IZMIR KATIP CELEBI UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

SOCIAL MEDIA ANALYSIS BASED ON MOBILE SYSTEMS

M.Sc. THESIS

Levent YAVUZ

Department of Computer Engineering

Thesis Advisor: Assoc. Prof. Dr. Ayşegül ALAYBEYOĞLU

APRIL 2017

IZMIR KATIP CELEBI UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

SOCIAL MEDIA ANALYSIS BASED ON MOBILE SYSTEMS

M.Sc. THESIS

Levent YAVUZ
(601514009)

Department of Computer Engineering

Thesis Advisor: Assoc. Prof. Dr. Ayşegül ALAYBEYOĞLU

APRIL 2017

İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL TABANLI SİSTEMLERDE SOSYAL MEDYA ANALİZİ

YÜKSEK LİSANS TEZİ

Levent YAVUZ
(601514009)

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Ayşegül ALAYBEYOĞLU

NİSAN 2017

Levent Yavuz, a M.Sc. student of iKCU Graduate School of Science Engineering and Technology student ID 601514009, successfully defended the thesis entitled "SOCIAL MEDIA ANALYSIS BASED ON MOBILE SYSTEMS", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :   **Assoc. Prof. Dr. Ayşegül ALAYBEYOĞLU** ....
Izmir Katip Celebi University

Jury Members :   **Assoc. Prof. Dr. Femin Yalçın** ....
Izmir Katip Celebi University

                            Assist. Prof. Dr. Rıfat Benveniste ....
Avrasya University

Date of Submission   **: 14.03.2017**
Date of Defense   **: 07.04.2017**

To my family of Yavuz,

FOREWORD

Primarily, I would like to express my thanks to my advisor, Assoc. Prof. Dr. Ayşegül Aleybeyoğlu, for her guidance, good intentions, supports and ideas throughout the research.

I would like to thank to my parents, Faruk Yavuz and Birsen Yavuz, for their supports.

APRIL 2017 Levent YAVUZ

x

TABLE OF CONTENTS

ABBREVIATIONS

SDK	**:**	Software Development Kit
API	**:**	Application Programming Interface
GPS	**:**	Global Positioning System
IDE	**:**	Integrated Development Environment
SQL	**:**	Simple Query Language
HTTP	**:**	Hyper-Text Transfer Protocol
JSON	**:**	JavaScript Object Notation
APP	**:**	Application
PIN	**:**	Personel Identification Number
XML	**:**	Extensible Markup Language
OS	**:**	Operating System
SD-Card	**:**	Secure Digital Card

LIST OF TABLES

LIST OF FIGURES

SOCIAL MEDIA ANALYSIS BASED ON MOBILE SYSTEMS

SUMMARY

The use of internet and social media has increasing with the developing new technologies and large number of people have actively using of the internet. At the same time the development of mobile technologies and the widespread usage of the mobile devices that can be shown as the reason for that. In this way, social media has become a virtual environment in which people interact with each other.

Some of social media platforms such as Twitter have become very popular and most of people share some information about themselves on these platforms. They also share their ideas, thoughts, lives and much more things with other people through on these platforms. On this thesis that is aimed to take meaningful results from the information that people share on the Twitter platform. In this way, very useful information can be revealed. Furthermore, can be obtained which people have shared about a specified keyword and what they think that about it. All these references play an important role with about people, universities, companies, events and so on.

Likewise, all these social media analyses are performed as thesis product on Android based mobile environment and thanks to this android app, can be displayed the important information of users who post tweet or tweets with about specified keywords. Their locations and countries can be also displayed. The other a different inference is what time periods of users generally post tweet on Twitter. People use some operating systems to connect from internet to Twitter platforms and these are mobile and desktop operation systems. Therefore, can be analysed to which operating system is used when posting a tweet on Twitter with specified keyword. This feature includes different results according to different keywords.

Besides all this, the information all the people in the world shared on Twitter that is can be view on the mobile application. The main goal of at this point is to analyse all the interactions on Twitter and as a result the getting significant results. In this way, meaningful results can be extract and these results can be accessed with Android mobile devices.

MOBİL TABANLI SİSTEMLERDE SOSYAL MEDYA ANALİZİ

ÖZET

Gelişen yeni teknolojiler ile birlikte internet ve sosyal medya kullanımı artmakta ve çok sayıda insan aktif olarak internet kullanmaktadır. Aynı zamanda mobil teknolojilerdeki gelişmeler ve mobil cihazların yaygın kullanımı buna bir sebep olarak gösterilebilmektedir. Bu sayede sosyal medya insanların birbirleriyle etkileşim kurduğu bir sanal ortam haline gelmektedir.

Twitter gibi sosyal medya platformları çok popüler bir hale geldi ve çok sayıda insan bu platformlarda kendileri hakkında bilgiler paylaşmaktadır. Ayrıca bu platformlar aracılığıyla fikirlerini, düşüncelerini, yaşamlarını ve çok daha fazlasını diğer insanlarla paylaşmaktalar.

Bu tezde insanların Twitter platformu üzerinde paylaştıkları bilgilerden anlamlı sonuçlar elde etmek hedeflemektedir. Bu sayede çok önemli bilgiler açığa çıkabilmektedir. Dahası, belirli bir anahtar kelime hakkında hangi insanların ne paylaştıkları ve bu konuda ne düşündükleri hakkında bilgi alınabilir. Tüm bu referanslar, insanlar, üniversiteler, şirketler, etkinlikler ve benzeri konular hakkında önemli bir rol oynamaktadır.

Benzer şekilde, bütün bu sosyal medya analizi Android tabanlı mobil ortamda tez ürünü olarak geliştirildi ve bu Android uygulaması sayesinde, belirtilen anahtar kelimeler hakkında tweet ya da tweetler gönderen kullanıcıların önemli bilgileri görüntülenebilmektedir. Ayrıca bu kullanıcıların bulundukları konumlar ve ülkelerde görüntülenebilmektedir. Diğer bir başka çıkarım ise, kullanıcıların genel olarak Twitter üzerinde tweet attıkları zaman aralıklarıdır.İnsanlar internet üzerinden Twitter platformuna bağlanmak için bazı işletim sistemlerini kullanıyorlar ve bu işletim sistemleri mobil ve masaüstü işletim sistemleridir. Bu nedenle, belirtilen bir kelime hakkında tweet atan kullanıcıların hangi işletim sistemini kullandığının analizi de yapılabilmektedir. Bu özellik farklı anahtar kelimelere göre farklı sonuçlar içermektedir.

Bütün bunlara ek olarak, Twitter üzerinde paylaşılan ve dünyanın her yerindeki insanlara ait olan bilgiler, bu mobil uygulama aracılığı ile de izlenebilmektedir. Bu noktada asıl hedef Twitter üzerindeki tüm etkileşimleri analiz etmek ve bunun sonucunda değerli sonuçlar elde etmektir. Bu yolla, anlamlı sonuçlar çıkarılabilmekte ve bu sonuçlara mobil Android cihazlarıyla erişim sağlanabilmektedir.

1. INTRODUCTION

The emergence of smart devices and high speed internet, internet users are able to engage with social media platform like Twitter, Facebook and etc. The volume of social data is rapidly growing and Global Web Index statistics says the number of mobile social users are increased with 17 % of yearly. [1] In this way, social media platforms have become very popular websites to connect all the people in the world. Most important feature of social media environments is provides the opportunity to reach lots of people in a short period of time. This characteristic of social media is appealing to people. The result of increasing number of social media users is social media channels have become popular environment to discuss ideas and interact with people worldwide.

With such large volume of social media data being generated, lots of challenges have occurred. The most important of them are accessing social media data, storing these data, processing and getting meaningful results. Also 80% of generated data is unstructured. [2] High quality hardware resources are needed to process data such as social media data. Therefore, mobile devices, desktop machines or servers etc. cannot be compared with each other in terms of hardware resources because mobile devices have limited hardware resource. For these reason the social media analysis on mobile platforms is hard to implements and usage.

With the developing technology, people were found more ways to interact with people, from simple text messages, picture and video sharing, posting tweets, writing blogs and etc. Nowadays, social media platforms are usually aggregate feedback about current events, companies and the other things, provides some opportunity to evaluate their products or other things. When any event occurs or new technologies or a product is launched, most of users start posting tweet, retweeting, posting comments and etc. on social media of Twitter. After all of these, Twitter becomes a social data repository. There are most of user information about their opinion, user names, their place and etc. in that the data repository.

Increasing tendency in big data problems itself and solutions in 21th century makes technology shifts though data computing and technologies and this makes project an up-to-date topic and an interest area. Main goal of this project is to develop and provide end user with efficient tool for finding intelligence information from the Twitter social network data and all these processes have made through mobile technologies..

1.1 Purpose of Thesis

The purpose of this thesis is to provide end users with adaptable situation awareness and efficient tools for finding intelligent information using social media of Twitter based on internet sources. Such projects getting more interesting and common day by day in nowadays due to social data processing and the other processing types due to big data problem and technological advances.

Since technology is taking a big part in human lives in modern century, communication and blogging through well-known platforms such as Twitter, is on the rise. And these social platforms have become an important key source of information during evaluate, protest, crisis and disaster situations.

With the android application, the users can displayed everything about keyword which after inserting a desired keyword. Tweets that posted on Twitter can displayed by the mobile application with easily. But not only tweets but also users' user names, their nick names, their genders, their places and most of thing like this are visualized within mobile application. These tweets are actually a text message and also sender is user and receiver is people who user of Twitter.

The another purpose is observe what the users are talking about and what is happening in the world right now. However, in such events, these platforms are sometimes misused for false publicity and also propaganda. In order to avoid such events before it happens and after or during such events, common opinions and talks on platforms and those who are responsible for such actions need to be investigated though their posts.

There exist example systems in business world for analysing thoughts on a specific company such as banks or product or states or security systems for such problem that mentioned above. These example systems are generally based on desktop or server

based applications. The mobile android application provides to a system that collects stores and analyses tweets for users to gathering important information. Users can easily conduct data analysis with the app that gives the opportunity use in any place where available internet. During the app is running, the internet connectivity is mostly important because Twitter data is obtained over from internet.

1.1.1 Thesis scope and outcomes

The scope of the project of thesis is building mobile software which consists of a crawler that collects social network data and an analyser that analyses founded data to find important knowledge for user. Crawler feature is collect Twitter data in real time and in the same time collects past time tweets about given keyword or keywords. Hereby, the end users can display streaming tweets and posted tweets in past time with easily.

Main outcomes are providing mobile software that is able to run automated search on tweets with given keyword and stores in regularly and then end user capability of searching tweets with given keywords. The most important one is analysing tweets with visualization. These visualizations are user interactions of each other. Such as used mobile operating system when posting tweets with specified keywords. Also gender analysis of users, posting tweet time intervals, users profile images, user names and nicknames and user tweet texts are visualized with on Android based mobile application.

1.2 Some Definitions and Knowledge

1.2.1 Twitter

Twitter is a social networking platform where its users post tweets and sending message with each other and all people of using Twitter. Tweet means 140 characters of text message but feature of tweet's is posting text message with 140 characters to everyone on Twitter. [3]

Figure 1.1 : Offivial Twitter logo. [4]

Users can access Twitter through with web site interface or its mobile application. Twitter has gained worldwide popularity since 2006, the year it was founded, and the number of active users increased day by day. Nowadays, Twitter have 313 million monthly active users [5] and the 82% of this active users are active mobile users. It has language support with over 40 and so Twitter have worldwide usage and worldwide users.

Due to all these feature, Twitter data is carrying of significant value. Shared data on Twitter platform is important for analysers because there are too many active users and too many opinions and etc. on there. In this way, the data that analysed within from Twitter data is gives the most accurate result.

Many companies, commercial organizations, governments consider Twitter data about their products, services, politics and something like this. Many of people and companies are using Twitter to affect other people and taking important feedback.

1.2.2 Android operating system

Android operating system is a kind of mobile operating system and developed by Google. [6] Android operating system is under open source licence. The first Android mobile operating system released in 2008. [7] Android operating system has been constantly updating since 2008. There are totally 11 versions. These version names are Alpha, Beta, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow and Nougat. There are some definition of Android operating system versions in Table 1.1 and there are

organized into code name, version number, initial release date, API level and support status features.

Table 1.1 : Android Operating System Versions [8].

Code name	Version number	Initial release date	API level	Support status
Alpha	1.0	September 23, 2008	1	Unsupported
Beta	1.1	February 9, 2009	2	Unsupported
Cupcake	1.5	April 27, 2009	3	Unsupported
Donut	1.6	September 15, 2009	4	Unsupported
Eclair	2.0 – 2.1	October 26, 2009	5 – 7	Unsupported
Froyo	2.2 – 2.2.3	May 20, 2010	8	Unsupported
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10	Unsupported
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13	Unsupported
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15	Unsupported
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18	Unsupported
KitKat	4.4 – 4.4.4	October 31, 2013	19	Unsupported
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22	Unsupported
Marshmallow	6.0 – 6.0.1	October 5, 2015	23	Supported
Nougat	**7.0 – 7.1.1**	**August 22, 2016**	**24 – 25**	**Supported**

The Nougat is last version of Android operating system that has been supported with Marshmallow version. Lollipop of them is most widely used version of Android operating system. Figure 1.2 represent usage ratio of Android operating system versions.
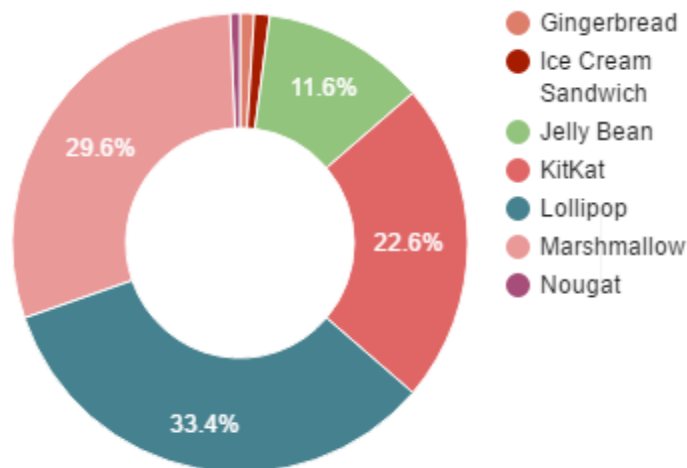
Figure 1.2 : Usage percentages of Android Operating System versions. [9]

To creating Android based application that needed to Android Software Development Kit. Mobile application developers create own applications using with

the Android SDK and also applications are written using the Java programming language.

1.2.3 Application Programming Interface

Application programming interface [10] is an interface provided by an application that shares its capabilities so that an application's capabilities can be used in another application. APIs are commonly used by web developers and program developers. The API refers to the software's operation, input, output and underlying types. Also API defines functionality that is independent of its own applications. APIs are a set of structures that enable two separate databases or software to interact smoothly and quickly and provides developer's job easier. Provides the opportunity to be more functional within the project.

There are many different API types for operating systems, applications or web sites. Examples of the leading APIs are Twitter, Google Maps, Facebook and YouTube.

1.2.4 Android System Architecture

Android is a comprehensive mobile software stack that includes the operating system, middleware and important applications. This rich software package is used in mobile technology through the Android Software Development Kit. The Android SDK provides the tools and APIs needed to get started developing applications that use the Java programming language on the android platform. [11]

Android provides application framework, Java support, handset layouts, storage, media support, additional hardware support, rich development environment, connectivity, market and multi-touch features.

SQLite database software is used for data storage purposes. Android can use GPS, accelerometers, cameras and etc. to provide additional hardware support. Android market is a store for applications that can be downloaded and installed to target hardware, without the using of any desktop operating system. Also Android include a device emulator, performance profiling, debugging tools, most of plugins to using with IDEs such as Android Studio, Eclipse. The Figure1.3 shows the components of the Android operating system architecture.
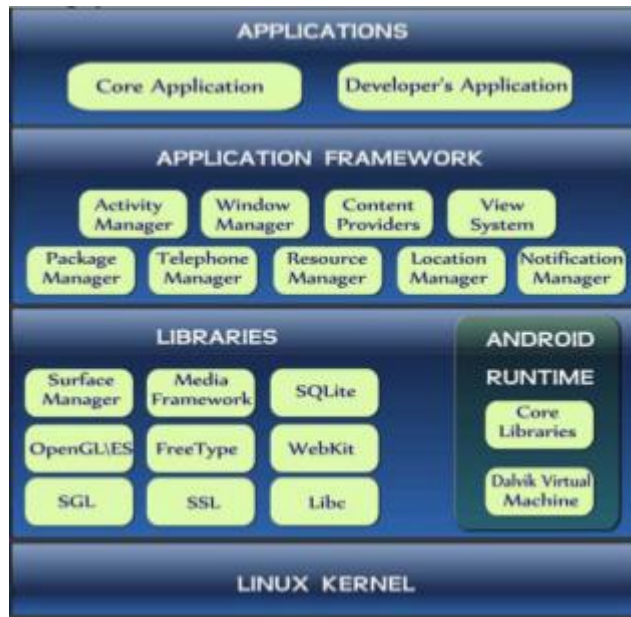
Figure 1.3 : Android System Architecture. [12]

The Android operating system has core applications including, browsers, contacts, calendar and others. All applications are implemented with Java programming language and also developers can develop their application using Java programming language. Developers can access some APIs with used by the core applications. The most of libraries used in the Android system are is the C/C++ library. Some of libraries include 3D libraries, SQLite. [13] Android contains core libraries that provide more of the functionality found in the core libraries of the Java programming language. Android applications runs on own instance of Dalvik virtual machine. Android is Linux based and the kernel provides a virtualization layer between core hardware and software.

