PRIVACY CONCERNS ON MOBILE APPLICATIONS VIS-À-VIS THE NUMBER OF PERMISSIONS REQUESTED BY ANDROID APPS

ALA RABEA

BOĞAZİÇİ UNIVERSITY
2019

PRIVACY CONCERNS ON MOBILE APPLICATIONS VIS-À-VIS THE NUMBER OF PERMISSIONS REQUESTED BY ANDROID APPS

Thesis submitted to the
Institute for Graduate Studies in Social Sciences
in partial fulfillment of the requirements for the degree of

Master of Arts
in
Management Information Systems

by
Ala Rabea

Boğaziçi University
2019

DECLARATION OF ORIGINALITY

I, Ala Rabea, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature .....

Date 09-08-2019

ABSTRACT

Privacy Concerns on Mobile Applications

Vis-À-Vis the Number of Permissions

Requested by Android Apps

Since the introduction of smartphones and applications, privacy concerns have been rapidly rising. Today, over one billion people use smartphones with millions of apps. These apps may request different permissions from users. Whether it's Android, IOS, Windows, or any other mobile operating system, apps may mostly request permissions more than necessary. In this study, Android apps from Google Play Store are examined. Python scraping code was used to collect details for over 5,000 apps including different parameters such as permissions requested and number of installs. The data was then analyzed using SPSS, AMOS, and Power BI. The analyses made varied from simple descriptive, to one-way ANOVA, multiple regression, and correlation. Also, graphs were constructed. Data analysis was fruitful and different hypothesis were significantly. The results showed that number of permissions requested is correlated with several other variables such as number of reviews, number of installs, and review score. Also, ANOVA tests showed that different developers and categories can possess statistically different number of permissions requested. The study has several limitations such as the IP address of the computer used in Turkey. Turkish apps were suggested. Also, the number of apps collected was 5,264 which relative to total number of apps on the store might be considered as small and non-representative.

ÖZET

Mobil Uygulamalar Kapsaminda Android Uygulamalari Tarafindan Istenen İzin Sayilari Hakkinda Gizlilik Endişeleri

Akıllı telefonların ve mobil uygulamaların ortaya çıkışından beri özel hayatın gizliliği ile ilgili kaygılar hızla artmaktadır. Günümüzde, bir milyardan fazla insan akıllı telefon kullanmaktadır. Günlük bazda milyonlarca farklı uygulama yüklenmekte ve kullanılmaktadır. Teknik mimarilerinden dolayı, bu uygulamalar işletim sistemindeki farklı kaynakları kullanmak zorundadır. Bu işletim sistemi; Android, IOS, Windows, ya da herhangi bir mobil işletim sisteminden hangisi olursa olsun; uygulamalar telefondaki verilere erişmek için ekstra izin almaya ihtiyaç duyabilir. Hatta çoğunlukla da pazarlama amaçlı kişisel veri toplamak için gereğinden fazla erişim hakkı isteyebilirler. Bu çalışmada Google Play Store'daki Android uygulamaları incelenmiştir. Uygulamada istenen izinler, yüklenme sayısı, incelemeler ve farklı birçok detay gibi parametreleri içeren 5.000'den fazla uygulama hakkındaki detayları toplamak için Python veri kazıma kodu kullanılmıştır. Daha sonrasında veriler SPSS ve AMOS'ta analiz edilmiştir. Analizler; temel göstergelerden tek yönlü varyans analizine, çoklu regresyon ve korelasyona kadar farklılaşmaktadır. Ayrıca, verilerin dağılımını göstermek için analizlere ek olarak grafiklerden de yararlanılmıştır. Veri analizleri, hipotezleri yüksek anlamlılıkla ispatlamış ve çok yararlı olmuştur. Sonuçlar, bir uygulamanın gerektirdiği izin sayısı ile inceleme sayısı, uygulama yüklenme sayısı ve inceleme skoru gibi farklı değişkenlerle ilişkili olduğunu göstermektedir.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

According to Statista (2019), number of applications available on Google Play Store has plunged from 16,000 apps in December 2009 to over 3.5 million apps in December 2017. That's more than 218 folds increase in less than a decade. Of course, this dramatic increase comes in parallel with the rapidly growing adaptation of smartphones. In 2015, Ericsson, a Swedish multinational telecommunication company, stated that the last decade has also witnessed a stable growth for smartphone users, who by 2020 would be 70% of the world's population. However, Felt et al. (2012) found that no more than 17% of the users actually give attention to what permissions are requested by applications when installed, and up to 42% are even unaware of the concept itself and base their decisions on the reviews and installs count. With such rapid development, concerns are raised regarding users' privacy and the extent of access being granted to apps with permissions.

There are more than 200 different permissions an Android application can request from user. In its documentation, Android classifies the permissions into 4 Protection levels (Android 2019):

- Normal (ex: BLUETOOTH – ACCESS NETWORK STATE)
- Signature (ex: MANAGE DOCUMENTS – READ VOICEMAIL)
- Dangerous (ex: READ PHONE NUMBERS – SEND SMS)
- Special (ex: SYSTEM ALERT WINDOW – WRITE SETTINGS)

Scholars and research centers tried to examine different aspects of the "Apps Permissions" issue such as awareness and behavior (Felt et al., 2012), types of permission and risk implications (Chia, Yamamoto, & Asokan, 2012; Atkinson, & Olmstead, 2015), or suggestions of technical improvements (Enck, Ongtang & Mcdaniel, 2009; Barrera, Kayacik, Oorschot & Somayaji, 2010; Felt, Chin, Hanna, Song, & Wagner, 2011).

The main objective of this study is to examine the relation between some variables of an app on the Google Play Store and the number of permissions requested when installing it. Since some previous studies pointed at the behavior of referring to reviews and installs count instead of permissions when installing, the focus was set on the relatively related variables such as:

- Reviews (number of reviews made on an app)
- Review Score (average score of reviews made; 1 to 5 scale with 0.1 increments)
- Installs (number of installs made)

The study is mainly composed of six classic chapters. The first one is this specific introduction for the study. Then, Chapter 2 includes a literature review of app stores, android app permissions, and Google play store. Chapter 3 tackles the theoretical model with its hypothesis. Next, in Chapter 4, the methodology is explained in details for all phases: coding the Python script, preparing the categories links, scrapping / collecting the data, cleaning and proofing the data, and statistical analysis using different software. Later, Chapter 5 explains detailed analysis results of the findings of all tests and measurements such as correlation, regression, and ANOVA. Finally, in Chapter 6, the

results and findings are interpreted and research limitations are briefly specified and explained.

CHAPTER 2

LITERATURE REVIEW

2.1 App stores

The App Store is the famous abbreviation that replaced “Application Store”. It is basically an online doorway through which users can download software programs, as mentioned by Rouse (2013) on her blog post. Mobile Operating Systems (which are software that allow devices like smartphones, PCs, tablets and others to run programs and applications) like Apple iOS, Microsoft’s Windows 8, Google Android, Nokia’s Symbian and others, manage their own app stores, and thus have control over the software available, as mentioned by Rouse (2013). The concept of an App store became famous with time, as the numbers of smartphone and tablet users have increased significantly.

2.1.1 Apple store

The Apple Store was released in July of 2008, opening the way to a new platform of applications. At first, it started with just 500 apps, and has continued to increase significantly. According to Statista Research Department (2016), in 2015, the number of active applications was 1,750,000 and has reached 4,670,000 in 2019. It is also estimated to reach 5,060,000 in 2020. According to Silver (2018), in his article published on apple insider, the Apple Store has been an integral addition to the e-commerce innovations during the past decade. Silver (2018) also mentions that the Apple Store has dramatically increased the growth in the business of developing apps.

"In its first decade, the App Store has surpassed all of our wildest expectations — from the innovative apps that developers have dreamed up, to the way customers have made apps part of their daily lives — and this is just the beginning," Apple's Phil Schiller said at the release. "We could not be more proud of what developers have created and what the next 10 years have in store."

2.1.2 Google play store

The Google Play Store was originally named the Android Market, as mentioned by Szul (2019) on his blog post. Google Play Store is Google's official store for Android apps on all Android devices. According to Clement (2019), the number of available applications in the Google Play Store was 1 million in July of 2013 and has reached a peak of 3,600,000 applications in March of 2018.

2.2 Android app permissions

2.2.1 Permissions and privacy policies in literature

Application permissions, as one can predict by the name, control what the application is allowed to do and even access on the user's device. It ranges from data that is stored on the device, for example contacts or media, to the user's camera or even the microphone. When the user approves, he/she gives permission for the application to use or have access to the feature, and when the user denies, he/she denies or prevent the application from doing so, as applications cannot by themselves have the permissions, the user has to give the permission him/herself. Olmstead & Atkinson (2015), mention that "Once that permission is granted, the apps can amass insights from the data collected by the apps on things such as the physical activities and movements of users, their browsing

and media-use habits, their social media use and their personal networks, the photos and videos they shoot and share, and their core communications". According to Olmstead & Atkinson (2015), in their study done by Pew Research Center, it was concluded that 60% of app downloaders actually chose not to install an application after they knew the extent of personal information it required in order for them to be able to use it and 43% of the users had even uninstalled the application after they have already downloaded it, also for the same reason. Their study has also found out that 90% of app downloaders said that it is "very" or "somewhat" important for them how the application will be using their personal data. Those results are critical as they suggest that the users or app downloaders actually don't know the extent to which the applications are having access to their personal information available on their phones as they grant access to those application, actually giving them the permission to do so.

