UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

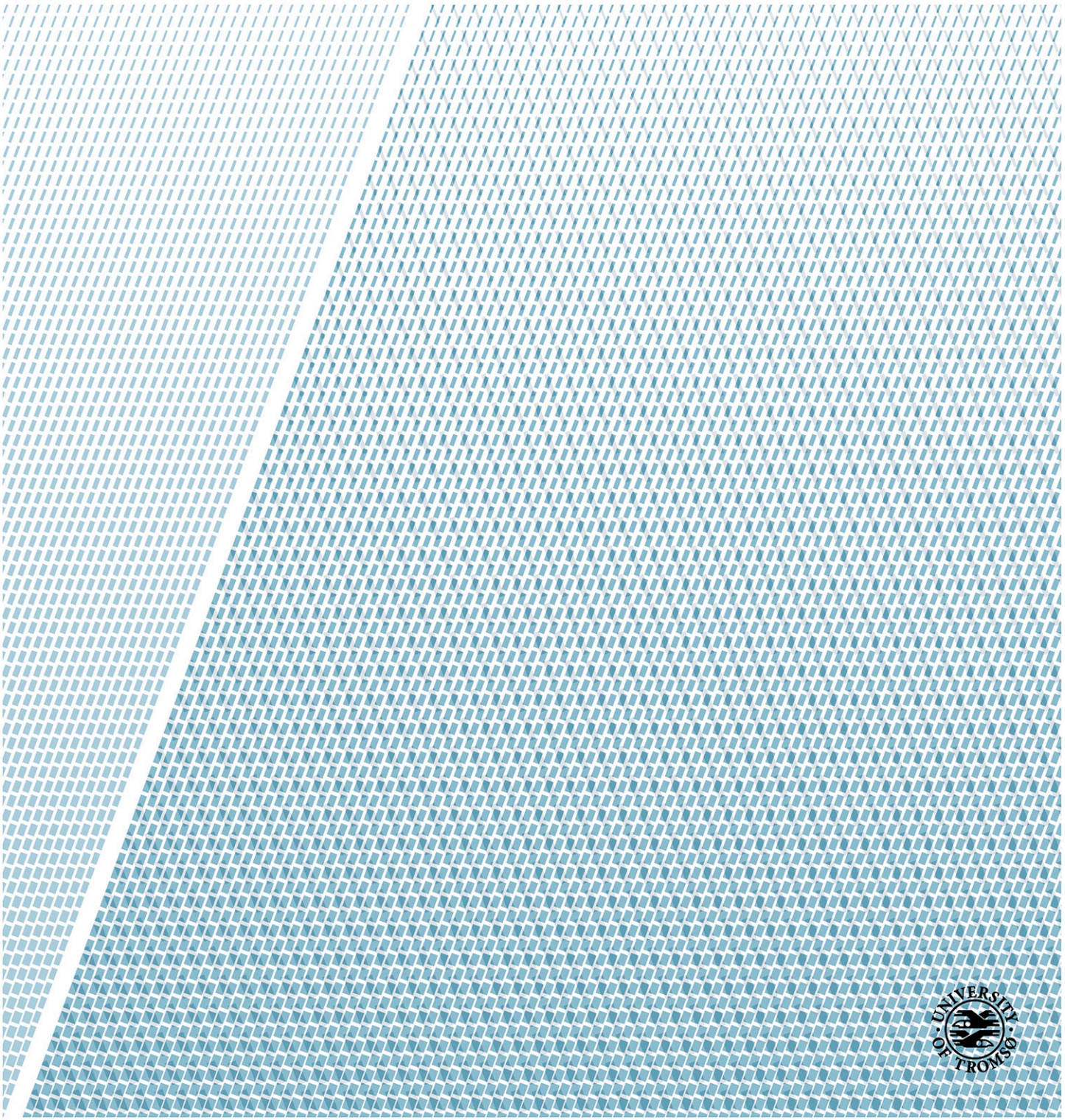
Faculty of Science and Technology
Department of Computer Science

User-Based Information Search across Multiple Social Media

—

Marte Lise Gåre

INF-3981 Master's Thesis in Computer Science - June 2015



Abstract

Most of todays Internet users are registered to one or more social media applications. As so many are registered to multiple application, it has become difficult to locate friends, former colleagues, peers and acquaintances. Reasons for this include private profiles, name collisions, multiple usernames, lack of profile attributes and profile picture.

The system designed and implemented in this thesis enable automatic user-based information search across multiple social media without relying on explicit user account registration as a basis for combining information. It uses information from a posting on a social media site to find a specific user on multiple other social media. Twitter is used as a starting point where the end-user can choose a tweet of interest, press the tweet and automatically be presented with information about the author of the tweet, including profiles on other social media.

The application was tested and evaluated by running tweets from a set of 100 Twitter users, where 14 of these were public figures, through the system and documenting the results.

Acknowledgements

I would like to thank my supervisor Professor Randi Karlsen for her guidance, feedback, ideas and advice during all stages of this thesis. It has been great working with you and I really appreciate your feedbacks.

Further I would like to thank my classmates for 5 great years of study. I wish you all the best of luck in the future.

Finally I would like to thank my friends and family for their support, and a special thanks to my boyfriend for his feedback and support.

Table of Contents

List of Figures

List of Tables

1 Introduction

Social media is a source of huge amounts of information, news updates, online collaboration, networking, viral marketing and entertainment. Many people use social media, and they are often registered on multiple social media sites.

Technologies used in social media are many, and they take on different forms including wikis, blogs, microblogs (e.g. Twitter), social networking sites (e.g. Facebook), content communities (e.g. YouTube), virtual worlds, social tagging, Internet forums, photo sharing, music sharing, and more. The different kinds of social media applications all have specific goals, characteristics and user communities. These applications differ in their content, extensiveness of information provided in user profiles, relationships among users, communication, search etc. Some of the applications overlap in user groups, as users tend to use or be registered to multiple social media applications. However automatically identifying a specific user on different social networks can be difficult due to the fact that many Social medias are largely closed applications that do not encourage interoperation, and therefore searching social media content across multiple social media applications can be challenging.

