Bilkent University

Department of Computer Engineering

Object Oriented Software Engineering

CS 319 Project: Instagram Data Analysis Application

Design Report

Orçun Gümüş
Cansu Demiryürek
Nihan Varilci
Melih Ekmekçi

Course Instructor: Hüseyin Özgür Tan

October 28, 2015

Table of Contents

Introduction

Instagram Data Analysis Application is a data analyzing program which can make data analysis and draw graphs among the user interactions. It is a web scale data operation and the data are provided by the Instagram API. In this program a user can see his/her social Instagram environment such as people from high school, university friends, work friends etc.. They are going to be clustered and easily identified. In addition to this, they can also get information about who like their photo regularly and who don't. This will show who the most popular one among friends is in terms of the follower number and the amount of like they get. At the same time people can easily identify who are close to each other and interested in their photos. Users' can do this analysis for themselves and at the same time if they are curious about anyone else, they can get the same information for others. To make this operation, you have to log in to our program with your Instagram account. After this process, users data will be automatically saved to the database and the analysis will be shown by a graph. In this graph, nodes are set as Instagram users, follows and likes are considered as edges.

In today's world, people are so curious about everyone's social environment and instead of what they share, they are mostly interested in who they know and how close they are. There are some mobile applications that just show their own data about who they follow and who gave up following them. By Instagram Data Analysis Application without knowing any of the people, users will be able to know which society they belong to and can know what kind of people they are.

2.1 Overview

In Instagram Data Analysis Application several functional requirements, nonfunctional requirements and constraints will be included. After the introduction part, this part will start to give more specific information about the program such as functionalities and technical information. Possible scenarios while using the program will be shown. In addition to this, they will be explained by the use case model and in the end the user interface will be demonstrated.

2.2 Functional Requirements

1) Help: Users are able to get all kind of information about application as an instructive explanation.

This instructive explanation contains the information about:

- How to use this application.
- What is the advantage of this application
- How to interpret graphs
- How to save the graphs.

This help document mainly provides Users a chance to learn the process of this application. To use this application, users have to have Instagram account. Moreover, this application works only with Instagram.

2) Login: This function is required and the most basic feature of the application for the users. Firstly, the user fills the user name and password boxes, these boxes include Instagram username and password. Then user should press the login button to access the main page of the application. If user has not got Instagram account, he/she must sing in new account from Instagram.

3) Follow Analyze

When users login the application, they see main page. Main page includes flows analyze button. Users can make an analyze over their follows. They can reach their follows graph and see their friends' communities via their follows information

4) Like Analyze

When user login the application, they see main page. Main page includes like analyze button. Users can make an analyze over their likes. They can reach their like graph and see their like communities via their like information

5) Graph and Suggestion

This application's aim is to produce graphs. Graphs show the users communities by like or follow. These communities include users' groups of various friends, their relatives and coworkers… etc. Moreover, users can see people who are more closed to them.

Users can reach people who are closed. According to graphs, this application makes suggestions to users about people who users want to follow. Analyzes produces graphs but system is always saving the graphs. They can reach their graphs whenever they want.

5) Save Downloaded Items from external API in Database

When user login to program and produce a graph, their all information (follows, likes) saved in a database. Users can access these graphs whenever they want to look in the future. Moreover, system is also using this information. During analyses system is always controlling the database first. If the required item is existing in the database, the system is not making any external API request on redundant situations.

6) Change Setting

Setting is the part of the main menu. It includes

Profile picture and background. When users click the setting button, setting page is opened. Users can choose picture from his or her photo library as profile picture. Moreover, users can choose background for application.

7) Log out

When user wants to exit the application, he or she can click the logout button. Log out button will delete all temporarily view information.

8) Delete account

User can delete their account from system. However system will delete only the information related to this user.

Use Case Description

Use Case 1:

Use Case Name: Firstly, Use Application

Participating Actors: User

Pre-condition: User must not be using this application before and have an Instagram account.

Entry condition: User presses “Login” button in the main page.

Exit condition: User presses “Logout” button in the main menu.

Main Flow of Events:

1) User press the help button and read the information about the application.

2) User fill the user name and password boxes according to his or her Instagram account.

3) After filling boxes user press “Login” button.

4) User can see his or her follower numbers in the main page.

5) User can look the graph which is about his or her followers.

6) User can save his or her graph for future.

7) To exit the program user press “Logout” button.

8) System goes to Step 2.

Use Case 2

Use Case Name: Use Application

Participating Actors: User

Pre-condition: User must have an Instagram account.

Entry condition: User presses “Login” button in the main page.

Exit condition: User presses “Logout” button in the main menu.

Main Flow of Events:

1) User fill the user name and password boxes according to his or her Instagram account.

2) After filling boxes user press “Login” button.

3) User can see his or her like and follower numbers in the main page.

4) User can look the graph which is about his or her likes and followers which he or she saved before.

6) To exit the program user press “Logout” button.

7) System goes to Step 2.

Use Case 3

Use Case Name: Change Setting

Participating Actors: User

Pre-condition: User must have an Instagram account.

Entry condition: User presses “Login” button in the main page.

Exit condition: User presses “Logout” button in the main menu.

Main Flow of Events:

1) User press the help button and read the information about the application.

2) User fill the user name and password boxes according to his or her Instagram account.

3) After filling boxes user press “Login” button.

4) User can see his or her like and follower numbers in the main page.

5) User can look the graph which is about his or her likes and followers.

6) User can save his or her graph for future.

7) User press “Settings” button in the main menu.

8) User can change his or her profile picture from photo library.

9) User also change the background of application.

10) To exit the program user press “Logout” button.

11) System goes to Step 2.

2.3 Non Functional Requirements

Usability

Without creating a new account user can directly continue with their Instagram Account.

Performance

When users start to analyze their account or any other one, it will be completed less than in 5 minutes.

Reliability

Only users can see their analysis and except the authorized user, no one can reach the analysis report. Analysis report privacy for each user will be provided.

Supportability

This program will work in all the internet browsers and graphs can also be shown in every browser. For the mobile side of this program, graphs will not be seen. Only the interaction numbers will be available.

2.4 Pseudo Requirements

Planned server provider is Amazon Web Services.

AWS will provide elastic beanstalk.

AWS will provide distributed and increasable size relational database system.

AWS will provide scalable computable power for analyze computations.

Files will have distributed over machines thanks to S3 service of AWS.

Required RAM, Hard Drive Space and computational power will be handled by AWS.

Python 3.4 will run on the server.

MySQL 5.6 will be existing as RDBS.

PHP 5.5 will accept web page requests.

2.5 Analysis

2.5.1 Scenarios

Scenario Name: Occurring Instagram Data Analysis Application

Melih Gökçek is phenomenon of social media such as Twitter. Melih Gökçek wants to determine friends who are closed than others because although he has lots of flowers, he feels alone. He needs program that draw a graph. This program takes data from Instagram and creates community according to likes and flowers. Melih shares this desire with us. We occur instagram data analysis application and thus, when Melih looks graph, which friends are close, which communites have follows and likes. When we present this application, Melih could not know to use this application. Fortunately, there is help button. When Melih enter to use this application, he clicks the help button. Therefore, he did not have difficulty in this application. First of all, he logins this application with instagram username and password. He can observe these graphs and he reached real friends and communities.