According to Obar & Hirsch (2018), their experimental study that did an empirical investigation on privacy policies and terms of service policy reading behaviors concluded that 74% of app downloaders actually skipped the privacy policy as they joined a fictitious social networking service, and just chose the "quick join" click that granted permissions to the application. Obar & Hirsch (2018) also concluded in their experimental survey that since the average adult reading speed is 250 to 280 words per minute, the privacy policies in their study should have taken 29 to 32 minutes approximately and the terms of service policy should have taken 15 to 17 minutes approximately to be read; however, reading time was almost 73 seconds for the privacy policies and 51 seconds for the terms of service policies, and actually, most participants granted the application permissions and agreed to the policies, 97% agreed to the

privacy policies and 93% to the terms of service policies. Obar & Hirsch (2018) have also qualitative findings that suggest that users actually view policies as a barrier that they ignore because they want to reach the ends of their digital production, without being blocked on the way.

2.2.2 Permissions documentation

On their official website, Android specifies types of permissions in their documentation. The permissions are categorized into 4 categories:

- Normal (ex: BLUETOOTH – ACCESS NETWORK STATE)
- Signature (ex: MANAGE DOCUMENTS – READ VOICEMAIL)
- Dangerous (ex: READ PHONE NUMBERS – SEND SMS)
- Special (ex: SYSTEM ALERT WINDOW – WRITE SETTINGS)

  According to the permission type, the user will be notified when first opening the app for the first time. A pop message will show asking to grant access to certain data according to permissions type like what is shown in figure 1.

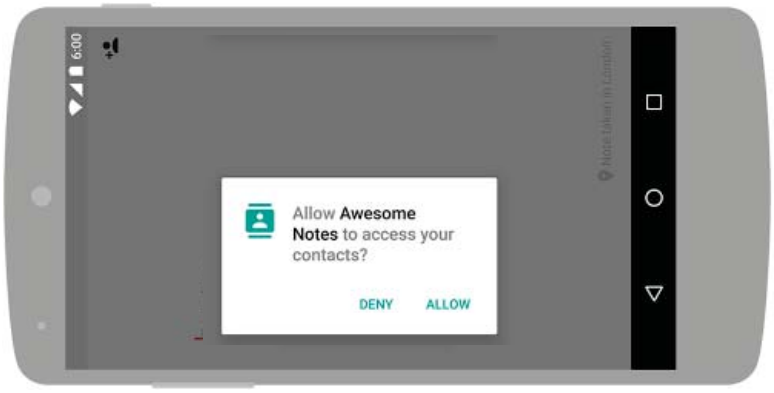

Figure 1 Android app requesting access to contacts

2.3. Google play store

2.3.1 Privacy policy

According to Iubenda, Google Play has made it a priority to disclose privacy issues to users, in accordance to law. Those disclosures are given to users in a form of a privacy notice that is easily made available to the user from within the application. The Developer Policy Center's User Data guidelines quotes: "You must be transparent in how you handle user data (e.g., information provided by a user, collected about a user, and collected about a user's use of the app or device), including by disclosing the collection, use, and sharing of the data, and you must limit use of the data to the description in the disclosure. If your app handles personal or sensitive user data, there are additional requirements described later. This policy establishes Google Play's minimum privacy requirements; you or your app may need to comply with additional restrictions or procedures if required by an applicable law". For the app developer, Google Play mandates that a link to the privacy policy be visible to the users. The app developer also needs to disclose his use to permission groups like the calendar, camera, contacts, location, SMS, and others. Google Play also asks app developers to limit their collection and use of data for the exclusive purpose of providing and improving the features of their applications and deal with all the personal user data securely. Google Play also promises the users regarding their privacy, security, and deception as they mention on their official page: "We're committed to protecting user privacy and providing a safe and secure environment for our users. Apps that are deceptive, malicious, or intended to abuse or misuse any network, device, or personal data are strictly prohibited."

2.3.2 Apps details

As the user reaches an application on Google Play store, he/she will have access to a range of information including the title of the application which is the application's name on Google Play, a short description which is the first text that the users will notice as they look for the app details, full description, graphic assets like images, videos, screenshots, and icons that can describe the various features of the application. A user can also find under what category does the application fall, contact details of the app developer, the star-rating of the application, reviews, and additional information like the date of the last update, version, size, interactive elements, report, size, in-app products, number of installs, content rating and permissions.

CHAPTER 3

THEORETICAL MODEL AND HYPOTHESES

In this chapter, the theoretical model is explained and along with the different hypotheses formulated base on it. The model was originally based on the classic SOR (stimulus-organism-response) model as a concept where there is set of behavior derived from set of stimuli. However, after running multiple fit-tests and trials, the model was modified from a multiple-mediator SOR model into the model shown next in figure 2 with no mediating or moderating variables. Nevertheless, to ensure maximum accuracy and model fit possible, the variables were kept as control variables which will be explained in details in Chapter 4, methodology chapter, analysis section. Figure 2 shows the model with no error or control variables for clarity and simplicity. Whereas full model with error and control variables will be shown in the Chapter 4 also.

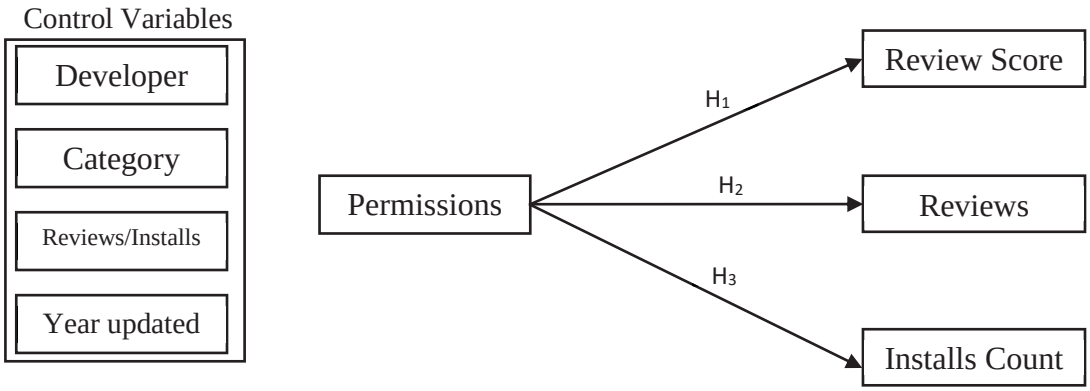

Figure 2 Theoretical framework

3.1 Model Variables

In this section the variables of the model are explained in details. It explains the meaning of each variable with respect to an android app, where it is found, and type of data.

3.1.1 Permissions

In the model, ‘Permissions’ variable is the cardinality (count / number) of permissions requested by the app from the user. This detail is found at: the app page > bottom section ‘Additional Information’ > ‘Permissions’ heading > ‘View Details’. The pop-up window shows the permissions as string and classified under different types such as: Storage, Wi-Fi connection information, and others. However, in the methodology section, there will be an explanation on how the cardinality was obtained.

3.1.2 Review score

In the model, ‘Review Score’ variable is the average score of reviews made for a specific app using a ‘1-to-5’ scale with 0.1 increments. The score is calculated based on the Play store users. This detail is found in two different places. The first one is at the top of the app page in form of five stars. The stars are shaded in dark gray to show the score. Due to graphical representation, this specific parameter wasn’t used for data collecting. The second place is right after the app description under the gallery. A relatively big counter shows the review score in numbers of 1-digit decimal with the ‘5-stars’ graphical representation beneath it again.

3.1.3 Reviews

In the model, ‘Reviews’ variable is the cardinality (count / number) of reviews made on a specific app. This detail is also found twice in the app page; both are shown as a counter in number form next to the two review score indicators mentioned earlier. The number is exact with no rounding like in the case of other variables such as number of installs which will be shown in the coming sub-section.

3.1.4 Installs count

In the model, ‘Installs Count’ variable is the cardinality (count / number) of installs made from a specific app. This detail is found at: the app page > bottom section ‘Additional Information’ > under ‘Installs’ heading. However, it is important to note that this number is ordinal/categorical. The indicator shows the record broken by the app instead of actual number of installs (such as: 5000+ instead of 5,435 or 1.7M instead of 1,700,956). Further explanation will be made in the methodology and limitations sections.

3.2 Hypotheses

- Hypothesis 1: ‘Permissions’ (number of permissions) has a significant impact on ‘Review Score’.
- Hypothesis 2: ‘Permissions’ (number of permissions) has a significant impact on ‘Reviews’ (number of reviews).

- Hypothesis 3: ‘Permissions’ (number of permissions) has a significant impact on ‘Installs Count’.
- Hypothesis 4: there is a significant relationship between ‘Permissions’ (number of permissions) and ‘Review Score’.
- Hypothesis 5: there is a significant relationship between ‘Permissions’ (number of permissions) and ‘Reviews’ (number of reviews).
- Hypothesis 6: there is a significant relationship between ‘Permissions’ (number of permissions) and ‘Installs Count’.
- Hypothesis 7: there is a significant relationship between ‘Permissions’ (number of permissions) and ‘Year Updated’.
- Hypothesis 8: there is a significant relationship between ‘Permissions’ (number of permissions) and ‘Reviews/Installs’ (ratio of reviews number / number of installs).
- Hypothesis 9: there is a significant relationship between ‘Reviews/Installs’ (ratio of reviews number / number of installs) and ‘Year Updated’.
- Hypothesis 10: there is a statistically significant difference among ‘Developer’ with respect to ‘Permissions’
- Hypothesis 11: there is a statistically significant difference among ‘Category’ with respect to ‘Permissions’
- Hypothesis 12: there is a statistically significant difference among ‘Year Updated’ with respect to ‘Permissions’

CHAPTER 4

METHODOLOGY

In this chapter, a detailed technical methodology will be explained regarding coding the scraping Python script, choosing the sample (categories / links) of the apps, data scraping process, data proofing and cleaning, and statistical analysis.