Social network aggregation based on user registration is available through services such as FriendFeed, Alternion and Flavors.me. Such services help users to combine their different social networking profiles into one, and enable search across multiple social networks. However, a limitation of these services is the requirement that users must explicitly register their different social media accounts to enable combination of social media information.

1.1 Problem definition

"The goal of this project is to enable automatic user-based information search across multiple social media, without relying on explicit user account registration as a basis for combining information."

1.2 Motivation

A motivation for cross social media search is the desire to find relevant information and possibly more postings and discussions on an interesting topic. For example, after reading an interesting tweet, we may be interested in finding more postings on the topic from the same user, on some other social media. The user-based cross social media search will combine information from multiple social media and make it possible for a user to find and follow a specific user on different social media.

1.3 Methodology

The discipline of computing is in the final report [1] of the ACM Task Force on the Core of Computer Science divided into three major paradigms:

1. **Theory** is rooted in mathematics, and the approach is to study objects, define problems, propose theorems, and attempts to prove the theorems in order to find new relationships between the objects and progress in computing.
2. **Abstraction** is rooted in experimental scientific method, and seeks to investigate a phenomenon in computing. The approach is to form hypotheses, collect data, designing experiments, and analyzing the results.
3. **Design**, is rooted in engineering. It seeks to construct a system to solve a given problem. The approach is to state requirements and specifications, design and implement the system, and finally test and analyze the system.

The design paradigm is most suitable for this thesis. There is stated a problem and a vision for what the system should do. Based on this there should be designed and implemented a prototype system to solve it. The system will be tested by end-users who will evaluate the accuracy of the prototype system.

1.4 Approach

The system designed and implemented in this thesis automatically use information from a posting on a social media site to find a specific user on multiple other social media. Twitter is used as a starting point where the end-user can choose a tweet of interest, press the tweet and automatically be presented with information about the author of the tweet, including profiles on other social media.

The system is designed given a set of functional requirements such as integrating the application with an existing social media application, retrieving data associated with a given user, identifying a user across multiple social media, and presenting the results. The approach for identifying users includes gathering profile attributes and presenting them to the end-user. Other non-functional requirements such as usability, security and privacy, performance and coupling have been considered throughout the entire process.

1.5 Contribution

To our knowledge there is no related work in enabling automatic user-based information search across multiple social media, without relying on explicit user account registration as a basis for combining information. The contribution of this work is the creation of a system that allows the user to choose a posting on a social media site, and then the system will use the username and full name of the posts author to perform a search across multiple social media. The results will then automatically be presented to the user through a web interface.

1.6 Organization

This thesis is structured as follows:
Chapter 2 presents relevant background information for the thesis.
Chapter 3 specifies the requirements of the system.
Chapter 4 presents the architecture and design of the prototype system.
Chapter 5 describes implementation details of the prototype system.
Chapter 6 presents test cases and results.

Chapter 7 discusses and evaluates the system.

Chapter 8 presents the conclusion.

2 Background

This thesis focuses on search across multiple social media applications whereas this chapter will present some background related to social media and social media applications. It will also present some background concerning tools and methods used in this thesis. Finally some related work will be presented.

2.1 Social Media

The term social media refers to a range of services, both mobile and internet-based, that allow users to socially interact with people in which they create, share or exchange information and ideas with, in virtual communities or networks. Kaplan and Haenlein [2] have defined social media as "a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content".

2.1.1 Twitter

Twitter is a microblogging service where the blog posts are referred to as tweets with a maximum length of 140 characters. All twitter users have a personal profile that contains some information about the user, and they form relationships between each other through following. Users can follow anyone, and there is no requirement to follow back. The follower will receive new tweets from the person they follow on their homepage. Twitter is built up of user created content in the form of tweets. People can use a hashtag, "#", to highlight a word in their tweet. This allows search to find tweets concerning a specific topic. Twitter also uses hashtags or most frequently used words in tweets to show what´s trending on twitter. Hashtags are treated as topics while searching, allowing users to find all microblog posts containing the tag. [3]

In this thesis Twitter will be used as a starting point of any search. A user will click an interesting tweet, and the system will gather information from this tweet and use it in a search across multiple other social media applications.

2.1.2 Facebook

Facebook is an online social networking site where users can register to connect with friends and colleagues or follow the pages of a business or famous people. To be able to use Facebook, users have to register and create their own personal profile. Once a user is registered, she can add other users as friends, publish status updates, photos and videos, and exchange messages with other users and chat with her friends. Users are also allowed to comment or like information shared by their friends or public users. Facebook also support the creation of event pages or groups, and any user can create such an event or group that is either public or private. Facebook have also added support for hashtags in posts where each hashtag is given a link that the users can press to view all other posts containing that same hashtag. [4]

2.1.3 Flickr

Flickr is a content community in the form of a website for hosting images and videos. Flickr can be viewed as a free service for bloggers who wish to post photos where they can get feedback and recognition from other Flickr users. Once registered, Flickr users can vote on photos, add them to their favorite list and add comments. Users communicate with each other through comments on photos, or they can add contacts to their list of friends and send FlickrMail to each other. When a user uploads photos, she provides information such as title, description and tags. Users also have the possibility of creating albums and groups for their photos. Albums are displayed on the users profile, and only contain photos from that user, while groups contain photos from multiple users. [5]

2.1.4 Instagram

Instagram is a content community that is an online application for mobile devices. It is a service that allow photo- and video-sharing. The users of this application have to register and create a basic profile. Once they are registered they can share pictures and videos on a variety of social networking platforms, including Facebook, Twitter, Tumblr and Flickr. Instagram uses hashtags for easily discovering photos and people. Like twitter, relationships between users are formed through following. Users can follow anyone, and there is no

requirement to follow back, although some users have private profiles where you have to send a request, which they must approve in order to follow them. Users interact or communicate through comment and likes on the photos or videos. [6]

