

Marquette University
e-Publications@Marquette

Master's Theses (2009 -)

Dissertations, Theses, and Professional Projects

A Comprehensive Context-Aware Interruption Management System

April Chunjiao Song
Marquette University

Recommended Citation

Song, April Chunjiao, "A Comprehensive Context-Aware Interruption Management System" (2017). *Master's Theses (2009 -)*. 398.
http://epublications.marquette.edu/theses_open/398

A COMPREHENSIVE CONTEXT-AWARE INTERRUPTION
MANAGEMENT SYSTEM

by

April Chunjiao Song, B.S.

A thesis submitted to the Faculty of the Graduate School,
Marquette University
in Partial Fulfillment of the Requirements for
the Degree of Master of Science.

Milwaukee, Wisconsin

May 2017

ABSTRACT
A COMPREHENSIVE CONTEXT-AWARE INTERRUPTION
MANAGEMENT SYSTEM

April Chunjiao Song, B.S.

Marquette University, 2017

The interruption system is an application that prevents the user from noticing phone calls when he/she is busy, by turning off the ringtone. In a previous project, the user can enter his/her class and work schedule on Google Calendar. The intelligent interruption system can detect if the current time matches the range of one of the events in the user's Google Calendar. Other contexts considered were: driving, relationship of the callers, and proximity of Bluetooth devices.

This project is a continuation of the interruption system. We consider additional context, social media such as Twitter. Research is done on when is the best time to turn off the ringer when the user is using Twitter. If the user is using social media, the user isn't as busy compared to, if the user is in class or at work. We further granularize social media activity such as reading messages, writing messages, and use these to help predict interruptions. We use the feedback provided by the user and employ machine learning approach which takes as input the different contexts and predicts if the user should be interrupted. We implemented a prototype application on Android operating system.

ACKNOWLEDGEMENTS

April Chunjiao Song, B.S.

I would like to thank my thesis advisor professor Praveen Madiraju for giving guidance for this research. I would also like to thank professor Sheikh Iqbal Ahamed for guiding me. Thank you to professor Thomas "Tom" Kaczmarek for also being a committee member. Thanks to the people who took my survey. Last but certainly not least, I thank my parents Linsheng Song and Qi Zhang. I would not have made it this far without their support.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
LIST OF TABLES	iii
LIST OF FIGURES	iv
CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Scopes and Objectives of the Research	3
1.4 Organization of the Thesis	4
CHAPTER 2: PREVIOUS RESEARCH	6
2.1 Literature Review	6
2.2 Proposals	7
2.3 Contexts	8
CHAPTER 3: MODEL OF THE PROGRAM	11
3.1 Summary of Code	11
3.2 Data Mining	21
CHAPTER 4: IMPLEMENTATION OF THE PROGRAM	26
4.1 Diagrams	26
4.2 Twitter Activities	27
CHAPTER 5: SYSTEM EVALUATION	32
5.1 Test Cases and Time	32
5.2 Survey Results	34
5.3 Survey Summary	46
CHAPTER 6: CONCLUSIONS	49
6.1 Conclusions	49
6.2 Future Works	50
BIBLIOGRAPHY	51
APPENDIX	53

LIST OF TABLES

Table 1: Fixed and Mobile Subscriptions in Millions	3
Table 2: Fixed and Mobile Subscriptions Per 100 Inhabitants	3
Table 3: SQLite Database Example	22
Table 4: SQLite Database Example Continued	23
Table 5: Survey Question 1	47
Table 6: Survey Question 3	47
Table 7: Survey Question 4	47
Table 8: Survey Question 5	47
Table 9: Survey Question 6	47
Table 10: Survey Question 7	48
Table 11: Survey Question 8	48

LIST OF FIGURES

Figure 1: Architecture diagram	26
Figure 2: UML aggregation diagram	27
Figure 3: Screenshots 1	28
Figure 4: Screenshots 2	29
Figure 5: Screenshots 3	30
Figure 6: Screenshots 4	31

CHAPTER 1

Introduction

1.1 Introduction

The purpose of this project is to research how useful an intelligent interruption application will be when people are using Twitter. If the users still prefer to have their ring tones turned off sometimes when they're using social networking, it will be a useful app that people can download.

Research has been done on an interruption system that turns off the ringer when the user is in class or at work. My project will be a continuation of this. [1]

A related work includes Adamczyk, Iqbal, and Bailey at University of Illinois-Urbana Champaign on an interruption system that makes judgments based on the users' pupils. [2]

Mewhort, Cann, Slater, and Naughton wrote a book called High Performance Computing Systems and Applications in which they also measured the users' pupils to detect whether or not the users are busy. [3]

This application uses data mining to make predictions on when is the best time to turn the ringer off. The program has awareness of what kind of activity the user is currently doing.

Previous students have contributed to this project by having the program turn the ringtone off if the users are busy, and make that judgment by the events the users have entered into their calendar. The user has the option of using a white list to make exceptions when it comes to turning off the ringtone. This

interruption system is connected to Twitter and uses data mining to learn when is the best time to turn off the ringer.

1.2 Motivation

More people have mobile phones than before. It used to be a luxury, now it's considered by many to be a necessity. According to the International Telecommunication Union, the number of fixed-telephone subscriptions in 2005 in developed countries is 570 million, and decreased to approximately 471 million subscriptions in 2016. The number of fixed-telephone subscriptions worldwide is 1,243 million in 2005. It decreased to about 1,013 million in 2016. The number of mobile-cellular telephone subscriptions in developed countries in 2005 was 992 million, and increased to approximately 1,600 million in 2016. Overall, there were 2,205 million mobile subscriptions worldwide in 2005. It increased to an estimated 7,377 million in 2016. Since mobile telephones have become more popular, it's more likely that people will be distracted by them, since they can be carried with them instead of being restricted in their homes. Cell phones have made it easier to communicate, but can be a nuisance because of the distractions and addiction it can potentially cause. The motivation behind an intelligent interruption system is to reduce the distractions. [4]

Table 1. Fixed and mobile subscriptions in millions

	(millions)											
	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016*
Fixed-telephone subscriptions												
Developed	570	565	546	544	562	553	540	526	510	498	484	471
Developing	673	696	708	705	692	676	661	652	628	593	565	542
World	1,243	1,261	1,254	1,249	1,254	1,229	1,201	1,178	1,138	1,090	1,049	1,013
Mobile-cellular telephone subscriptions												
Developed	992	1,127	1,243	1,325	1,383	1,404	1,411	1,447	1,481	1,536	1,577	1,600
Developing	1,213	1,618	2,125	2,705	3,257	3,887	4,453	4,785	5,185	5,470	5,638	5,777
World	2,205	2,745	3,368	4,030	4,640	5,290	5,863	6,232	6,666	7,006	7,216	7,377

Table 2. Fixed and mobile subscriptions per 100 inhabitants

	Per 100 inhabitants											
	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016*
Fixed-telephone subscriptions												
Developed	47.2	46.6	44.8	44.3	45.5	44.6	43.4	42.2	40.8	39.7	38.5	37.3
Developing	12.7	13.0	13.0	12.8	12.4	11.9	11.5	11.2	10.6	9.9	9.3	8.8
World	19.1	19.2	18.8	18.5	18.4	17.8	17.2	16.7	15.9	15.1	14.3	13.7
Mobile-cellular telephone subscriptions												
Developed	82.1	92.9	102.0	107.8	112.1	113.3	113.5	116.0	118.4	122.7	125.7	126.7
Developing	22.9	30.1	39.1	49.0	58.2	68.5	77.4	82.1	87.8	91.4	93.0	94.1
World	33.9	41.7	50.6	59.7	68.0	76.6	83.8	88.1	93.1	96.8	98.6	99.7

About 1 in 4 accidents were caused by using cell phones while driving, including but not limited to texting. [6] According to some employers, the main

cause of distractions in the workplace are cell phones, including texting. Other causes include the internet, gossip, social media, and emails. These causes are not as common as cell phones, but they may be related, because some people use cell phones for the internet. Internet includes social media and emailing. This study surveyed 2,175 employers and human resources managers. [7] Studies using driving simulators have shown:

- Talking on the cell phone while driving slows the reaction time.
- Using a hands-free cell phone can potentially cause as many accidents as handheld cell phones, because they create equal distractions.
- Using cell phones while driving increases the chance of crashing by about 4 times.
- About more than 4 out of 5 people believe that using cell phones while driving is seriously unwise.
- Over half of the people surveyed said they have occasionally used cell phones while driving. [8]

The economic cost of accidents caused by cell phones is about \$46 billion. [9]

1.3 Scopes and Objectives of the Research

In this research, we read literature of research related to intelligent interruption systems. We implemented a prototype of an intelligent interruption system, and also used a survey to have a better idea of people's opinions of when they prefer when to prevent interruptions. Overall, we have:

- a. Read papers of related research projects
- b. Decided to focus this research on Twitter activities

- c. Implemented a program
- d. Surveyed some Marquette students and employees
- e. Used data mining with Weka to make predictions based on feedback from the user

1.4 Organization of the Thesis

The beginning of this thesis contains the abstract and acknowledgements. Chapter 1 contains the potential negative impact of the increasing use of mobile phones, the motivation behind this project, and the scopes of this research. Chapter 2 contains information about previous related research and plans for this current research. Chapter 3 gives a description of the code in the program. Chapter 4 has screenshots of the app. Chapter 5 includes the survey results. The bibliography is at the end of this thesis. The appendix contains the remainder of the survey responses.

CHAPTER 2

Previous Research

The first mobile phone was invented and created in 1973 by Martin Cooper. It cost Motorola \$10,000 to create it, which is proportionally equivalent to about \$1 million today. It weighed 2 and a half pounds. In 2008, about half of the people in the world owned mobile phones. [10]

2.1 Literature Review

As mentioned before, Zulkernain, Madiraju, Ahamed, and Stamm did research on an interruption system that turns off the ringer based on the schedule the user entered. University of Illinois-Urbana Champaign did research on an interruption system that makes judgments based on the users' pupils. Their study indicates that the pupils are more dilated when the user is busier. Their interruption system focuses on making judgments by physiological signs rather than the user's entered schedule. [2] Brian Bailey and Shamsi Iqbal also tested research subjects by having them do tasks such as route planning, document editing, and email classification. Their goal was to allow phone interruptions when people have lower workloads. They believed the best time is when people are at subtask boundaries. [11] The textbook High Performance Computing Systems and Applications mentioned about also measuring the users' pupils to detect whether or not the users are busy. In addition, they measured other physiological signs, such as facial expressions, that indicate whether or not the user feels stressed out. The interruption system turns off the ring tone if there are

physiological signs that the user feels preoccupied and stressed. When the user feels stressed, the data entered will be nonlinear. Their program uses machine learning to learn the user's habits. [3]

Kaykobad from Marquette University did research on an interruption system that makes predictions using the Bayesian model. The user can enter the probability of interruption for a time range to decrease or increase the chances of the phone ringing, on a scale of 1 to 10. They can enter different predicted levels of interruptions based on groups they're created, such as Friends, Family, and Colleagues. The user can also set a probability when driving, and set up Bluetooth devices. So if it senses the entered Bluetooth device nearby, it will decrease the chances of it ringing. There is an option for setting the predicted level of interruption when the user is using Twitter or Facebook. [4]

2.2 Proposals

This thesis proposes to reduce interruptions by using data mining. Data mining examines data in databases to create useful information. For example, data mining may reach conclusions about how to cut costs and to find correlations between different types of data in the database. A grocery chain in the Midwest concluded from data mining with Oracle that men that shopped at those stores on Thursdays who bought diapers usually bought beer too. So they put the locations of the diaper and beer displays closer to each other and didn't give discounts to those items on Thursdays. Walmart also uses data mining to observe the customers' buying patterns.

Data mining can use classes, clusters, associations, or sequential patterns. An example of association is the diaper and beer example. Classes are groups, similar to classes in programming. Sequential patterns are similar to associations in which both make predictions of what customers will buy based on what they already bought.

Data mining extracts the data into a database warehouse, stores the data in a database, gives access to the data to business analysts, they analyze the data using software, and present the results in a graph or table.

There are different types of data mining, such as artificial neural networks, genetic algorithms, decision trees, nearest neighbor method, rule induction, and data visualization. Artificial neural networks are nonlinear and have a similar structure as biological neural networks. Genetic algorithms use combinations and mutations to make predictions. Decision trees can also be used for data mining. The branches in decision trees are similar to if, else if, and else statements in programming. The nearest neighbor method uses classes, and makes predictions based on the other classes that are most similar to the current class. If there are more data and queries, a larger and more powerful system is needed. [12]

This thesis will do data mining using Weka. If a user receives a call and the ringtone gets turned off, the program will ask him/her whether it was useful. It uses this feedback to make predictions of whether or not to turn off the ringer in the future. The feedback in the database also includes information about what activity the user was doing at the time. Weka is a program that does machine learning. It can be run itself or its library can be called using Java. Its machine

learning includes regression, clustering, and classification. It was created by University of Waikato, and is named after a bird. [13]

The lazy function delays making generalizations of data until it receives a query asking for the information, which is the moment it is required to process the data. The lazy function takes more space since it will have to store the data to be processed later. [14]

The structure of an artificial neural network is similar to a biological nervous system. Each path has a different weight, and can be changed by learning. [17]

I initially tried artificial neural networks, and continued using it because it usually has a 0% error rate when used with a training set.

2.3 Contexts

This program mostly focuses on Twitter activities. The Twitter activities are: Update status, Show status, Get direct messages, Send direct messages, Get friends IDs, View profile, Get followers IDs, and Show information. For example if the user says that turning off the ringer when he/she is viewing someone else's profile, which is View profile, isn't helpful, then it's more likely that their ringer will be on when they're viewing other people's profiles on Twitter. As mentioned before, the program uses SQLiteDatabase in Android Studio for storing the data, and Weka to do data mining. This program enables the user to use Twitter features, such as viewing their Tweets and messages by using the Twitter4J library. Twitter4J is used by Java programs to communicate and receive information with Twitter. [15]

This program also includes features from the previous version of the intelligent interruption system. Those features include the user's schedule, their preferences for the weekdays and weekends, and time of day. For example, the user might prefer having their ringer turned off when they're in class. They're okay when their phone rings during the weekends because they're less busy during that time. Or the user might prefer having their ringer turned off late at night and early in the morning because that's when they're sleeping.

Surveying was also done about sentiment analysis. Sentiment analysis will attempt to collect information about the user's emotions based on their Tweets and messages. For example, if their recent Tweets or messages contain words such as "sad", "upset", or "depressed", then the probability that their phone rings when they receive a call will be smaller.

CHAPTER 3

Model of the Program

3.1 Summary of Code

This program gives the user the option of doing some activities on Twitter. Each activity that's available on this program is in a separate class. The MainActivity.java class is the index activity called "Nothing". It contains buttons that can activate the other activities. The `setOnClickListener(...)` method used on each button looks like this:

Main activity:

```
@Override
public void onClick(View v)
{
    Intent intent = new Intent(MainActivity.this,
Activity.class);
    intent.putExtra("Consumer key", CONSUMER_KEY);
    intent.putExtra("Consumer secret", CONSUMER_SECRET);
    intent.putExtra("Access token", ACCESS_TOKEN);
    intent.putExtra("Access secret", ACCESS_SECRET);
    intent.putExtra("Screen name", MY_SCREEN_NAME);

    activity = "Activity";

    startActivityForResult(intent, 1);
}
```

The consumer key, consumer secret, access token, access secret, and screen name are information that are unique to each Twitter account. The user needs to create an application on Twitter App and connect their Twitter account to it to generate the consumer key, consumer secret, access token, and access secret. The screen name is the user's username. For example, if the user presses View Profile, the ViewProfile.java file will be opened.