4.1 Coding – Python

This section is technical and requires fundamental knowledge regarding coding and programming. The main language used was Python using version 2.7.10 operated on MAC OS MOJAVE 10.14.5. In the Python code, Selenium library was imported and used to be able to control internet browser. The code requires and works along with:

- JS – JavaScript
- Node
- ChromeDriver (since Google Chrome was chosen).

The code would first get 3 parameters from the user on the terminal / command line: location / directory of the chrome driver, URL of the apps category needed to be scrapped, and, optionally, an integer indicating the number of scrolls (will be explained); if the number of scrolls is not indicated, the code scrolls down enough to load all possible apps. Then, when the code is executed, the Google Chrome web browser will lunch and go to the indicated URL. The browser was first cleared from cookies and any

user's data to avoid suggestions and recommendations. Only IP address was an affecting factor which relatively displayed more Turkish local apps (the location of the study was Istanbul). The first set of apps will be loaded and the code will start scrolling down to stimulate the browser to load more apps at the bottom of the page (based on the third parameter). When the apps are loaded, the code scraps/collects all the apps' Id's and initiate a CVS file using CSV library imported. The CVS file will start with columns based parameters needed to be collected. First column would be app id. After it comes any parameters needed. To specify the parameters, 'Container Id' was used. Container Id were previously recorded for the needed parameters. The collected parameters in this study were:

- ID: App unique Id (ex: com.whatsapp)
- Name: App Name (ex: WhatsApp)
- Category: the category which the app belongs to (ex: Communication)
- Reviews: the number of reviews made on the app (ex: 135,846 reviews)
- Installs: the number of installs made from the app (ex: 35M)
- Last Updated: the date of last updated released (ex: 25/01/2019)
- Developer Name: the name of developer or developing company (ex: Whatsapp Inc.)
- Permissions: all the permissions requested by the app from the user delimited with comma ',' (ex: access to contacts, access to Wi-Fi status)

The last parameter, 'Permissions', required an extra step to be collected as it was the only parameter not found in the page and required opening a link "details". In the code, Try-Catch / Try – Except blocks where needed. If the needed parameter or

specified container was not found, error was printed on the screen. In the case of missing detail, the code was designed to simply skip the record and move to the next one. However, in the case of wrong container, usually due to opening from different IP address (different IP addresses may reach different page designs), the container name was checked, updated, and the code was reran. The code also was designed to display the number of apps found and number of apps successfully collected.

When the process was over, the CSV file would be found in the same directory of the script. Microsoft Excel was used for primary and basic data editing and grouping.

*Check Appendix C for details about the code.

4.2 Preparation of links / categories – Choosing sample

In Google Play Store, there are several classifications and categorizations for the apps. In the homepage, even if the user is visiting for the first time, the store recommends some apps. Also, top rated games, popular games, and other top charts are displayed.

Since number of installs, number of reviews, and review score were involved in the study, all top charts and recommendations were avoided. Only categorical charts were used which rely only on the nature of the app (ex: Games > Action, Business). Scrapping was determined by displayed order (‘Daydream’ chart till ‘Word’ chart under ‘Games’). ‘Family’ chart was not scraped as the size needed for the sample was achieved (5000+ record). However, in some charts, there were sub-charts with top charts included. All top charts were ignored. A total of 39 categories / subcategories for 4803 apps (5,264 app before data cleaning / proofing) were scrapped as in Table 1:

Table 1. Categories Scraped and Cardinality

Category		Number of Apps collected	Category		Number of Apps collected
1	Art & Design	113	21	News & Magazines	102
2	Auto & Vehicles	112	22	Parenting	75
3	Beauty	74	23	Personalization	120
4	Board	124	24	Photography	256
5	Books & Reference	82	25	Productivity	115
6	Business	112	26	Puzzle	208
7	Casual	62	27	Racing	94
8	Comics	80	28	Role Playing	121
9	Communication	102	29	Shopping	102
10	Education	228	30	Simulation	108
11	Educational	169	31	Social	122
12	Entertainment	191	32	Sports	204
13	Finance	156	33	Strategy	144
14	Food & Drink	94	34	Tools	159
15	Health & Fitness	78	35	Travel & Local	136
16	House & Home	98	36	Trivia	67
17	Lifestyle	94	37	Video Players & Editors	125
18	Maps & Navigation	124	38	Weather	130
19	Music	55	39	Word	159
20	Music & Audio	110	Total 4803		

4.3 Scrapping / Collection process

Since the URLs had to be written manually on the terminal/Cmd every time, a list of all valid URLs was manually prepared. Then ten different tabs were opened on the terminal/Cmd. Directory was changed into ten identical folders pre-prepared and the command was written. Later, only the URL was modified in each tab and all were executed I parallel. On average, a URL took 22 seconds to fully load and scroll. An app took 5 seconds to be scrapped. A link contained 29 to 200 app. So each run of ten parallel scrapings took approximately 13 minutes on average.

The scraping was done between 3 June, 2019 and 20 July, 2019. A total of 52 CSV files were obtained by six iterations (5 iterations x 10 parallel tabs + 2 tabbed iteration). The CSV files were then merged using a single command on terminal:

```
cat *.csv  > merged.csv
```

Example: "cat" space "*.csv" space " > All_Apps.csv"

Final CSV file was ready to be cleaned and processed.

4.4 Data proofing and cleaning

In this phase, different stages of data proofing, cleaning, and preparation were done. The primary stage included basic proofing. First, using MS Excel, all duplicates were deleted. Duplicates were found using the first parameter, the unique ID of the app (i.e. no two apps can have same app ID). Then, the cells were defined according to the data held. For all numerical data, it was changed to ‘Number’; for all string data, it was changed to ‘Text’. For ‘Number’ data, all formatting was removed such as commas and

decimals. However, for some records, reviews or installs were abbreviated with ‘M’ for millions and ‘K’ for thousands. A simple ‘Find and Replace’ scanning was made to replace them with proper number of zeros.

The second stage of data preparation was to make it more useful and easy to process on the data-analysis software, SPSS and AMOS. For the date updated, a full-format date is complicated to process, especially when not being a major variable. For this purpose, another column was inserted where an Excel function extracted the year only from the date.

Also, since the permissions were collected in string form (all permissions in one cell) with only comma between the permissions, there was a need to count the permissions as the study is quantitative not qualitative. A simple function in Excel named ‘Text to Column’ was used. The function simply divides the string in the cell into the next empty cells in the row using the commas as delimitation. The biggest record was for an app with 133 permissions. Another column was added where an Excel function would count the blanks next to the cell to know number of permissions.

Advanced data cleaning was done in the analysis phase for statistical purposes. Some categories with very low number of apps were eliminated. Categories with more than 34 apps were kept. Also, for ANOVA analysis, apps with last updated year 2011 (2 apps), 2012 (3 apps), and 2013 (6 apps) were eliminated as it is impossible to detect statistical difference among such small groups. This approach will be explained in more details in statistical analysis section and analysis and results chapter.

4.5 Statistical analysis

After the data was almost ready to be analyzed, there were 4803 apps with full app details. The scale variables were standardized (z-bin) and annotated with ‘Std’. For the string values, non-ASCII characters were coded into new variables annotated with ‘New’. The following analyses were made:

4.5.1 Model-Fit Analysis

The approach of modeling was mainly based on SOR model. The number of permissions was considered as a stimuli for users where number of reviews, review score, and number of installs were considered as indicators of response behavior. A conservative Model-Fit analysis was made using SPSS and AMOS, originally an SPSS module, to ensure best fit possible for the model until getting the best match. Throughout the analysis, a new variable was suggested to the model for a better fit, Reviews/Installs ratio (a ratio dividing number of reviews by number of installs). This new variable was added to rationalize the relation between the number of reviews made on an app and number of installs and make a better statistical fit. Statistically, the ratio implies the percentage of the users who made a review among those who actually installed the app instead of just measuring number of reviews previously. This approach allows the analysis to compare rate of reviewing apps with relatively extremely low number of installs and reviews to apps with relatively extremely high number of installs and reviews. However, the variable was kept as control variable where different effects were measure.

4.5.2 Descriptive analysis

A general Descriptive analysis was made using SPSS for the collected parameters stating their ‘N’, Mean, Median, Mode, Std. Deviation, Variance, Minimum, and Maximum. Categorical variables (Category and Developer) were processed separately due to statistical and string-processing limitations.

4.5.3 Correlation analysis

Correlation analysis was made using SPSS to examine the existence and strength of relationship between variables mentioned in the hypotheses: Permissions and Review Score – Permissions and Reviews Count – Permissions and Installs Count. Moreover, ‘Year updated’ and the suggested ratio of ‘Reviews/Installs’ were added to the correlation and all correlations were examined.

4.5.4 Multiple regression analysis

Multiple regression analysis was done to examine the impact of Permissions on Review Score, Reviews Count, and Installs Count. The analysis was planned on SPSS, but due to the interference of other variables in the relationship, it was carried on by AMOS were direct and indirect effects were also calculated and bootstrapping (N = 2000) was done for better measuring.

4.5.5 One-way ANOVA analysis

The one-way ANOVA is made to measure weather there is a statistically significant difference among different categories with respect to a certain variable. In this study,

one-way ANOVA was conducted from three different aspects on SPSS, all with respect to ‘Permissions’:

- If there is a statistically significant difference among app developers with respect to number of permissions*.
- If there is a statistically significant difference among apps categories with respect to number of permissions**.
- If there is a statistically significant difference among different years of last update with respect to number of permissions***.

*Since SPSS doesn’t conduct ANOVA on more than 50 groups, only developers with 10 apps or more (23 developers for 455 apps) were selected.