2.2 Web Technologies

2.2.1 HTML

HyperText Markup Language (HTML) is a standard markup language used for creating web pages. Markup languages are sets of markup tags. An HTML document is written using HTML elements consisting of markup tags. Markup tags are used to describe different content in the document such as paragraphs, lists, etc. They are enclosed in angle brackets "<html>" and they mostly come in pairs " " and " ", where the fist marks the start and the last marks the end of a content section. [7]

2.2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used to define the look and formatting of a document written in a markup language such as HTML. It is used for creating great visual experiences on websites, mobile applications, webpages, user interfaces, and web applications. CSS was designed to separate the content and presentation of a document. [8]

2.2.3 JavaScript

JavaScript is a scripting language that runs on a browser and helps to improve the functionality and user experience. It allows creation of client-side functionality of a web site. It can be viewed as the programming language of HTML, as it has the ability to change HTML content. [9]

2.3 Tools

This section describes tools and methods that have been used in the implementation part of this thesis.

2.3.1 APIs

Many web applications provide an application-programming interface (API), which is a programming recipe on how to access the application. It is a set of routines, protocols, and tools for building software applications. It is a software-to-software interface that allows applications to talk together. By providing a public API, applications allow developers to design products and applications that utilize the services accessible exposed through the API.

2.3.1.1 Twitter API

Twitters API[1] allows other web services and applications to integrate with Twitter. They use a REST API that allows programmatic access to read and write Twitter data, and to issue queries against the indices of recent or popular tweets. The API can be used to search for users by their user ID or their screen name. Table 1 lists a request and response to and from the Twitter API. The request is to show information about a user based on username.

Table 1: Twitter API request and response

Request	https://api.twitter.com/1.1/users/show.json?screen_name=rsarver

Response	{
	"name": "Ryan Sarver",
	"profile_image_url":
	"http://a0.twimg.com/profile_images/1777569006/image1327396628_normal.png",
	"location": "San Francisco, CA",
	"url": null,
	"description": "Director, Platform at Twitter. Detroit and Boston export. Foodie and over-the-hill hockey player. @devon's lesser half",
	"statuses_count": 13728,
	"friends_count": 1780,
	"screen_name": "rsarver"
	…
	}

[1] https://dev.twitter.com/rest/public

2.3.1.2 Facebook API

Facebook uses a social graph to represent object and connections on Facebook. Objects include people, photos, events, pages, etc., and connections include friendships, shared content, photo tags, etc. Facebook provide a Graph API[2] that allows other applications to integrate with their open social graph to read from and write data to Facebook. Their Graph API is a low-level HTTP-based API that allows other applications to issue queries, post new stories, upload photos, and perform other tasks on Facebook. The API allows search for people based on their names. Table 2 lists requests and responses to and from Facebook's graph API. The first request is a search for users by name, and the second request is to show information about a user based on the user id.

Table 2: Facebook API request and response

Request	https://graph.facebook.com/v2.2/search? type=user&q='Ola Norman'
Response	{ "data": [{"name": "Ola Norman", "id": "946295895410273"}, {"name": "Ola Kari Norman", "id": "768011266652780"}, {"name": "Ola Norman", "id": "10206421464676506"}, …], "paging": { "next": "https://graph.facebook.com/v2.3/search?type=user&q='Ola Norman'&limit=25&offset=25&__after_id=enc_AdAP67HJXAcr30T2ZBHrTZCk4tJ2Dk0Tl0np8vzLQUfJsIcZA21bnGBZAMQzJLYQXiFC72b6ZCnOTkNtgLbmfZByTzTNur" } }
Request	https://graph.facebook.com/v2.2/946295895410273
Response	{"id": "946295895410273",

[2] https://developers.facebook.com/docs/graph-api

	"first_name": "Ola", "last_name": "Norman", "link": "https://www.facebook.com/app_scoped_user_id/946295895410273/ ", "name": "Ola Norman", "updated_time": "2012-06-07T08:15:10+0000"}

2.3.1.3 Flickr API

Flickr provides an API[3] that allows other application to read and write data from Flickr. This may include displaying photos from specific users or groups, viewing, manipulating, and searching for photo tags, getting photos from a specific location etc. They allow different request and reply formats including REST, XML-RPC, SOAP, JSON and PHP. The API allows users to be searched by email or username. Table 3 lists requests and responses to and from The Flickr API. The first request is a search for users by username, and the second request is to show information about a user based on the user id.

Table 3: Flickr API request and response

Request	http://api.flickr.com/services/rest/?&method=flickr.people.findByUsername&username=Stewart
Response	<user nsid="12037949632@N01"> <username>Stewart</username> </user>
Request	http://api.flickr.com/services/rest/?&method=flickr.people.getInfo&user_id=nsid
Response	<person nsid="12037949754@N01" ispro="0" iconserver="122" iconfarm="1"> <username>bees</username> <realname>Cal Henderson</realname> <mbox_sha1sum>eea6cd28e3d0003ab51b0058a684d94980b727ac</mbox_sha1sum> <location>Vancouver, Canada</location> <photosurl>http://www.flickr.com/photos/bees/</photosurl>

[3] https://www.flickr.com/services/api/

	<profileurl>http://www.flickr.com/people/bees/</profileurl> <photos> <firstdate>1071510391</firstdate> <firstdatetaken>1900-09-02 09:11:24</firstdatetaken> <count>449</count> </photos></person>

2.3.1.4 Instagram API

Instagram provides an API[4] that allows other applications to pull photos from Instagram and display them in their own application. The API can be used to search tags, view photos from specific locations, pull popular or trending photos etc. The API allow search for users by name. Table 4 lists a request and response to and from the Instagram API. The request is a search for users by username.