The MainActivity class detects if the phone is ringing by using a phone call listener. If the user receives a phone call, the Alert class is opened if the ringer is silent. Here is some code from the Alert.java file:

```
@Override
public void onClick(View view)
{
    int feedback = 1;
    User user = new User(database.getCount(), feedback,
activity);
    database.createUser(user);
    Toast.makeText(getApplicationContext(), "Added 'Yes'.",
Toast.LENGTH_SHORT).show();
    finish();
}
```

The Alert window asks the user if turning off the ringer was helpful that time. The above code is when the user presses "Yes". The feedback can be 1 or 0. 1 is yes, and 0 is no. The information is entered into the database with the feedback and current activity. Then a message appears saying that the data has been added, and the Alert window closes. The Database.java file contains the following code for the createUser(...) method:

```
public void createUser(User user)
{
    SQLiteDatabase database = getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(user.getColumn(), user.getWasHelpful());

    values.put("ColumnName", user.getColumn());
    database.insert(TABLE_NAME, null, values);
    database.close();
}
```

This method inserts the feedback into the matching column. The columns in the database consists of the primary key, the activity names, and the column the feedback was inserted into. For example, suppose the user was sending a message on Twitter, his/her phone didn't ring when someone was calling, and he/she said it

was helpful. Then 1 will be put in the "SendDirectMessages" column and "SendDirectMessages" will be put in the "ColumnName" column.

Get direct messages:

```
@Override
public void run()
{
    try
    {
        messages = twitter.getDirectMessages();
        sentmessages = twitter.getSentDirectMessages(paging);

        runOnUiThread(new Runnable()
        {
            public void run()
            {
                messageadapter = new MessagesAdapter();
                listview.setAdapter(messageadapter);

                sentmessageadapter = new
                SentMessagesAdapter();
                listview2.setAdapter(sentmessageadapter);
            }
        });
    }
    catch (TwitterException exception)
    {
        exception.printStackTrace();
    }
}
```

The getDirectMessages() method is used to get the received messages.

The getSentDirectMessages(...) method gets the messages the user sent. The message adapter and sent message adapter sets the format for how the messages are displayed. Most functions of the Twitter4J API can potentially throw a Twitter exception.

Get followers IDs:

```
long cursor = -1;

try
{
    friend = twitter.getFollowersIDs(MY_SCREEN_NAME, cursor);
    long[] friendsarray = friend.getIDs();
}
```

```

    User user;
    for(int i=0; i < friendsarray.length; i++)
    {
        user = twitter.showUser(friendsarray[i]);
        users.add(user);
    }
}
catch(TwitterException exception)
{
    exception.printStackTrace();
}

```

The `getFollowersIDs(...)` is used to retrieve the followers, and the `getIDs()` method converts it to long format. The `showUser()` method displays the username of each user ID.

Get friends IDs:

```

friend = twitter.getFriendsIDs(MY_SCREEN_NAME, cursor);

long[] friendsarray = friend.getIDs();
User user;

for(int i=0; i < friendsarray.length; i++)
{
    user = twitter.showUser(friendsarray[i]);
    users.add(user);
}

```

The `getFriendsIDs(...)` method retrieves the people the user is following, and `getIDs()` converts the information to numbers. `showUser(...)` displays the usernames of those IDs.

Phone call listener:

```

@Override
public void onCallStateChanged(int state, String number)
{
    phonenumber = number;
    switch(state)
    {
        case TelephonyManager.CALL_STATE_IDLE:
            callstate = "Call state idle";
            break;
        case TelephonyManager.CALL_STATE_OFFHOOK:
            callstate = "Call state offhook";
            break;
        case TelephonyManager.CALL_STATE_RINGING:
            callstate = "Call state ringing";

```

```

        break;
    }
}

```

The `onCallStateChanged(...)` method inherits the `onCallStateChanged(...)` method from the `PhoneStateListener` class and detects if the call state has changed. If the call state is ringing, it means the user has received a call, and it can detect the phone number that the user received the call from.

Send direct messages:

```

try
{
    screenname = editText.getText().toString();
    message = editText2.getText().toString();

    twitter.sendDirectMessage(screenname, message);
    runOnUiThread(new Runnable()
    {
        public void run()
        {
            Toast.makeText(getApplicationContext(), "Your
message has been sent.", Toast.LENGTH_SHORT).show();
        }
    });
}
catch (TwitterException twitterexception)
{
    twitterexception.printStackTrace();
    runOnUiThread(new Runnable()
    {
        public void run()
        {
            Toast.makeText(getApplicationContext(),
"Twitter exception error.", Toast.LENGTH_SHORT).show();
        }
    });
}

```

`getText().toString()` gets the text of the username the message is for and the message and converts them to strings. The `sendDirectMessage(...)` method sends the message. If it's successfully sent, a pop up message appears saying that it was sent.

Show status:

```

private class TweetsAdapter extends ArrayAdapter<Status>

```

```

{
    public TweetsAdapter()
    {
        super(ShowStatus.this, R.layout.layout, statuses);
    }

    @Override
    public int getCount()
    {
        return statuses.size();
    }

    @Override
    public View getView(int position, View view, ViewGroup
parent)
    {
        if(view == null)
        {
            view =
getLayoutInflater().inflate(R.layout.layout, parent, false);
        }
        TextView textviewadapter = (TextView)
view.findViewById(R.id.helloworldadapter);
        TextView textviewadapter2 = (TextView)
view.findViewById(R.id.helloworldadapter2);
        textviewadapter.setText(statuses.get(position).getUser().ge
tScreenName());
        textviewadapter2.setText(statuses.get(position).getText());

        return view;
    }
}

```

The TweetsAdapter class uses the ArrayAdapter<Status> interface. It takes the text views of the screen name and status and sets them to the screen name and status of each element in the statuses array. In other words, it formats each element displayed in the list to have the status and the screen name of the person who posted the status.

Show user:

```

final User user2 = twitter.showUser(MY_SCREEN_NAME);
runOnUiThread(new Runnable()
{
    public void run()
    {

```

```

        name.setText(user2.getName());
        description.setText(user2.getDescription());
        location.setText(user2.getLocation());

        URLEntity urlentity = user2.getURLEntity();
        website.setText(urlentity.getExpandedURL());
        String join = "Joined " + user2.getCreatedAt();
        joined.setText(join);
    }
});

```

The `showUser(...)` method is used to get basic information about the user. `getName()` gets the user's username, `getDescription()` gets the description in the user's profile, the `getLocation()` method gets the location, the `getURLEntity()` method gets information about the website, and `getCreateAt()` gets the date the profile was created. `getExpandedURL()` converts the URL information to a string.

Update status:

```

String message = edittext.getText().toString();

twitter.updateStatus(message);
runOnUiThread(new Runnable()
{
    public void run()
    {
        Toast.makeText(getApplicationContext(), "Your Tweet has
been posted.", Toast.LENGTH_SHORT).show();
    }
});

```

The `updateStatus(...)` method adds the Tweet the user enters and posts it on his/her Twitter page. If it was successfully posted, a popup message appears confirming that. If no message appears, most likely the program isn't connected to the internet.

User:

```

public User(int rowid, int feedback, int feedback2, int feedback3,
int feedback4, int feedback5, int feedback6, int feedback7, int
feedback8, int feedback9, String columnname)
{
    id = rowid;
    washelpful = feedback;
}

```



```

        washelpful2 = feedback2;
        washelpful3 = feedback3;
        washelpful4 = feedback4;
        washelpful5 = feedback5;
        washelpful6 = feedback6;
        washelpful7 = feedback7;
        washelpful8 = feedback8;
        washelpful9 = feedback9;

        column = columnname;
    }

```

The User class contains information about the row number, feedback, and column name. feedback refers to the first feedback column, feedback2 refers to the second feedback column, feedback3 refers to the third, etc. The first feedback column is Nothing, the second feedback column is GetDirectMessages, the third is GetFollowersIDs, etc. The User(int rowid, int feedback, String columnname) constructor is used when inserting information into the database. The other constructor is used when retrieving data from a database.

View profile:

```

button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        Intent intent2 = new Intent();
        username = editText.getText().toString();
        activity = "View profile";
        intent2.putExtra("Username", username);

        intent2.putExtra("Activity", activity);

        setResult(RESULT_OK, intent2);
        isviewprofile = true;
        finish();
    }
});

```

If the user wants to view someone's profile, they can enter the person's username and press View Profile. The username is sent to MainActivity and the activity is set to "View profile". The MainActivity.java file can detect if the

activity has been changed to "View profile". If that happens, it will open the ViewProfile2 which displays the requested profile.

