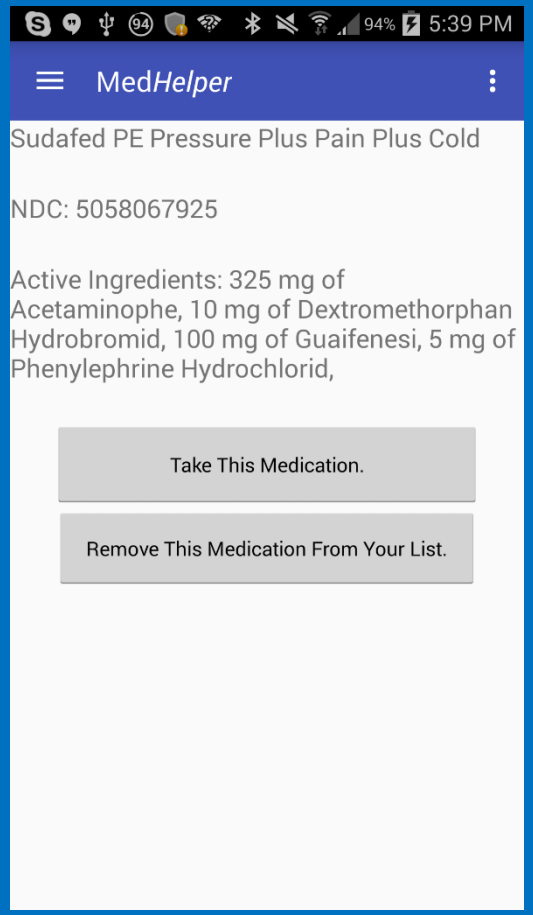


# “My Medications” List




More Options to select

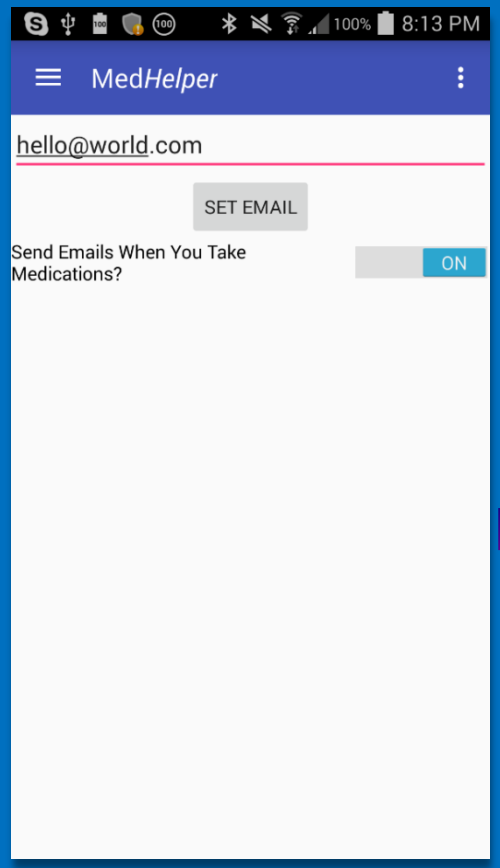



## Drug Information

- Drug Name
- NDC
- Ingredients

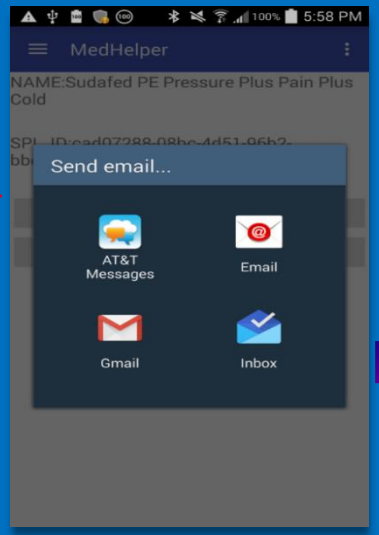
# Setup Email Preferences

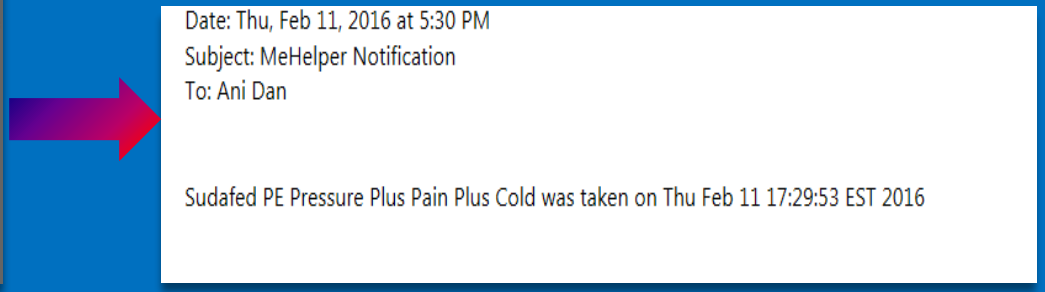
If the email notification preference is on, *MedHelper* notifies the care taker with the **drug name, date, and time** the drug was taken.