2. RELATED WORK

The emergence of mobile devices and the reduction of the cost of the internet made it easier for people to use more than one social networking service at same time. Web search methods such as Google search are not enough for Twitter content search. Many companies, researches, developers focused on tweets content analysis by appliying some techniques on tweets.

Some works are being done and developed in the area of Twitter analysis. In 2012, a group of researchers identified a software architecture that aggregated tweets submitted in a given geographical location and within a specified time periods. [14] This tweet gathering process was performed using Twitter Streaming API.In another study at Arizona State University, Twitter data was analyzed and the detection of people using harmful substances was studied. [15] The other study in 2012, [16] a group of reaserchers described their architecture to analyse Twitter data with using Twitter Rest API and Twitter Streaming API in PHP language and then MySQL database is used for storing data that coming from Twitter. In 2010, a study is conducted to determine trends using the Twitter Streaming API. [17] In this way, the most posted words in the tweets were instantly analyzed on Twitter. In paper, [18] tweets are posted by the users are analyzed and displayed user's city level locations as a results.

The common feature of all these studies are that they are implemented on desktop based platforms. These kinds of works require high speed internet connection and rich hardware resource. Such studies are possible with cloud computing technologies [19] [20] that are popular with the developing technologies. The cloud computing[21] technologies include inside of some technologies such as Apache Hadoop platform, [22] [23] Apache Spark platform, [24] [25] and R programming language. [26] As s result of the literature survey, there was no social media analysis work on mobile platforms. However, since the use of 4.5G has become widespread, mobile devices can actually perform as good quality work as some desktop devices.

3. TWITTER DATA ANALYSIS

3.1 Twitter APIs

Twitter provides a website for developers [27] and API resources [28] to the usage of developers. Programmers are use the Twitter API to develop applications, websites and other projects that interact with Twitter. Applications talk to the Twitter API over HTTP, the same protocol that browser uses to visit and interact with web pages. For that reason, when using Twitter API, internet connection is needed. Generally, these APIs are like as a bridge between application and Twitter.

There are many Twitter API libraries built for the Twitter platform. ASPTwitter [29] is a library to implement Twitter API within an ASP language. Twitcurl [30] is a C++ Twitter API library. LINQ2Twitter [31], TweetSharp [32] libraries are for .Net libraries. Twitter4J [33] is a Twitter API library for Java platform. TwitterJSClient [34] is Twitter client library written with Javascript and packaged as a node module. STTwitter [35] and FHSTwitterEngine [36] are Objective-C libraries. Python-twitter [37] and tweepy [38] libraries are a Python libraries for interacting with Twitter. And then, there are many other libraries similar to that. All libraries have a single purpose is interact Twitter with different programming languages.

Twitter basically provides two API to programmers. These are Streaming API [39] and Rest API. [40] The Streaming API is for application developers who want to receive a real-time stream of the public Tweets on Twitter. In this way, Streaming API is provides to observe what is going on in the world. Tweets which is displayed with Streaming API is displayed in real time.

The Rest API provides implementing access to read and write Twitter data. Rest API provides especially creating a new tweet, reading user profile and displaying follower data, and more. In addition to the Rest API, the Twitter API includes the Restful methods to send and receive twitter data. These methods are "Get" and "Post" methods. Get methods provides to obtain information such as user direct messages and post methods provides to sending information such as sending to new direct messages from an application with Rest API and the Rest API generally

provides user information such as friendships, messages, statuses and etc. so tweet text analysis is not suitable with Rest API. Both Streaming API and Rest API is important APIs for using and implementing Twitter data.

3.2 OAuth

The OAuth protocol is an open protocol that provides simple, standardized safety way authorization for web, mobile and desktop applications. With the OAuth so the protected data in a secure manner to deliver services have been simplified. [41] There are two protocol of OAuth. These are OAuth 1.0 and OAuth 2.0 protocols. The Oath 2.0 protocol is a version that released in 2006 after a series of edits to provide facilities for developers. [42] OAuth 2.0 is an open authorization protocol that allows applications to access data between each other. OAuth protocols used for application authorization not for user authentication.

Today, many social media platforms such as Twitter, Facebook, Google, etc. support the OAuth 2.0 protocol. For example, a website or an application offers login options to its users through other accounts such as social media accounts, which is possible because these applications implement user authentication and application authorization using the OAuth protocol.

OAuth is not a client or server library. It is only a protocol, that created for authorization between applications. However, there are server and client libraries developed on various platforms (Java, Python, etc.) in accordance with the OAuth standard. There are two type of applications that can be implemented with these libraries. First is a server-side authorization application. With this application is created a validation and authorization server. In this way the other applications or softwares can also use this application to authorize with this application, just like Twitter and Facebook. The second is a client application that conforms to the OAuth 2.0 protocol. With this application, the resource where found in a server application with OAuth 2.0 support is easily accessible. Own mobile application works like as the client application of second one.

Sides that communicate with each other within the OAuth 2.0 protocol are resource owner, client application, authorization server and resource server. Resource owner is a person or application that has the data stored on the system. For example,

12

tweets, images or videos which shared on Twitter. Client application is the application that has reached the resource owner's ownership. For instance, if a person wants to share own tweets on own personal blog page, who need to access own blog page's resource server such as Twitter servers and in this case client becomes personal blog page. Authorization server is a server that authorized by applications designated as client applications. Resource server is the place where preserved data is desired to be accessed. Such as API endpoints that provide tweets are called resources. The abstract OAuth 2.0 flow illustrated in Figure 3.1 describes the interaction between the four sides.
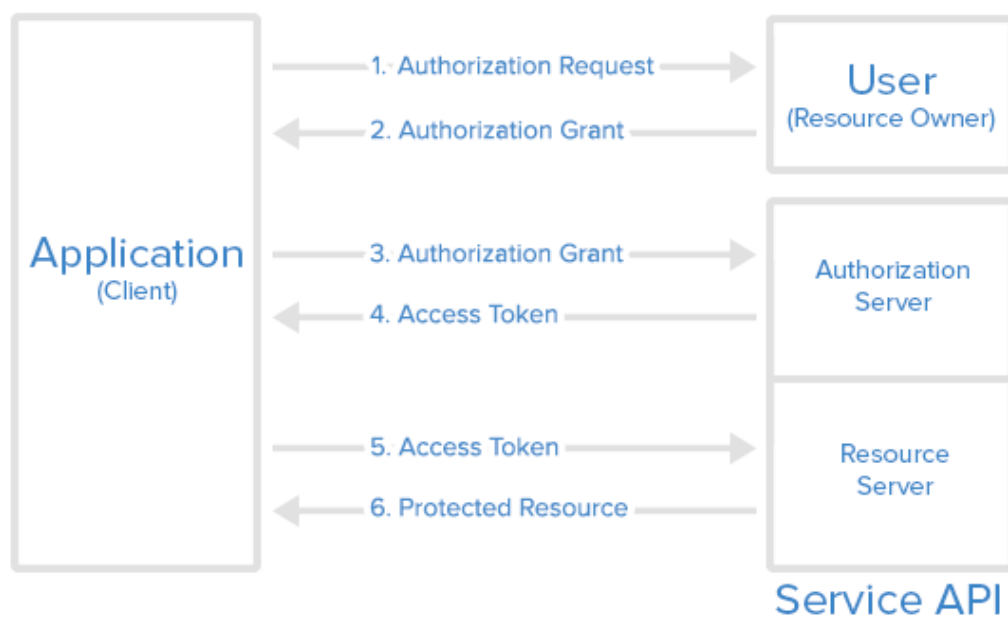
Figure 3.1 : Abstract OAuth protocol flow. [43]

The client application sends authorization request to access service resources from the user and this request can be made directly to the resource owner. The starting point of the roaming system is the resource owner. Resource owner initiates the authorization flow through from the client application. The authorization grant request is transferred to the authorization server through the application. When the user is authenticated, an access token is sent by the client application to access the resource. It is now possible to access the information on the resource server with the access token.

Twitter uses OAuth to provide authorized access to its API while sending secure authorized requests to the Twitter API. In this wise, Twitter provides security and standardization. Because, the users of Twitter are not required to share their

passwords and any other information with other applications or software. Many client libraries are compatible with Twitter's OAuth implementation. There are two types Twitter API authentication model. These are user authentication and application-only authentication. [44] With application-only authentication, the applications can reach Twitter server to obtain any information about it in. But with user authentication is allows to reach only user information such as reach own Twitter profile, sending direct message, displaying coming message, displaying friends list and follower list and etc. To making analysis on Twitter, application-only authentication is needed.

3.3 Working with Twittter Data

Twitter data is a repository about getting some answers from tweets written all around the world. When data is used correctly and effectively, Twitter data will carry a great value. To realize the analysis of Twitter data is needed to searching, storing, analysing and visualizing steps. Tweets are needed to be downloaded for an efficient data analysis. So that it uses Twitter API to collect tweets from Twitter server database. Before searching data, we can specify to what we want to work on as follows; a user's profile information; a user's followers or friends, a user's tweets, search results of Twitter database, location information about a tweet, place information of a tweet.

It is better to store crawled data for efficient analysis. Non-relational database systems can be used for storing crawled data. Relational databases are inadequate for indexing a large number of documents, intensive data applications such as web pages with has more traffic and streaming data. On the other hand, non-relational database system is more successful and faster in these matters.

Analysing Twitter data focuses on two key aspects of Twitter data: network analysis and text analysis. Network analysis is based on actions between users about following, followers and retweets. It tries to find answers out of user network on Twitter and network analysis is important for individual analyses.

Text analysis is based on tweets which users posted on Twitter. Text analysis is tries to find tweet's topics, any special words, owners and obtain any other information.

Visualizing Twitter data is done to gain insight into how and why users interact and visualization is important for displaying the results. Visualization focuses on visualizing network, temporal, geo spatial and textual information. Network information focuses on retweets which helps identifying information propagation paths. Temporal information is time sensitive information about variable like volume of tweets and interaction between users. Geospatial visualization helps that tracking place of events which occurs and likely to be occurred. Textual information includes word clouds which helps highlighting important information. For successful Twitter data analysis that searching, storing, analysing and visualization steps are needed. After from the first three steps, Twitter data was downloaded, stored and analysed so obtained significant data. Actually at this point, the purpose of analysis has completed. On visualization step, obtained data that coming from Twitter can be display with some visualization tools.

4. PROPOSED SYSTEM

4.1 Overview

In order to satisfy the needs, needs to develop a smart crawler. The typical architecture of a smart crawler based on data is given following figure below.
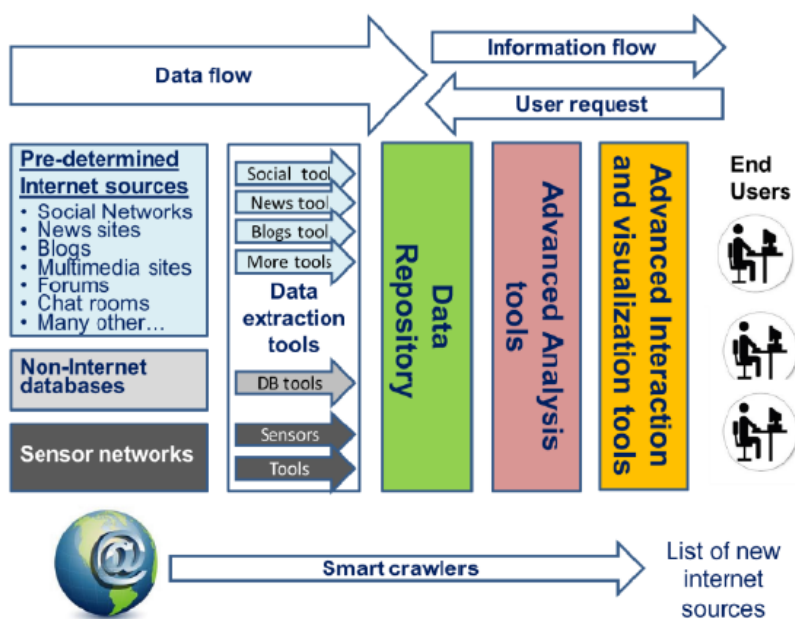
Figure 4.1 : The typical architecture of a smart crawler based on big data.[53]

As seen in figure 4.1, the first step of developing a smart crawler is the actual crawling step. The information can be gathered from pre-determined internet sources like social networks, news sites, blogs, non-internet databases, and sensor networks. In this case, Twitter which is a social network platform is selected as source.

The next step, gathered raw data from internet must be extracted and formatted with social, news, blogs, DB tools and etc. At this point, Twitter as source and Twitter APIs as extraction tools.

The other step is forming of data repository. Crawled and formatted data needs to be stored for next processing steps. When the app run in each time, it will give different results to different requests of the user. As the final step in the analysing phase, the gathered data, according to required user measures and parameters, is analysed and

17

presented with visualization tools for non-technical end-users. Various types of data are presented to the end-user through various stages.

4.2 Open Source Visualization Tools

MPAndroidChart is a project which was created at 2014 by PhilJay. MPAndroidChart is one of widely used chart library for Android with Java that makes it easy for developers to display professional quality charts in applications. MPAndroidChart library works on Android API 8 and above, but for using animation that it work on API 11 and above. MPAndroidChart is under the Apache 2.0 licence and it can draw various types of charts. [45] These chart types are line chart, combined chart, bar chart, pie chart, scatter chart, candle stick chart, bubble chart and radar chart. Also an iOS version of this library is available.

4.3 Proposed System Architecture

As mentioned in previous section, the system has four basic processes which are storing, searching, analysing and visualizing. Application searches Twitter cloud and results are downloaded from Twitter cloud. After that data goes into storing process to be stored in application's data structures. In storing process, data may be applied to some filtering operations to eliminate spams, advertisements and other unrequired and unrelated data. Due to time limitations of Twitter API to reach its own database, crawling and storing operations are batch processing operations performed in background at regular intervals. The system architecture is shown at Figure 4.2 on below.
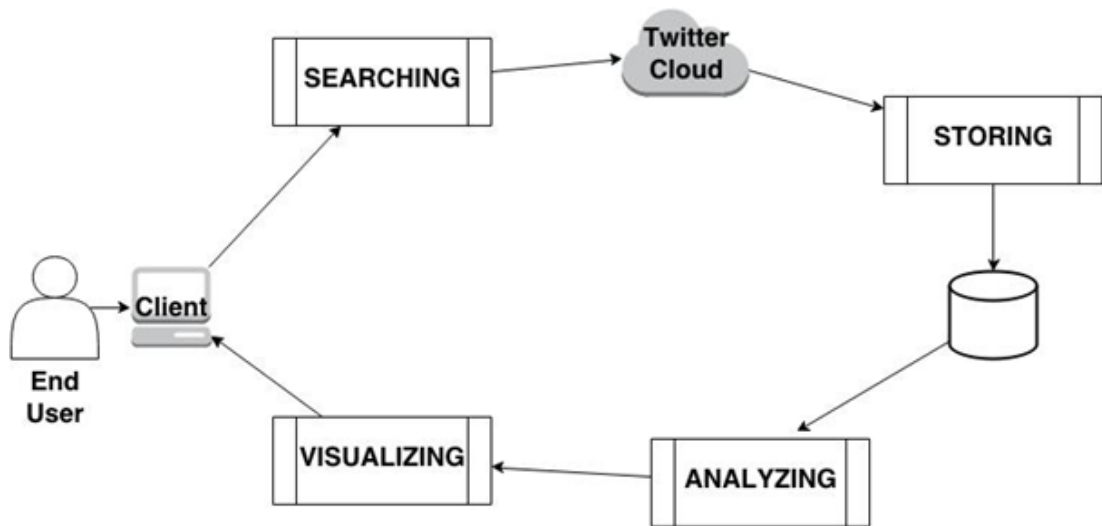
Figure 4.2 : The proposed system architecture.