View profile 2:

```
Thread thread = new Thread(new Runnable()
{
    @Override
    public void run()
    {
        try
        {
            responselist =
twitter.getFavorites(MY_SCREEN_NAME);
            if(responselist.size() == 0)
            {
                like = "Like";
            }
            for (int i = 0; i < responselist.size(); i++)
            {
                if (responselist.get(i).getId() == id2)
                {
                    like = "Unlike";
                    break;
                }
                else
                {
                    like = "Like";
                }
            }
        }
        catch (TwitterException exception)
        {
            exception.printStackTrace();
            runOnUiThread(new Runnable()
            {
                @Override
                public void run()
                {
                    Toast.makeText(getApplicationContext(), "Error.",
Toast.LENGTH_SHORT).show();
                }
            });
        }
    }
});
thread.start();
```

The `getFavorites(...)` method retrieves the Tweets the user "liked." If it's empty, that means there are no Tweets the user has "liked" so far, so the option will say "Like". Otherwise, there is a for loop that examines each Tweet the user "liked" to see if there is one with an ID that matches the ID of the selected Tweet. If it found a match, that means that the Tweet the user selected was already "liked" and the user will be given the option to "Unlike" the Tweet.

Weka:

```

file = new File(Environment.getExternalStorageDirectory() +
File.separator + "twitter2.txt");
filearff = new File(Environment.getExternalStorageDirectory() +
File.separator + "twitter2.arff");

InputStream inputstreamarff = null;
try
{
    inputstreamarff = new FileInputStream(filearff);
}
catch(FileNotFoundException filenotfoundexception)
{
    filenotfoundexception.printStackTrace();
}

BufferedReader bufferedreaderarff = null;
try
{
    bufferedreaderarff = new BufferedReader(new
InputStreamReader(inputstreamarff, "UTF-8"));
}
catch(UnsupportedEncodingException exception)
{
    exception.printStackTrace();
}
instances = null;
try
{
    instances = new Instances(bufferedreaderarff);

    bufferedreaderarff.close();
}
catch(IOException exception)
{
    exception.printStackTrace();
}

instances.setClassIndex(instances.numAttributes() - 1);

MultilayerPerceptron multilayerperceptron = new MultilayerPerceptron();

Evaluation evaluation = null;

try

```

```

{
    multilayerperceptron.buildClassifier(instances);
    evaluation = new Evaluation(instances);

    evaluation.evaluateModel(multilayerperceptron, instances);
}
catch(Exception exception)
{
    exception.printStackTrace();
}

```

The Weka class opens the twitter.arff file, which also contains the feedback data. This type of file is needed for Weka. This code used the multilayer perceptron, which is also known as artificial neural networks. The classifier is created using the instances created from reading the file. The evaluateModel(...) method returns the predictions as an array. The errorRate() method is used to print the error rate. When it was run, it usually said the error rate was 0.

3.2 Data Mining

The SQLite database is used to store data in the application. The data is also saved in ARFF files to be used when using Weka for data mining. Here is a preview of what an ARFF file looks like:

```

% April Song
@RELATION twitter
@ATTRIBUTE id INTEGER
@ATTRIBUTE nothing INTEGER
@ATTRIBUTE getdirectmessages INTEGER
@ATTRIBUTE getfollowersids INTEGER
@ATTRIBUTE getfriendsids INTEGER
@ATTRIBUTE senddirectmessages INTEGER
@ATTRIBUTE showstatus INTEGER
@ATTRIBUTE showuser INTEGER
@ATTRIBUTE updatestatus INTEGER
@ATTRIBUTE viewprofile INTEGER
@ATTRIBUTE columnname {Nothing, GetDirectMessages, GetFollowersIDs,
GetFriendsIDs, SendDirectMessages, ShowStatus, ShowUser, UpdateStatus,
ViewProfile}
@DATA
1, -1, -1, -1, -1, -1, -1, -1, -1, 1, ViewProfile
2, -1, -1, -1, -1, -1, -1, -1, 1, -1, UpdateStatus

```

3, -1, -1, -1, -1, -1, -1, 1, -1, -1, ShowUser
 4, -1, -1, -1, -1, -1, 1, -1, -1, -1, ShowStatus
 5, -1, -1, -1, -1, 1, -1, -1, -1, -1, SendDirectMessages
 6, -1, -1, -1, 1, -1, -1, -1, -1, -1, GetFriendsIDs
 7, -1, -1, 1, -1, -1, -1, -1, -1, -1, GetFollowersIDs
 8, -1, 1, -1, -1, -1, -1, -1, -1, -1, GetDirectMessages
 9, 1, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, ViewProfile

@RELATION is used to state the name of the file, which in this case is twitter.arff. @ATTRIBUTE refers to the name of each column. Most of the attributes are integers. The attribute columnname is nominal and has the column names as the options. @DATA is where the data is stored. For example, the first row means that 1 is entered into the column with the unique key, and 1 is inserted into the column ViewProfile which means that it was helpful turning off the ringer when the user was viewing someone's profile. The rest of the columns in that row contain -1, which means information weren't entered there.

Here is what the SQLite database looks like:

Table 3. SQLite database example

Id	Nothing	GetDirectMessages	GetFollowersIDs	GetFriendsIDs	SendDirectMessages
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1
4	-1	-1	-1	-1	-1
5	-1	-1	-1	-1	1
6	-1	-1	-1	1	-1

7	-1	-1	1	-1	-1
8	-1	1	-1	-1	-1
9	1	-1	-1	-1	-1
10	-1	-1	-1	-1	-1

Table 4. SQLite database example continued

ShowStatus	ShowUser	UpdateStatus	ViewProfile	ColumnName
-1	-1	-1	1	ViewProfile
-1	-1	1	-1	UpdateStatus
-1	1	-1	-1	ShowUser
1	-1	-1	-1	ShowStatus
-1	-1	-1	-1	SendDirectMessages
-1	-1	-1	-1	GetFriendsIDs
-1	-1	-1	-1	GetFollowersIDs
-1	-1	-1	-1	GetDirectMessages
-1	-1	-1	-1	Nothing
-1	-1	-1	0	ViewProfile

The code used for making the predictions:

```

if(activity.equals(elementsfinal.get(i)))
{
    elementsfiltered.add(elementsfinal.get(i));
    if(activity.equals("Nothing"))

```

```

    {
        elementsfiltered.add(elementsfinal.get(i-9));

        position = 1;
    }
    else if(activity.equals("GetDirectMessages"))
    {
        elementsfiltered.add(elementsfinal.get(i-8));
        position = 2;
    }
    else if(activity.equals("GetFollowersIDs"))
    {
        elementsfiltered.add(elementsfinal.get(i-7));
        position = 3;
    }
    else if(activity.equals("GetFriendsIDs"))
    {
        elementsfiltered.add(elementsfinal.get(i-6));
        position = 4;
    }
    else if(activity.equals("SentDirectMessages"))
    {
        elementsfiltered.add(elementsfinal.get(i-5));
        position = 5;
    }
    else if(activity.equals("ShowStatus"))
    {
        elementsfiltered.add(elementsfinal.get(i-4));
        position = 6;
    }
    else if(activity.equals("ShowUser"))
    {
        elementsfiltered.add(elementsfinal.get(i-3));
        position = 7;
    }
    else if(activity.equals("UpdateStatus"))
    {
        elementsfiltered.add(elementsfinal.get(i-2));
        position = 8;
    }
    else
    {
        elementsfiltered.add(elementsfinal.get(i-1));
        position = 9;
    }
    foundmatch = true;
}

```

At this point, the program has retrieved the predicted values and stored them into the elementsfinal array list. Then the information is filtered so that the elementsfiltered list contains the information with the activity that matches the current activity. Then one of the information will be randomly selected to create the prediction. For example, if the current activity is Nothing, the information in

the filtered array list will be the following predictions if the information is predicted completely correctly:

```
11, 1, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
12, 1, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
13, 1, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
14, 0, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
15, 0, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
16, 0, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
17, 1, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
18, 0, -1, -1, -1, -1, -1, -1, -1, -1, Nothing
```

Then the program will randomly choose a row. In this case, there is a 50% chance that the ringer will be turned off when the user is receiving a call while doing nothing on Twitter.

```
Instance denseinstance = new DenseInstance(11);
denseinstance.setDataset(instances);
denseinstance.setValue(position, feedback);
denseinstance.setValue(0, (previouskey+1));

denseinstance.setValue(10, activity);
for(int i=1; i < 10; i++)
{
    if(i != position)
    {
        denseinstance.setValue(i, -1);
    }
}

instances.add(denseinstance);
```

Then the data in the ARFF file will be updated. The DenseInstance object instantiated has 11 elements for 11 columns. The setDataset(...) method enters the instances, which is the information that already existed before the app was run. The setValue(...) method adds the new feedback in the corresponding position. The unique key number is added into the first column. The name of the activity is added in the last column. The rest of the columns are filled with -1.