**As mentioned in the previous section, only categories with 34 apps or more (39 categories for 4803 apps) were selected.

***As mentioned in the previous section, apps with last updated year 2011 (2 apps), 2012 (3 apps), and 2013 (6 apps) were eliminated as it is impossible to detect statistical difference among such small groups.

Also, for one-way ANOVA, there are assumptions made for the test to be valid:

- Normality of sample (normally distributed)
- Equality of homogeneity of variance, which is also called as homoscedasticity
- Independence of cases

However, as the results will show, both normality and homoscedasticity assumptions were violated. Hence, ‘Robust Tests of Equality of Means’ analysis was made using the conservative approaches of ‘Welch’ and ‘Brown-Forsthe’ where the

results were valid and ‘Games-Howell’ approach was then applied for the multiple comparison.

4.5.6 Errors and controls effect analysis (direct and indirect)

Since the modeling and multiple regression analyses were made using AMOS, several conservative measurements were made to ensure best statistical accuracy:

- Variance Error for Review score, Reviews count, and Installs count
- Direct and indirect effect and relationships of different errors and control variables

CHAPTER 5

ANALYSES AND RESULTS

In this chapter, descriptive, model fit, correlation, one-way ANOVA, normality testing, homogeneity test, direct effect, indirect effect, total effect, and multiple regression analyses are shown. With the findings, hypotheses suggested in chapter 3 are tested.

Descriptive findings include ‘N’, Mean, Median, Mode, Std. Deviation, Variance, Minimum, and Maximum of scale variables. Some figures are also shown for categorical variables.

In the model-fit section, the results of the model-fit of the suggested model values are shown and discussed.

The correlation analyses made it possible to test the previously suggested hypotheses H4 to H12 and understand the significance of the relationship between used variables.

Also, the nature of data distribution was explored to better understand the data. Graphical representations and graphs were prepared.

One-way ANOVA is done to test hypothesis H10 to H12 and understand whether there is a significant statistical difference in variance between different categories with respect to a certain variable, ‘Permissions’. As mentioned earlier, some of the one-way ANOVA assumptions were violated. Further analyses were made using conservative approaches.

Direct effect, indirect effect, and total effect are all analyses by AMOS. Instead of ignoring a control variable, an indirect effect of another variable, or an error in the variance of a certain variable, AMOS calculates the mentioned possible noise and separates it from the analysis.

After accounting for relatively unwanted effects, multiple regression is done to test the suggested hypotheses H1 to H3. These hypotheses test weather a certain variable ‘Permissions’ has a significant impact on another variable: ‘Review Score’, ’Reviews Count’, and ‘Install Count’.

5.1 Descriptive findings

Descriptive findings from the SPSS are shown in table 2. The variables are respectively: ‘Reviews’ for the number of reviews on an app, ‘ReviewScore’ for the review score on an app, ‘InstallCount’ for the number of installs made from an app, ‘RevInsRatio’ for the ratio of reviews to installs, ‘Yearupdated’ for the year of last update, and ‘perms’ for the number of permissions requested by an app. The total number of apps after cleaning was 4803 with no missing data.

In table 3, similar descriptive analysis was made for categorical variable ‘Category’ which describes the category the app belongs to according to Google Play Store categorization.

Table 2. Descriptive Analysis Results for Scale Variables

		Reviews	ReviewScore	InstallCount	RevInsRatio	Yearupdated	perms
N	Valid	4803	4803	4803	4803	4803	4803
	Missing	0	0	0	0	0	0
Mean		407942.72	4.288	24528306.97	.024007084966868	-	12.42
Median		25537.00	4.300	1000000.00	.013237000000000	-	10.00
Mode		121[a]	4.5	1000000	.006260000000000	2019	7
Std. Deviation		2897528.629	.3320	193390448.773	.033461556712170	-	8.094
Variance		8395672157834.840	.110	37399865676707400.000	.001	-	65.517
Minimum		100	2.2	5000	.000009920000000	2011	0
Maximum		92888725	5.0	5000000000	.554374000000000	2019	133

a. Multiple modes exist. The smallest value is shown

Table 3. Descriptive Analysis Result for 'Category'

Category Descriptive	
Number of categories	39
Number of apps	4803
Max	256
Min	55
Mean	123.2051
Std. Dev.	45.46353
Mode	102
Median	113

In table 4, similar descriptive analysis was made for categorical variable 'Developer' which describes the developer or developing company of the app.

Table 4. Descriptive Analysis Result for ‘Developer’

Developer Descriptives	
Number of developers	3295
Number of apps	4803
Max	80
Min	1
Mean	1.45827
Std. Dev.	2.264995
Mode	1
Median	1

In Appendix A, frequency table is shown for ‘Category’. For each category, frequency, percent, valid percent, and cumulative percent are shown.

5.2 Model fit

As mentioned earlier, Model-Fit analysis was done using additional software. Multiple models were tested and iterated. There was a clear need for customization and the use of conservative analysis. For this purpose, SPSS wasn’t sufficient at covering statistical limitations and concerns. Hence, AMOS, originally an SPSS module, was used to test model-fit and account for errors and other possible effects or cross-contributions. Figure 3 shows the full model including four controlled variables (Developer - Category – Year Updated – Reviews/Installs ratio), three error sets (e1 for Review Score - e2 for Reviews Count - e3 for Install Count), and the effects and correlations for them.

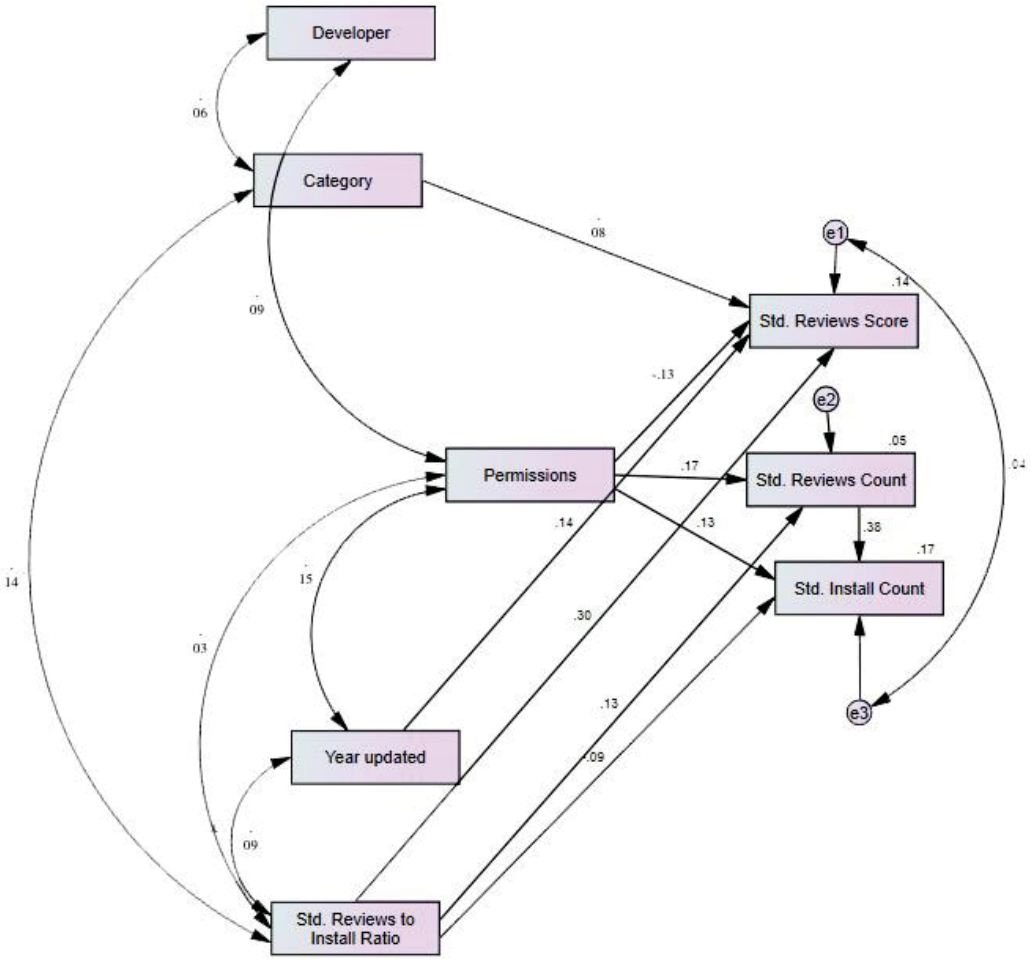

Figure 3 The full model from AMOS

For this specific model, the Model-Fit results are displayed. Table 5 and Table 6 show the model fit measures and cut-off criteria. According to both table, the model is significantly valid to be studied.

Table 5. AMOS Model Fit Measures for final Model

Measure	Estimate	Threshold	Interpretation
CMIN	47.565	--	--
DF	12	--	--
CMIN/DF	3.964	Between 1 and 3	Acceptable
CFI	0.984	> 0.95	Excellent
SRMR	0.02	< 0.08	Excellent
RMSEA	0.025	< 0.06	Excellent
PClose	1	> 0.05	Excellent
Congratulations, your model fit is acceptable.			

Table 6. AMOS Cutoff Criteria for Model Fit Measures

Measure	Terrible	Acceptable	Excellent
CMIN/DF	> 5	> 3	> 1
CFI	< 0.90	< 0.95	> 0.95
SRMR	> 0.10	> 0.08	< 0.08
RMSEA	> 0.08	> 0.06	< 0.06
PClose	< 0.01	< 0.05	> 0.05

*Note: Hu and Bentler (1999, "Cutoff Criteria for Fit Indexes in Covariance Structure Analysis: Conventional Criteria Versus New Alternatives") recommend combinations of measures. Personally, I prefer a combination of CFI > 0.95 and SRMR < 0.08. To further solidify evidence, add the RMSEA < 0.06.