Scenario Name: Who is liking my phonos more

Meliz Ozenat is a college student. He has a Instagram account. Everytime he has uploaded any photo his friends started to like them. He have hundred average like on his photos. He is realy curious about finding the most liker amoung his friend. He login with his Instagram account to the application and make a like analyze request. After 5 min, he can now see his best fans from his scool and family. Thanks to see like communities. After that, he log off from the application.

2.5.2 Use-Case-Model

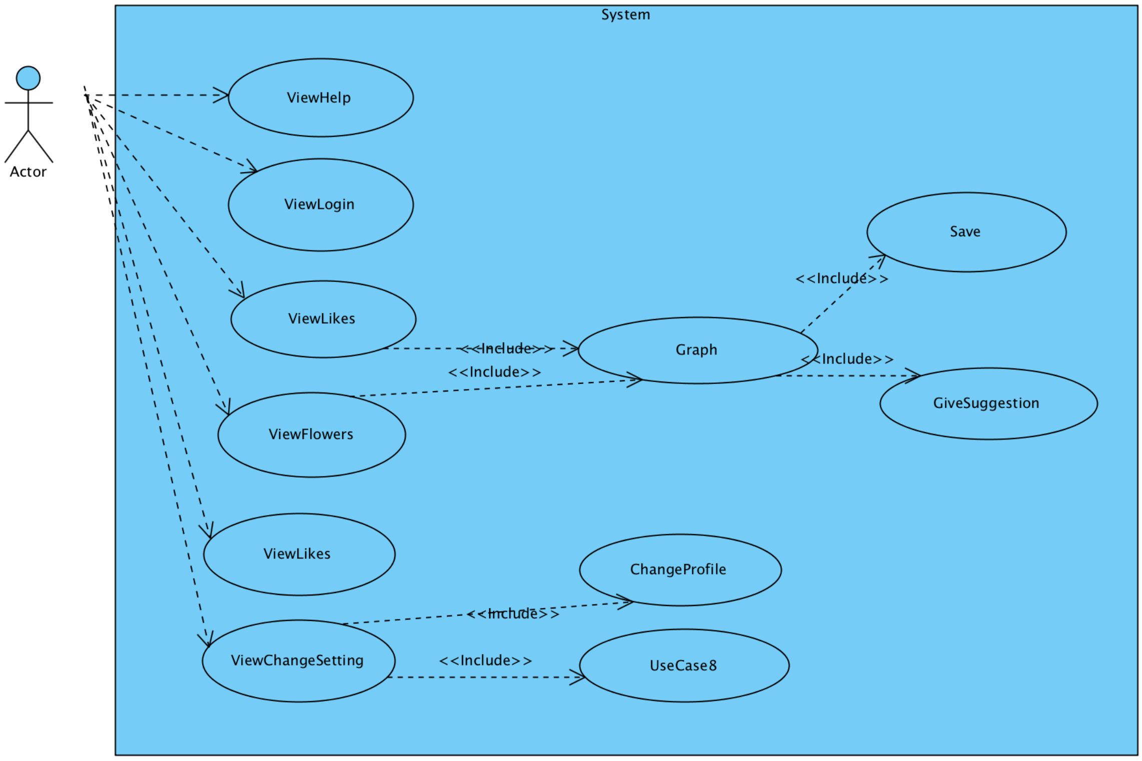


Figure 1:Use-Case Diagram

2.5.3 Object and Class Model

2.5.3.1 Domain Lexicon

SocialBaseObject: All database tables will have class equivalent in application. All column of database tables will exist as attributes in python application. SocialBaseObject will be the parent class for all of them, and socialbase object controller classes will have the ability of getting, inserting, modifying of socialbaseobejcts. Thanks to this plan application will not have different controllers for all of the socialbase objects.

User: A person who interacts with the program. It will have access token sto be able to make interact with Instagram API.

InstaUser: An instagram user. It has username, id, profile_picture etc.

Analyze: A standard analyze object which is the child of socialbaseobject. It will hold the user id which want to make this analyze, request time and etc.

FollowAnalyse: One of the analyze types. It is one of the child of the Analyze class. This is only the ticket of the analyze. It is holding the analyze type, analyze purposed user, specific analyze proprietes.

InstaRelation: Child of SocialBaseObject. It is holding source_id and target_id of a following event.

InstaLike: Child of SocialBaseObject. It is holding source_id and target_id of a like event and also media_id of the liked media.

IntaUserGroup: It holds list of instauser but also it has attributes about gender information about the instauser group that it has.

Follows: Child of InstaUserGroup. Like its parent it has a list of InstaUser and information about genders. However, it has also a InstaUser who is follows.

Our class model divided 3 main parts. Main parts are sub systems named InstaModel, InstaController, InstaView

Model Controller Objects: Controller objects are respobsible for all model objects. They are getting data from API or database and they are responsible for inserting data to database.

Analysis Controller Objects: They are responsible for the main analysis part. They have “do()” method which have the ability of processing data to make an analysis over them.