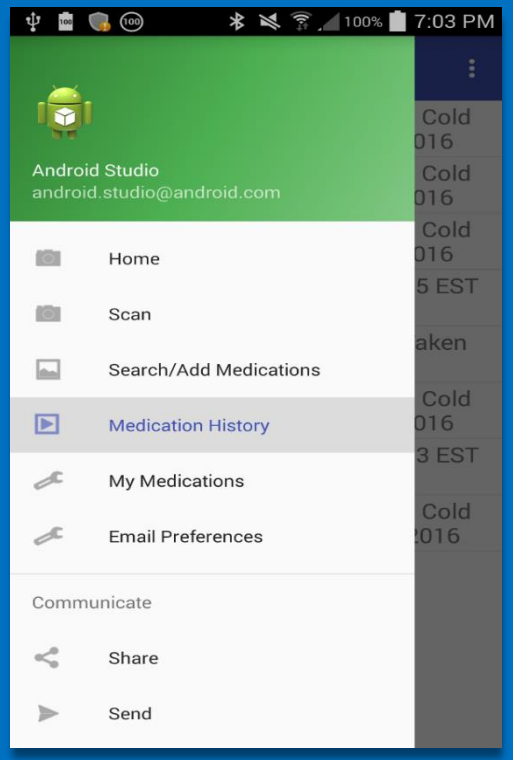

Enter caretaker's email

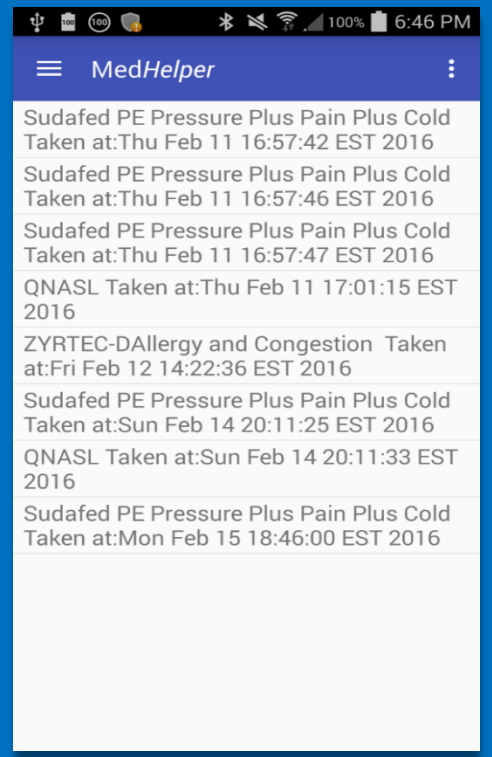
Toggles prompt to send email




# Medication History



# Sample Drugs Tested Using The Scan & Search Methods

# Med*Helper* Real World Applications

My project's rationale is to develop an easy to use app by leveraging the power of smartphones and the Cloud.

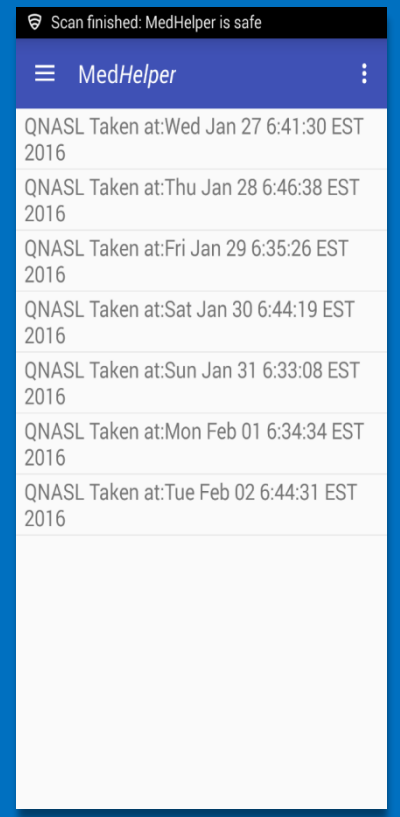
The app will help medication users, to adhere to prescription compliance with the below features:

- **My Medications** - Allow users to add their frequently taken medications
- **Medication History** - Record when users take the medications
- **Secure Login –** User authentication feature through Google sign-in
- **Secure Data Transfer** – Data is encrypted during transit using https
- **Secure Data Storage** – *MedHelper* data is *stored* securely on Amazon Cloud
- **Email Notification** - Email medication usage notifications to assigned care takers

## Conclusion

- In conclusion, the app met the project goals to keep prescription compliance.
- The app is able to search/scan medications and retrieve information from online databases.
- Data is securely stored on Amazon Cloud.
- The app offers an option to notify a caretaker.
- The app is able to keep track of the medication usage. I utilized the app to track my Qnasl prescription and to record it daily as shown here.
- Prescription compliance is most likely to increase with Med*Helper*.