5.3 Correlation

For correlation analysis, hypotheses H4 to H12 were tested to check the significance of relationship between different variables. Table 7 shows all the correlation results from SPSS output.

Table 7. Correlation values from SPSS

Correlations							
		Reviews	ReviewScore	InstallCount	RevInsRatio	Yearupdated	perms
Reviews	Pearson Correlation	1	.048**	.389**	.139**	.033*	.176**
	Sig. (2-tailed)		.001	.000	.000	.023	.000
	N	4803	4803	4803	4803	4803	4803
ReviewScore	Pearson Correlation	.048**	1	.010	.317**	.140**	-.095**
	Sig. (2-tailed)	.001		.506	.000	.000	.000
	N	4803	4803	4803	4803	4803	4803
InstallCount	Pearson Correlation	.389**	.010	1	-.028	.023	.191**
	Sig. (2-tailed)	.000	.506		.053	.118	.000
	N	4803	4803	4803	4803	4803	4803
RevInsRatio	Pearson Correlation	.139**	.317**	-.028	1	.082**	.046**
	Sig. (2-tailed)	.000	.000	.053		.000	.001
	N	4803	4803	4803	4803	4803	4803
Yearupdated	Pearson Correlation	.033*	.140**	.023	.082**	1	.154**
	Sig. (2-tailed)	.023	.000	.118	.000		.000
	N	4803	4803	4803	4803	4803	4803
perms	Pearson Correlation	.176**	-.095**	.191**	.046**	.154**	1
	Sig. (2-tailed)	.000	.000	.000	.001	.000	
	N	4803	4803	4803	4803	4803	4803
**. Correlation is significant at the 0.01 level (2-tailed).							
*. Correlation is significant at the 0.05 level (2-tailed).							

As shown in the table, the following relationships are significant:

- Reviews and Review Score
- Reviews and Installs Count
- Reviews and Reviews to Installs ratio
- Reviews and Reviews to Year Updated
- Reviews and Reviews to Permissions
- Review Score and Reviews to Installs ratio
- Review Score and Reviews to Year Updated
- Review Score and Reviews to Permissions
- Installs Count and Year Updated
- Installs Count and Permissions
- Reviews to Installs ratio and Year Updated
- Reviews to Installs ratio and Permissions
- Year Updated and Permissions

Relationships that are NOT significant:

- Review Score and Installs Count
- Installs Count and Reviews to Installs ratio

While examining the correlations results, it is noticed that ‘Permissions’ are significantly correlated with all other variables with a value up to 0.191 (Permissions and Installs Count).

5.4 Explore – Distributions

This part of the analysis is a prerequisite for the upcoming ANNOVA test. In this section, the results of 'Tests of Normality' are shown and analyzed. Table 8, shows the significance results of Kolmogorov-Smirnova and Shapiro-Wilk normality tests.

Table 8. Tests of Normality

Tests of Normality						
	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
perms	.149	4803	.000	.812	4803	.000
a. Lilliefors Significance Correction						

As shown in the table, according to both tests' results, the data is not normally distributed. Thus, distribution graphs were conducted for different variables to better understand nature of data. Figure 4, shows the data distribution according to number of permissions.

As the graph in Figure 4 shows, the data is not normally distributed. However, a positively skewed pattern is detected indicating that more apps require higher number of permissions.

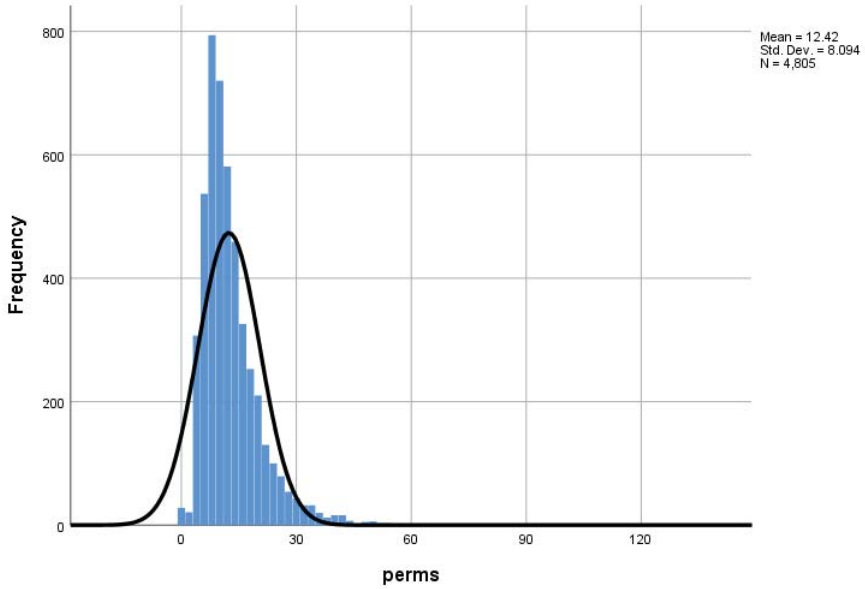

Figure 4 Data distribution according to permissions

Figure 5, shows the data distribution according to Review score.

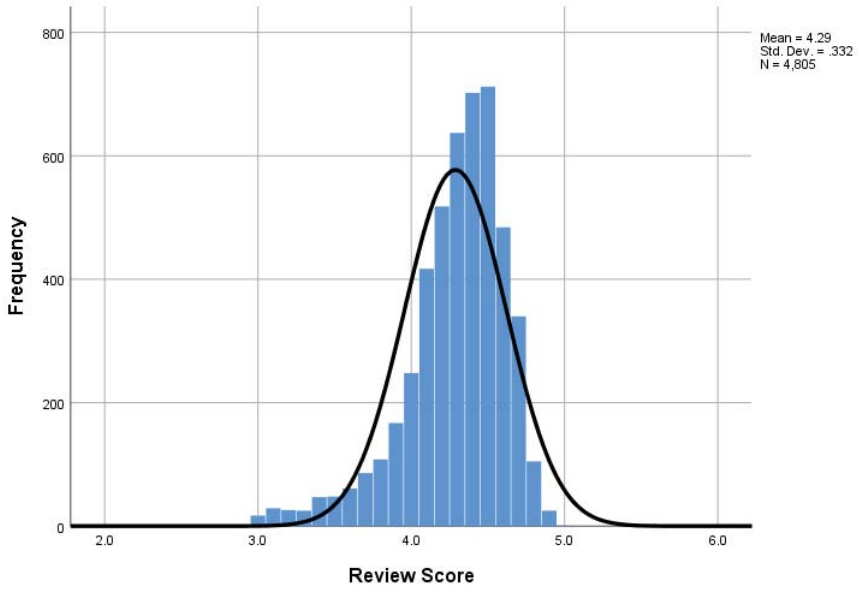

Figure 5 Data distribution according to review score

As the graph in Figure 5 shows, the data is not exactly normally distributed; the pattern seems to have similarity with normal distribution patterns. However, a negatively skewed pattern is detected indicating that users tend to give lower scores.

Finally, figure 6, shows the data distribution according to Review / Installs ratio which was suggested.

As the graph in Figure 6 shows, the data is not normally distributed. However, a Pareto pattern is observed indicating a heavily tailed data.

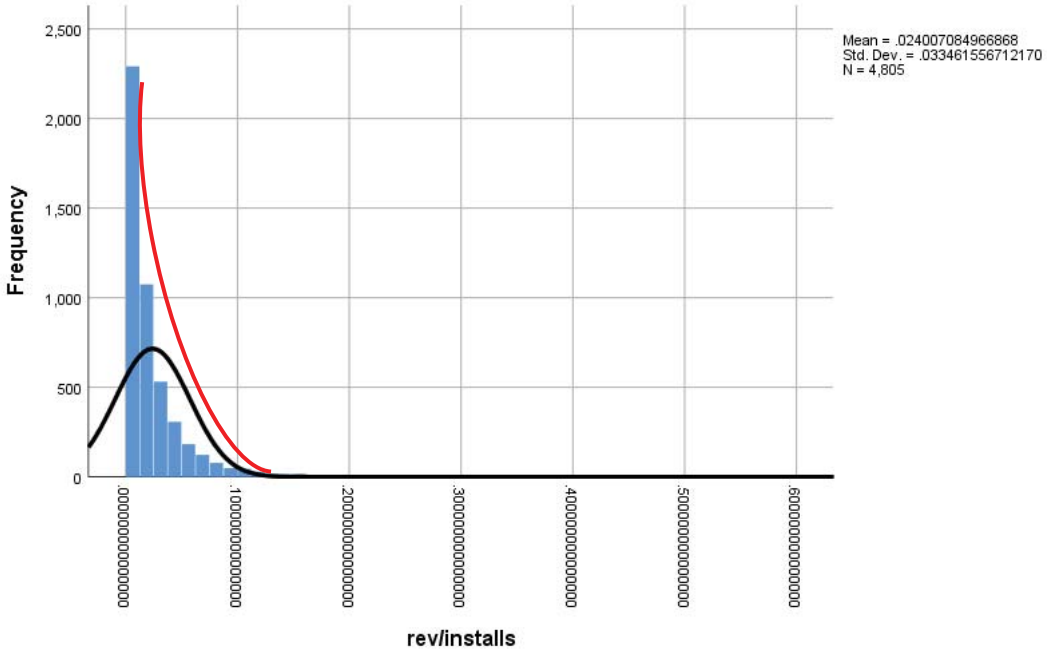

Figure 6 Data distribution according to review / installs ratio

Finally, in scope of exploring visualized data, Figure 7 and Figure 8, respectively, show the top permissions requested and number of permissions requested by different categories. The data is color-coded to show privacy concerns and risks.