2.5.3.2 Class Diagrams

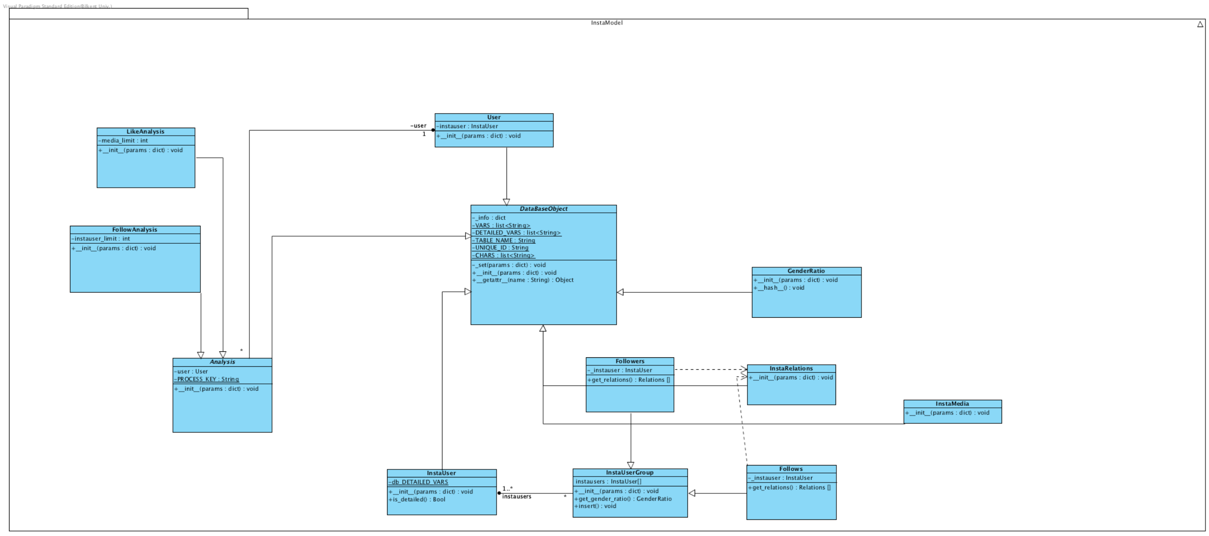

Visual Paradigm Standard Edition(Bilkent Univ.)
InstaModel
LikeAnalysis
-media_limit : int
+__init__(params : dict) : void
FollowAnalysis
-instauser_limit : int
+__init__(params : dict) : void
Analysis
-user : User
-PROCESS_KEY : String
+__init__(params : dict) : void
*
-user
1
User
-instauser : InstaUser
+__init__(params : dict) : void
DataBaseObject
-_info : dict
-VARS : list<String>
-DETAILED_VARS : list<String>
-TABLE_NAME : String
-UNIQUE_ID : String
-CHARS : list<String>
-_set(params : dict) : void
+__init__(params : dict) : void
+__getattr__(name : String) : Object
GenderRatio
+__init__(params : dict) : void
+__hash__() : void
Followers
-_instauser : InstaUser
+get_relations() : Relations []
InstaRelations
+__init__(params : dict) : void
InstaMedia
+__init__(params : dict) : void
InstaUser
-db_DETAILED_VARS
+__init__(params : dict) : void
+is_detailed() : Bool
1..*
instausers
*
InstaUserGroup
instausers : InstaUser[]
+__init__(params : dict) : void
+get_gender_ratio() : GenderRatio
+insert() : void
Follows
-_instauser : InstaUser
+get_relations() : Relations []

Diagram 1: Model Sub Package Class Diagram

Model classes have a common main parent. SocialBaseObject is common parent. All attributes can be accessible and settable by getter and setter which are not shown in UML diagram.

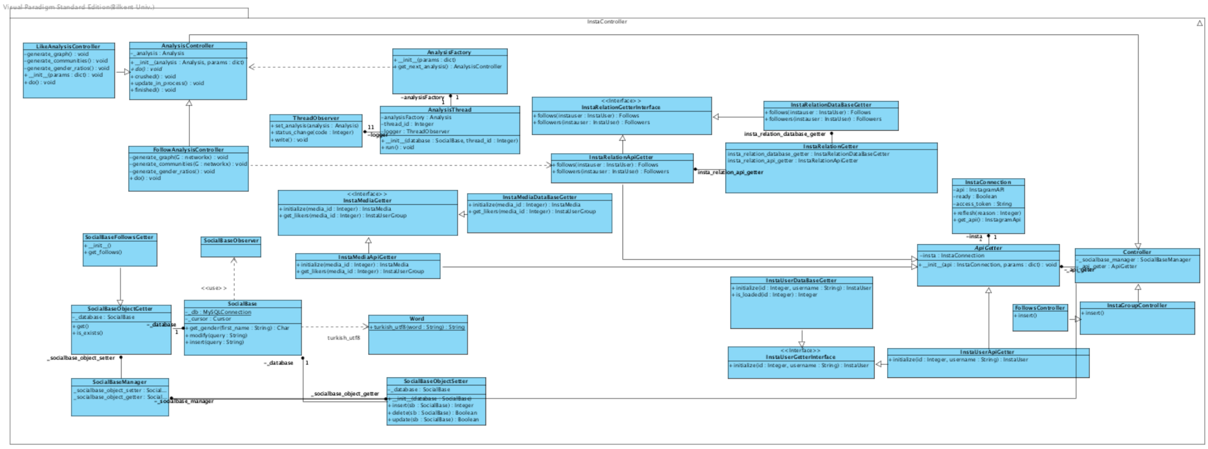
All attributes of model classes is saved in a –info attribute which is a dict(Python base class for Hashed Map)

For example, InstaUser will have

'instauser_uid', 'username', 'profile_picture', 'full_name', 'gender', 'is_public', 'total_followers','total_media', 'total_follows','is_loaded', 'total_media', 'total_followers', 'total_follows', 'follows_gr_uid'

These attributes will be saved inside *info* attribute like:

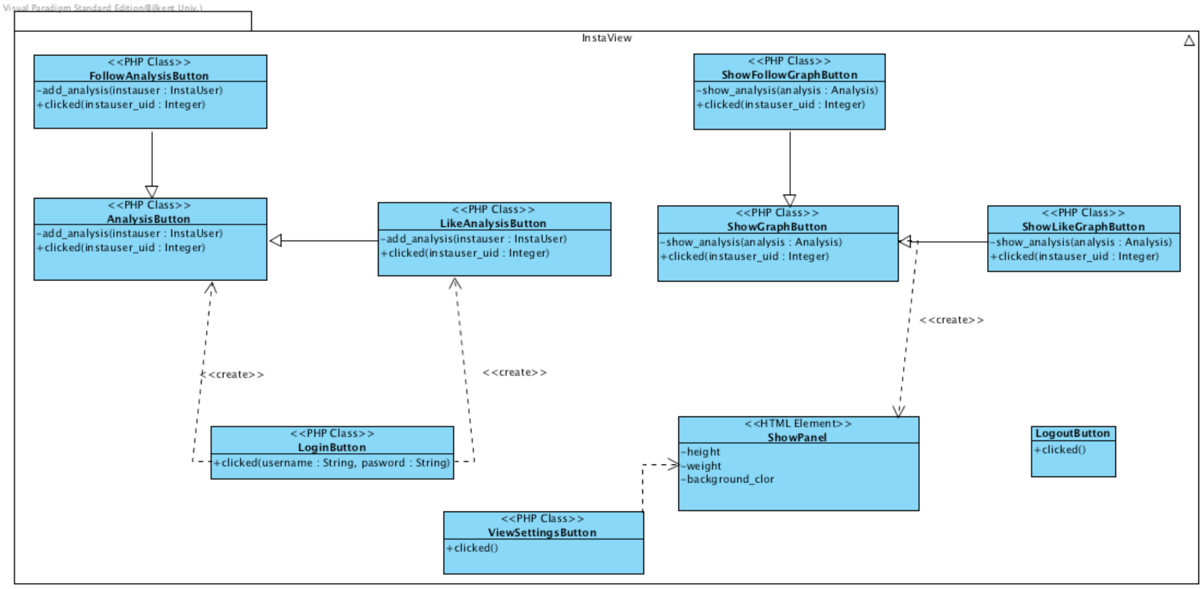
Example info attribute: {instauser_uid: 123123, username: ‘aylakadam’, profile_picture: ‘http://…/’, full_name: ‘Ayca Abancı’, gender: ‘f’, total_media: 56…}