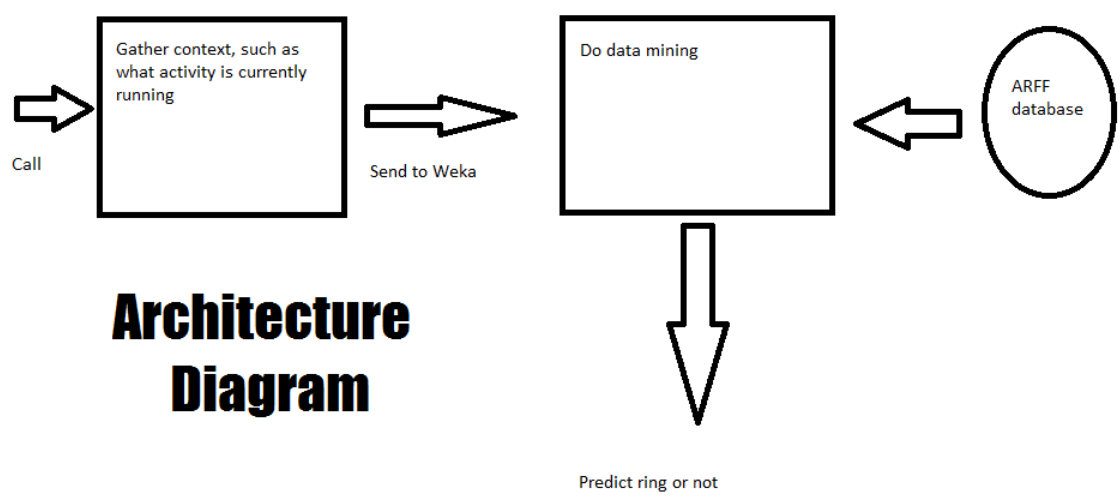
CHAPTER 4

Implementation of the Program

4.1 Diagrams

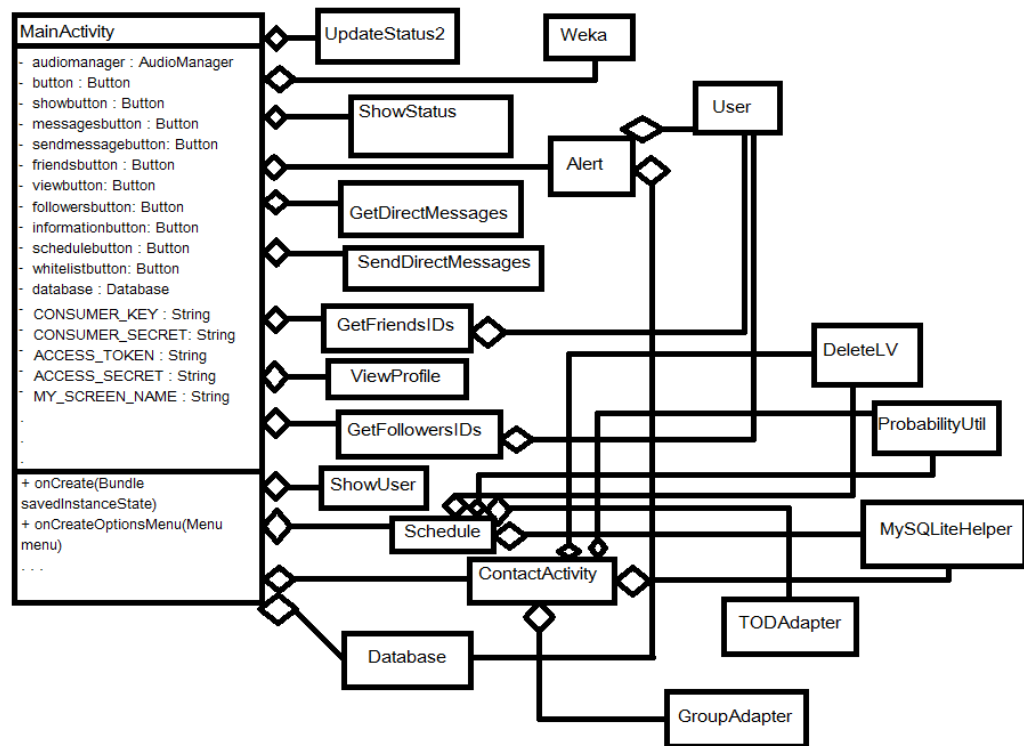
Here is the architecture diagram of the app:

Figure 1. Architecture diagram



The MainActivity class keeps track of what activity the user is currently doing, even if the user has opened a different class. If the MainActivity receives a call, it sends the context of the current activity to the Weka class. The Weka class uses the Weka library to make a prediction using data mining. It retrieves the data from an ARFF file and writes new data to the ARFF file if the user enters feedback. Its prediction is either 1 or 0.