Figure 7 Top permissions requested with color code

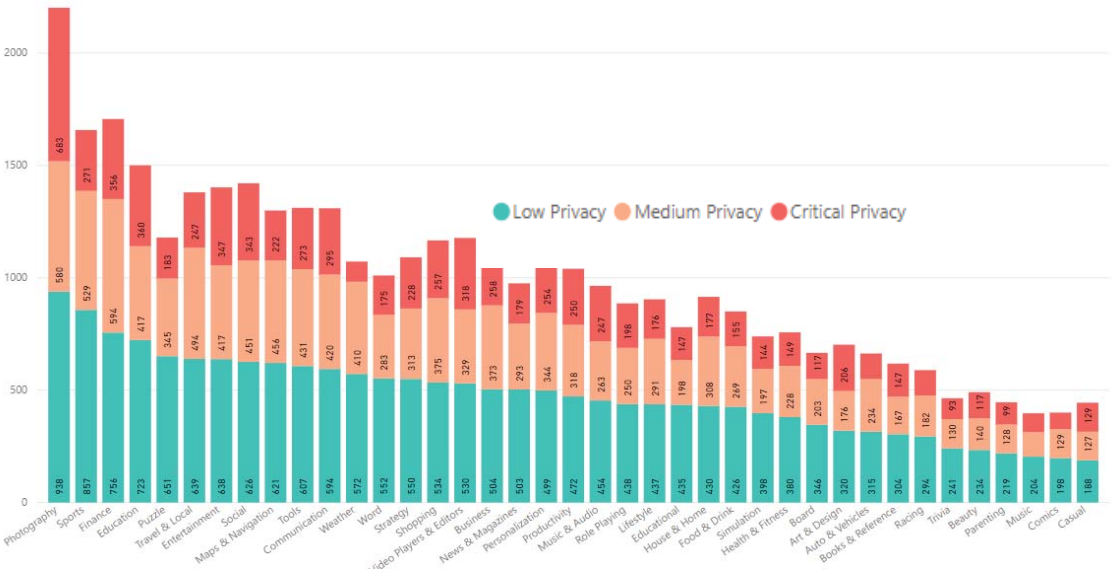

Figure 8 Permissions requested by categories

5.5 One-way ANOVA

The one – way ANNOVA was conducted on three aspects: difference among Categories with respect to permissions, difference among Developers with respect to permissions, and difference among Years of update with respect to permissions. Before conducting the ANOVA with respect to permissions, Test of Homogeneity of Variance (Levene's Test) was conducted. Table 9 shows the results among Categories.

Table 9. Levene's Test Results among Category

Test of Homogeneity of Variances					
		Levene Statistic	df1	df2	Sig.
Zperms	Based on Mean	26.446	38	4766	.000
	Based on Median	23.111	38	4766	.000
	Based on Median and with adjusted df	23.111	38	1855.887	.000
	Based on trimmed mean	24.515	38	4766	.000

As shown in the table, the data doesn't meet the assumption of homogeneity. To proceed, 'Robust Tests of Equality of Means' analysis was made using the conservative approaches of 'Welch' and 'Brown-Forsthe' where the results were valid and 'Games-Howell' approach was then applied for the comparison. In Table 10, Robust Tests of Equality of Means results are shown.

Table 10. Robust Tests of Equality of Means among Category

Robust Tests of Equality of Means				
Zperms				
	Statistic[a]	df1	df2	Sig.
Welch	52.879	38	1463.683	.000
Brown-Forsythe	45.493	38	1964.077	.000
a. Asymptotically F distributed.				

As the table shows, both tests were significant and ensuring the possibility of conducting one – way ANNOVA test.

In Table 11, the result of one-way ANNOVA among Category with respect to Permissions.

Table 11. ANOVA results among Category

ANOVA					
Zperms					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1293.468	38	34.039	46.212	.000
Within Groups	3510.532	4766	.737		
Total	4804.000	4804			

As the table shows, there is a significant difference among different categories with respect to number of Permissions with F = 64.21 and df = 38.

Again test of Homogeneity of Variance (Levene's Test) was conducted. Table 12 shows the results among Developers.

Table 12. Levene's Test Results among Developers

Test of Homogeneity of Variances			
Zscore(perms)			
Levene Statistic	df1	df2	Sig.
17.986	22	432	.000

As shown in the table, the data doesn't meet the assumption of homogeneity. To proceed, 'Robust Tests of Equality of Means' analysis was made using the conservative approaches of 'Welch' and 'Brown-Forsthe' where the results were valid and 'Games-Howell' approach was then applied for the comparison. In Table 13, Robust Tests of Equality of Means results are shown.

Table 13. Robust Tests of Equality of Means among Developers

Robust Tests of Equality of Means				
Zscore(perms)				
	Statistic[a]	df1	df2	Sig.
Welch	75170181880428 24000000000000 000.000	22	113.791	.000
Brown-Forsythe	20.516	22	133.000	.000
a. Asymptotically F distributed.				

As the table shows, both tests were significant and ensuring the possibility of conducting one – way ANNOVA test.

In Table 14, the result of one-way ANNOVA among Developer with respect to Permissions.

Table 14. ANOVA results among Developer

ANOVA					
Zscore(perms)					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	286.071	22	13.003	15.348	.000
Within Groups	365.993	432	.847		
Total	652.064	454			

As the table shows, there is a significant difference among different developers with respect to number of Permissions with F = 15.35 and df = 22.

Again test of Homogeneity of Variance (Levene's Test) was conducted. Table 15 shows the results among Year of update.

Table 15. Levene's Test Results among Years of update

Test of Homogeneity of Variances			
Zscore(perms)			
Levene Statistic	df1	df2	Sig.
9.995	5	4788	.000

As shown in the table, the data doesn't meet the assumption of homogeneity. To proceed, 'Robust Tests of Equality of Means' analysis was made using the conservative approaches of 'Welch' and 'Brown-Forsthe' where the results were valid and 'Games-Howell' approach was then applied for the comparison. In Table 16, Robust Tests of Equality of Means results are shown.

Table 16. Robust Tests of Equality of Means among Years of update

Robust Tests of Equality of Means				
Zscore(perms)				
	Statistic[a]	df1	df2	Sig.
Welch	46.116	5	117.599	.000
Brown-Forsythe	49.682	5	278.027	.000
a. Asymptotically F distributed.				

As the table shows, both tests were significant and ensuring the possibility of conducting one – way ANNOVA test.

In Table 17, the result of one-way ANNOVA among years of update with respect to Permissions.

Table 17. ANOVA results among Years of update

ANOVA					
Zscore(perms)					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	v	5	25.686	26.397	.000
Within Groups	4658.998	4788	.973		
Total	4787.426	4793			

As the table shows, there is a significant difference among different developers with respect to number of Permissions with F = 26.4 and df = 5.

After finishing ANOVA tests, Appendix B shows the descriptive of the one-way ANOVA conducted among different categories with respect to number of Permissions. Since the data is standardized, using ‘Zperms’ instead of ‘perms’, the mean and StDev in

this case are 1. For example, Communication, has one of the highest variance from other categories (Mean = 1.77 i.e. 77% difference and StDev = 1.38 i.e. 38% difference).

5.6 Direct, indirect, and total effect

As explained in previous sections, AMOS was used for modeling. In AMOS, it is possible to calculate error of variance and indirect effect before conducting any analysis such as multiple regression. In Table 18 and Table 19, standardized indirect effects lower and upper bounds respectively.

Table 18. Standardized Indirect Effects - Lower Bounds (PC) (Group number 1 - Default model

	ZRevInsRatio	NewCategory	Yearupdated	Zperms	ZReviews
ZReviews	0	0	0	0	0
ZReviewScore	0	0	0	0	0
ZInstallCount	0.038	0	0	0.046	0

Table 19. Standardized Indirect Effects - Upper Bounds (PC) (Group number 1 - Default model

	ZRevInsRatio	NewCategory	Yearupdated	Zperms	ZReviews
ZReviews	0	0	0	0	0
ZReviewScore	0	0	0	0	0
ZInstallCount	0.072	0	0	0.092	0

In Table 20 and Table 21, standardized direct effects lower and upper bounds respectively.

Table 20. Standardized Direct Effects - Lower Bounds (PC) (Group number 1 - Default model)

	ZRevInsRatio	NewCategory	Yearupdated	Zperms	ZReviews
ZReviews	0.11	0	0	0.14	0
ZReviewScore	0.283	0.057	0.111	-0.159	0
ZInstallCount	-0.11	0	0	0.079	0.294

Table 21. Standardized Direct Effects - Upper Bounds (PC) (Group number 1 - Default model)

	ZRevInsRatio	NewCategory	Yearupdated	Zperms	ZReviews
ZReviews	0.167	0	0	0.203	0
ZReviewScore	0.318	0.1	0.164	-0.112	0
ZInstallCount	-0.073	0	0	0.179	0.516

On the other hand, Table 22 and Table 23, standardized total effects lower and upper bounds respectively.

Table 22. Standardized Total Effects - Upper Bounds (PC) (Group number 1 - Default model)

	ZRevInsRatio	NewCategory	Yearupdated	Zperms	ZReviews
ZReviews	0.11	0	0	0.14	0
ZReviewScore	0.283	0.057	0.111	-0.159	0
ZInstallCount	-0.047	0	0	0.147	0.294

Table 23. Standardized Total Effects - Upper Bounds (PC) (Group number 1 - Default model)

	ZRevInsRatio	NewCategory	Yearupdated	Zperms	ZReviews
ZReviews	0.167	0	0	0.203	0
ZReviewScore	0.318	0.1	0.164	-0.112	0
ZInstallCount	-0.025	0	0	0.246	0.516

5.7 Multiple regression – Bootstrapping

Finally, the last conducted analysis was multiple regression based on the model to test the hypotheses suggested earlier H1 to H3. The multiple regression analysis was conducting to check if there a significant impact by ‘Permissions’ on other variables such as ‘Reviews Count’, ‘Installs Count’, and ’Review Score’. Moreover, Bootstrapping was done (n = 2000 iterations) to increase confidence. Table 24 shows the multiple regression results.