Table 4: Instagram API request and response

Request	https://api.instagram.com/v1/users/search?q=jack

Response	{
	"data": [{
	"username": "jack",
	"first_name": "Jack",
	"profile_picture":
	"http://distillery.s3.amazonaws.com/profiles/profile_66_75sq.
	jpg",
	"id": "66",
	"last_name": "Dorsey"
	},
	{
	"username": "sammyjack",
	"first_name": "Sammy",
	"profile_picture":
	"http://distillery.s3.amazonaws.com/profiles/profile_29648_75
	sq_1294520029.jpg",
	"id": "29648",
	"last_name": "Jack"
	}]
	}

[4] http://instagram.com/developer/

2.3.2 OAuth

For authentication most applications use OAuth, which is an open standard for authentication. The resource owners use OAuth in order to grant access to third-party client application without sharing their credentials. Once a client application is authenticated they are given an access token that is used when issuing queries to the resource owners APIs. Users only have to request the access token once. When the access token is granted it is valid until it expires. Once a token is expired, the user will have to request a new access token. The lifetime of an access token varies, and it can expire in hours, months or it can be permanent. Information about he lifetime of the access token is usually given when it is granted.

2.3.3 REST

REST, Representational State Transfer, is a style of architecture based on interaction between client and server. Besides client/server, the other formal constraint of the rest protocol is that it is stateless, layered, and supports caching. It usually runs over HTTP for communication. Clients make requests to servers, using the HTTP operations GET, PUT, DELETE and POST, and the server's processes the requests and returns appropriate responses.

2.3.4 Bookmarklets

Bookmarklets [10] are combinations of bookmarks and applets. A bookmark is a Uniform Resource Identifier (URI) that is stored in the browser in order to retrieve it later. An applet is a small program that usually runs as a plug-in within a larger program and performs a specific task. Bookmarklets are smart bookmarks that you can add to your bookmarks bar, and with a single click, it will perform actions or enhance the website you are on. Bookmarklets are usually JavaScript programs that are stored as the URL of a bookmark in a web browser, or as a hyperlink on a web page.

2.3.5 Bootstrap

Bootstrap is a front-end-framework for HTML, CSS and JavaScript, and it is free and open-source. It contains a collection of tools that are used for developing web applications. Bootstrap is designed for faster and easier front-end web development, and it is hosted on GitHub. The framework works on devices of different sizes from personal computers to smartphones. They have a responsive layout that scales to the size of the screen. [11]

2.4 Information retrieval

Within the field of computer science, Manning et. Al. [12] has given a definition of Information retrieval (IR):
"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."
On a daily basis people engage in information retrieval when they use IR applications such as a web search engine or when they search their email. The activity is initiated when a user enters a query into the IR system. The query states the information need, and the system will present objects that match the query.

When using Information retrieval, a common goal is to index the documents for efficient retrieval. In order to achieve an index as presented by Manning et. al., a few normalization techniques must be used on the documents including:

Tokenizing: meaning that the individual words must be identified.
Stopword removal: meaning that common words such as "and", "of", "the", etc. must be eliminated.
Stemming: meaning that words will be reduced to a common root.

After these techniques are used, the documents are ready for indexing. There are differed models for building index structures, but the most common is inverted index file structure.

Once there is a set of index terms for the documents, they are usually weighted according to their relevance to different search queries. The reason for this is that not all terms are of equal relevance in describing the documents content.
There are different approaches for assigning weights to index terms and measure their relevance to different search queries. The most common models are Boolean, vector and probabilistic.

In this thesis IR is used to collect information about users. All useful information from a given tweet is used to issue queries to the different social media applications using their APIs. Meaning that this thesis will issue queries to the different social media applications Information Retrieval systems.

2.5 Human-Computer Interaction

Human-computer interaction [13] is the study of the interaction between humans and computers. The goal for this study is to improve the interactions by making computers more usable and receptive to the users needs. Important elements of this study involves the design of both hardware and software, that results in a "product" where factors like aesthetics, usability, ergonomics, cognitive engineering, design, psychology and sociology plays a big role in how the end user perceives the interaction.
In HCI there are guidelines [13] for designing web pages, the most important are:

- The page should be readable and there should be a focus on the presentation of information, meaning that there should be a clean visual structure.
- There should be appropriate use of color, meaning that the number of main colors should be limited and important information should be highlighted. In addition the colors chosen should be compatible with each other.

In this thesis HCI guidelines will be used in the design of the web interface. There will be a focus on page design where the goal is to present the data in a

readable and neat structure, with colors that are pleasing to look at and that complement each other as well as highlighting important elements. This will be discussed further in chapter 4.2.1.

2.6 User Experience

User experience, UX, is a term that refers to a users overall experience when using a product, service or system. The term originates from Human-computer interaction. A goal for UX design is to understand the users, their needs, abilities, and limitations. A key goal for UX designers is to ensure that users find value in what they are providing. Peter Morville represents this through a User Experience Honeycomb. [14]

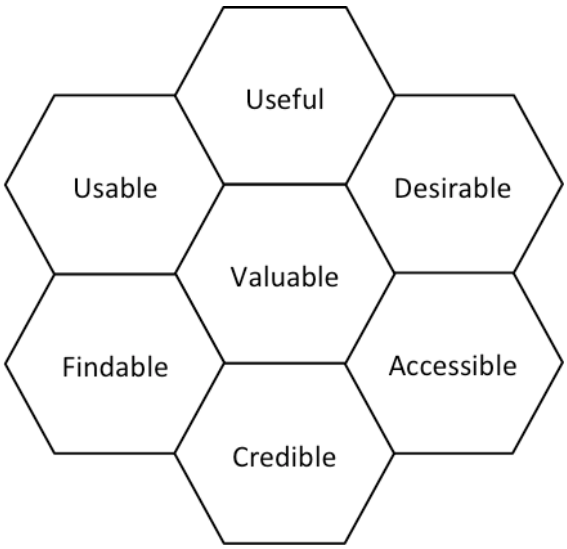


Figure 1: User Experience Honeycomb

In order to create a meaningful and valuable user experience the information must contain the qualities presented in Figure 1. It must be **useful** in that the content should be original and fulfill a need. It must be **usable** in that the application must be easy to use. The application should be **desirable**, which is achieved by using design elements such as image, identity, brand, and others are used to evoke emotion and appreciation. It should be **findable**, which means that the content needs to be navigable and locatable, in order for users to find what they need. The application and all of its content must be **accessible** to all. The information presented must be **credible**, meaning that users must

trust and believe what the application tells them. It must be **valuable** in that it must deliver value to the end-users. [14]