An end user starts the software connects to the system requesting analysis of tweets about a specific subject or company which the user wants to learn what other people think about it. If software has necessary data in inside, user sends own request for analysis, if not, user may starts auto search to store data to be analysed later with given parameters like language, region, date and keyword. When an analysis request is given, system takes this request, runs a query applying user parameters and restrictions and returns textual based result with a chart to support results as visualization if requested.

The developed application is a mobile based android application as crawler that has analysis and visualisation operations and then it uses the native API of Twitter. The mobile application is giving services to a wide range of people, organizations or companies which need in public opinion pool about any subjects. The planned principal purpose is testing by running application it on subjects in our country like thoughts about universities in general.

4.4 Required APIs

There are so many libraries to access the Twitter with different platforms. However, android applications must be implement in Java language. Twitter4J [33] is an unofficial open source Java library for Twitter API which is developed by Yusuke Yamamato in 2007. Twitter4J can easily integrate any application with the Twitter service. Twitter4J is a an open-sourced that is released under the Apache Licence

2.0. Twitter4J has more features such as, completely runs on Java platform version 5 or later, zero dependency, no additional jar required, built in OAuth support. Twitter4J hides low-level details like setting-up a connection with twitter via http from developer. It also includes software to parse JSON response which is returned by Twitter API, meaning that developer do not need to learn how to parse and deal with JSON which makes it developer-friendly. These simple features make it more easily to integrate Java applications with Twitter service.

Twitter4J library has some packages of inside own it. These packages are twitter4j, twitter4j.api, twitter4j.api, twitter4j.auth, twitter4j.media and twitter4j.util. Twitter4J package is a general package for Twitter4J. Twitter4j package has 51 interfaces, 36 class, 6 enum types and 1 exception. These interfaces and classes are frequently used in the mobile Android application. The package of twitter4j.auth is include OAuth related classes. Twitter4j.conf package is an important package that has include twitter4j configurations. It has 2 interfaces and 5 classes inside of own it. Twitter4j offers a content-rich library with integrating Twitter services.

4.5 Android Twitter Integration using OAuth

Twitter4J is an unofficial Java library for Twitter API. Java Android applications can easily integrate with the Twitter service. Firstly, twitter4j libraries are need to be downloaded from the official web site of Twitter4J. After downloaded the required libraries that needs creating new developer account on Twitter developer console. Twitter developer console as shown at Figure 4.3.
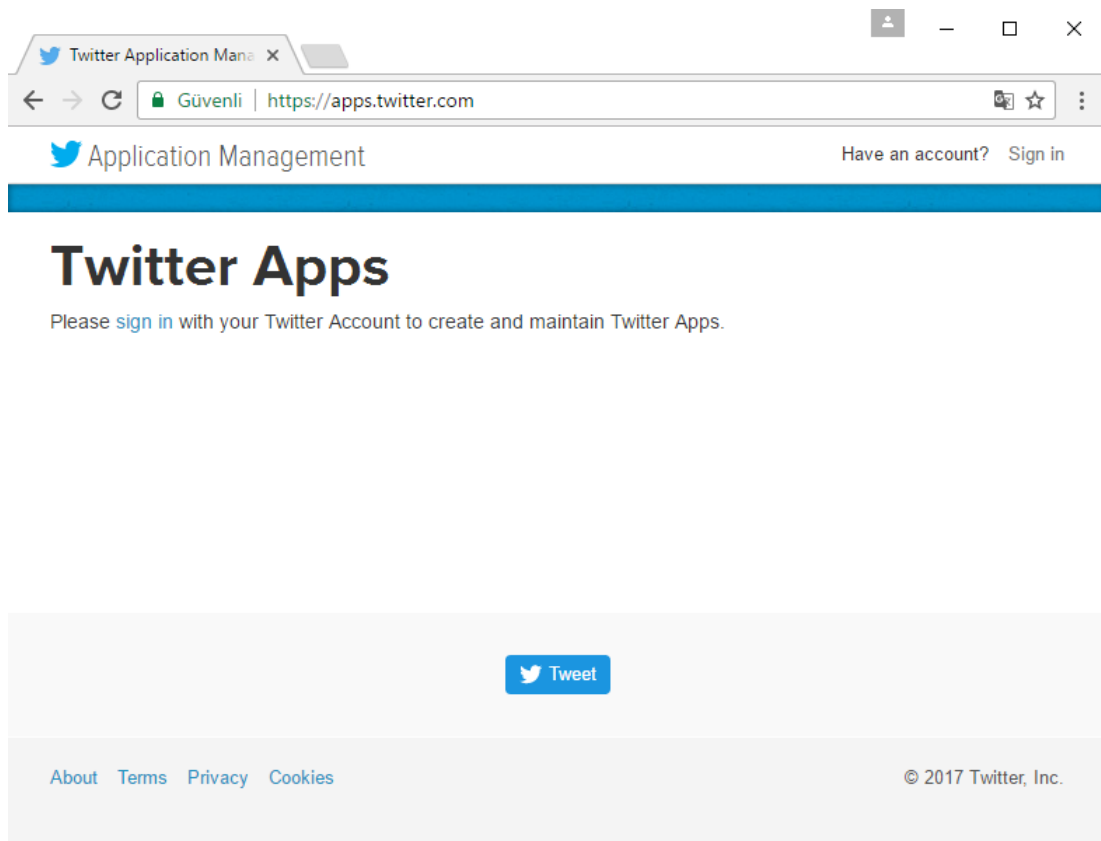
Figure 4.3 : Twitter Developer Console.

After login operations, the android application must register while creating a new application on Twitter console and provide necessary information to create application id. Creating application page shown as at Figure 4.4. In this page, the areas to filled that are name, description, website and Callback URL areas. Name is application name and this area is use to attribute the source of a tweet and in userfacing authorization screens. It has 32 characters max. Description field will filled with android application description. Between 10 and 200 characters as a maximum. This field will also be shown on the authorization screens for Twitter's users. Website field is applications publicly accessible home page, where users can go download or find out more information about Android application. Callback URL field can leave blank. Just agree Twitter's terms and conditions, and then click the "Create your Twitter application" button. On this point application registration operations have completed successfully.
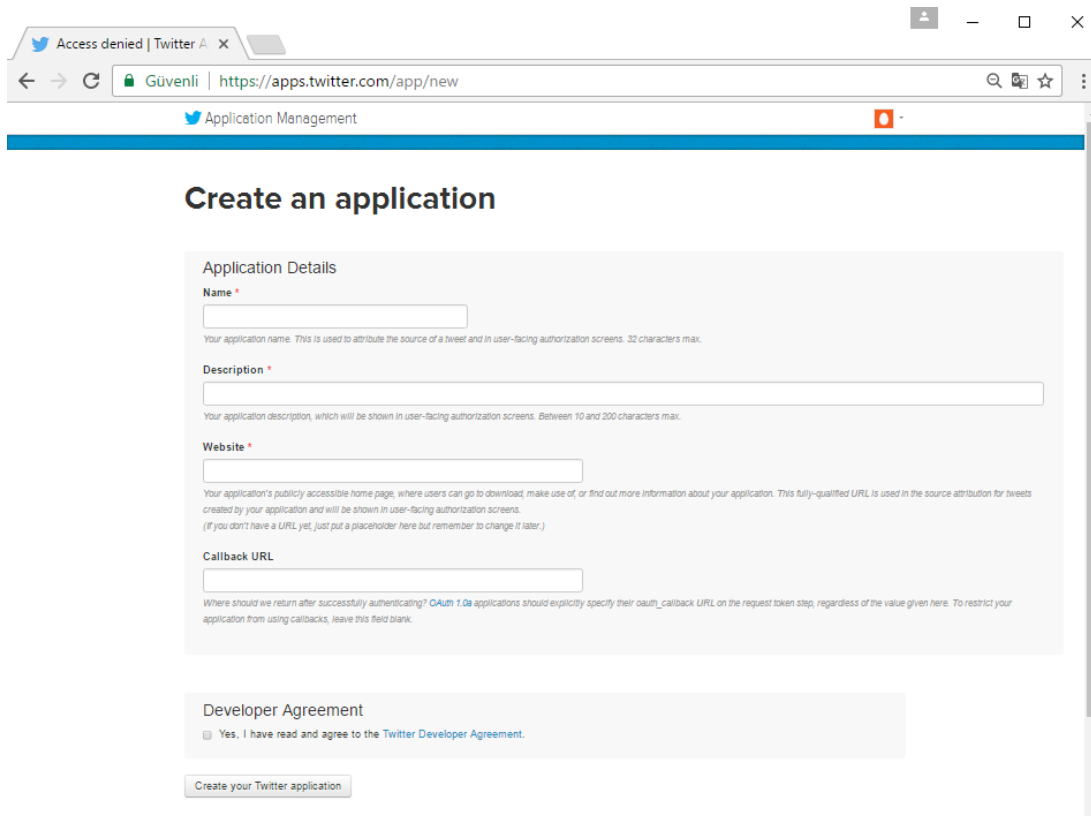
Figure 4.4 : The page of creating application on Twitter.

The next step is configuring the Android application. After creating Twitter application, the access level for application have to set. There are three options for setting level. These options are read only, read & write and read, write & Access direct messages and shown as in figure 4.5. The options of read only that is provides only read data and don't post anything. The second one is provides read as well as post the data. The read, write & access direct messages option is allowing an application to read or delete a user's direct messages.
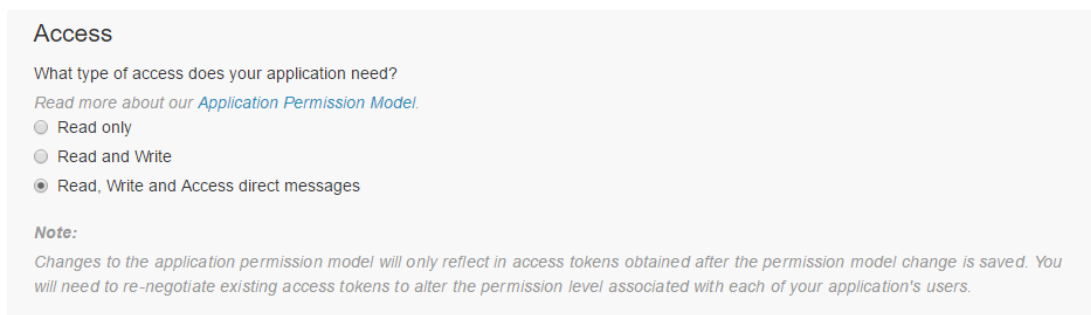
Figure 4.5 : Access permission page on Twitter.

There are two type of authentication and these are authentication and authorization on Twitter services. Authentication which uses the consumer key and consumer

secret to identify the client and be sure that it is a valid account. The authorization that is allows the resource server to identify which kind of actions have the permission to do with data or call a resource. This operation uses consumer key, consumer secret, access token and access token secret keys.

The final step is creating access token for new Twitter application. Under the "Details" tab, simply scroll down to the bottom of the page and click "Create my access token" button. If all steps are completed correctly, the application's access token should have the access level of read, write, and direct messages.

Figure 4.6 : Application settings for new Twitter Application.

Under "Keys and Access Token" tab, there are application settings and access token sections. Figure 4.6 illustates the application settings for the new Twitter application. Application settings section has consumer key, consumer secret key, access level, owner and owner id. Consumer secret key is a secret that should never be humanreadable in applications.

Access token section has access token and access token, access level, owner and owner id information. Access token can be used to make API request on own account's behalf. Access token secret is a secret that don't be shared with anyone. Application settings and access token sections are shown as in figure 4.7.

Figure 4.7 : Consumer Keys and Access Tokens of Twitter Application.

All operations are completed and the android application is registered on Twitter. After that actions, implementing will be started. Generated tokens and keys will used with configuring Twitter4J library while implementing the mobile android application.

Before creating the application is needed to register the application on Twitter developer console. Twitter developer console provides some keys and tokens. These tokens are required working with Twitter APIs. The application never interacts with Twitter without keys and tokens. The same process applies to all Twitter APIs. In thesis work, the Android mobile application was registered to Twitter. And then access token, access token secret, consumer key and consumer secret key are taken from Twitter developer console.

5. SYSTEM ARCHITECTURE AND IMPLEMENTATION

5.1 Twitter4J Configuration

There are a few features that can be used to configure Twitter4J. To configuring the Twitter4J is required with consumer key, consumer secret key, access token and access token secret. The first way is to creating a file in the application file system and putting all required parameters that will be used for configuration within this file. This operation is shown at in figure 5.1.

```
debug=true
oauth.consumerKey=*********************
oauth.consumerSecret=******************************************
oauth.accessToken=***************************************************
oauth.accessTokenSecret=******************************************
```

Figure 5.1 : First Way for Configuring Twitter4J.

The second way is using ConfigurationBuilder class to configure Twitter4j with programmatically. ConfigurationBuilder class is under Twitter4J's conf package. This feature is shown at Figure 5.2.

```
ConfigurationBuilder cb = new ConfigurationBuilder();
cb.setDebugEnabled(true)
  .setOAuthConsumerKey("*********************")
  .setOAuthConsumerSecret("******************************************")
  .setOAuthAccessToken("*************************************************")
  .setOAuthAccessTokenSecret("******************************************");
TwitterFactory tf = new TwitterFactory(cb.build());
Twitter twitter = tf.getInstance();
```

Figure 5.2 : Configuring Twitter4J with programmatically.

The third way is configuring Twitter4J via system properties. This feature is shown at Figure 5.3.

```
$ java -Dtwitter4j.debug=true
    -Dtwitter4j.oauth.consumerKey=*********************
    -Dtwitter4j.oauth.consumerSecret=******************************************
    -Dtwitter4j.oauth.accessToken=*************************************************
    -Dtwitter4j.oauth.accessTokenSecret=******************************************
    -cp twitter4j-core-4.0.4.jar:yourApp.jar yourpackage.Main
```

**Figure 5.3 :** Configuring Twitter4J via system properties.

In thesis work, ConfigurationBuilder class is used to configure Twitter4J. The values of consumer key, consumer secret, access token and access token secret are passing

with the methods of setOUauthConsumerKey, setOAuthConsumerSecret, setOAuthAccessToken and setOAuthAccessToken through ConfigurationBuilder object. Twitter API currently consists of two distinct APIs and both of them are based on HTTP protocol. GET requests are used to retrieve data from Twitter and POST requests are used to submit, and change.Twitter has fully integrated the OAuth authentication. OAuth is token passing mechanism that allows users to control which application has access their data. Twitter have PIN-based authorization is differently from the others. The PIN-based authorization method that uses the following flow for clients as shown at Figure 5.4.

1: The application obtains a request token from Twitter

2: The application directs the user to authorize on twitter.com

3: After obtaining approval from the user, a prompt on twitter.com will display a7 digit PIN.

4: The user is instructed to copy this PIN and return to the application

5: The application will prompt the user to enter the PIN.

6: The application uses the PIN to exchange a request token for an access token.

7: Twitter will return an access token for the application to generate subsequent OAuth signatures.

**Figure 5.4 :** PIN-Based authorization with Twitter.

In the web applications, users that can direct to Twitter to complete the authorization flow. But the applications without web, users make copy and paste the printed URL into a browser and then after user that users received PIN-code from developer Twitter website shown as at Figure 5.5, application will be send with Token class object for an argument.
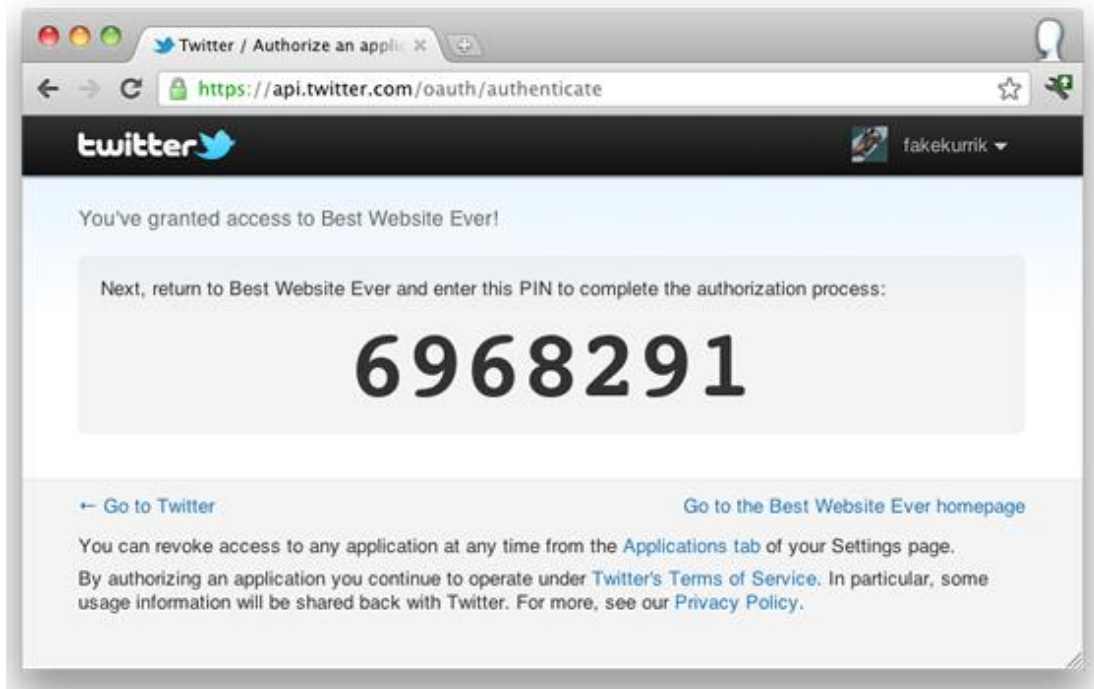
Figure 5.5 : PIN-Code from Twitter.