Sample Record of my QNASL usage



# Challenges And Future Improvements

## Project Challenges

- Due to lack of standards related to drug packaging, the scanning only works when the NDC of the medication is embedded within the UPC code.
- Working with openFDA has been a challenge as the data presented is complex and not directly related to the drug that was searched.
- Converting UPC to NDC for non-prescription/generic drugs has been a challenge and was unable to test those drugs.
- Adding accounts and a login screen to protect users data and make data transfer over different devices more efficient took a lot of time to research and implement.

## Future Improvements

- I want to add functionality to schedule automatic reminders.
- I would like to setup app to app communication where a patient and a  caretaker can exchange medication schedules. This could also tie into notifications as the app could notify if a patient has or hasn’t taken a specific medication.

# Works Cited


- "Amazon Elastic Compute Cloud Documentation." *Amazon Web Services*. Amazon Web Services, n.d. Web. 29 Mar. 2016. <http://aws.amazon.com/documentation/ec2/?icmpid=docs_menu_internal>.
- "Apache Tomcat 8 Version 8.0.33 - Documentation Index." *Apache Tomcat*. Apache Software Foundation, 18 Mar. 2016. Web. 29 Mar. 2016. <http://tomcat.apache.org/tomcat-8.0-doc/index.html>.
- "Application Development Support: Web Services." *DailyMed*. U.S. National Library of Medicine, n.d. Web. 2 Mar. 2016. <https://dailymed.nlm.nih.gov/dailymed/app-support-web-services.cfm>.
- "APPLICATION DEVELOPMENT SUPPORT: Web Services: RESTful Resources." *DailyMed*. U.S. NATIONAL LIBRARY OF MEDICINE, n.d. Web. 15 Dec. 2015. <https://dailymed.nlm.nih.gov/dailymed/webservices-help/v1/ndc_spls_api.cfm>.
- "AsyncTask." *Android Developers*. Google, n.d. Web. 2 Mar. 2016. <http://developer.android.com/reference/android/os/AsyncTask.html>.
- Fung, Brian. "The $289 Billion Cost of Medication Noncompliance, and What to Do About It." *The Atlantic*. The Atlantic Monthly Group, 11 Sept. 2012. Web. 16 Feb. 2016. <http://www.theatlantic.com/health/archive/2012/09/the-289-billion-cost-of-medication-noncompliance-and-what-to-do-about-it/262222/>.
- "Getting started with openFDA." *openFDA*. U.S. Food and Drug Administration, n.d. Web. 15 Dec. 2015. <https://open.fda.gov/api/reference/>.
- "Introduction to Android." *Android Developers*. Google, n.d. Web. 15 Dec. 2015. <http://developer.android.com/guide/index.html>.
- "Introduction to SQL." *w3schools.com*. Refsnes Data, n.d. Web. 29 Mar. 2016. <http://www.w3schools.com/sql/sql_intro.asp>.
- "Java™ Platform, Standard Edition 8 API Specification." *Oracle*. 8th ed. Oracle, n.d. Web. 15 Dec. 2015. <https://docs.oracle.com/javase/8/docs/api/>.
- "JSON Formatter & Validator." *Curious Concept*. Curious Concept, n.d. Web. 16 Feb. 2016. <https://jsonformatter.curiousconcept.com/>.
- "National Drug Code Directory." *U.S Food and Drug Administration*. U.S. Department of Health and Human Services, 6 Jan. 2016. Web. 2 Mar. 2016. <http://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm>.
- "National Drug Codes." *Drugs.com*. Drugs.com, n.d. Web. 2 Mar. 2016. <http://www.drugs.com/ndc.html>.
- Owen, Sean. "ZXing." *ZXing*. Github, n.d. Web. 15 Dec. 2015. <http://zxing.github.io/zxing/>.
- "Package Index." *Android Developers*. Google, 11 Feb. 2016. Web. 16 Feb. 2016. <http://developer.android.com/reference/packages.html>.
- "Start Integrating Google Sign-In into Your Android App." *Google Developers*. Google, n.d. Web. 29 Mar. 2016. <https://developers.google.com/identity/sign-in/android/start-integrating>.
- "UPC Symbols." *Global Trade Item Number*. Bar Code Graphics, n.d. Web. 2 Mar. 2016. <http://www.gtin.info/upc/>.
- Whitsel, Eric, et al. "Medication Survey Form." *Collaborative Studies Coordinating Center*. U of North Carolina at Chapel Hill, n.d. Web. 16 Feb. 2016. <http://www.cscc.unc.edu/carmri/ctrtrain/UNLICOMMCarotidMRIMedicationSurveyPresentation02152005.pdf>.