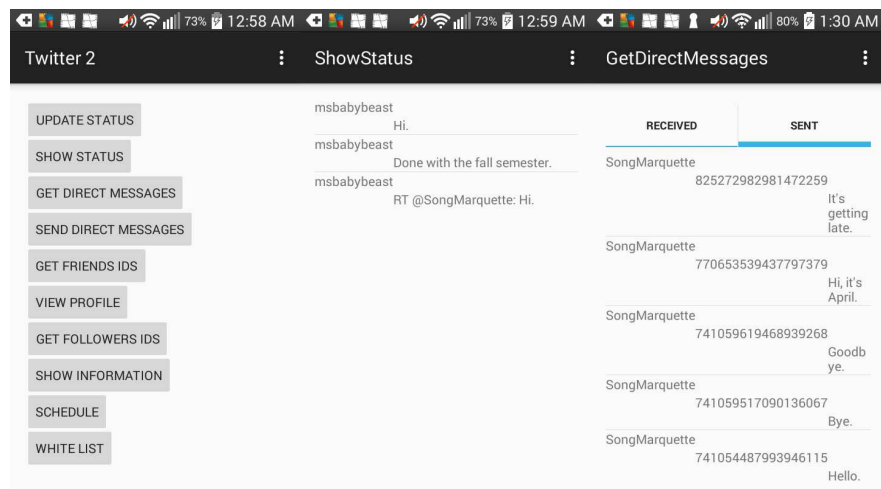
Figure 2. UML aggregation diagram

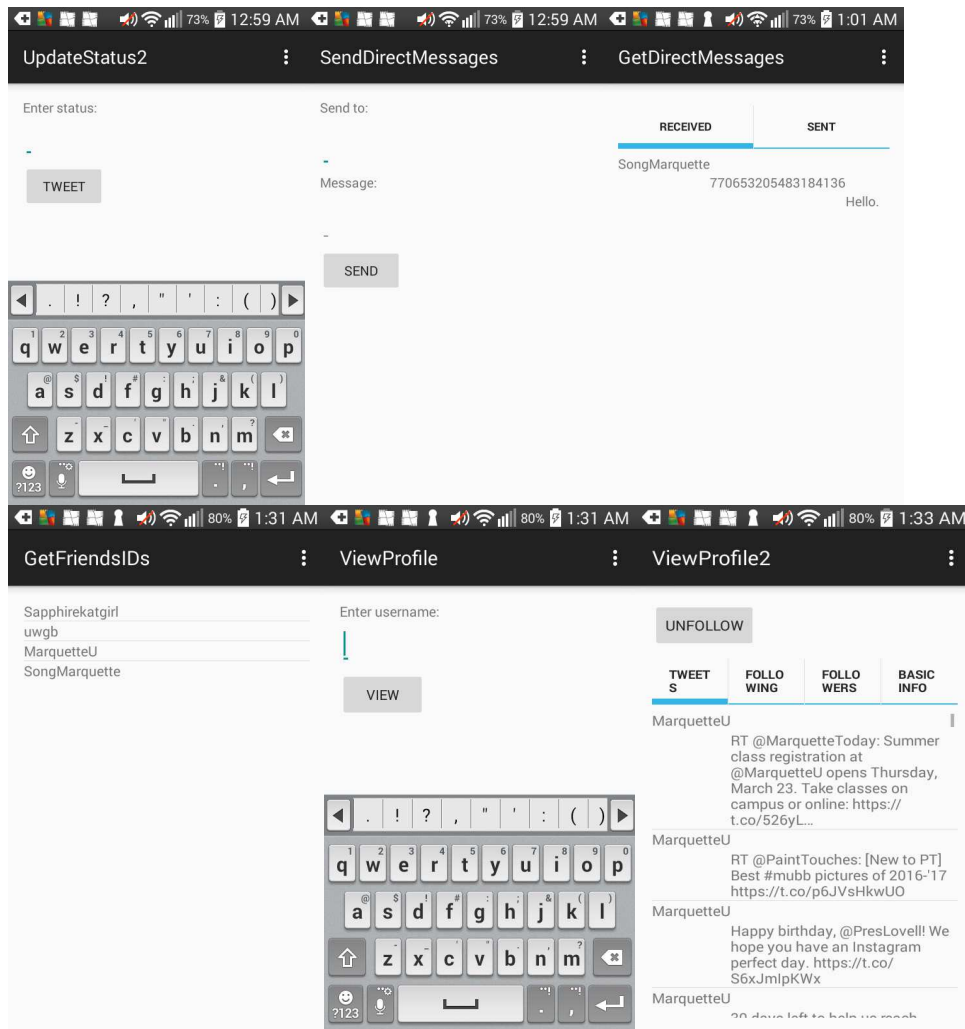


The MainActivity class aggregates UpdateStatus2, ShowStatus, GetDirectMessages, SendDirectMessages, GetFriendsIDs, ViewProfile, GetFollowersIDs, ShowUser, Schedule, ContactActivity, Weka, and Alert. Some of the classes were taken from the previous version of the intelligent interruption system that wasn't created by me. [5] The classes are included in Intents. Not all of the variables and methods in the MainActivity were included in the diagram. Some of the classes MainActivity has also aggregates some of the classes created in the program. When it comes to inheritance, all of the activity classes inherit from the AppCompatActivity class.

4.2 Twitter Activities

Figure 3. Screenshots 1





The app allows the user to do activities considered to be important, such as viewing someone's profile, adding and viewing the user's Tweets, and viewing and sending messages. Unlike the official Twitter app, this application can detect what kind of activity on Twitter the user is doing.

Figure 4. Screenshots 2

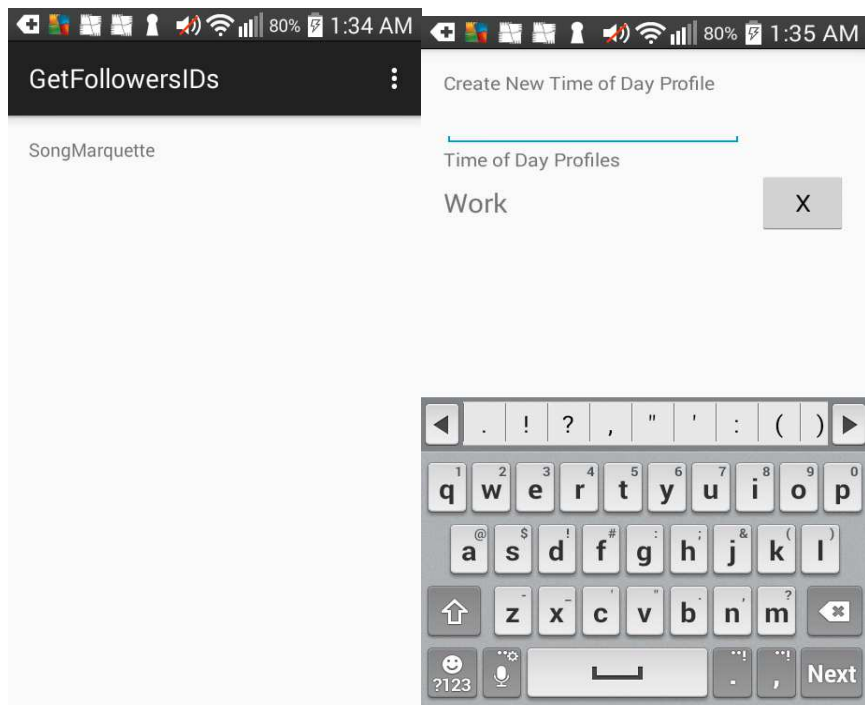


Figure 5. Screenshots 3

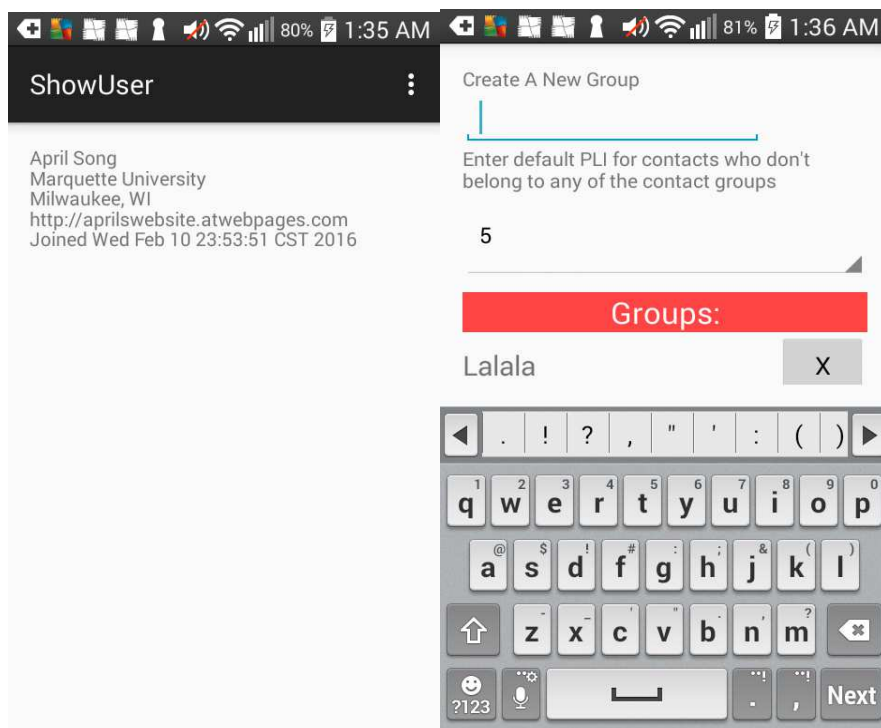
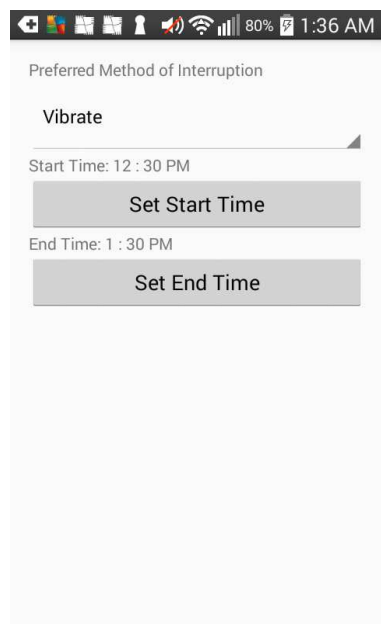
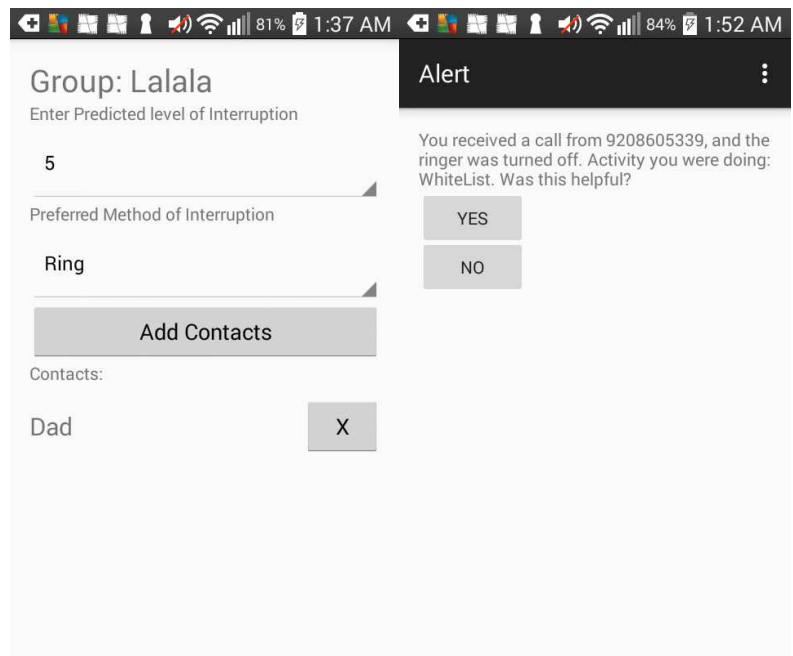