2.7 Related Work

Many individuals register on social media application, and many are registered to more than one. It has become difficult to locate friends, former colleagues, peers and acquaintances. One reason it is hard to locate friends is that there can be name collisions, meaning that people can have the same name and it might be difficult to determine which is correct if there are no additional information. Another reason is that people use different aliases or usernames. If a person knows one alias that a friend uses and is looking for the friend on another social media application using this alias, the person might not be able to find the friend if she has chosen to use a different alias on this site. People have the ability to choose which profile attributes they add to their profiles. Some might choose not to add any additional information but their name or username, which make it harder for their friends to discover them.

A big part of this thesis is trying to solve this problem by identifying users across multiple social media applications. Many others have also addressed the problem; this section presents a few of them and what approaches they have chosen when addressing the problem.

Motoyama et. al. [15] developed a system for searching and matching individuals in online social networks. They extracted attributes from a profile in one Online Social Network, which they used when searching on other Online Social Networks.

Jain et. al. [16] present two search algorithms for identifying users. They base their algorithms on content and network attributes in order to improve on traditional identity search algorithms that are based on the profile attributes of a user. Given a users identity on Twitter, they try to find the users identity on Facebook. They divide their matching process into two steps. The first step is to find a set of possible candidates, and the second step is to match the candidates to the given individual based on the usernames.

Cheng et. al. [17] base their approach on user-profile comparison but expand by considering additional social relationships, i.e., groups of friends. They propose a two-phase clustering algorithm where the first phase is to select strongly connected groups as seeds by removing critical nodes iteratively, and the second phase is to assign other nodes to the cluster based on social structure and profile.

Zafarani et. al. [18] present a methodology for finding a mapping among identities of users across social media sites. Their approach consists of three components: the first component identifies a users unique behavioral pattern that leads to redundancies across sites; the second constructs features that exploit these redundancies; and the third employs machine learning for effectively identifying the user.

Iofciu et. al. [19] present a solution for identifying users across social tagging systems. They examine user profiles from Flickr, Delicious and StumbleUpon where they exploit tagging behavior and lightweight profile information such as usernames. They present different approaches where they match users based on tags, username and both, and they also use aggregated user profiles in one of their approaches.

Nie et. al. [20] present a method to relate users identities across social media sites by mining users behavior information and features. Their method has two components: the first component distinguishes different users by analyzing their common social network behaviors and finding strong opposing characters, while the second component constructs a model of behavior features that helps to obtain the difference of users across social media sites.

In [15] and [16] they exploit profile attributes when identifying users, while in [17] they use profile comparison and expand by considering social relationships. In [18] and [20] they base their user identification on behavioral modeling and in [19] they exploit tagging behavior when identifying users.

The approach presented in this thesis is most similar to the approaches of [15] and [16], where they make use of profile attributes in order to identify users. This approach will be discussed further in chapter 4.3.2.

Most of the related work presented in this section only identify users across two ([15], [16], [17]) or three ([19]) social media applications, while in [18] and [20] they do not specify which social media sites they use. The system presented in this thesis uses one social media site as a starting point and identify the user on three other social media sites. One thing the system does differently than the others is that it complements the identity search by using Google.

3 Requirement Specification

In order to design and implement the application there must be outlined a system model. This chapter will present an overview of the system and its components, as well as functional and non-functional requirements.

3.1 Overview

The vision for the system is that a user on one social media site should be able to find more information about the author of a post on that site on multiple other social media sites.

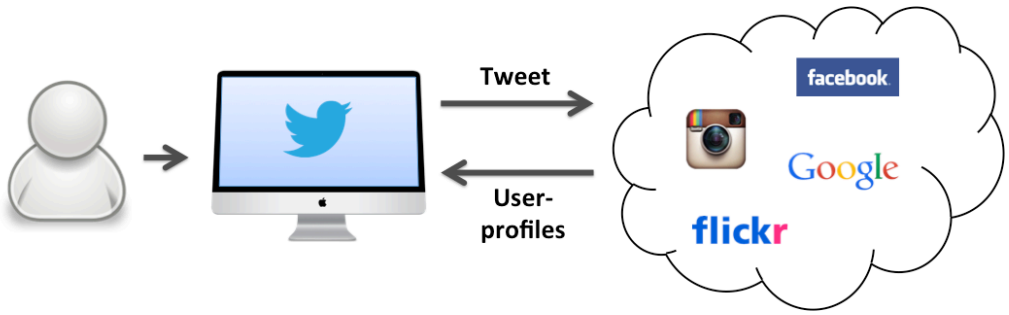


Figure 2: High-level overview of the system.

The idea is that if a user reads an interesting tweet and wants to find more postings from the same user on some other social media, she should be able to click the tweet and the information should automatically be presented to her. A high-level overview of the system is illustrated in figure 2. The vision is that the application will function as a tool that is embedded on an existing social media site in order to obtain more information about a specific user from other social media sites. A main objective for the application is to be as easy as possible to use. Anyone should be able to use the application, and therefore the system needs to have a neat and simple user interface, and all features must be understandable and easy to use.

3.2 Functional Requirements

3.2.1 Integration with existing social media applications

Since the vision is that an end-user should be able to click a post on a social media site and automatically be presented with postings from the same or other users on other social media sites, the systems needs to be federated with existing social media sites.

3.2.2 Retrieval of data associated with the user

In order to search for postings by the same or other users, the system needs to gather all available information associated with the post of interest. This includes information about the person that posted such as name, username, location etc., and hashtags from the post in order to identify the topic.

3.2.3 User Identification

In order to find postings by the same user on other social media, the user needs be identified on other social media sites. The information retrieved about the user should be used in order to identify the user.