Table 24. Results of Multiple Regression from AMOS

Regression Weights: (Group number 1 - Default model)						
Parameter			Estimate	Lower	Upper	P
ZReviews	< -- -	Zperms	0.17	0.109	0.246	0.001
ZReviews	< -- -	ZRevInsRatio	0.131	0.09	0.184	0.001
ZInstallCount	< -- -	Zperms	0.129	0.069	0.197	0.001
ZReviewScore	< -- -	Zperms	-0.135	-0.165	-0.109	0.001
ZReviewScore	< -- -	Yearupdated	0.182	0.149	0.214	0.001
ZReviewScore	< -- -	NewCategory	0.007	0.005	0.009	0.001
ZInstallCount	< -- -	ZReviews	0.378	0.243	0.672	0.001
ZInstallCount	< -- -	ZRevInsRatio	-0.086	-0.126	-0.058	0.001
ZReviewScore	< -- -	ZRevInsRatio	0.301	0.271	0.334	0.001
Bootstraping = 2000						

As shown in the table, all regressions are significant. Since bootstrapping was done, lower and upper values are also shown indicating valid estimations for all (no zero value in the range).

5.8 Hypotheses results

In table 25 are the results of the hypotheses suggested. All suggested hypotheses were significantly supported. As mentioned earlier, some hypothesis are driven directly from theoretical framework, other ones were done for further analysis.

Table 25. Result of Hypotheses Suggested

No	Hypothesis	Result
Hypothesis 1	'Permissions' (number of permissions) has a significant impact on 'Review Score'.	Supported
Hypothesis 2	'Permissions' (number of permissions) has a significant impact on 'Reviews' (number of reviews).	Supported
Hypothesis 3	'Permissions' (number of permissions) has a significant impact on 'Installs Count'.	Supported
Hypothesis 4	There is a significant relationship between 'Permissions' (number of permissions) and 'Review Score'.	Supported
Hypothesis 5	There is a significant relationship between 'Permissions' (number of permissions) and 'Reviews' (number of reviews).	Supported
Hypothesis 6	There is a significant relationship between 'Permissions' (number of permissions) and 'Installs Count'.	Supported
Hypothesis 7	There is a significant relationship between 'Permissions' (number of permissions) and 'Year Updated'.	Supported
Hypothesis 8	There is a significant relationship between 'Permissions' (number of permissions) and 'Reviews/Installs' (ratio of reviews number / number of installs).	Supported
Hypothesis 9	There is a significant relationship between 'Reviews/Installs' (ratio of reviews number / number of installs) and 'Year Updated'.	Supported
Hypothesis 10	There is a statistically significant difference among 'Developer' with respect to 'Permissions'	Supported
Hypothesis 11	There is a statistically significant difference among 'Category' with respect to 'Permissions'	Supported
Hypothesis 12	There is a statistically significant difference among 'Year Updated' with respect to 'Permissions'	Supported

CHAPTER 6

INTERPRETATION AND CONCLUSION

6.1 Findings

As seen in the results of the analyses done, the model stands valid against all the tests. The variable ‘Permissions’ that stands for the number of permissions requested by an app from user, is correlated with the other variables:

- Reviews: number of reviews
- Review Score: review score of the app based on users’ reviews
- Year Updated: the year of the last updated made
- Installs Count: the number of installs made from the app
- Reviews/Installs Ratio: the ratio of number of reviews made to number of installs, stating the rate of reviewing an app by installers

Also, one-way ANOVA tests showed that number of permissions is actually a difference among different developers, apps categories, and apps updated in different years. This analysis may tell that different type (categories) require different number of permissions to serve the user and vice versa. Also, the number of permissions is significantly changing with time where recently updated apps seem to request more permissions. Finally, also different developers seem to request different number of permissions and vice versa. The fact that apps by same developers seem to request similar number of permissions for different apps raises the concern of whether is due to

similar app nature (category and architecture) or due to the demand of developers for different purposes. Such concerns require deeper and more specific research.

On the other hand, the multiple regression results showed a significant impact of number of permissions requested on variables such as reviews, review score, and installs count. This means that the variance in permissions numbers requested for an app can actually estimate or account for part of the variance in these variables.

6.2 Limitations

For this study, there were several limitations that had affected the results directly and indirectly. Although 5,264 apps is statistically big enough to conduct such analyses, the sample size is small compared to the huge number of apps on Google Play Store, over 3 million apps.

Also, as the scrapping was made in Turkey and no VPN was used, some of the apps were displayed as local apps such as banking apps of local banks. Local may have an indirect effect on the result due to demographics and language differences.

Another limitation was collecting the installs count. This variable shows the total number of installs made. It doesn't show the number of the currently active installs or the number of uninstalls. Users may have installed an app multiple times on different devices from different accounts. This could affect the results.

Similarly, the date collected was the date of last updated, not the date of lunching he app. The study doesn't take into consideration how old or new an app could be. Older apps may possess much bigger number of installs and reviews.

APPENDIX A

FREQUENCY TABLE OF 'CATEGORY'

Category					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Art & Design	113	2.4	2.4	2.4
	Auto & Vehicles	112	2.3	2.3	4.7
	Beauty	74	1.5	1.5	6.2
	Board	124	2.6	2.6	8.8
	Books & Reference	82	1.7	1.7	10.5
	Business	112	2.3	2.3	12.8
	Casual	62	1.3	1.3	14.1
	Comics	80	1.7	1.7	15.8
	Communication	102	2.1	2.1	17.9
	Education	228	4.7	4.7	22.7
	Educational	169	3.5	3.5	26.2
	Entertainment	191	4.0	4.0	30.2
	Finance	156	3.2	3.2	33.4
	Food & Drink	94	2.0	2.0	35.4
	Health & Fitness	78	1.6	1.6	37.0
	House & Home	98	2.0	2.0	39.0
	Lifestyle	94	2.0	2.0	41.0
	Maps & Navigation	124	2.6	2.6	43.6
	Music	55	1.1	1.1	44.7
	Music & Audio	110	2.3	2.3	47.0
	News & Magazines	102	2.1	2.1	49.1
	Parenting	75	1.6	1.6	50.7
	Personalization	120	2.5	2.5	53.2
	Photography	256	5.3	5.3	58.5
	Productivity	115	2.4	2.4	60.9
	Puzzle	208	4.3	4.3	65.2
	Racing	94	2.0	2.0	67.2
	Role Playing	121	2.5	2.5	69.7
	Shopping	102	2.1	2.1	71.8
	Simulation	108	2.2	2.2	74.1

	Social	122	2.5	2.5	76.6
	Sports	204	4.2	4.2	80.9
	Strategy	144	3.0	3.0	83.9
	Tools	159	3.3	3.3	87.2
	Travel & Local	136	2.8	2.8	90.0
	Trivia	67	1.4	1.4	91.4
	Video Players & Editors	125	2.6	2.6	94.0
	Weather	130	2.7	2.7	96.7
	Word	159	3.3	3.3	100.0
	Total	4803	100.0	100.0	

APPENDIX B

DESCRIPTIVES OF ANOVA AMONG CATEGORIES

WITH RESPECT TO PERMISSIONS

Descriptives

Zperms

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
Art & Design	113	-.4691713	.61845308	.05817917	-.5844459	-.3538967	-1.16342	2.54290
Auto & Vehicles	112	-.4850341	.64505235	.06095172	-.6058140	-.3642542	-1.16342	1.55455
Beauty	74	-.3804210	.62034815	.07211401	-.5241440	-.2366980	-1.16342	2.66645
Board	124	-.5915341	.47535422	.04268807	-.6760325	-.5070357	-1.53406	1.18392
Books & Reference	82	-.1539774	.72761748	.08035191	-.3138525	.0058977	-1.28697	2.54290
Business	112	.6753996	1.20373185	.11374197	.4500123	.9007869	-1.16342	7.23759
Casual	62	-.4261442	.33478458	.04251768	-.5111636	-.3411248	-1.53406	.68974
Comics	80	-.6121082	.53615966	.05994447	-.7314247	-.4927917	-1.53406	1.92518
Communication	102	1.7689352	1.38459188	.13709502	1.4969755	2.0408948	-1.53406	5.38442
Education	228	-.3863961	.57940989	.03837236	-.4620076	-.3107845	-1.28697	1.92518
Educational	169	-.7174955	.24749724	.01903825	-.7550805	-.6799105	-1.28697	.07202
Entertainment	191	-.2468683	.74259161	.05373204	-.3528563	-.1408804	-1.53406	4.76670
Finance	156	.5559000	.92056689	.07370434	.4103054	.7014946	-.91634	4.51961
Food & Drink	94	.0220748	.60505623	.06240678	-.1018527	.1460023	-1.16342	1.67809
Health & Fitness	78	.3745433	1.05107397	.11901069	.1375627	.6115239	-1.03988	3.90189
House & Home	98	.1136199	.63788384	.06443600	-.0142678	.2415075	-1.28697	2.54290
Lifestyle	94	.2455059	.82415689	.08500529	.0767023	.4143096	-1.28697	3.53126
Maps & Navigation	124	.3579634	.70547560	.06335358	.2325589	.4833680	-.91634	2.54290
Music	55	-.3188310	.51751341	.06978149	-.4587344	-.1789275	-1.16342	1.55455
Music & Audio	110	.1102046	.82778733	.07892643	-.0462250	.2666342	-1.53406	3.28417
News & Magazines	102	.2004074	.72266822	.07155482	.0584619	.3423529	-1.53406	2.91354
Parenting	75	-.4056863	.81782037	.09443376	-.5938497	-.2175229	-1.16342	3.28417
Personalization	120	.9141784	2.02972263	.18528748	.5472907	1.2810661	-1.53406	10.57328
Photography	256	.0232761	.80680804	.05042550	-.0760273	.1225796	-1.28697	5.26088