Figure 6. Screenshots 4



CHAPTER 5

System Evaluation

5.1 Test Cases and Time

I used a second phone to call the phone with the program. I used SECOND in the Calendar class to do the timing. It takes about 1 second for everything to be created and instantiated when I open my app.

Here are some test cases:

Scenario 1:

April is busy chatting with her friend via messaging. While she's chatting, her phone rings. April is preoccupied with her Twitter conversation and ignores the call. Therefore, the purpose of the ring would have been more meaningful if it didn't ring. By default, the phone's ringer will be turned off. However, if the user receives a phone call, it's possible to send a feedback saying that the silence was helpful in the future when receiving a call in that circumstance.

Activities: Get Direct Messages and Send Direct Messages

Predicted: Ringer off

Actual: Ringer off

Scenario 2:

Ashley is busy messaging her team members about a group project they're been working on. She usually communicates about the group project via Twitter because the other team members check their Twitter accounts more frequently

than their school accounts. When her phone rings, she ignores it and also feels slightly irritated because she's busy discussing about the project.

Activity: Get Direct Messages

Predicted: Ringer off

Actual: Ringer off

Scenario 3:

Michael is looking at posts from the people he's following. His phone rings. He doesn't have a problem with it and answers his phone. After all, he's only looking at people's posts and pictures, which is very unproductive. He also realizes that he has spent more time looking at Tweets than he initially planned.

Activity: View Profile

Predicted: Ringer on

Actual: Ringer off

5.2 Survey Results

24 people who were at least 18 years old were surveyed for the first survey.

Here are some of the results [16]:

Respondent #1:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.5
- Show status 0.8
- Get direct messages 0.1
- Send direct messages 0.9
- View profile 0.5
- Miscellaneous 0.2

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Bad interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Bad interruption

Q9: If you think another activity/activities should be included, what are they?
Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

25

Respondent #2:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.4
- Show status 0.5
- Get direct messages 0.6
- Send direct messages 0.7
- View profile 0.2
- Miscellaneous 0.6

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Bad interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

5

The other survey results can be found at the end of this thesis.

Respondent #3:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.8
- Show status 0.7
- Get direct messages 0.0
- Send direct messages 0.0
- View profile 0.5
- Miscellaneous 0.8

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

7

Respondent #4:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.0
- Show status 0.0
- Get direct messages 0.0
- Send direct messages 0.0
- View profile 0.0
- Miscellaneous 0.0

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

0

Respondent #5:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.5
- Show status 0.1
- Get direct messages 0.1
- Send direct messages 0.5
- View profile 0.1
- Miscellaneous 0.1

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

The option to view replies to Tweets

Q10: How many bad interruptions did you receive in the past week?

0

Respondent #6:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.5
- Show status 0.5
- Get direct messages 0.3
- Send direct messages 0.4
- View profile 0.7
- Miscellaneous 0.2

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Bad interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

3

Respondent #7:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.5
- Show status 0.5
- Get direct messages 0.7
- Send direct messages 0.5
- View profile 0.5
- Miscellaneous 0.5

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

n/a

Q10: How many bad interruptions did you receive in the past week?

zero

Respondent #8:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.8
- Show status 0.3
- Get direct messages 0.3
- Send direct messages 0.7
- View profile 0.5
- Miscellaneous 0.5

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Bad interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Bad interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

1

Respondent #9:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.9
- Show status 0.8
- Get direct messages 0.1
- Send direct messages 0.7
- View profile 0.2
- Miscellaneous 0.5

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

10

Respondent #10:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.9
- Show status 0.1
- Get direct messages 0.7
- Send direct messages 0.7
- View profile 0.2
- Miscellaneous 0.1

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

0

The other survey results can be found in the appendix.

5.3 Survey Summary

Table 5. Survey question 1

Answer Choices –	Responses –
–	95.83%
Yes	
–	4.17%
No	

Most of the people who took the survey said "Yes" to question 1.

The average value for Update Status in question 2 is 0.5364. The average value for Show Status in question 2 is 0.3591. The average answer for Get Direct Messages is 0.2868. The average answer for Send Direct Messages is 0.4505. The average value for View Profile is 0.3955 for question 2. The average answer for Miscellaneous is 0.3682. Based on these responses, people generally prefer their phone to be silent when typing Tweets. Overall, they mind the least when their phone rings when reading messages they received.

Table 6. Survey question 3

Answer Choices –	Responses –
–	50.00%
Good interruption	
–	50.00%
Bad interruption	

There were mixed responses to question 3: Update status.

Table 7. Survey question 4

Answer Choices –	Responses –
–	65.00%
Good interruption	
–	35.00%
Bad interruption	

The majority of people thought that receiving a call when doing Show Status was a good interruption.

Table 8. Survey question 5

Answer Choices –	Responses –
–	40.00%
Good interruption	
–	60.00%
Bad interruption	

The majority of people thought that receiving a call when doing Get Direct Messages was a bad interruption.

Table 9. Survey question 6

Answer Choices –	Responses –
–	54.17%
Good interruption	
–	45.83%
Bad interruption	

There were slightly more people who said that the ringer ringing was a good interruption than those who said it was bad, when doing Send Direct Messages.

Table 10. Survey question 7

Answer Choices –	Responses –
–	65.00%
Good interruption	
–	35.00%
Bad interruption	

The majority said that it's okay for their phones to ring if they are doing View Profile.

Table 11. Survey question 8

Answer Choices –	Responses –
–	65.00%
Good interruption	
–	35.00%
Bad interruption	

The majority said that it's okay for their phones to ring if they're doing Miscellaneous. Question 9 was optional, so most of the respondents skipped it. 4 or them said:

* Call while in a game, Google services popup while trying to type or in game (to re-authenticate into google), On desktop - cursor/mouse stealing notifications/popups.

* None

* n/a

* The option to view replies to Tweets

For question 10, the average number of bad interruptions in one week was around 5 calls. There was a second survey that asked about sentiment analysis. The average values are 0.5071 for Angry, 0.2643 for Sad, 0.2929 for Okay, and 0.2571 for Happy. Based on these results, it's generally preferable for their ringer to be on when they're feeling happy, and least preferable when they're feeling angry.

CHAPTER 6

Conclusions

6.1 Conclusions

Overall, interruption systems can be useful in preventing unwanted ringing phone calls. There are interruption systems that make judgments based on the physiological stress of the users. There are interruption systems that make judgments based on the user's calendar and white list. Additionally, Bluetooth devices, driving, and time of day can be taken into consideration. Distractions from phone calls can potentially result in accidents and loss of productivity. This research focused on Twitter activities, and briefly focused on the previous work and sentiment analysis.