5.2 Data Collecting

The following architecture is Figure 5.6 shows the convergence of different advanced features such as collecting, storing and analysing tweets and managing the user interface. Beyond combining complementary functions, this generic architecture also allows to highlight these related systemic features. For example, tweets that are not explicitly requested but have a chronological or semantic connection with the initial request may be aggregated to have a degree of platform autonomy. Such functions are only possible during the long time period and when collecting tweets that are moving away from the traditional timed strategy.
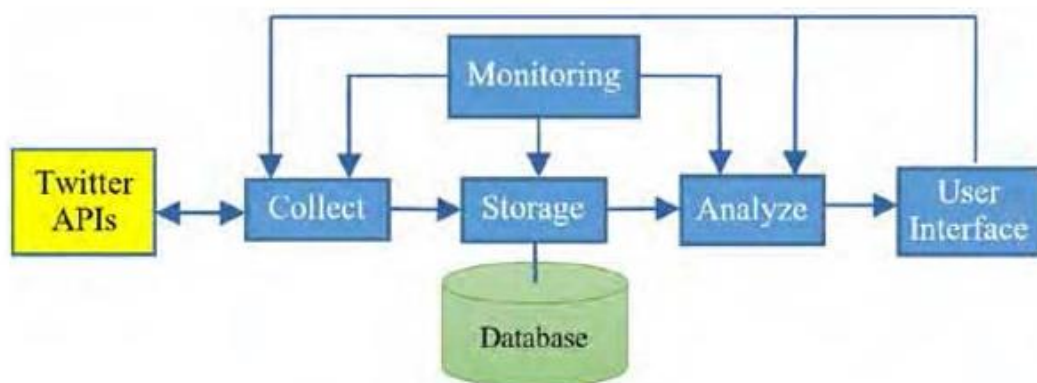

Figure 5.6 : General architecture of Twitter data analysis. [46]

The architecture and its performance must be oversized. Besides this, the operation should be monitored to quickly identify and correct problems that prevent the collection of data.

The data is provided via the Twitter API [47] and also to collect real time data that used with Twitter Streaming API. After getting authentication, Twitter4J library used in the android application for sending queries and collecting data.

The design of the data acquisition application module distinguishes application interface logic from background processes for data collection and storage. It uses the Twitter4J library for connecting to the Streaming API and stores received tweets within a type of data structure or database. Twitter4j connects Twitter Streaming API and application with each other like as a bridge. Data connection requests are coming from the application interface to main application. And then application stores the received data is coming from the Twitter Streaming API via Twitter4j. The mobile application interface allows users to configure data collection processes and track its process. Figure 5.7 illustrates the architecture of this module.
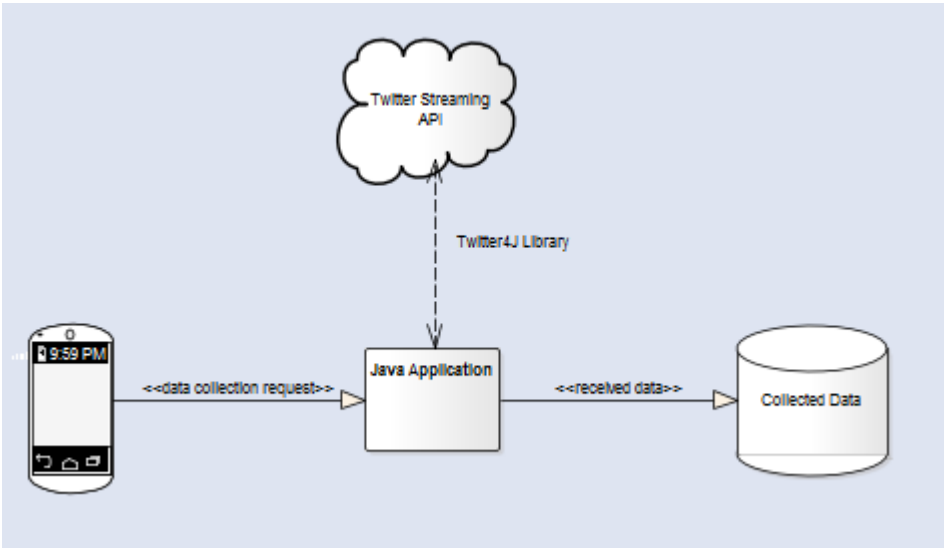

Figure 5.7 : Architecture of streaming data collection module.

After the data has been obtained and stored, the parameters of interest are the tweet text with given keyword, username and user profile names like as nick names. However, there are many of attributes that can be obtained, such as language preference, user's geometric coordinates, friend list, follower counts, profile description, and etc. Table 5.1 illustrates examples of tweets that posted on Twitter with about "Katip Çelebi" keyword.

Table 5.1 : Example of Obtained Tweet with about "Katip Çelebi" Keyword.

Tweet Text	Tweet User Name	Tweet User Profil Name
Bi ameliyat olup gelcem 😈 (@ İzmir Kâtip Çelebi Üniversitesi Atatürk Eğitim ve Araştırma Hastanesi)	Beyzanur Oğlak	@OglakBeyzanur
Ilim Çin'de de olsa gidiniz diye artistlik yapmama gerek yokmuş. Katip Çelebi Üniversitesi'ni, Izmir'e göre Çin'e inşa etmişler zaten. 😒	Gökçe Ergençiçeği	@_melonik_
İzmir Kâtip Çelebi Üniversitesi Yönetimi, bünyesine hainlerin sızmasına ve içinde barınmasına asla izin vermeyecektir!	İzmir Kâtip Çelebi	@izmkatipcelebi
Nerde okuyorsun diye soranlara katip çelebi dedikten sonra devlet üniversitesi diye söylüyorum ki özel mi diye sormaya kalkmasınlar	Buse Şen	@busesen393

The mobile application captures an input parameter and this input parameter is character based keyword which is inserted by the user of application for colletting data. Character based keyword is used to extract tweets. The figure 5.8 describes the data collection process.
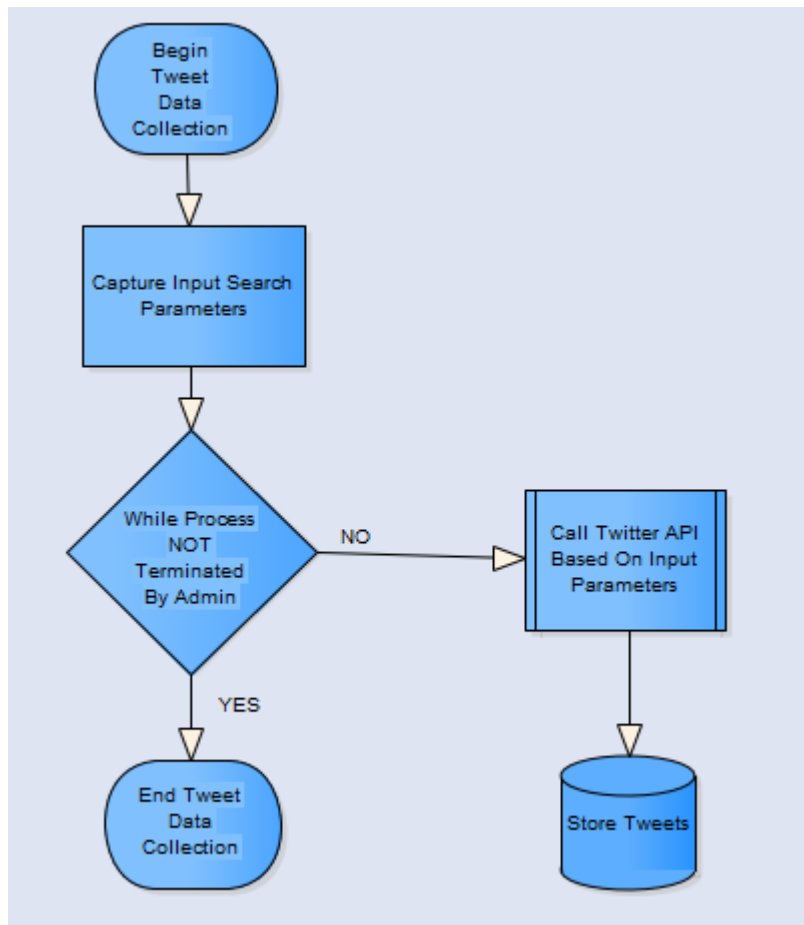
Figure 5.8 : Twitter data collection procedure.

5.3 Data Storing

The Twitter client based on Android System implements the following basic functions of Twitter:

- The application builds a search interface that collects data from Twitter by using Twitter API.

  o Information about a user. This information is user name, user profile name, user profile image, location information about user, geometric coordinates of users and mobile operating system when users used to post a tweet which is used.

  o Tweets published by a user.

  o Search results on Twitter.

- Crawled Twitter data will be stored.

- It then analyses this data and visualizes with a chart to be able to reach a result from posted tweets by users.

The data that is coming from Twitter needs to be stored in the Android Operating System. There are several different ways to store data in Android OS as shown at Table 5.2.

Table 5.2 : Different Storage Options in Android.[48]
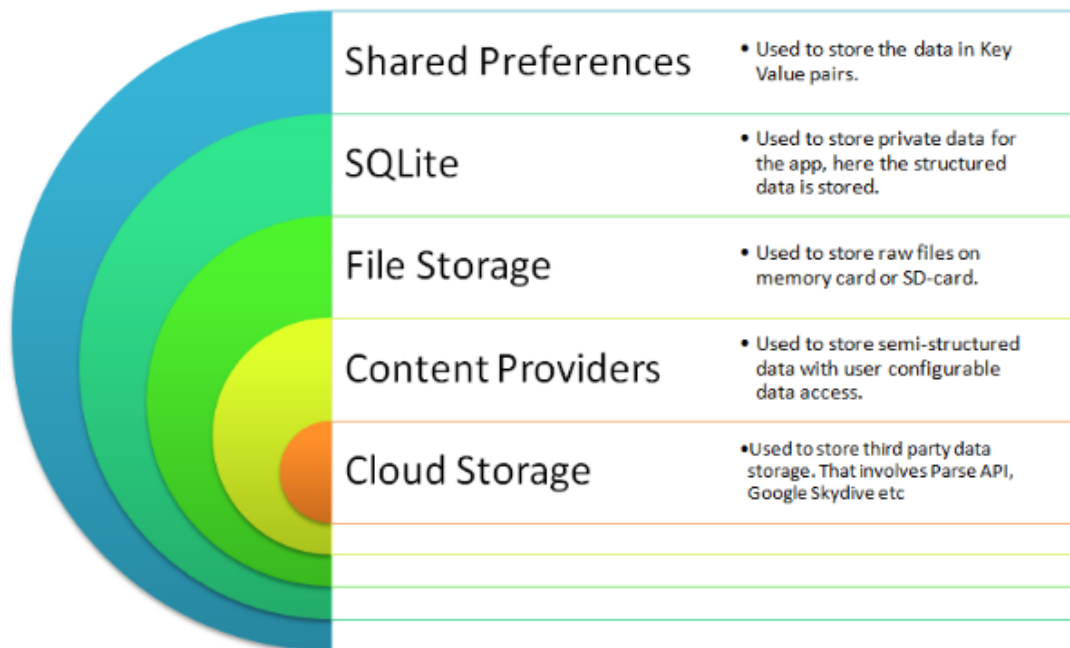

The first one is using SharedPreferences. SharedPreferences is a XML file that stores data as a combination of key value pairs and similar structure to the HashMap structure (key-value) that is used to store and retrieve the small sized data. The values can be any valid data types such as integer, double, boolean and etc. Figure 5.9 illustrates SharedPreferences usage.
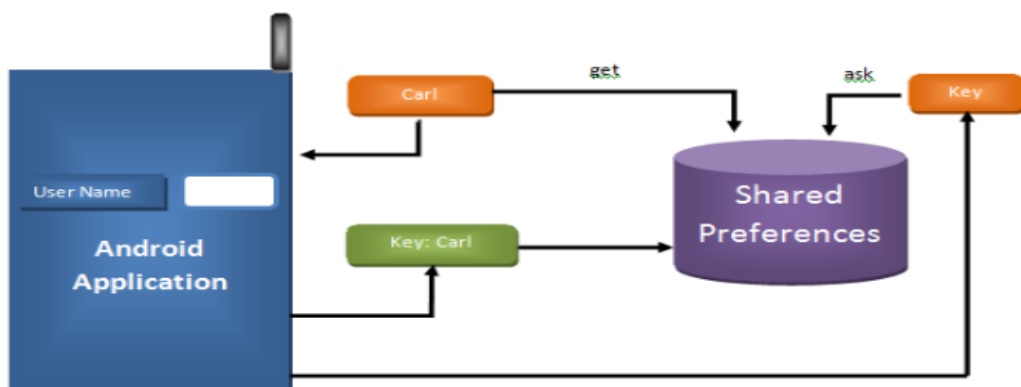

Figure 5.9 : Shared Preference usage.[48]

File storage is the other way of storing data. Firstly, the data is written in a file and then the file can be stored on mobile internal built-in memory or external memory such as SD-Card. The types of files can be text file, image file, user-defined files and etc. As seen in the Figure 5.10 as seen that there are two place to store data on it. The first way is Primary memory is the mobile devices' own memory. Another memory that is Secondary memory is the external memory space. SD-card are generally used for an external memory unit.
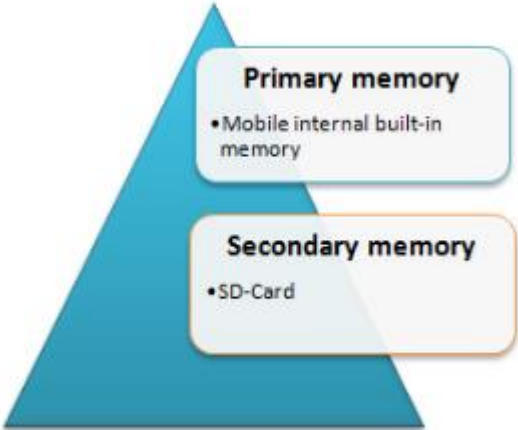
Figure 5.10 : File storage system.[48]

The other one is using database to store data as shown at Figure 5.11. The Android platform also uses SQLite as the database, [49] [50] just like any other mobile platforms. SQLite has methods to create, delete, insert, execute SQL commands, and fulfil other database management tasks. SQLite has similar functionality like MySQL. SQLite is very useful in storing complex and large data can be used repeatedly until the application is running.
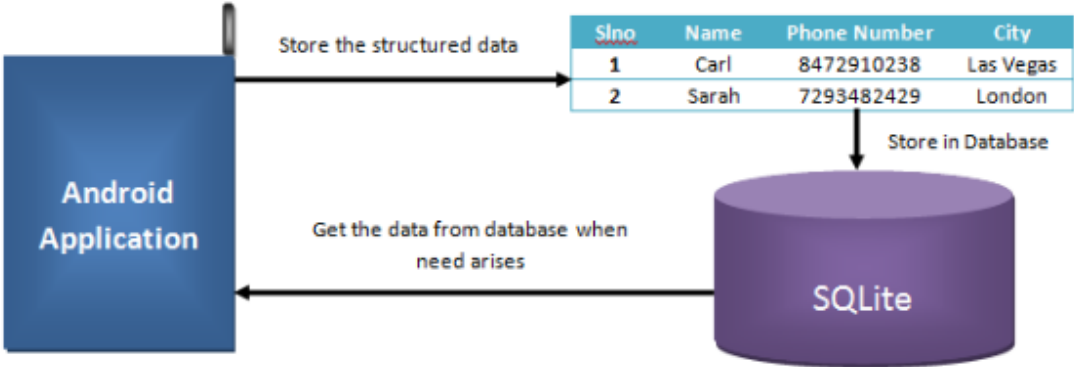
**Figure 5.11 :** Android Database System.[48]

Using Content Providers is another way to store data. Content Provides provide that data sharing between the other applications on the same mobile device. Cloud storage is used to storing data on cloud systems such as Google Drive and etc. [51]

5.4 System Models

The system is collecting data which coming from Twitter and analyse them with about the user requested topics. As a results of display some results with visualization tools. The system must:

- Allow users to collect Twitter data posted with given keyword.