3.2.4 Result presentation

For presenting the results to the end-user a good solution is to implement a web interface. The interface will need to present data from different social media applications at the same time. The presentation of the results must be in a way that is clear and understandable for the user.

3.3 Non-Functional Requirements

3.3.1 Usability

The end-users can be anybody who uses social media meaning that the technical expertise of the end-users may vary. Most end-users will probably not have any technical background, and therefore it is important that the system is easy to use and the interface must have an intuitive layout.

3.3.2 Security and Privacy

When searching through user generated data in the form of posts and personal information about the user, security and privacy must be taken into account. The system should only use publicly available information that can be accessed through the social media applications APIs. No information should be stored or cached by the system itself.

3.3.3 Loose Coupling

The system should be loosely coupled in that the components should have little or no knowledge of the definitions of the other components other than APIs defined between them. The system might need to add, remove or replace components. One example might be to add support for a new social media application. Another advantage by having loose coupling between the components it allows for the system to serve multiple front-ends from the same backend.

4 Architecture and Design

This chapter will present the architecture and design of the application based on the requirements outlined in chapter 3.

4.1 System Architecture

The system architecture is based on the overview and requirements outlined in chapter 3. The system is divided into three layers, a presentation layer, federation layer and a data layer that consist of different components. The presentation layer interacts with the user and communicates with the federation layer. The federation layer communicates with both the presentation and data layer. The architecture for the system is illustrated in figure 3.

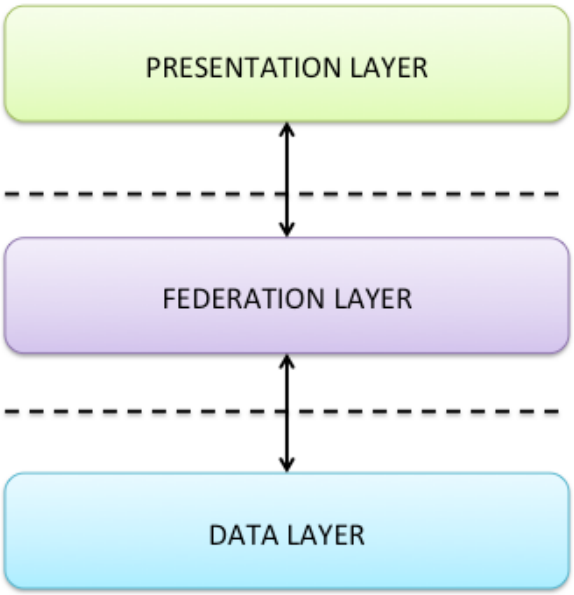


Figure 3: Layered Architecture of the system

4.2 Presentation Layer

This section describes the design of the presentation layer including how the system is integrated with existing social media pages and the presentation of the web interface. The presentation layer is illustrated in figure 4.

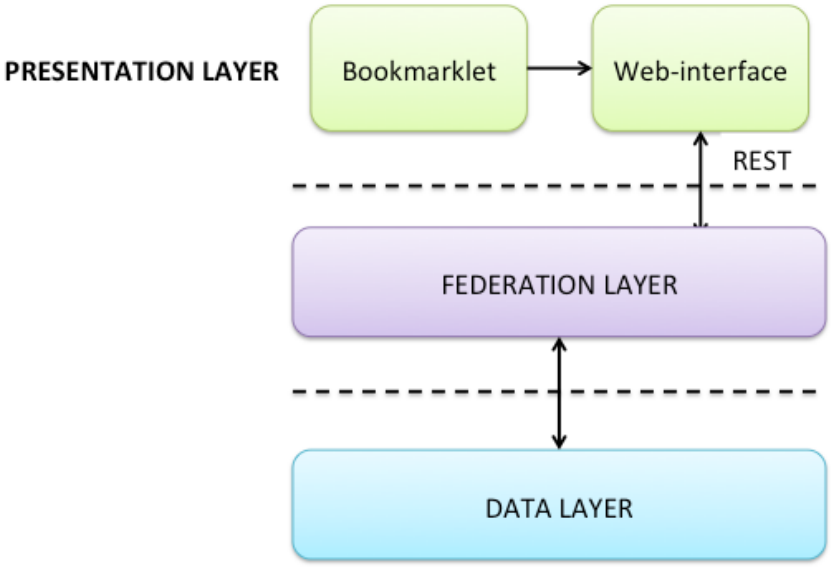


Figure 4: Presentation Layer

Bookmarklet:

The application is integrated with twitter through a bookmarklet, which allows the user to choose an interesting tweet. The information from this tweet will be passed on to the web interface, which will further handle it.

Web Interface:

The web interface is communicating with the bookmarklet and the federation layer. It receives information about the post from the social media application, which it will use to query the federation layer. Later the web interface will receive the results from this query and present them to the user.

4.2.1 Integration to existing pages

Options for integrating the program on existing web pages includes bookmarklets, web applications, browser plug-ins and extensions[5].

Bookmarklets can be shared with multiple web browsers, while web applications, browser plug-ins and extensions are made for specific browsers and are not compatible with others. In addition, plug-ins and extensions will have to be installed on the browser which is a drawback since the more that is

[5] http://colonelpanic.net/2010/08/browser-plugins-vs-extensions-the-difference/

installed on the browser, the more memory the browser will have to use, and tings will become slower. Another advantage with bookmarklets is that they are easy for users to install, they just need to drag them to the bookmarks bar, while extensions and web applications must be downloaded and the browser may require a restart. For this thesis bookmarklets seems to be the way to go since they are compatible with multiple browsers, and easy to install. One drawback with bookmarklets is that they are not widely adopted and may be unfamiliar to many.

As mentioned in chapter 2 bookmarklets can perform actions or enhance the website you are on. The vision for the system is that a user should be able to click a tweet and the automatically be presented with the search results. Different options on how to achieve this were explored. Two solutions were chosen as the best options, and they both consist of 3 steps as shown in figure 5 on the next page.