Productivity	115	.7606433	1.84084542	.17165972	.4205867	1.1006999	-1.53406	14.89733
Puzzle	208	-.5730255	.36852068	.02555231	-.6234016	-.5226493	-1.16342	.44265
Racing	94	-.4747309	.42507995	.04384365	-.5617957	-.3876661	-1.53406	.56620
Role Playing	121	-.2873833	.40475287	.03679572	-.3602363	-.2145304	-1.03988	1.06037
Shopping	102	.6279675	.92529681	.09161804	.4462219	.8097130	-1.53406	4.51961
Simulation	108	-.4210150	.30001449	.02886891	-.4782442	-.3637857	-1.16342	.31911
Social	122	.9267016	1.39906003	.12666494	.6759349	1.1774682	-1.53406	10.20265
Sports	204	-.1181431	.57626586	.04034668	-.1976954	-.0385908	-1.03988	2.17227
Strategy	144	-.2162518	.58259690	.04854974	-.3122197	-.1202839	-1.16342	2.78999
Tools	159	.3206608	1.25952496	.09988683	.1233751	.5179465	-1.53406	6.74341
Travel & Local	136	.3699779	.88176611	.07561082	.2204430	.5195129	-1.28697	3.40771
Trivia	67	-.4092513	.42696059	.05216154	-.5133952	-.3051074	-1.16342	.68974
Video Players & Editors	125	.3299787	1.09606777	.09803528	.1359394	.5240179	-1.53406	3.77835
Weather	130	-.0287179	.61739273	.05414892	-.1358529	.0784171	-1.53406	2.91354
Word	159	-.4796575	.35513417	.02816397	-.5352839	-.4240310	-1.16342	.56620
Total	4803	0E-7	1.00000000	.01442625	-.0282820	.0282820	-1.53406	14.89733

APPENDIX C

PYTHON CODE

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import sys
import subprocess
import time
import csv
from selenium.common import exceptions


'''Getting command line arguments'''
chrome_executable_path  =  str(sys.argv[1])
google_play_link  =  str(sys.argv[2])

''' Setting up the browser '''
browser  =  webdriver.Chrome(executable_path = chrome_executable_path)

#Opening url in browser
browser.get(google_play_link)
time.sleep(1)

#Getting body element of the DOM
elem  =  browser.find_element_by_tag_name("body")

#Setting up the number of scrolls (infinity scroll)
```

```
try:
        no_of_pagedowns  =  int(sys.argv[3])
except Exception as e:
        print(e)
        no_of_pagedowns  =  50

#Scroll the page
while no_of_pagedowns:
        try:
                more_button  =  elem.find_element_by_id("show-more-button")
                display  =  more_button.get_attribute("style")
                if not display:
                        more_button.click()
                        time.sleep(0.5)
        except:
                pass
        elem.send_keys(Keys.PAGE_DOWN)
        time.sleep(0.5)
        no_of_pagedowns- = 1

app_links  =  set()

#Getting all the apps which are present in the given page
containers  =  browser.find_elements_by_class_name("poRVub") #This class needs to be changed in case page structure(app link class) is different.

#Iterate through the apps and getting each app's link and app id
for container in containers:
        try:
```

```
            link = container.get_attribute("href")
            app_links.add(tuple({"app_id":link.split(" = ")[1], "link":link}.items()))
        except Exception as e:
            print(e)
            pass

#Printing total number of apps found
print("Total Apps : " + str(len(app_links)))

#Opening a file for writing app details
file = open("apps.csv", "wb")
writer = csv.writer(file)
writer.writerow(["Id", "Name", "Category", "Reviews", "Review Score", "Logo",
"Installs", "Last Updated", "Developer Name", "Permissions"])
count = 0

for app_link in app_links:
    try:
        app_link = dict(app_link)
        app_id = app_link["app_id"]
        browser.get(app_link["link"]) #Visiting every app's url
        elem = browser.find_element_by_tag_name("body") #Getting body
element of app page

        app_name = elem.find_element_by_class_name("AHFaub") #Getting
App name
        logo = elem.find_element_by_class_name("T75of") #Getting App logo
        reviews = elem.find_element_by_class_name("AYi5wd") #Getting App
reviews
```

```
                review_score = elem.find_element_by_class_name("BHMmbe")
#Getting App review score

                dev_cat = elem.find_elements_by_class_name("hrTbp")

                developer = dev_cat[0] #Getting App developer

                app_category = dev_cat[1] #Getting App category


                additional_details = elem.find_element_by_class_name("IxB2fe")
#Additional details

                app_details = additional_details.find_elements_by_class_name("htlgb")

                last_updated = app_details[1] #Getting last updated

                installs = app_details[4] #Getting number od installs


                permissions = subprocess.Popen(["node", "npm_play.js", app_id], stdout
= subprocess.PIPE).stdout.read() #Getting permissions required
        except Exception as e:

                print ("couldn't get all data")

                count- = 1

                pass



        ''' Writing all the app detail to file '''

        try:

                writer.writerow([app_id, app_name.text.encode('utf-8'),
app_category.text.encode('utf-8'), reviews.text.encode('utf-8'),
review_score.text.encode('utf-8'), logo.get_attribute("src"), installs.text.encode('utf-8'),
last_updated.text.encode('utf-8'), developer.text.encode('utf-8'), permissions.encode('utf-
8')])

        except Exception as e:

                print ("couldn't write missing data")

        #       print(app_id, app_name.text.encode('utf-8'),
app_category.text.encode('utf-8'), reviews.text.encode('utf-8'),
```

```
review_score.text.encode('utf-8'), logo.get_attribute("src"), installs.text.encode('utf-8'),
last_updated.text.encode('utf-8'), developer.text.encode('utf-8'), permissions.encode('utf-
8'))
                pass
        count+ = 1
        #printing count
        print(count)
print("Total Apps : " + str(len(app_links)))
```

REFERENCES

Albrecht, U., Hasenfuß, G., & Jan, U. V. (2018). Description of cardiological apps from the German app store: semi-automated retrospective app store analysis. *JMIR mHealth and uHealth*, 6(11). doi:10.2196/11753

AppleInsider, S. S. (2018, July 05). Apple details history of app store on its 10th anniversary. Retrieved from https://appleinsider.com/articles/18/07/05/apple-details-history-of-app-store-on-its-10th-anniversary

Atkinson, M., & Atkinson, M. (2017, March 15). Apps permissions in the google play store. Retrieved from http://www.pewinternet.org/2015/11/10/apps-permissions-in-the-google-play-store/

Barrera, D., Kayacik, H. G., Oorschot, P. C., & Somayaji, A. (2010). A methodology for empirical analysis of permission-based security models and its application to android. Proceedings of the 17th ACM Conference on Computer and Communications Security - CCS 10. doi:10.1145/1866307.1866317

Chia, P. H., Yamamoto, Y., & Asokan, N. (2012). Is this app safe? Proceedings of the 21st international conference on world wide web - WWW 12. doi:10.1145/2187836.2187879

Clement, J. (2019, July). Google play store: number of apps 2019. Retrieved from https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

Enck, W., Ongtang, M., & Mcdaniel, P. (2009). Understanding android security. *IEEE Security & Privacy Magazine*, 7(1), 50-57. doi:10.1109/msp.2009.26

Ericsson Mobility Report: 70 percent of world's population using smartphones by 2020. (2017, June 22). Retrieved from https://www.ericsson.com/en/press-releases/2015/6/ericsson-mobility-report-70-percent-of-worlds-population-using-smartphones-by-2020

Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011). Android permissions demystified. Proceedings of the 18th acm conference on computer and communications security - CCS 11. doi:10.1145/2046707.2046779

Felt, A. P., Ha, E., Egelman, S., Haney, A., Chin, E., & Wagner, D. (2012). Android permissions. Proceedings of the eighth symposium on usable privacy and security - SOUPS 12. doi:10.1145/2335356.2335360

Global number of active apps in the apple store 2020. (n.d.). Retrieved from https://www.statista.com/statistics/604277/number-of-active-apps-in-the-apple-apple-store-worldwide/

Google play store: number of apps 2019. (n.d.). Retrieved from https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

Kochmann, M., & Locatis, C. (2016). Telemedicine in the apple app store: an exploratory study of teledermatology apps. 2016 international conference on collaboration technologies and systems (cts). doi:10.1109/cts.2016.0099

M. R. (2013, March). What is app store (application store)? - definition from whatis.com. Retrieved from https://searchmobilecomputing.techtarget.com/definition/app-store-application-store

Obar, J. A., & Oeldorf-Hirsch, A. (2018). The biggest lie on the internet: ignoring the privacy policies and terms of service policies of social networking services. *Information, Communication & Society*, DOI: 10.1080/1369118X.2018.1486870

Permissions overview: android developers. (n.d.). Retrieved from https://developer.android.com/guide/topics/permissions/overview?hl = en#normal_permissions

Privacy, security, and deception - developer policy center. (n.d.). Retrieved from https://play.google.com/about/privacy-security-deception/#!?zippy_activeEl = personal-sensitive#personal-sensitive