- Allows users to collect real time Twitter data.

- Allow users to retrieve relevant tweets from the collected data using textual queries.

- Allow users to display user profile image.

- Allow users to search tweets and retrieve user information.

- Allow users to analyse tweets which posted on Twitter.

- Allows users to visualize the results of analysed tweets.

The use case diagram in figure 5.12 illustrates how users are expected to interact with the system.
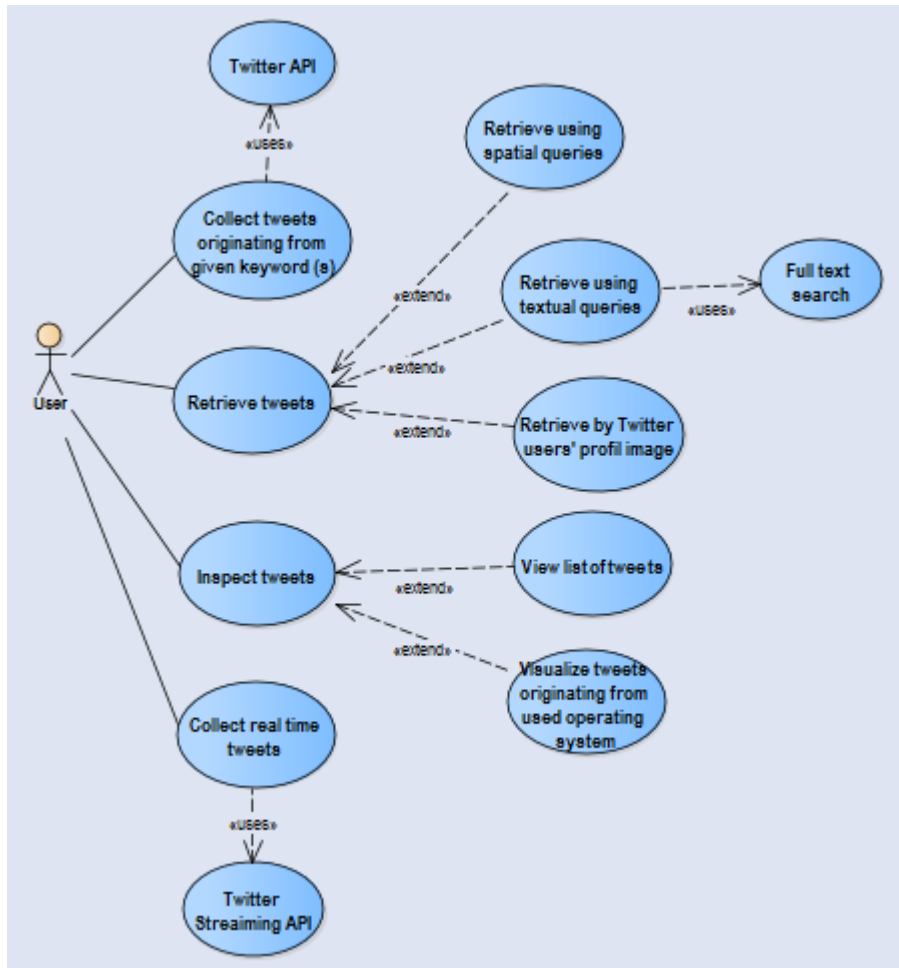
Figure 5.12 : Use Case Diagram showing the expected usage of the proposed system architecture.

The system has some functionalities. These functionalities are related with tweets that coming from Twitter. The first feature is getting tweet of a user. Used parameter that getting tweets of users is user id. After a tweet is obtained, the same operation is continuously repeated, and the captured data is displayed on the screen as listview. Figure 5.13 illustrates the getting tweets of a user.

```
Get the tweets of a user:

Input Parameters: user id.

Returns: list of tweets(id)

Initialize tweetsIDs list with the tweets of the user input id

ListTweets

Sleep()

Foreach tweet in tweetsIDs

Begin:

Tweets.add(tweet id)

End:

Return Tweets.

End
```

Figure 5.13 : The pseudocode of Getting Tweet of a User.

Required to avoid collecting same data for same tweet when getting tweets. The algorithm of avoid collecting same data for same tweet is shown at Figure 5.14.

```
Avoid collecting same data for same tweet:

Input Parameters: tweet id

Returns: Boolean, true if the tweet id exist, false otherwise

Begin:

List tweets;

Foreach user in users

Begin:

Tweets.add(user.tweetID);

End:

If (tweet id is in tweets)

Return true;

Else

Return false;

End
```

Figure 5.14 : The pseudocode of Avoid Collecting Same Data for Same Tweet.

Another situation to avoid is collecting same data for same user. To avoid unnecessary repetitions that used algorithm is shown at Figure 5.15.

```
Avoid collecting same data for same user:

Input Parameters: user id

Returns: Boolean, true if the user id exist, false otherwise

Begin:

List users;

Foreach user in users

Begin:

Tweets.add(user.userID);

End:

If (user id is in tweets)

Return true;

Else

Return false;

End
```

Figure 5.15 : The pseudocode of Avoid Collecting Same Data for Same User.

The another feature is getting the followers for a user. Used algorithm to getting followers for a user is shown at figure 5.16.

```
Get the follower for a user:

Input Parameters: user id

Returns: list of followers id(s)

Begin:

Initialize followerIds list with the followers of the input user id

List FOL;

Sleep();

For each follower in followerIds

Begin:

FOL.add(follower id);

End;

Return FOL;

End
```

Figure 5.16 : The pseudocode of Get the Followers for a User.

For this feature, Twitter provides users profile image URL for each user who post a tweet with given specified keyword. Firstly, since the URL operations are done, needed to work with Internet. To work with Internet, we need a separate thread otherwise we will get an error and this error type is NetworkOnMainThreadException. To handle this situation, AsyncTask could be used. Optionally can also use StrictMode.ThreadPolicy. StrictMode is most commonly used to catch network access on the main thread of application. When doing network operation on UI thread, [52] have to separate with AsyncTask from UI thread. So used with network operation within another thread from main thread. Same operation is done with displaying user's image. Figure 5.17 shows that how network operation is worked and displayed the images. Firstly, created Bitmap objects and assigned null for default. And then string URL variables send with URL constructor. In progress using URLConnection object, connection is provided. After InputStream object is send with BufferedInputStream constructor method. After stream setup, using the decodeStream method of BitMapFactory, Bitmap object is

loaded with BufferedImageStream object. As a conclusion Bitmap object is returned to displaying image on listview.
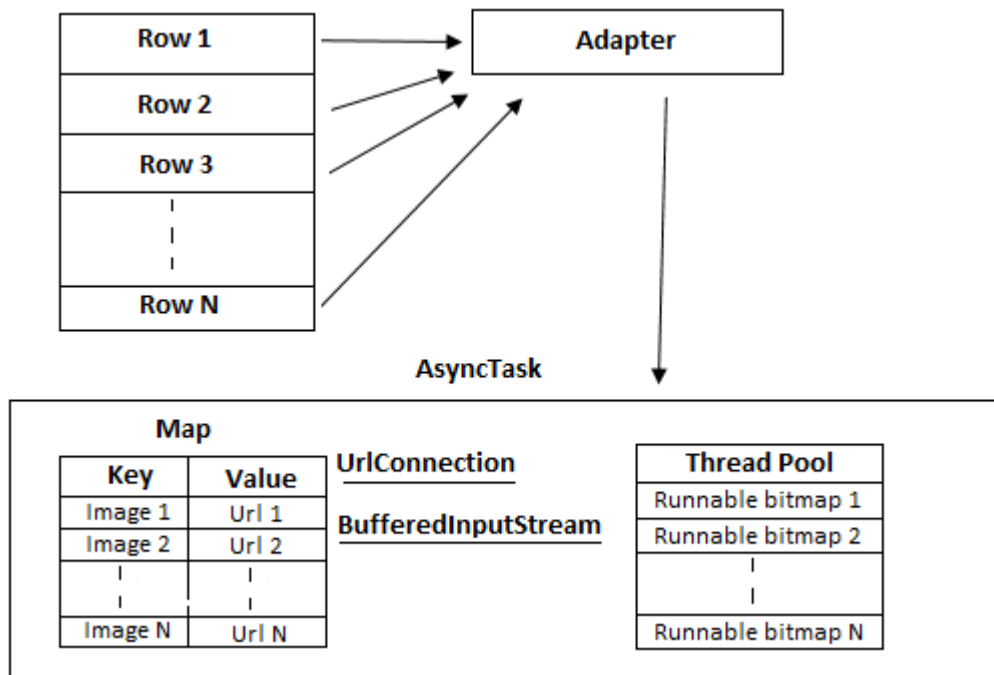
**Figure 5.17 :** Displaying user profile image.

5.5 User Interfaces

The mobile application of the analysis of Twitter data that developed on android platform and developed by using Java language. The application has 8 activities, 18 Java class and 16 xml files.   There are 8 choices in the main interface and shown as at Figure 5.18. These are Start Data Flow, Streaming Data, Location Analysis, User Analysis, User Information, Tweet Analysis, Settings and About.   The first six choices are used to displaying the results of the analysis.   The Settings choices provide to insert keyword that to make analysis on Twitter. After inserting keyword on the Settings menu, the other choices will be available to display results.

Figure 5.18 : Main interface of proposed system.

Main interface has 8 button and an option menu. Option menu provides to transition of between activities. When the Setting button is pressed, some filling fields will be seen. This action is shown at Figure 5.19.
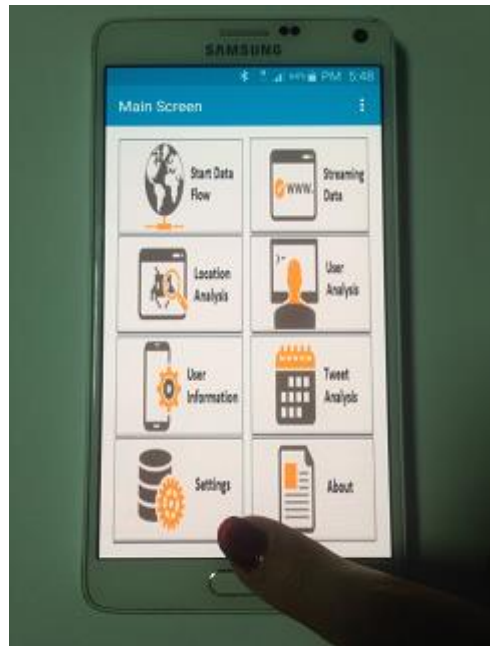
Figure 5.19 : Pressing settings button action.

There are five fields in the Settings activity. This fields are Consumer Key, Consumer Secret Key, Access Token, Access Token Secret and the last field of inserted keyword. The first four fields are required to authorization of the application on Twitter. But these fields are not enabled because these keys are implemented within the Java application code. The last field is used for inserting some word to display analyse result later. When the keywords are inserted and pressed the Confirm button, Return Main Screen button will be appeared. After pressed this button the application turns main screen. Figure 5.20 shows the inserting keyword action.
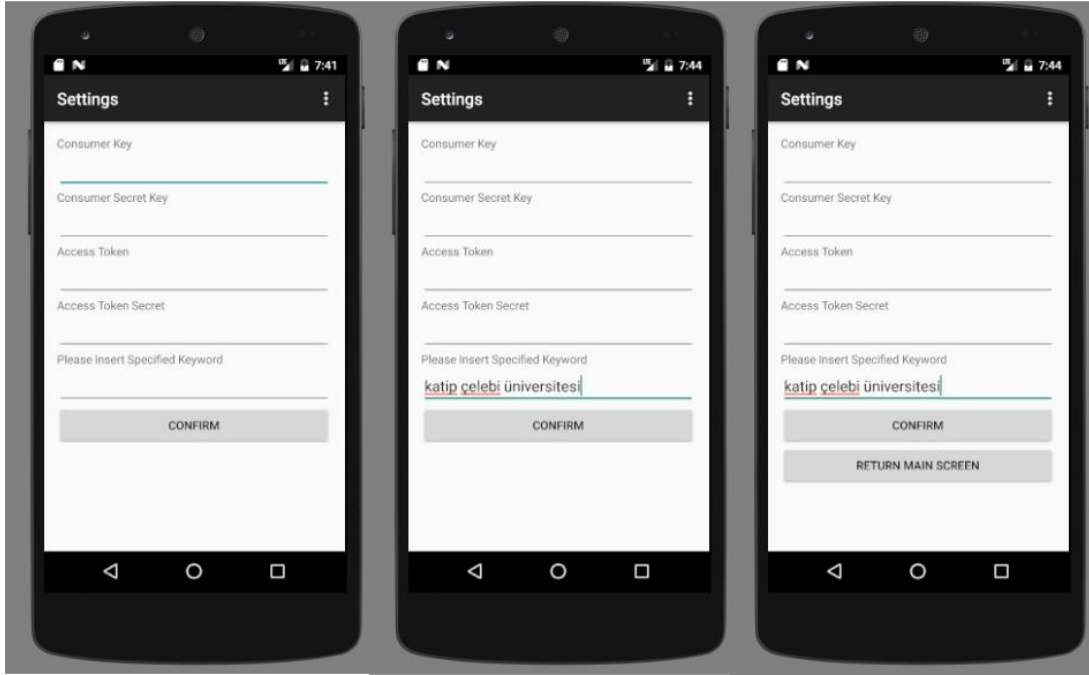
Figure 5.20 : Inserting keyword action.

When the Start Data Flow button is pressed from on main screen, some of analysis result will be displayed. These results are obtained data about keyword where inserted on the Settings screen. Figure 5.21 illustrates Start Data Flow activity page. The users can display the who posted tweet on Twitter about that inserted key and then tweet text the containing inserted keyword.
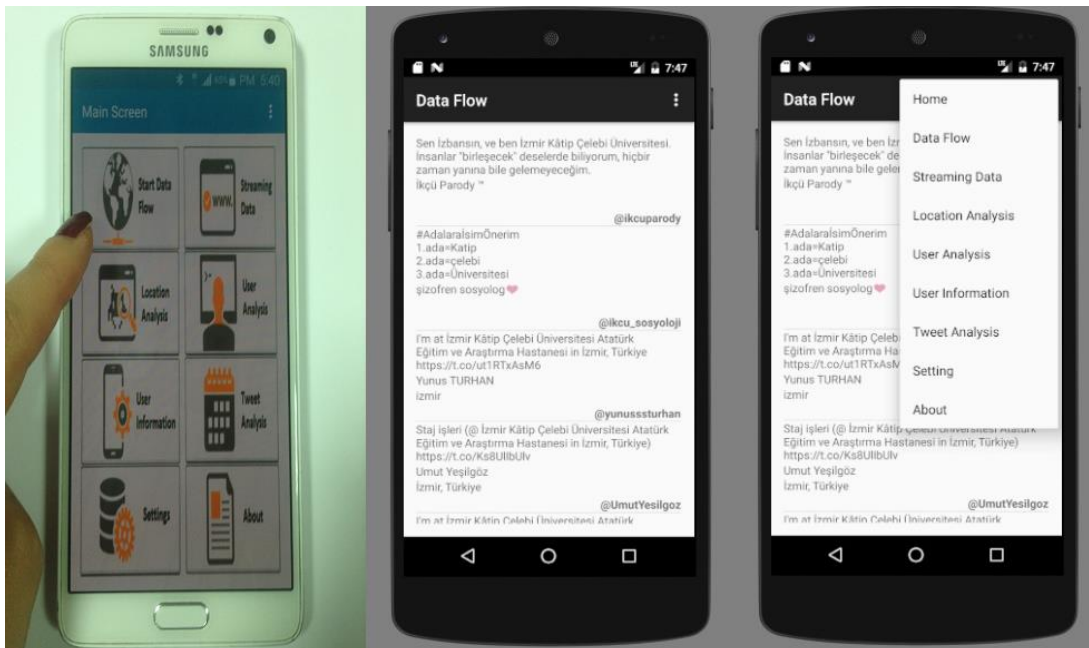

Figure 5.21 : Start Data Flow activity.