The previous version and research done at Marquette University made predictions based on the Bayesian model. This research did machine learning by using data mining. Examples of machine learning include artificial neural networks, the lazy function, and Bayes. The Weka library was used for the data mining.

The prototype was created using Android Studio, which mostly uses Java and XML. The MainActivity class is the index page that receives information from the other classes, such as what activity is currently running. The feedback is stored in a ARFF database. The data mining makes predictions, filters the predictions based on the current activity, and chooses a random prediction from the filtered set.

The MainActivity class aggregates many of the other classes. Most of the classes it aggregates are Twitter activities, which can be opened by pressing the buttons on the main class.

The most overhead on the prototype was the data mining code. Based on the survey results, there were mixed responses about whether it's a good or bad interruption when typing a Tweet. When they're reading messages, the majority said that it was a bad interruption. About 50% of them preferred their ringer to be turned off when they're sending messages on Twitter.

6.2 Future Works

This research briefly worked on sentiment analysis, by asking people what their opinions of them were. In the future, if there is sentiment analysis research related to this one, it will be done by reading the user's Tweets and messages sent. Or it can implement sentiment analysis in an app by using an API. A future app can also include more features from previous related research, such as driving, Bluetooth, time of day, and physiological stress, so that it makes predictions based on many factors. Research has already been done at Marquette University on measuring people's facial expressions to predict if they're in pain, which can potentially be combined with this app. [18]

BIBLIOGRAPHY

- [1] http://www.jucs.org/jucs_16_15/a_mobile_intelligent_interruption/jucs_16_15_2060_2080_zulkernain.pdf [Last accessed on 10/7/2016]
- [2] <http://interruptions.net/literature/Adamczyk-TAMODIA05-p123-adamczyk.pdf> [Last accessed on 10/7/2016]
- [3] Douglas Mewhort, Natalie Cann, Gary Slater, and Thomas Naughton. High Performance Computing Systems and Applications. Print.
- [4] <https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx> [Last accessed on 10/14/2016]
- [5] Mohammad Kayes Kaykobad. MIMS: Mobile Interruption Management System. Print.
- [6] <http://www.usatoday.com/story/money/cars/2014/03/28/cellphone-use-1-in-4-car-crashes/7018505> [Last accessed on 11/26/2016]
- [7] <http://www.businessnewsdaily.com/8098-distractions-killing-productivity.html> [Last accessed on 11/27/2016]
- [8] <https://www.aaafoundation.org/sites/default/files/CellPhonesandDrivingReport.pdf> [Last accessed on 1/13/2017]
- [9] Amy Schick, Debbie Ascone, Julie Kang, and Maria Vegega. Distraction By Cell Phones and Texting. Print.
- [10] http://news.bbc.co.uk/2/hi/programmes/click_online/8639590.stm [Last accessed on 11/28/2016]
- [11] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.1731&rep=rep1&type=pdf> [Last accessed on 12/2/2016]
- [12] <http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm> [Last accessed on 12/2/2016]
- [13] <http://www.cs.waikato.ac.nz/ml/weka> [Last accessed on 12/9/2016]
- [14] <https://web.archive.org/web/20120216183916/http://iridia0.ulb.ac.be/~lazy> [Last accessed on 12/20/2016]
- [15] <http://twitter4j.org/en> [Last accessed on 12/20/2016]
- [16] <http://www.surveymonkey.com> [Last accessed on 3/31/2017]
- [17] https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html [Last accessed on 4/6/2017]
- [18] <http://www.marquette.edu/mscs/facstaff-ahamed.shtml> [Last accessed on 4/11/2017]

APPENDIX

Survey results (Continued):

Respondent #11:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.0
- Show status 0.0
- Get direct messages 0.0
- Send direct messages 0.0
- View profile 0.1
- Miscellaneous 0.0

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Bad interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

None

Q10: How many bad interruptions did you receive in the past week?

3

Respondent #12:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.7
- Show status 0.6
- Get direct messages 0.6
- Send direct messages 0.7
- View profile 0.5
- Miscellaneous 0.5

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

1

Respondent #13:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.5
- Show status 0.4
- Get direct messages 0.2
- Send direct messages 0.2
- View profile 0.3
- Miscellaneous 0.2

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Bad interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Bad interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

7

Respondent #14:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.5
- Show status 0.3
- Get direct messages 0.2
- Send direct messages 0.7
- View profile 0.2
- Miscellaneous 0.4

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

6

Respondent #15:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.7
- Show status 0.6
- Get direct messages 0.7
- Send direct messages 0.5
- View profile 0.8
- Miscellaneous 0.8

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

0

Respondent #16:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.6
- Show status 0.4
- Get direct messages 0.2
- Send direct messages 0.3
- View profile 0.5
- Miscellaneous 0.6

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

4-5

Respondent #17:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.2
- Show status 0.2
- Get direct messages 0.1
- Send direct messages 0.2

- View profile 0.2
- Miscellaneous 0.2

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Bad interruption

Q9: If you think another activity/activities should be included, what are they?
Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

1

Respondent #18:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:
Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.8
- Show status 0.1
- Get direct messages 0.8
- Send direct messages 0.8
- View profile 0.5
- Miscellaneous 0.9

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Bad interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Bad interruption

Q9: If you think another activity/activities should be included, what are they?
Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

Respondent #19:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.9
- Show status 0.8
- Get direct messages 0.3
- Send direct messages 0.3
- View profile 0.9
- Miscellaneous 0.2

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Bad interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

Not sure how one could answer this honestly without actively keeping track....

But anything over one would be significant to me and it's definitely over that.

Respondent #20:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.1
- Show status 0.0
- Get direct messages 0.1
- Send direct messages 0.1
- View profile 0.0
- Miscellaneous 0.6

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Bad interruption

Q9: If you think another activity/activities should be included, what are they?
Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

3-5

Respondent #21:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.0
- Show status 0.0
- Get direct messages 0.01
- Send direct messages 0.01
- View profile 0.0
- Miscellaneous 0.0

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Call while in a game, Google services popup while trying to type or in game (to re-authenticate into google), On desktop - cursor/mouse stealing notifications/popups.

Q10: How many bad interruptions did you receive in the past week?

20

Respondent #22:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 1.0
- Show status 0.2
- Get direct messages 0.2
- Send direct messages 1.0
- View profile 1.0
- Miscellaneous 0.2

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Bad interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Bad interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

Too many

Respondent #23:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- Yes

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.8
- Show status 0.6
- Get direct messages 0.7
- Send direct messages 0.5
- View profile 0.3
- Miscellaneous 0.1

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Bad interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Bad interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

2

Respondent #24:

Q1: Do you think an application for preventing unwanted interruptions will be useful?

- No

Q2: Fill the textboxes with the percentage of bad interruptions, based on Twitter activities. For example, 0.7 means 70% of interruptions are bad.* Activity details:

Update status - Adding a Tweet.

Show status - View your Tweets.

Get direct messages - View messages received.

Send direct messages - Send a message.

View profile - View someone else's profile.

Miscellaneous - Get usernames of those you are following, get followers IDs, show your information, such as name, date joined, etc.

- Update status 0.0
- Show status 0.0
- Get direct messages 0.0
- Send direct messages 0.0
- View profile 0.0
- Miscellaneous 0.0

Q3: Instruction: For questions #3 - #8, pick a random person who has called you in the past 30 days. If they called when you're doing Update Status, how would you rate it?

- Good interruption

Q4: If they called when you're doing Show Status, how would you rate it?

- Good interruption

Q5: If they called when you're doing Get Direct Messages, how would you rate it?

- Good interruption

Q6: If they called when you're doing Send Direct Messages, how would you rate it?

- Good interruption

Q7: If they called when you're doing View Profile, how would you rate it?

- Good interruption

Q8: If they called when you're doing Miscellaneous, how would you rate it?

- Good interruption

Q9: If you think another activity/activities should be included, what are they?

Respondent skipped this question

Q10: How many bad interruptions did you receive in the past week?

0

Code: <https://drive.google.com/file/d/0B7F1-Q9y9Ml4djJXXzdIQjdjR00/view?usp=sharing>