Visual Paradigm Standard Edition(Bilkent Univ.)
InstaController
LikeAnalysisController
-generate_graph() : void
-generate_communities() : void
-generate_gender_ratios() : void
+__init__(params : dict) : void
+do() : void
AnalysisController
-_analysis : Analysis
+__init__(analysis : Analysis, params : dict)
+do() : void
+crushed() : void
+update_in_process() : void
+finished() : void
AnalysisFactory
+__init__(params : dict)
+get_next_analysis() : AnalysisController
-analysisFactory
1
AnalysisThread
-analysisFactory : Analysis
-thread_id : Integer
-logger : ThreadObserver
+__init__(database : SocialBase, thread_id : Integer)
+run() : void
ThreadObserver
+set_analysis(analysis : Analysis)
+status_change(code : Integer)
+write() : void
11
-logger
FollowAnalysisController
-generate_graph(G : networkx) : void
-generate_communities(G : networkx) : void
-generate_gender_ratios() : void
+do() : void
<<interface>>
InstaRelationGetterInterface
+follows(instauser : InstaUser) : Follows
+followers(instauser : InstaUser) : Followers
InstaRelationDataBaseGetter
+follows(instauser : InstaUser) : Follows
+followers(instauser : InstaUser) : Followers
insta_relation_database_getter
InstaRelationGetter
insta_relation_database_getter : InstaRelationDataBaseGetter
insta_relation_api_getter : InstaRelationApiGetter
InstaRelationApiGetter
+follows(instauser : InstaUser) : Follows
+followers(instauser : InstaUser) : Followers
insta_relation_api_getter
<<interface>>
InstaMediaGetter
+initialize(media_id : Integer) : InstaMedia
+get_likers(media_id : Integer) : InstaUserGroup
InstaMediaDataBaseGetter
+initialize(media_id : Integer) : InstaMedia
+get_likers(media_id : Integer) : InstaUserGroup
InstaMediaApiGetter
+initialize(media_id : Integer) : InstaMedia
+get_likers(media_id : Integer) : InstaUserGroup
InstaConnection
-api : InstagramAPI
-ready : Boolean
-access_token : String
+reflesh(reason : Integer)
+get_api() : InstagramApi
-insta
1
ApiGetter
-insta : InstaConnection
+__init__(api : InstaConnection, params : dict) : void
Controller
-_socialbase_manager : SocialBaseManager
-_api_getter : ApiGetter
-_api_getter
SocialBaseFollowsGetter
+__init__()
+get_follows()
SocialBaseObserver
<<use>>
SocialBaseObjectGetter
-_database : SocialBase
+get()
+is_exists()
-_database
1
SocialBase
-_db : MySQLConnection
-_cursor : Cursor
+get_gender(first_name : String) : Char
+modify(query : String)
+insert(query : String)
Word
+turkish_utf8(word : String) : String
turkish_utf8
InstaUserDataBaseGetter
+initialize(id : Integer, username : String) : InstaUser
+is_loaded(id : Integer) : Integer
<<interface>>
InstaUserGetterInterface
+initialize(id : Integer, username : String) : InstaUser
InstaUserApiGetter
+initialize(id : Integer, username : String) : InstaUser
FollowsController
+insert()
InstaGroupController
+insert()
_socialbase_object_setter
-_database
1
SocialBaseManager
_socialbase_object_setter : Social...
_socialbase_object_getter : Social...
-_socialbase_manager
_socialbase_object_getter
SocialBaseObjectSetter
-_database : SocialBase
+__init__(database : SocialBase)
+insert(sb : SocialBase) : Integer
+delete(sb : SocialBase) : Boolean
+update(sb : SocialBase) : Boolean

Diagram 2: Controller Sub Package Class Diagram

Controller objects are responsible for fulfill model objects by getting information from API, getting information from database or inserting them(SocialBaseObjects) to database



Visual Paradigm Standard Edition(Bilkent Univ.)
InstaView
<<PHP Class>>
FollowAnalysisButton
-add_analysis(instauser : InstaUser)
+clicked(instauser_uid : Integer)
<<PHP Class>>
ShowFollowGraphButton
-show_analysis(analysis : Analysis)
+clicked(instauser_uid : Integer)
<<PHP Class>>
AnalysisButton
-add_analysis(instauser : InstaUser)
+clicked(instauser_uid : Integer)
<<PHP Class>>
LikeAnalysisButton
-add_analysis(instauser : InstaUser)
+clicked(instauser_uid : Integer)
<<PHP Class>>
ShowGraphButton
-show_analysis(analysis : Analysis)
+clicked(instauser_uid : Integer)
<<PHP Class>>
ShowLikeGraphButton
-show_analysis(analysis : Analysis)
+clicked(instauser_uid : Integer)
<<create>>
<<create>>
<<create>>
<<PHP Class>>
LoginButton
+clicked(username : String, pasword : String)
<<HTML Element>>
ShowPanel
-height
-weight
-background_clor
LogoutButton
+clicked()
<<PHP Class>>
ViewSettingsButton
+clicked()

Diagram 3:View Sub Package Class Diagram

2.5.4 Dynamic Models

In the server side Python modules is always working and idling by waiting any analysis request. If any request come in database. One of the thread is taking it (by locking it for inhibit concurrency errors). In the Diagram 4 the satate diagram of a example AnalysisThread. It is idling while there is not any new analysis in the database. When it meet with new analysis, it collects the data about it and trying to do this analysis request. And return its idling and searching state.