The Streaming Data button is displays real time data that not related with inserted keyword. This activity displays all real time tweet the all on the world. After activity is opened, there is start and stop button on top of the page. Start button is enabled and stop button is disabled. However, after start button is pressed, the real time tweet flow will be started and start button will be disabled and stop button will be enabled. While tweets are displayed, another activity can be passed with using options menu. The options menu is located in the upper right corner of the page. Figure 5.22 illustrates streaming data flow.
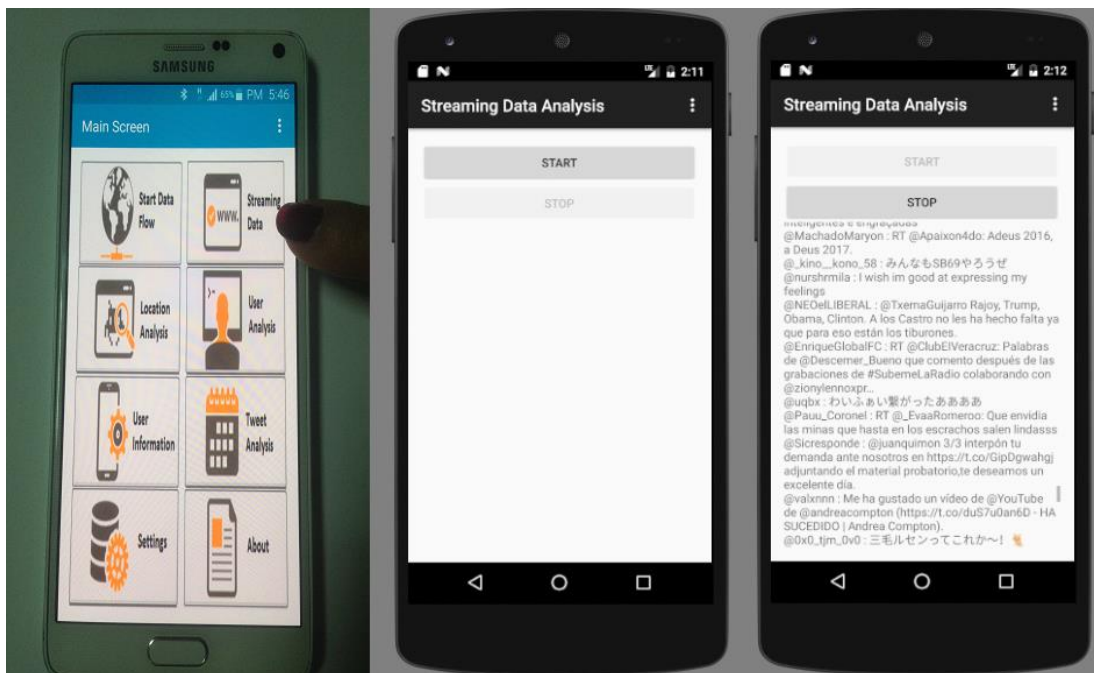

Figure 5.22 : Streaming Data activity.

The Location Analysis button provides to display users' names, profile names and their places names who posted tweets related with inserted keyword on Twitter. Figure 5.23 illustrates location analysis activity.
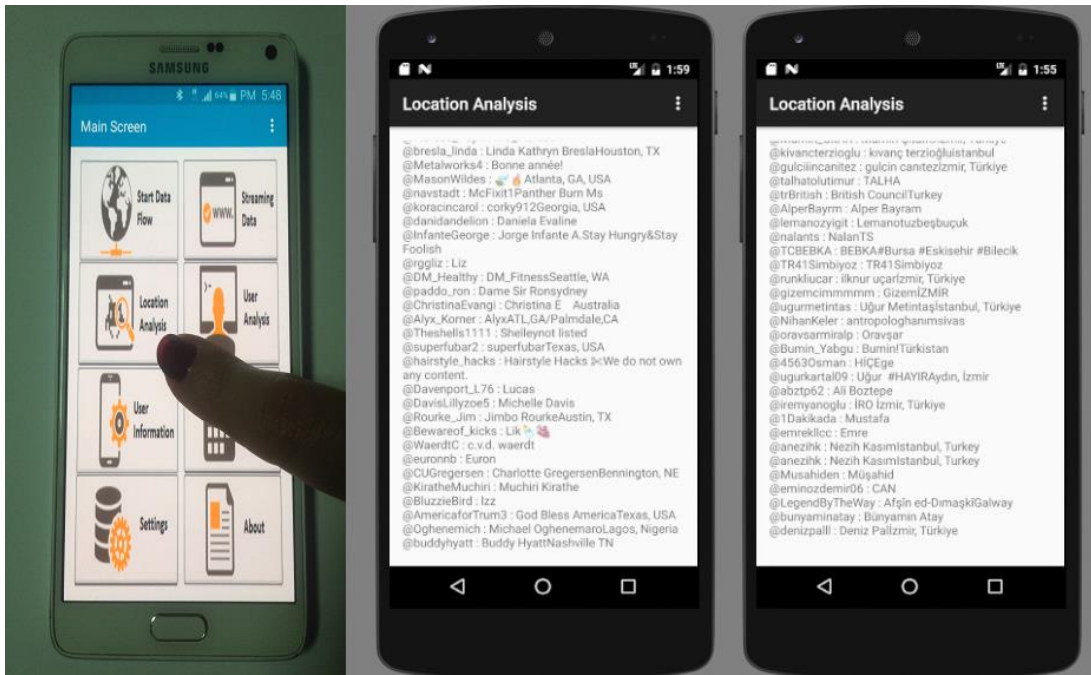
Figure 5.23 : Location Analysis activity.

The User Analysis button provides to displaying visualized results as shown at Figure 5.24. These results show the percentage of the users' operating system which they used. These operating systems are grouped into 3 categories. These are Android Operating System, IOS Operating System and desktop operating systems.
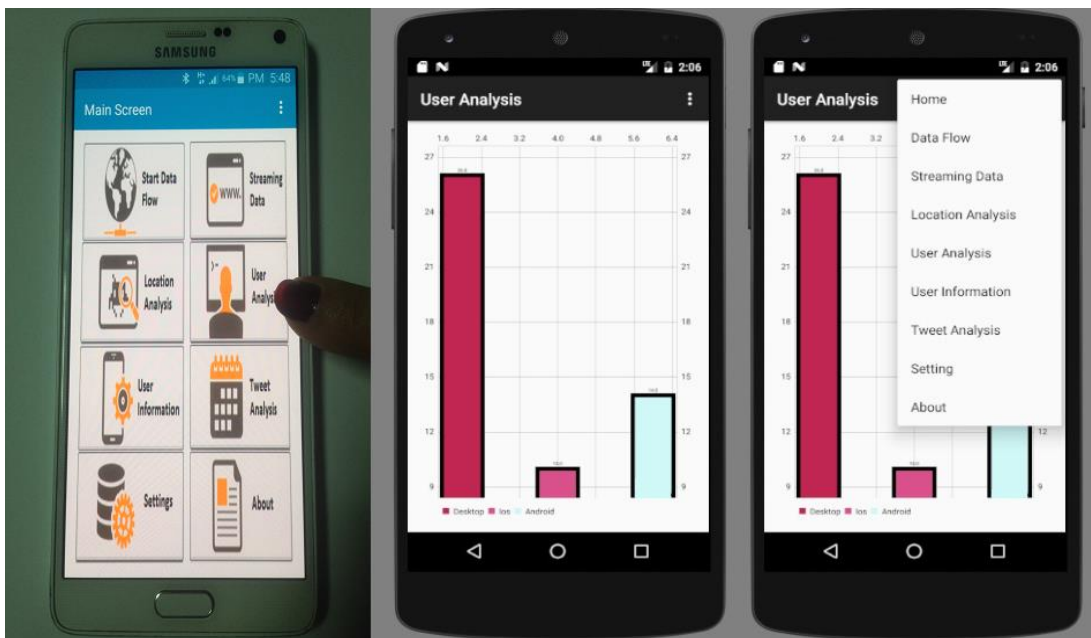
Figure 5.24 : User Analysis activity.

The User Information button provides to display users' profile image and their profile names. These displayed users that posted tweet or tweets with related inserted keyword. This activity is shown at Figure 5.25.
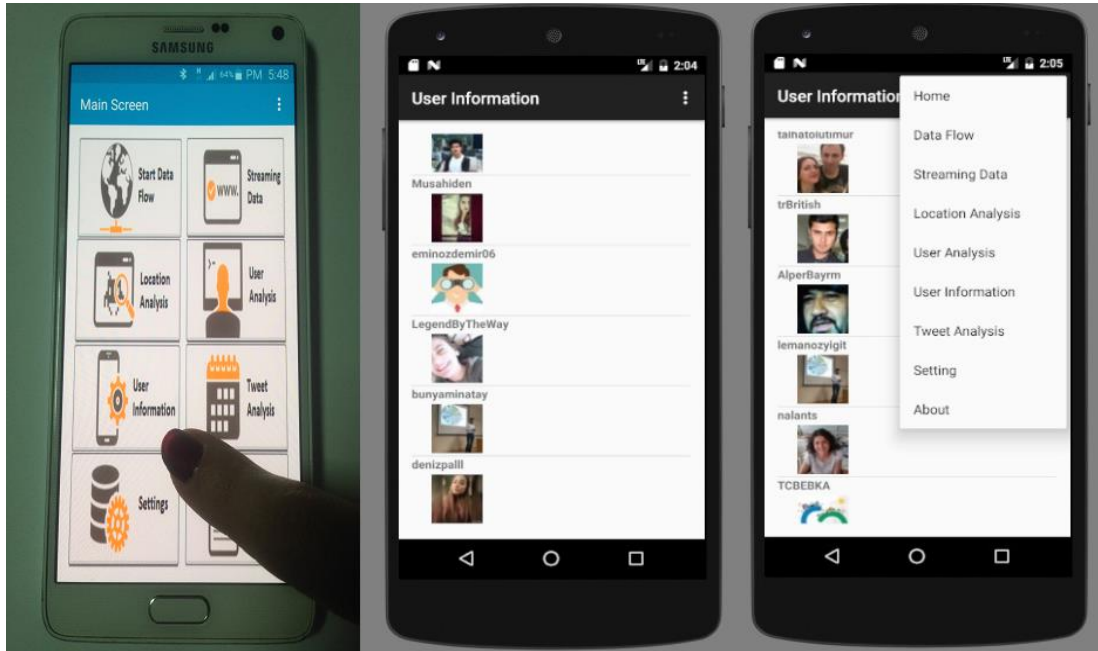

Figure 5.25 : User Information activity.

The Tweet Analysis activity shows that the follower counts number of user, retweet count number and friends counts number of users and tweet text. Figure 5.26 illustrates the Tweet Analysis activity.
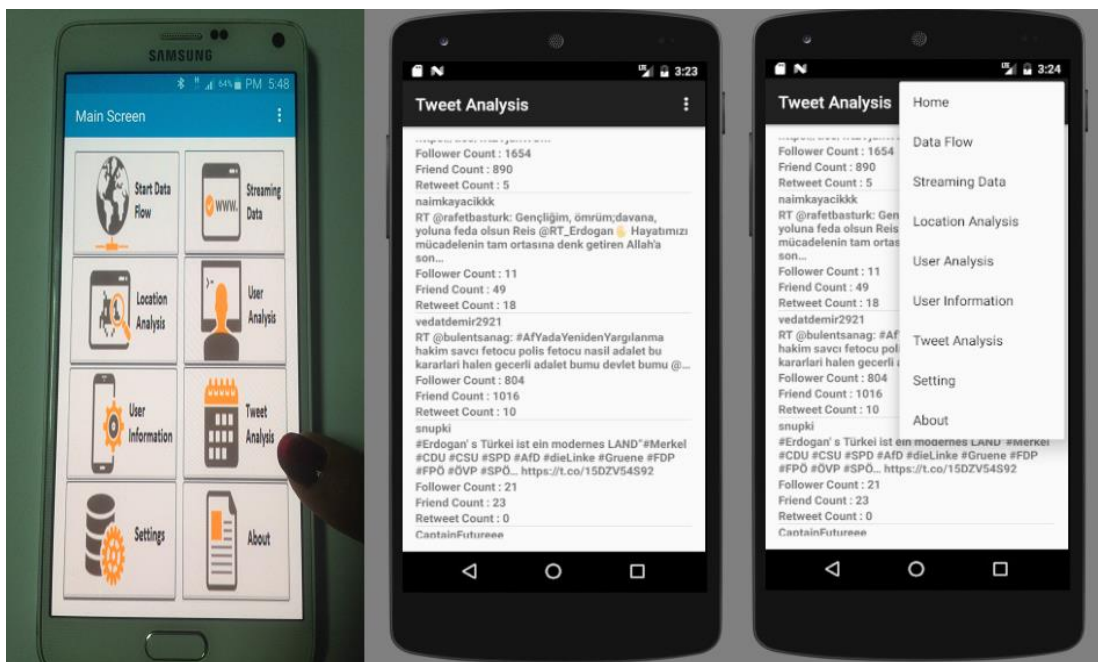

Figure 5.26 : Tweet Analysis activity.

The last activity is About activity. About activity is giving some information about the application and thesis and also gives information about the use of application. Figure 5.27 is shows About activity.
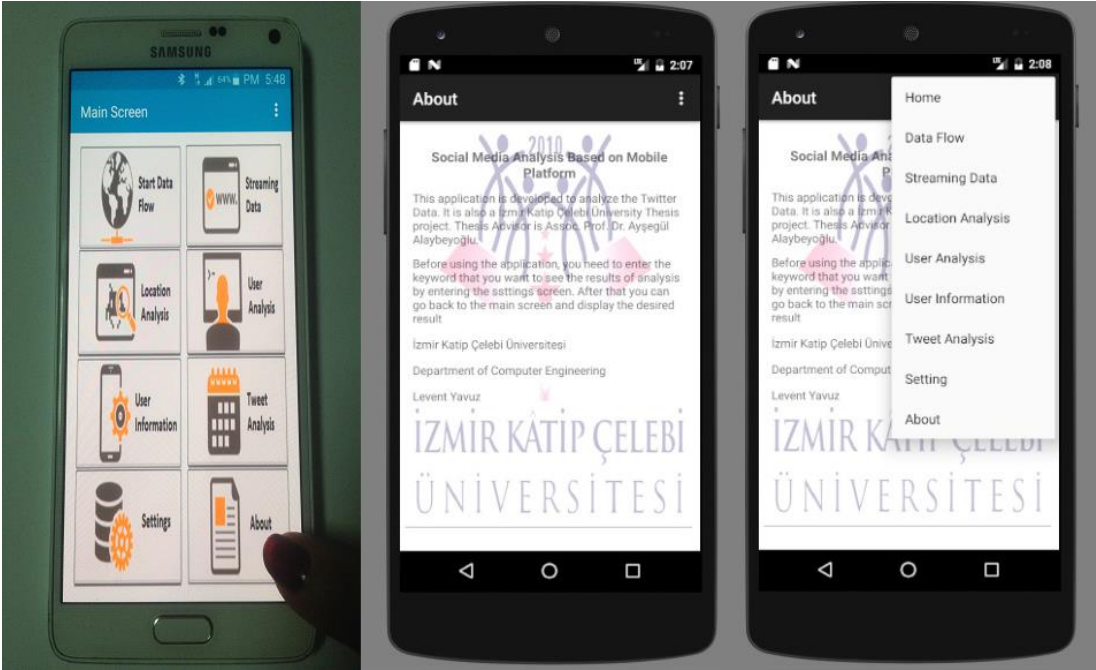

Figure 5.27 : About activity.

6. RESULTS AND DISCUSSIONS

The obtained tweet data have important meanings. It is important to take meaningful results as far as collecting data. Some received results have very different implications. All the results shown in below are related to the results of tweets that containing "Katip Çelebi Üniversitesi" keyword. Katip Celebi University is an university in Turkey. The following figure 6.1 shows the types of mobile operating systems used by users who posted the tweet about Katip Celebi University. Generally, Android and iOS operating systems are used on smart phones and tablets, while desktop operating systems are used on computers and laptops.
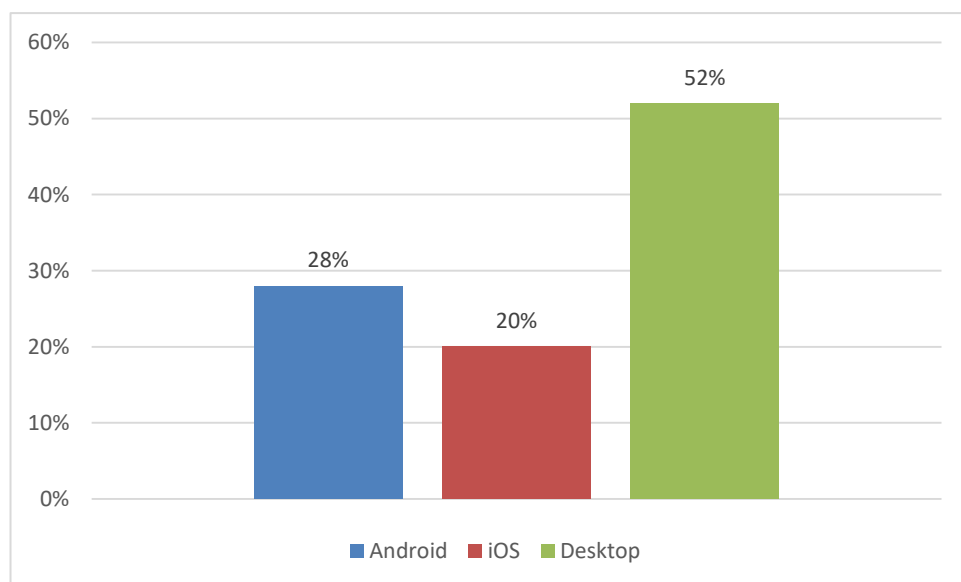
**Figure 6.1 :** Usage percentages of operating systems used by users.