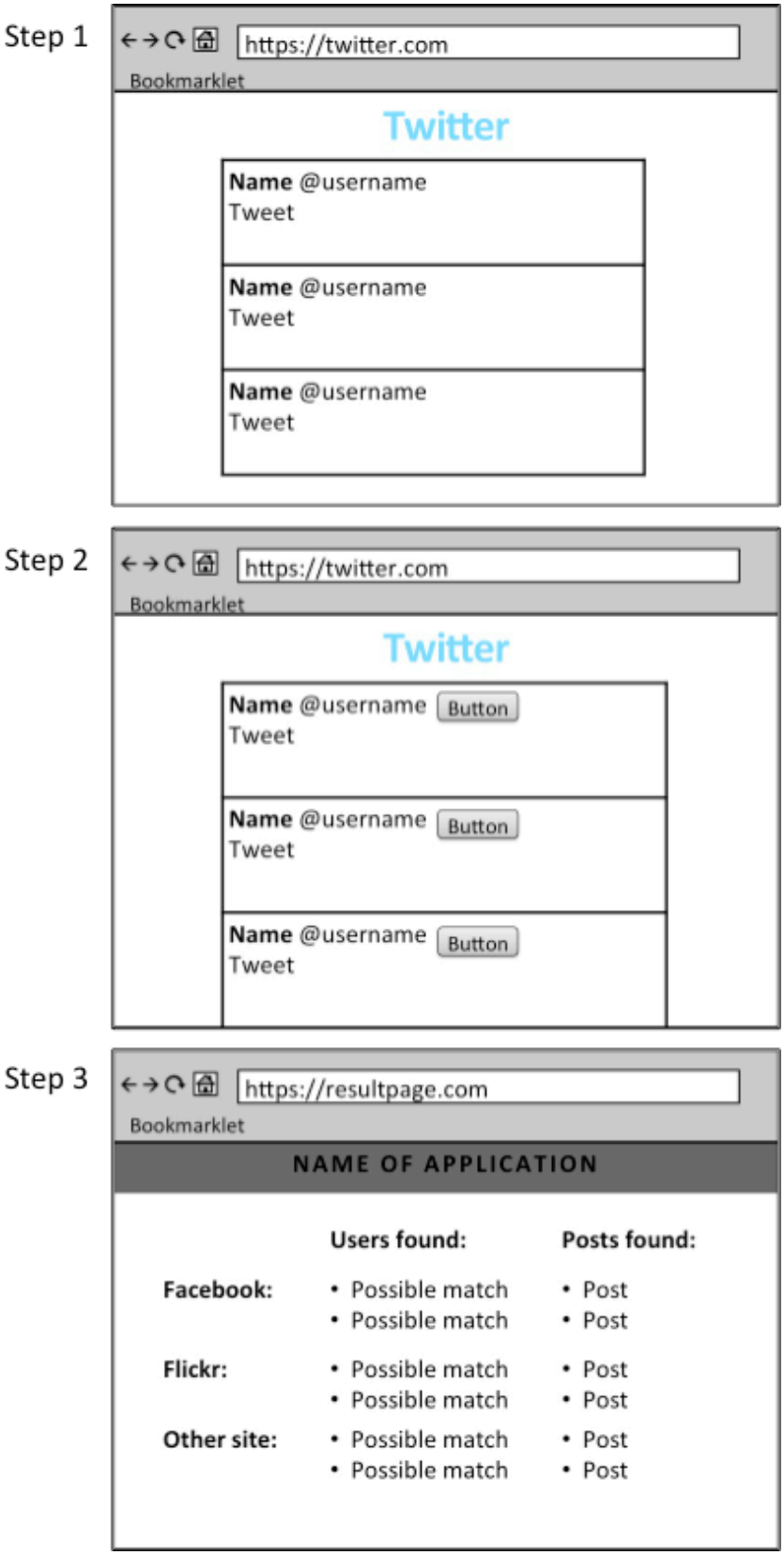


Figure 5: Design option for bookmarklet

First option:

1. User clicks the bookmarklet.
2. Links will appear on the tweets displayed on the existing page. The user clicks the link on the correct tweet and is redirected to a page.
3. The new page displays the search results.

Second option:

1. User clicks the bookmarklet and is redirected to a new page.
2. The new page displays the tweets. The user clicks the correct tweet and is redirected to a page.
3. The new page displays the search results.

Both options include a first step where the user needs to click the bookmarklet and a last step where the user is presented with the results. The difference between the two options is step 2. In the first option the user stays on the existing page in the same experience, and links are added to the tweets so the user can click the tweet of interest. In the second option, the user is redirected to a new page where all the tweets from the previous page are displayed, and the user can click the tweet of interest. Optionally in this scenario the existing page can be stripped and the tweets can be displayed on this page, which will give the same result without redirecting the user, but requires major changes to the existing page.

The first option was chosen as the best solution for this thesis since a common perception is that the user experience will be better with small changes instead of big ones. User experiences might also be better when there is no need for redirecting the user twice. Figure 4, illustrates the 3 steps of this solution.

4.2.2 Web interface

The application should have a web interface containing two different pages; one page being the application homepage and the other page should be the result page. Users should be able to visit the application homepage in order to get familiar with the application and to download the bookmarklet by dragging it to the bookmarks bar.

They should be redirected to the result page after using the bookmarklet. The result page will contact the federation layer with the query and display the results to the user.

According to HCI and UX the appearance of a web interface is important and it should contain good use of colors, the text should be easily read and it should be kept simple. With this in mind the color scheme chosen should only contain a few primary colors that complement each other. It is important not to use colors that are conspicuous and will distract the user from the content of the site. The text on the site should have text and background color with an appropriate contrast and the font should be easy to read and compatible with most browsers. In addition the font size should be kept within a standard range. The site should be simple with a tidy layout with appropriate white spaces that allows the user to focus on the content of the site. Both pages in the web interface should use the same color scheme and layout.

In order to achieve a neat and responsive layout, the Bootstrap CSS framework is used. Bootstrap makes it easy to add responsive components to the site, meaning that they resize and scale to fit the size of the screen.