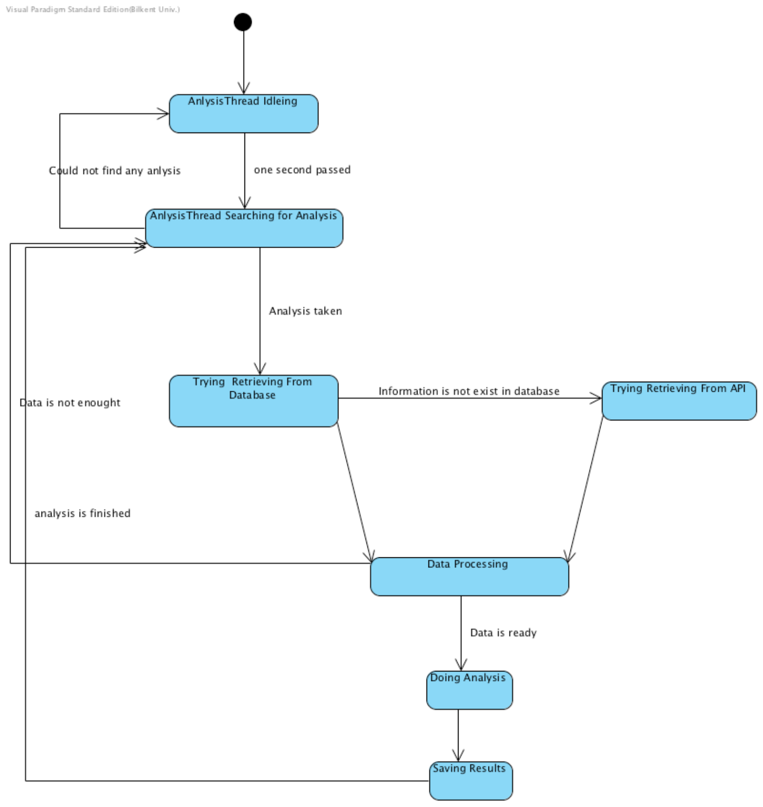
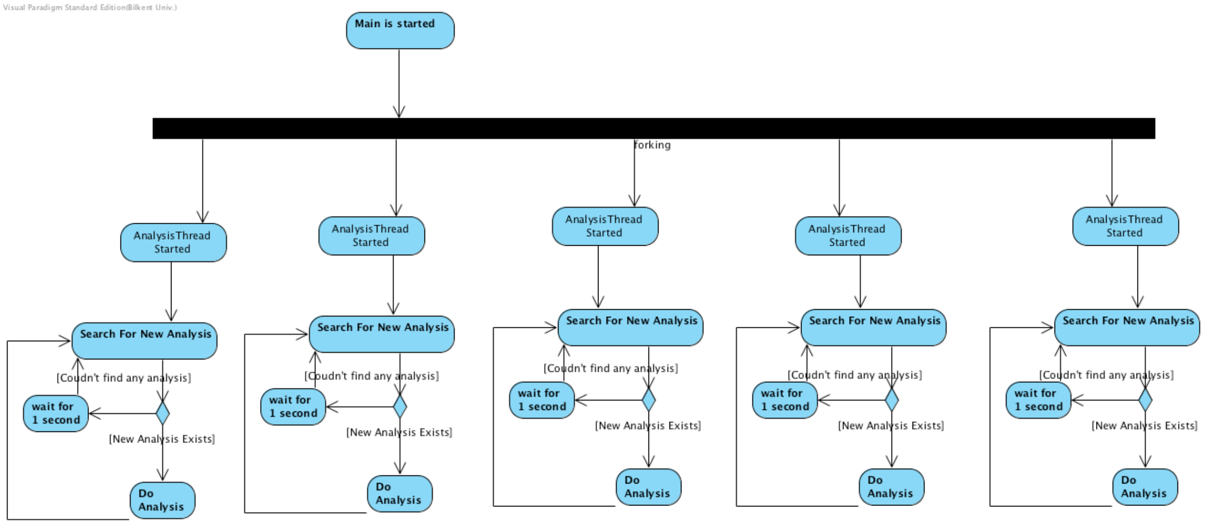

Visual Paradigm Standard Edition(Bilkent Univ.)
AnlysisThread Idleing
Could not find any anlysis
one second passed
AnlysisThread Searching for Analysis
Analysis taken
Trying Retrieving From Database
Information is not exist in database
Trying Retrieving From API
Data is not enought
analysis is finished
Data Processing
Data is ready
Doing Analysis
Saving Results

Diagram 4: State diagram of a AnalysisThread



There is not only one AnalysisThread. To benefit I/O and network usage we use multiple thread system. All system work is distributed parallel by separate different analysis to different threads.

AnalysisThread stars an single AnalysisController over an single analysis. It can be understood by (Diagram 4) that when it finished the analysis. It run finished method over the controller and seeking for any new analysis.

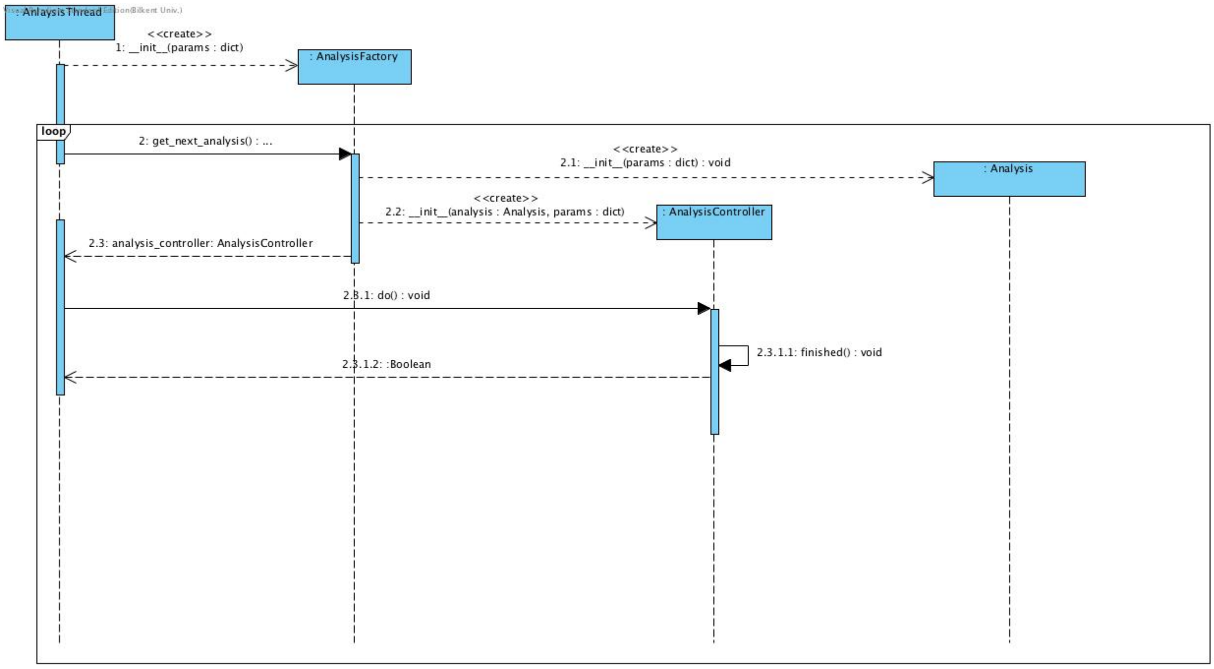

: AnlaysisThread
Visual Paradigm Standard Edition(Bilkent Univ.)
<<create>>
1: __init__(params : dict)
: AnalysisFactory
loop
2: get_next_analysis() : ...
<<create>>
2.1: __init__(params : dict) : void
: Analysis
<<create>>
2.2: __init__(analysis : Analysis, params : dict)
: AnalysisController
2.3: analysis_controller: AnalysisController
2.3.1: do() : void
2.3.1.1: finished() : void
2.3.1.2: :Boolean

Diagram 5: Sequence Diagram in Server Side after received any Possible Analysis Request from User Interface