52 percent of tweet users who posted tweets about the keyword of "Katip Çelebi University" use the desktop operating system. 20 percent of the tweet users use the iOS operating system and 28 percent of the users use Android operating system. These results show that the rates of mobile device usage and desktop devices are almost the same while users posting tweets on Twitter. Also shows that the use of android based devices is a bit more preferred than iOS based devices. However, the application in hand supports the mobility, namely it can be conveniently posted tweet while moving with mobile devices, for example on the bus, during a lecture and etc.

Since the desktop devices usually require a constant internet connection, they will be in a fixed position. Approximately, half the users post tweets while on the move while the other from a fixed position. Figure 6.2 shows the gender ratios of users who posted tweets. According to obtained graph, the number of female users seems a little more than that of the male users. There is no big difference between them.
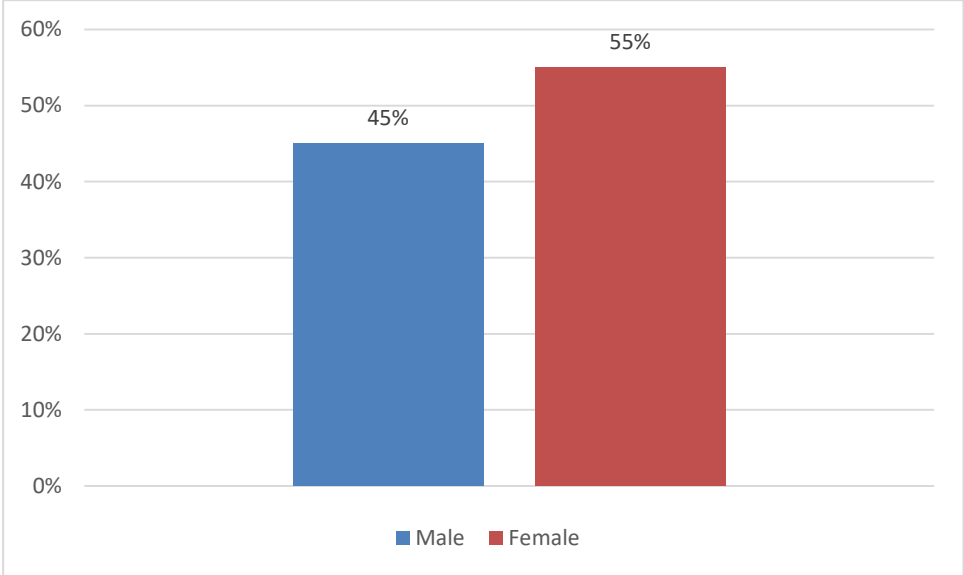
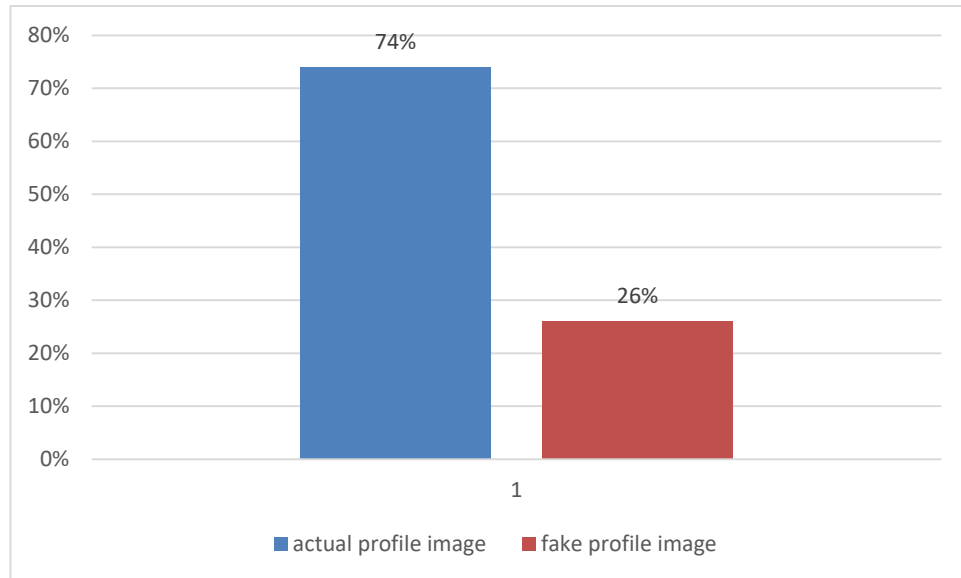**Figure 6.2 :** Gender percentages of users.

Another important result is the percentage of users who do not use their real names on Twitter. Figure 6.3 shows the percentage of who use or does not use their real names on Twitter.
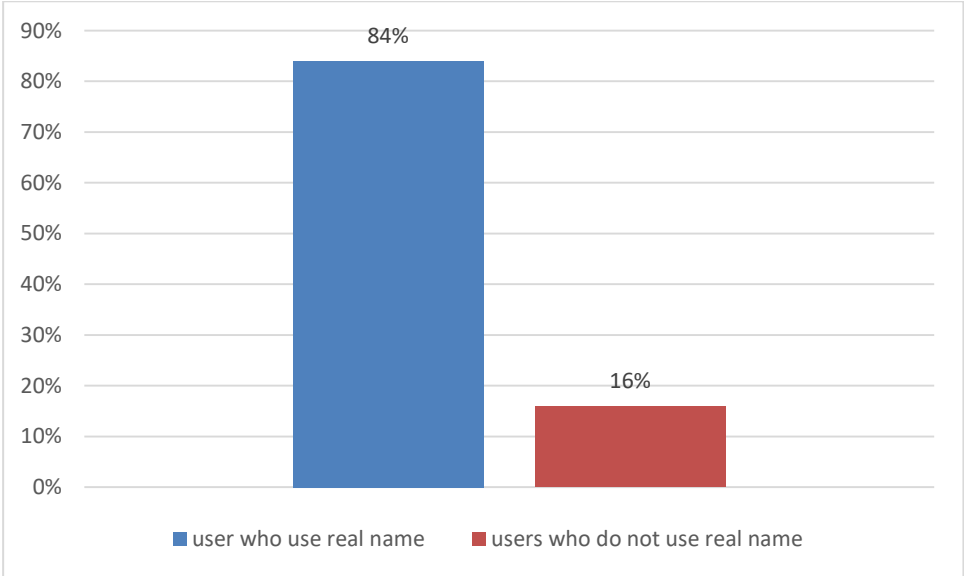
Figure 6.3 : Percentages of real name and fake name.

A small percentage of users are not use their actual names. At same time, they use fake names and fake accounts on Twitter as well as fake profile images and descriptions.

Figure 6.4 illustrates the percentage of reality of the users' preferred profile images.

Figure 6.4 : Percentages of real profile image and fake profile images.

This chart shows 26 percent of users use fake profile image on their Twitter profile. If the Figure 6.4 and Figure 6.3 are compared with each other, different meanings are emerged. Some of users who do not use fake names in their profiles use fake pictures.

Figure 6.5 shows the percentage of users by cities. The place where the most users are located is seems to be Izmir because of Izmir Katip Çelebi University is located in Izmir.
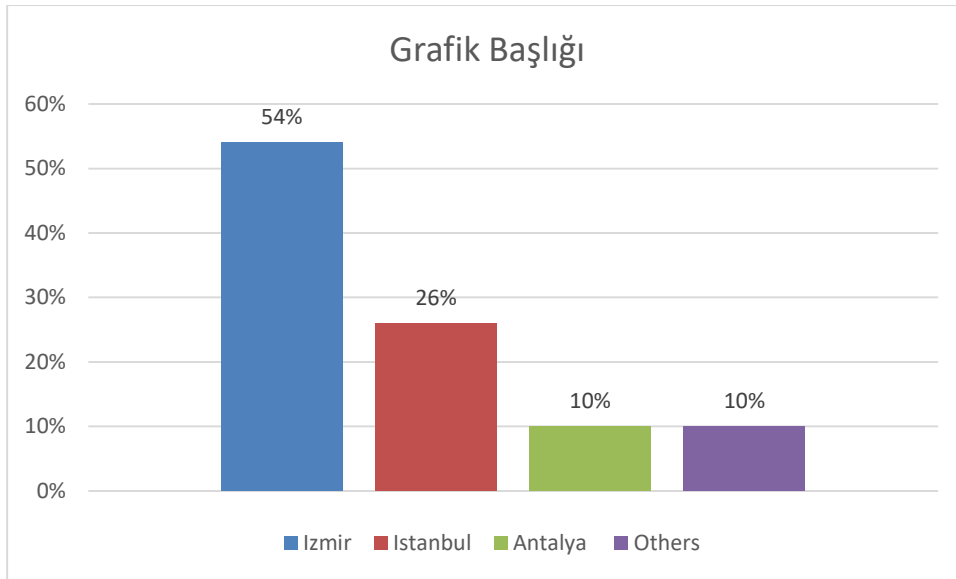
**Figure 6.5 :** Percentages of users' locations.

After Izmir, Istanbul is the place where the most users are located. However, Antalya is in third place, although it is not a big crowded city. The cities that come after Antalya are Ankara, Sakarya, Çanakkale, Urfa, Denizli and other cities of Anatolia.

The Figure 6.6 shows the percentage of users by countries. A large majority of users are posting tweet from Turkey. However differently from Turkey, there are a few users who posted tweet from Germany.
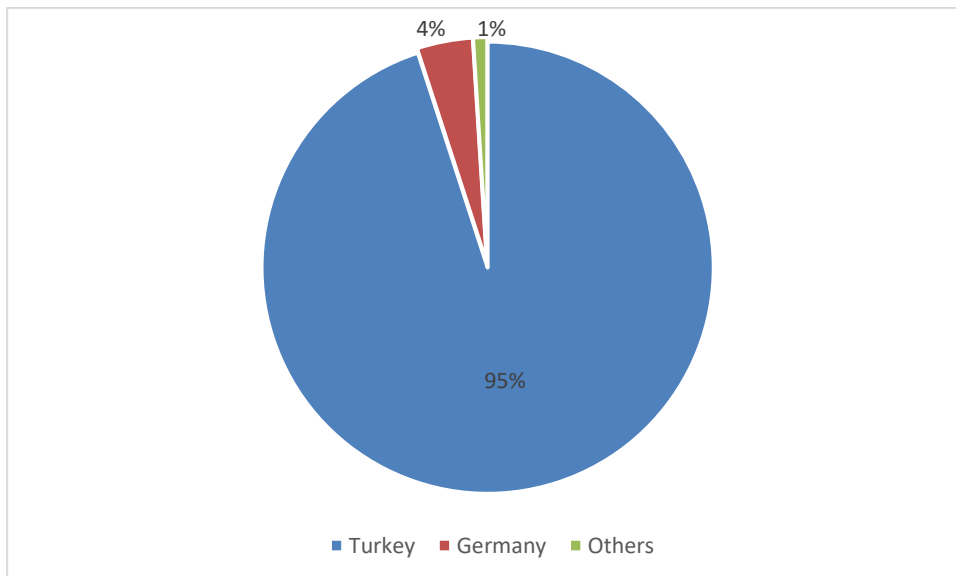
Figure 6.6 : Percentages of users by countries.

Figure 6.7 shows the percentages of users according to their follower counts. The numbers of followers of the users are grouped as the following 0 to 100, 100 to 200, 200 to 500, 500 to 1000 and so on.
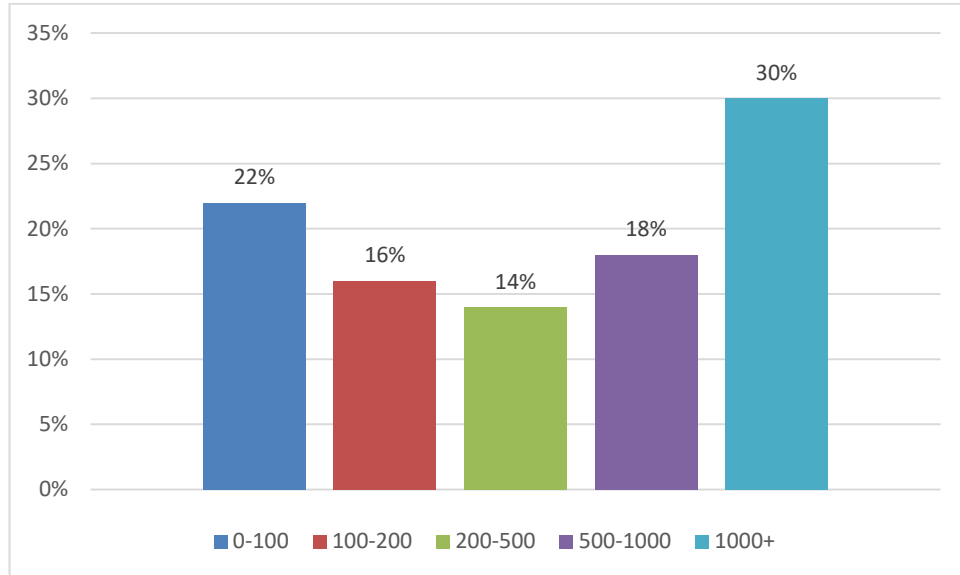
Figure 6.7 : Percentages of users by the number of followers.

The number of followers decreases from 0 to 500 and increases from 500 to 1000. 30 percent of users have more than 1000 followers while 22 percent of others have 0 to 100 followers.

The figure 6.8 shows percentages of users according to the number of friends. Figure 5.5 shows the opposite results to the existing results.
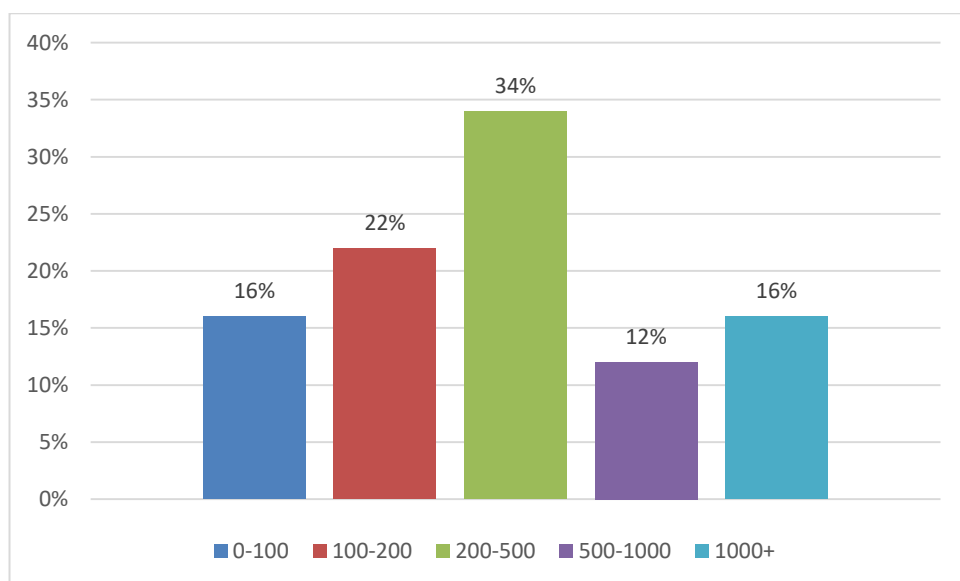
Figure 6.8 : Percentages of users by the number of friends.

The percentage of users who have between 200 and 500 friends is the highest. According to these results, there is an inverse proportion between the number of followers and the number of friends.