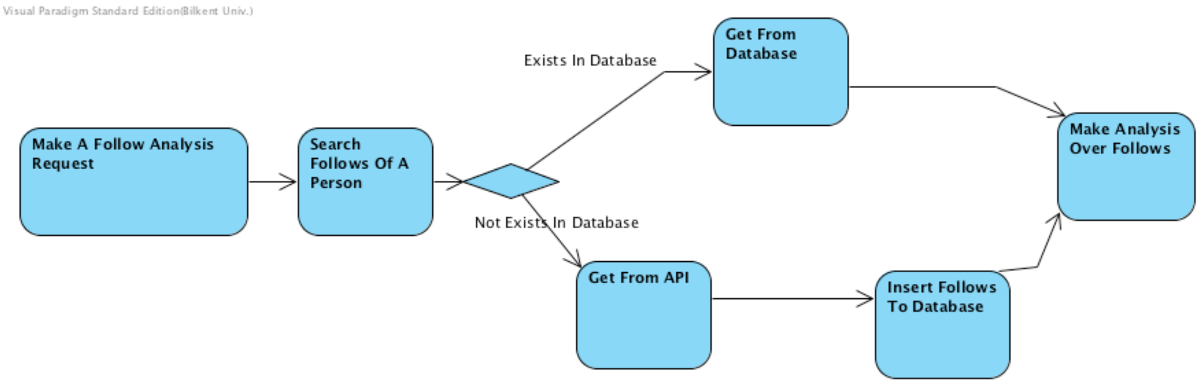

Visual Paradigm Standard Edition(Bilkent Univ.)
Make A Follow Analysis Request
Search Follows Of A Person
Exists In Database
Not Exists In Database
Get From Database
Get From API
Insert Follows To Database
Make Analysis Over Follows

Diagram 6: Activity Diagram of Information Retrieval

The application will be always preferring to use the information in the databases because of non functional requirements. So Socialbase controllers is always make a search over the database before they do a request from the external Instagram API.

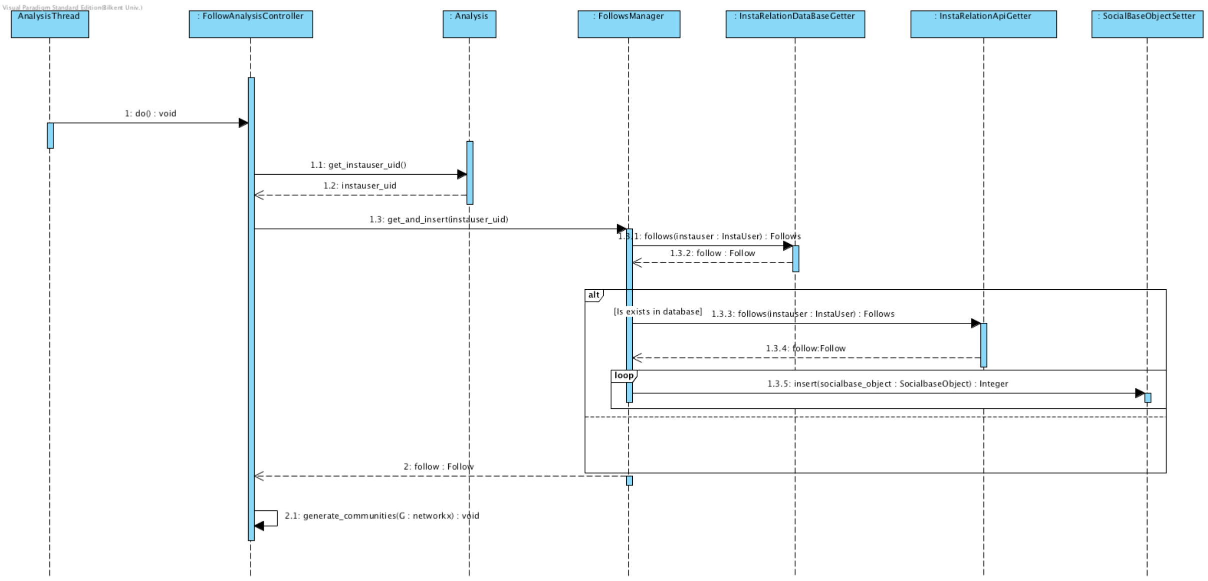

Visual Paradigm Standard Edition(Bilkent Univ.)
AnalysisThread
: FollowAnalysisController
: Analysis
: FollowsManager
: InstaRelationDataBaseGetter
: InstaRelationApiGetter
: SocialBaseObjectSetter
1: do() : void
1.1: get_instauser_uid()
1.2: instauser_uid
1.3: get_and_insert(instauser_uid)
1.3.1: follows(instauser : InstaUser) : Follows
1.3.2: follow : Follow
alt
[Is exists in database]
1.3.3: follows(instauser : InstaUser) : Follows
1.3.4: follow:Follow
loop
1.3.5: insert(socialbase_object : SocialbaseObject) : Integer
2: follow : Follow
2.1: generate_communities(G : networkx) : void

Diagram 7: Follow Analysis is Processing By Application

This behavior of application can be shown as a sequence diagram (Diagram 6). One of the analysis diagram is calling do method of a AnalysisController. FollowManager is responsible for information retrieval and store of follow information.

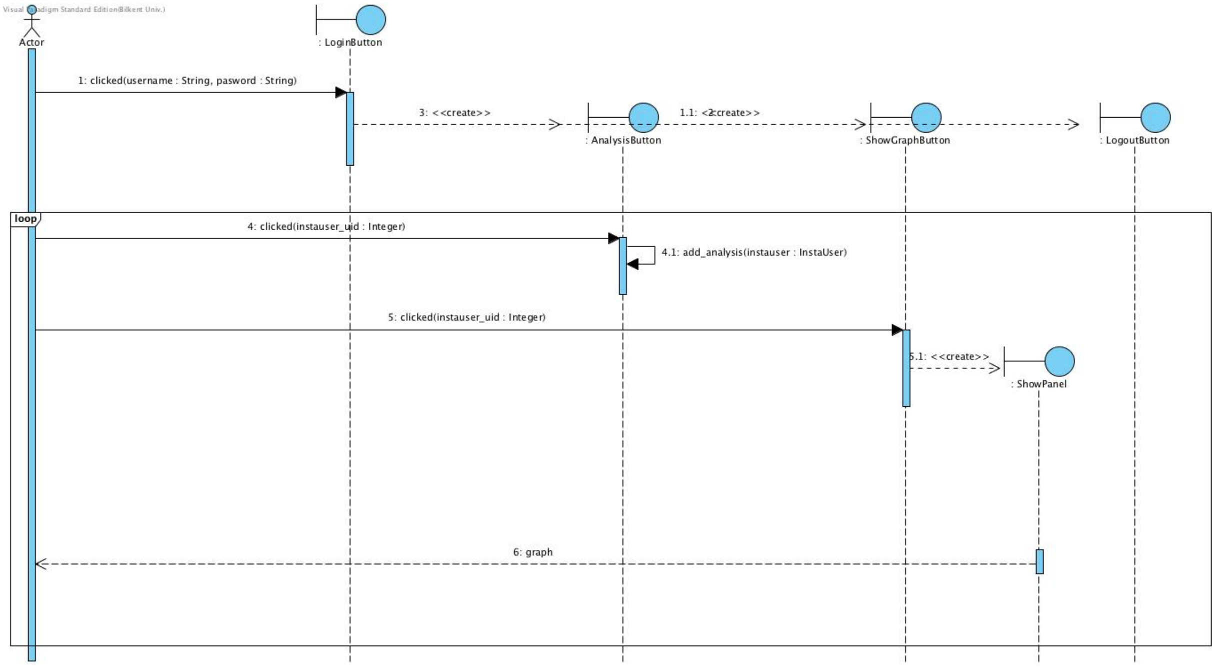
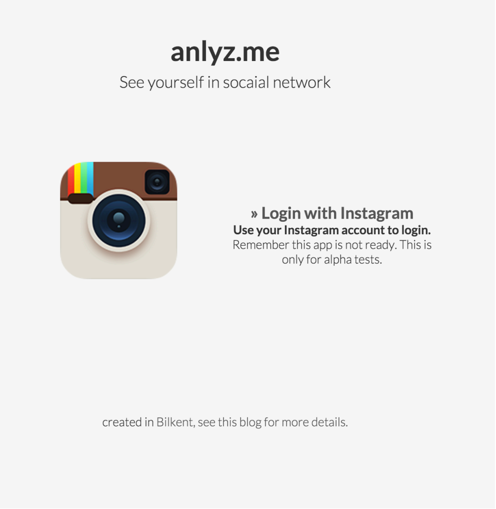

Visual Paradigm Standard Edition(Bilkent Univ.)
Actor
: LoginButton
1: clicked(username : String, pasword : String)
3: <<create>>
: AnalysisButton
1.1: <2create>>
: ShowGraphButton
: LogoutButton
loop
4: clicked(instauser_uid : Integer)
4.1: add_analysis(instauser : InstaUser)
5: clicked(instauser_uid : Integer)
5.1: <<create>>
: ShowPanel
6: graph

Diagram 8: Sequence Diagram for Standard Analysis Request from User Interface

User interface will contain login button. After pressed login button with correct authorization username and password, the page will change to a main menu which have analysis button and show graph button with a panel and logout button.

2.5.5 User Interface



anlyz.me

See yourself in socaial network

» Login with Instagram

Use your Instagram account to login.

Remember this app is not ready. This is only for alpha tests.

created in Bilkent, see this blog for more details.

Figure 2:Welcome Page

User will see be welcomed with a welcome page. He will basically informed about the application in this section.

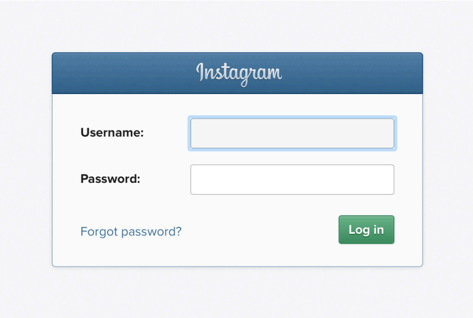


Figure 3: Login Page

User will login with a Instagram account to this application. So login page will be provided by Instagram.

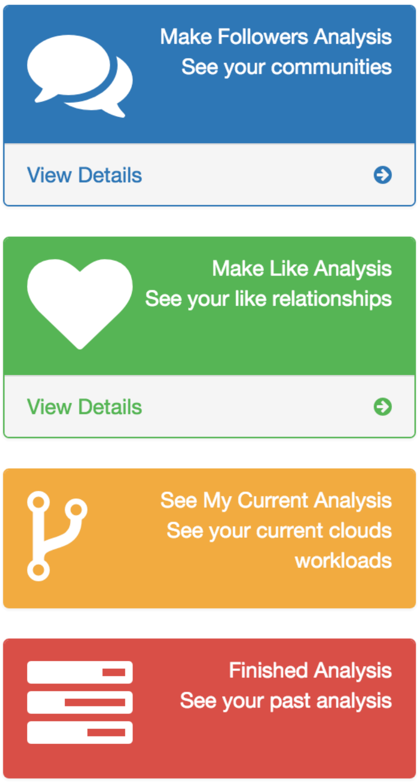


Figure 4:Job Selecting Page

User will select his analysis thought a main menu. There will be always a possibility to see past analysis.

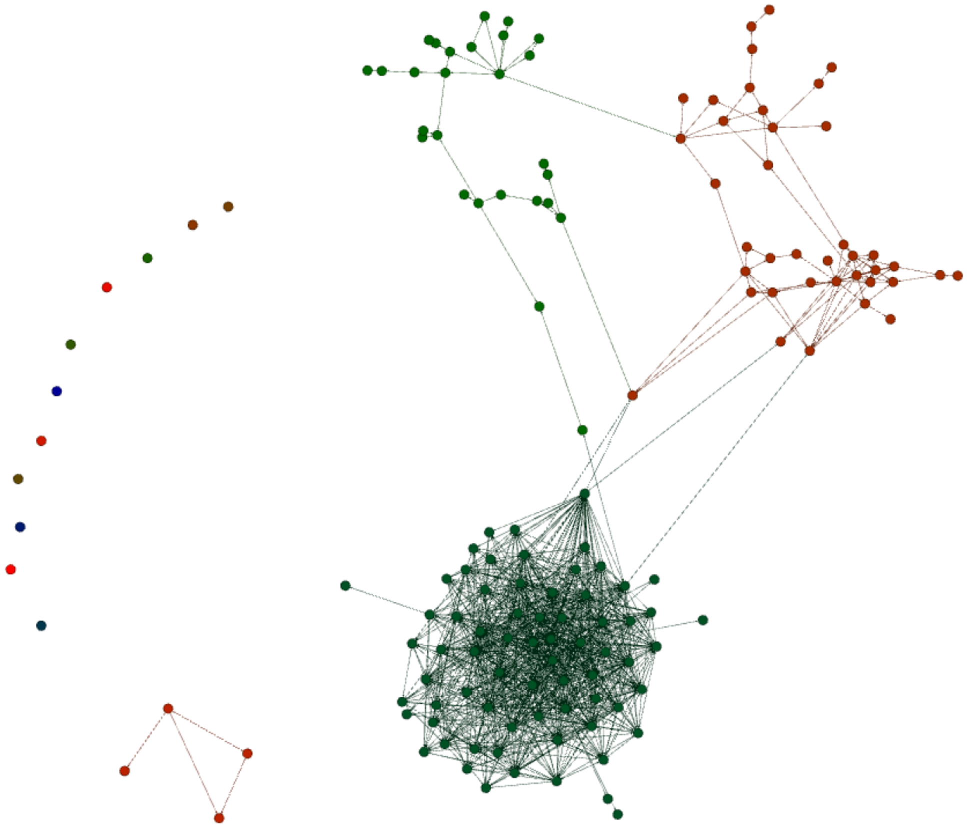


Figure 5: Follow Graph Example

Follow graph of follow information. User will see directly the communities that the person has. The graph will be drawn by ForceAtlas2 in server side and will be motionless in browser side but nodes will be clickable.