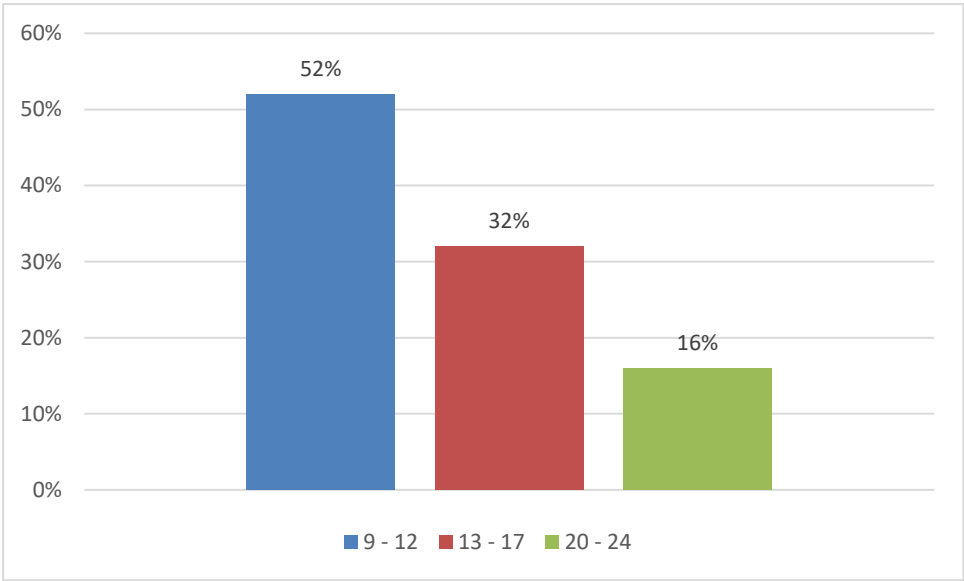The Figure 6.9 shows the percentage of time intervals in which users posted tweets. These time intervals are selected at morning, noon and evening hours. Morning hours are between 9 and 12, noon hours are between 13 and 17 and evening hours are between 20 and 24. According to the obtained results, a large majority of users post tweets on morning hours. In the evening hours, there is a decrease in the percentage of active users.

Figure 6.9 : Percentages of time intervals in which users posted tweet.

Figure 6.10 shows the percentages of users who post tweet in the morning according to their genders. The following chart shows that the female users are more active than male users at morning hours.
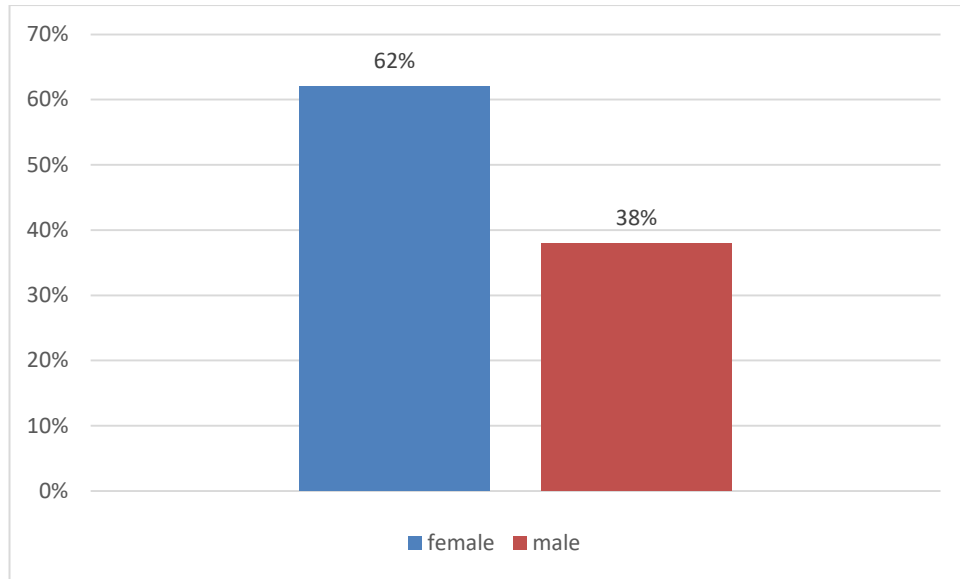
Figure 6.10 : Percentages of users who post tweet in the morning according to their genders.

As a result of this chart, it seems that female users are woken up earlier than male user. The Figure 6.11 shows the percentage of users who post tweets in the noon according to their genders.
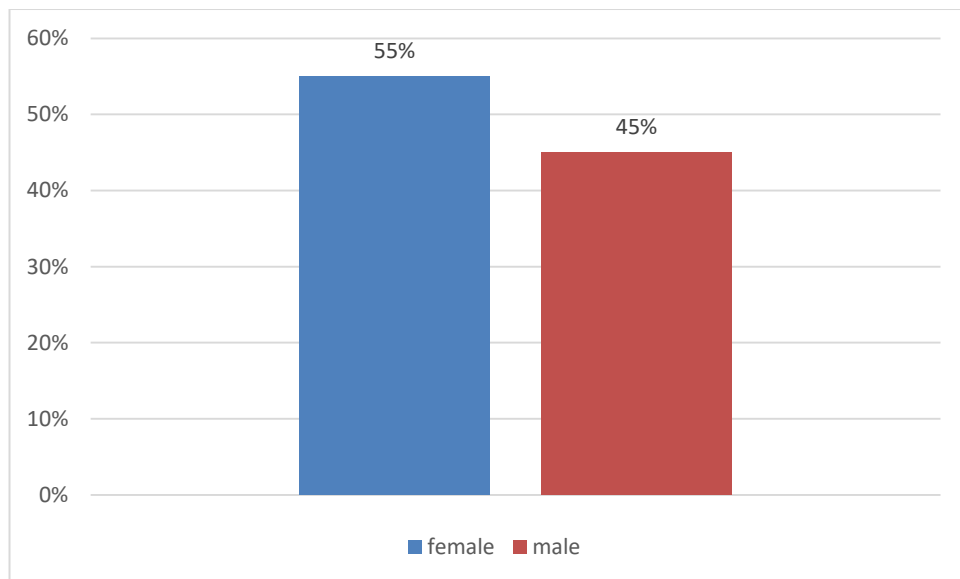
Figure 6.11 : Percentages of users who post tweet in the noon according to their genders.

This chart shows that the percentage of users who post tweet at noon hours is close to each other. The other figure 6.12 shows the percentage of users who post tweet in the evening according to their genders. It seems that there are more male users in the evening hours. The last 3 figure shows that the female users are more active during

the morning hours while male users are more active during the evening hours. Generally, on Twitter the female users spend more time than the male users.

Figure 6.12 : Percentages of users who post tweet in the evening according to their genders.

7. CONCLUSION
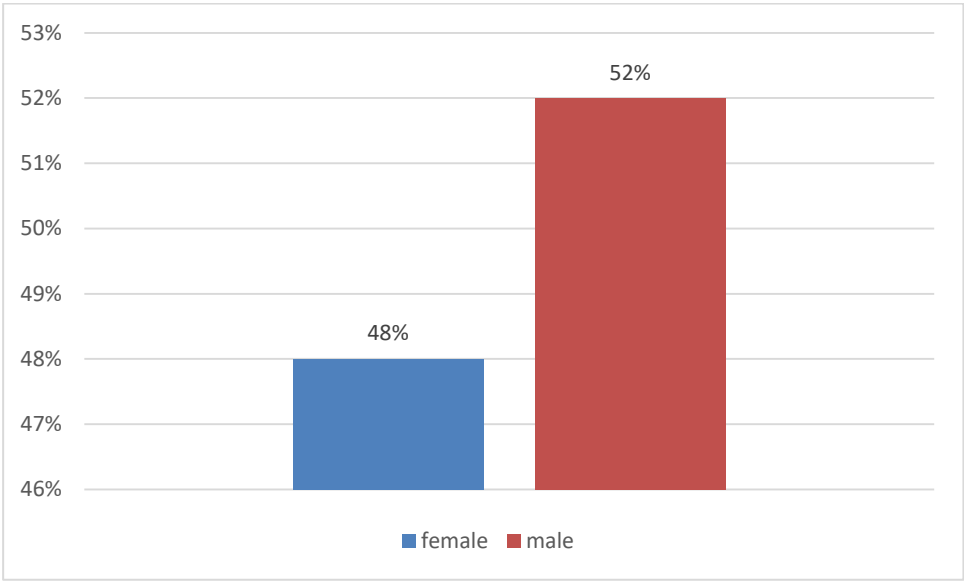
As a conclusion, technology developments in this century made communication and blogging shifts through social websites and platforms like Twitter. And this made social media platforms such as Twitter a valuable information resources in the important events like natural disasters, elections.

In this thesis study, it was aimed to gather important information via Twitter by making a smart crawler. This crawler will collect and store tweets and related information based on search with given keywords and other parameters. Then, the collected tweets were analyzed with textual measures and a simple graphic chart was provided to the end user as a visualization. An Android based mobile application was developed as a product of the thesis work. This mobile application was developed entirely using pure Java programming language. With the application, the data coming from Twitter has been evaluated in meaningful way. Very important results came out after the evaluation of the obtained data.

When the obtained results are evaluated, it seems that the data coming from Twitter has an important value. The results of the thesis work give the important clues in about many things such as the people's social behavior, personalities and characters. At this point, a lot of data can be obtained which may have beneficial results on behalf of our country. In this way, it can learn what Twitter users think about the developments in our country and take precautions accordingly. The topics that users are interested in can be determined based on tweet counts. Also the topics that users are not interested in can be determined by tweet count numbers in the same way.

As the years are progressed, Twitter and similar social platforms will become even more popular and the number of active users will increase. So, the data obtained from social platforms will be very important. However, not only data coming from Twitter and the other social media platform but also the application or programs that used for analyzing of the data that coming from social platform will carry important role for many organizations such as countries, governments, commercial organizations, universities and etc.

It is very important that the mobile application which the product of the thesis work, is made for the mobile devices because of the widespread use of mobile devices today. For this reason, achieving fast results and achieving this with a mobile application for easy use is of great importance. As the technology continues to evolve and social media usage increases, the more of this kind of work will be needed.

REFERENCES

[1] Global Social Media Resarch Summary, (2016), https://www.smartinsights.com/social-media-marketing/social-mediastrategy/new-global-social-media-research

[2] Zikolopoulos P., C. Eaton, (2011), Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill Osborne Media

[3] Boyd D., Golder S., Lotan G., (2010) Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter, HICSS-43, IEEE: Kauai

[4] Twitter Brand Resources, (2017), https://brand.twitter.com/

[5] Twitter Usage, (2017), https://about.twitter.com/

[6] Android Operating System, (2017), https://en.wikipedia.org/wiki/Android_(operating_system)

[7] Android Developer Blog, (2008), http://androiddevelopers.blogspot.com/2008/09/announcing-android10-sdk-release-1.html

[8] Android versions, (2017), https://en.wikipedia.org/wiki/Android_version_history#cite_note-1

[9] Hoog C., (2017), Global Android version usage as of January 2017

[10] Bierhoff K., (2009), API Protocol Compliance in Object-Oriented Software, CMU Institute for Software Reasearch

[11] Android Developers, (2009), https://developer.android.com/studio/commandline/index.html

[12] Wang C., Duan W., Ma J., (2011), The research of Android System Architecture and application programming, 2011 International Conference on Computer Science and Network Technology

[13] Android SDK, (2009), http://developer.android.com/sdk/index.html

[14] Oussalah M., Bhat F., Challis K., Schnier T., (2013), A software architecture for Twitter collection, search and geolocation services, Know.-Based Syst. 37, 205-120

[15] Tian Q., LAgissetty J., Li B., (2016), Finding Needles of Interested Tweets in the Haystack of Twitter Network, IEEE International Conference on Advances in Social Networks Analysis and Mining

[16] Black A., Mascaro C., Gallagher M., Goggins S., (2012), Twitter Zombie: Architecture for Capturing, Socially Transforming and Analyzing the Twittersphere

[17] Sakaki T.i Okazaki M., Matsuo Y., (2010), Eartquake shakes Twitter users: real-time event detection by social sensors, In Proceedings of the 19th international conference on World Wide Web (WWW '10)

[18] Cheng Z., Cavarlee J., Lee K., (2010), You are were you tweet: a contentbased approach to geo-locating twitter users, In preceedings of thw 19th ACM international conference on Information and knowledge management, pp. 759-768

[19] Demchenko Y., Bernstein D., Belloum A. Oprescu A., Wlodarczyk W., Laat C., (2013), New Instructional Models for Binding Effective Cirricula on Cloud Computing Technologies and Engineering, IEEE International Conference on Cloud Computing Technology and Science

[20] Bisdikian C., Mitschang B., Pedreschi D., Tseng V., (2011), Challenges for Mobile Data Management in the Era of Cloud and Social Computing, IEEE International Conference on Mobile Data Management

[21] Singh T., Darshan V. S., (2015), A Modern Data Architecture with Apache Hadoop, IEEE International Conference on Green Computing and Internet of Things

[22] Manikandan S. G., Ravi S., (2014), Big Data Analysis using Apache Hadoop, IEEE International Conference on IT Convergence and Security

[23] Alam N., Ahmed J., (2014), Hadoop Architecture and Its Issues, IEEE International Conference on Computing Science and Computational Intelligence

[24] Backhoff O., Ntoutsi E., (2016), Scalable Online-Offline Stream Clustering in Apache Spark, IEEE 16th International Conference on Data Mining Workshop

[25] Maheshwar C. R., Haritha D., (2016), Survey on High Performance Anaytics of Bigdata with Apache Spark, IEEE International Conference on Advanced Communication Control and Computing Technologies

[26] Kumar S., Singh P., Rani S., (2016), Sentimental analysis of social media using R language and Hadoop: Rhadoop, 5th International Conference on Reliability, Infocom Technologies and Optimization

[27] Twitter Application Page, (2017), https://apps.twitter.com

[28] Twitter Developer Page, (2017), https://dev.com/overview/api

[29] ASPTwitter Library, (2013), http://www.timacheson.com/blog/2013/jun/asptwitter

[30] Twitcurl Library, (2010), https://code.google.com/archive/p/twitcurl/

[31] LINQ2Twitter library, (2017), https://github.com/JoeMayo/LinqToTwitter

[32] TwitSharp Library, (2017), https://www.nuget.org/packages/TweetSharp/

[33] Twitter4j Library, http://twitter4j.org/en/index.html

[34] TwitterJSClient Library, https://github.com/BoyCook/TwitterJSClient

[35] STTwitter Library, https://github.com/nst/STTwitter

[36] FHSTwitterEngine Library, https://github.com/fhsjaagshs/FHSTwitterEngine

[37] Python-twitter Library, https://github.com/bear/python-twitter

[38] Tweepy Library, https://github.com/tweepy/tweepy

[39] Twitter Streaming API, (2017), https://dev.twitter.com/streaming/overview

[40] Twitter Rest API, (2017), https://dev.twitter.com/rest/public

[41] Leiba B., (2012), OAuth Web Authorization Protocol, IEEE Internet Computing

[42] OAuth 2.0 Protocol, https://oauth.net/2/

[43] Hardt D., (2012) RFC-6749 The OAuth 2.0 Authorization Framework

[44] Twitter OAuth, (2017), https://dev.twitter.com/oauth

[45]MPAndroidChart, https://github.com/PhilJay/MPAndroidChart

[46] Giovanetti R., Lancieri L., (2016) Model of Computer Architecture for Online Social Networks Flexible Data Analysis, IEEE International

[47] Jeremy D., Wells Randy, Richard V., Field Jr., (2016), On data collection, graph construction, and sampling in Twitter, IEEE/ACM International Conference on Advances in Social Netwoks Analysis and Mining

[48] Android data storage techniques, (2014), http://stackoverflow.com/questions/9986734/which-android-datastorage-technique-to-use/26333317

[49] Tuan D., Cheon S., Won Y., (2016), On the IO charecteristics of the SQLite Transactions, IEEE/ACM International Conference on Mobile Software Engineering and Systems

[50] Li Q., Hu X., Wu H., (2014), Database Management Strategy and REcıvery Methods of Android, 2014 5th IEEE International Conference on Software Enginnering and Service Science

[51] Nandugudi A., Nuessle C., Challen G., Milazzo E., Chen, Y., (2014), The PocketLocker Personal Storage System, IEEE 6th Conference on Mobile Computing, Applications and Services.

[52] Tang H., Wu G., Wei J., Zhong H., (2016), Generating Test Cases to Expose Concurrency Bugs in Android Applications, IEEE ASE'16

[53] Hwang K., Fox C., Dongarra J., (2012), Distributed and Cloud Computing From Prallel Processing to the Internet of Things

CURRICULUM VITAE

Name Surname	**: Levent Yavuz**
Place and Date of Birth	**: Bakırköy / 30.08.1991**
E-Mail	**: levent.yavuz@avrasya.edu.tr**

EDUCATION **:**

- **B.Sc.** **:** 2014, Gediz University, Engineering Faculty, Computer Engineering Department

Conference Papers

1. Alaybeyoglu A, Yavuz L, "Design of Social Media Analysis Based on Mobile Platforms", International Conference on Research in Education & Science, 2017(Accepted Paper)
2. Alaybeyoglu A, Yavuz L, "Android based Social Media Analysis", Cybernetics and Systems, 2017 (Submitted Paper)