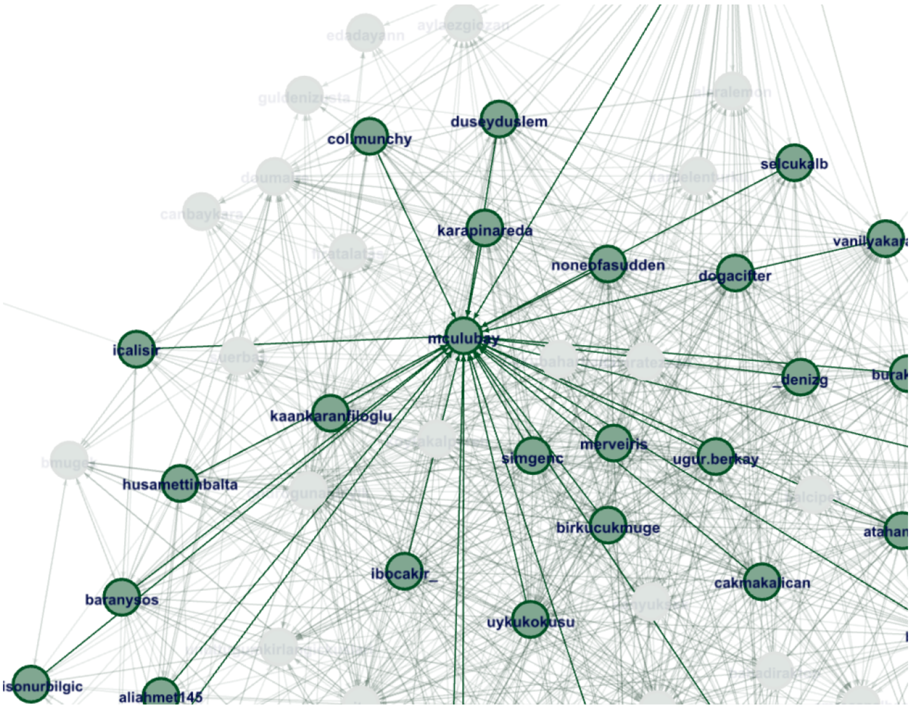


Figure 6: Follow Graph Zoomed, Focused to A User

When a node pressed, it will be possible to see the edges of this person more clearly.

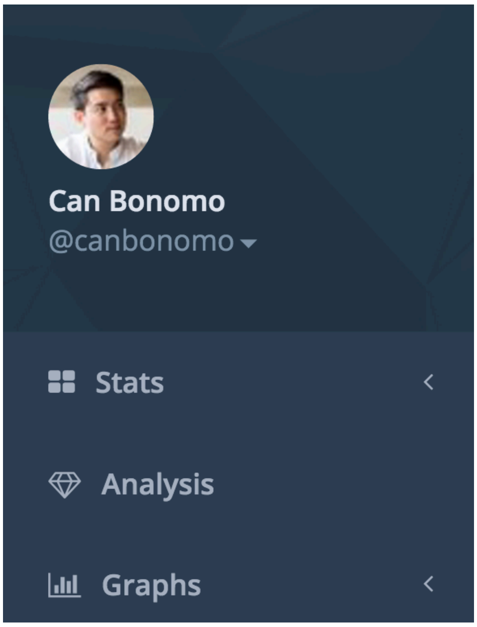
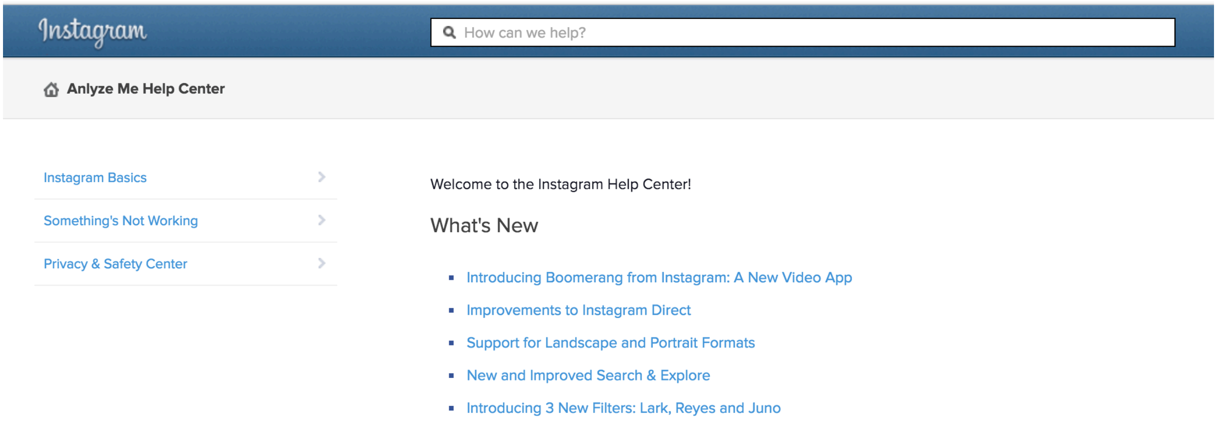


Figure 7:Sidebar Example

Sidebar will make enable to user to jump over sections directly.



Welcome to the Instagram Help Center!

What's New

- Introducing Boomerang from Instagram: A New Video App
- Improvements to Instagram Direct
- Support for Landscape and Portrait Formats
- New and Improved Search & Explore
- Introducing 3 New Filters: Lark, Reyes and Juno

Figure 8: Help Section

Help section is a HTML page to help users to understand better application dynamics.

3.Conclusion

Our aim occurs Instagram Data Analysis Application. This is a data-analyzing program that can make data analysis and draw graphs among the user interactions. Users are able to see their community and thus, users social interaction increases because they can see followers and likes. With the increased amount of data analyses and visualization of data are getting more attraction day by day. This application will be a pilot example of these nowadays subjects of computer science.

4.Referances

Wikipedia (2015) TERADATA 9 26,2015 tarihinde Wikipedia’dan : https://en.wikipedia.org/wiki/Teradata adresinden alındı

Sergey Brin and Lawrence Page (2002) 9 25, 2015 The Anatomy of a Large- Scale Hypertextual Web Search Engine, Stanford University, Stanford, CA 94305, USA http://infolab.stanford.edu/pub/papers/google.pdf adresinden alındı

D3. (2014). D3js. 9 25, 2015 tarihinde Data-Driven Documents: http://d3js.org/ adresinden alındı

Ramos, J. (2002). Using TF-IDF to Determine Word Relevance in Document Queries. http://citeseerx.ist.psu.edu/ adresinden alındı

TERADATA. (2014).0. 9 24, 2015 tarihinde teradata: http://www.teradata.com.tr/Teradata-Aster-Discovery-Platform adresinden alındı

AWS (2015)).0. 9 24, 2015 tarihinde Amazon Cloud Services:

http://www.cloudera.com/content/dam/cloudera/Resources/PDF/whitepaper/AWS_Reference_Architecture_Whitepaper.pdf