The basic layout of the site consists of four components:

1. A navigation bar that is at the top of the page with the option of going to the applications home page or an about page containing information about the site.
2. A Jumbotron. Jumbotron usually refers to a large-screen television, but in this case it is a large header that will contain the application name, and is placed at the top of the site below the navigation bar.
3. A sidebar column placed to the left.
4. A content column placed to the right.

The two columns are placed under the Jumbotron side by side. The sidebar column to the left contains the bookmarklet and the content column to the right contains the rest of the content on the site, which varies from which page it is on.

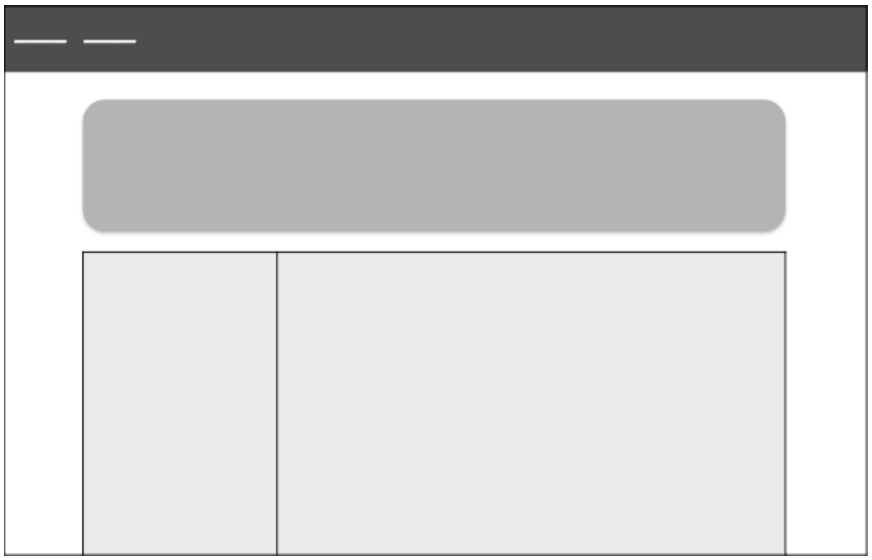


Figure 6: Layout of the home page.

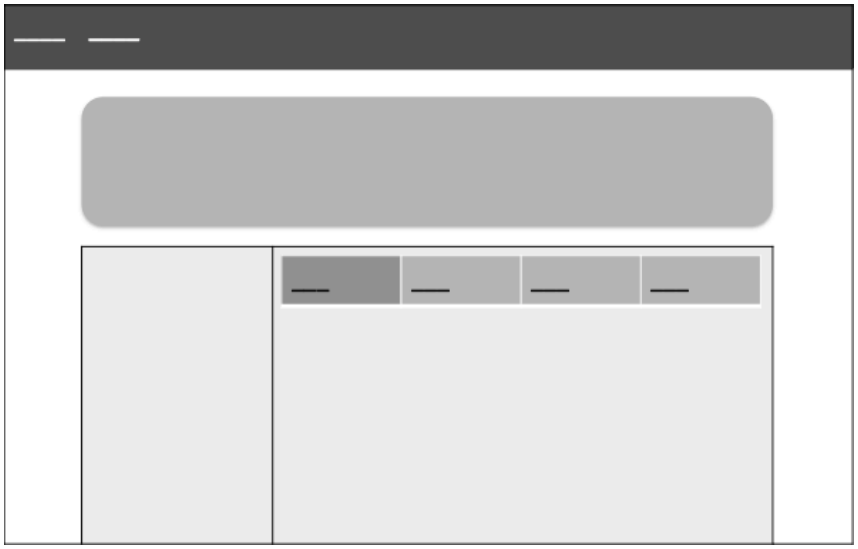


Figure 7: Layout of the result page.

At the home page illustrated in figure 6, the right column will contain information about the bookmarklet, including how to install and use it. At the result page illustrated in figure 7, the right column will contain a navigation bar where the user can choose which results to show. The results come from the different social media applications and Google. The results can contain photos, basic information and links to the URL that are found.

4.3 Federation and Data Layer

This section describes the design of the federation layer components including communication with the presentation and data layer, and the identification of users. The federation layer is illustrated in figure 8.

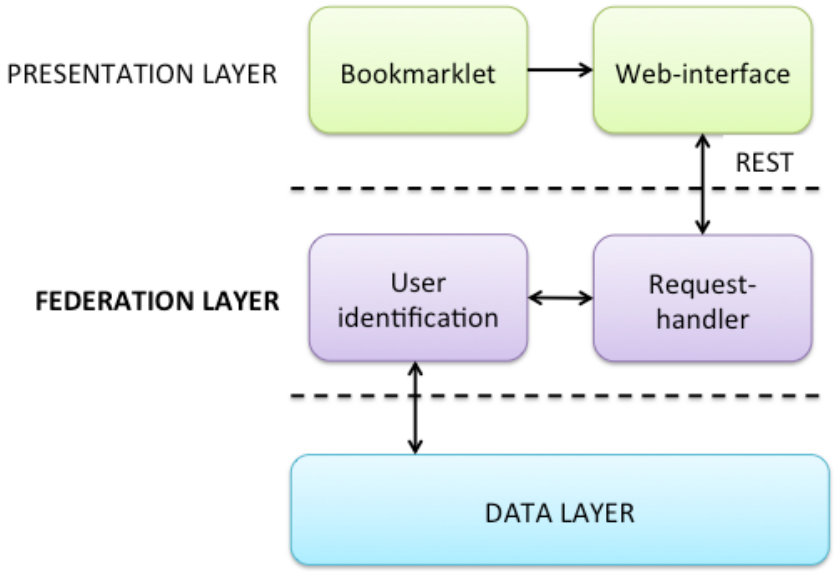


Figure 8: Federation Layer

4.3.1 Request Handler

The Request Handler receives request from the web interface through a RESTful interface. It will then contact the user identification component and pass along the request, wait for it to return any results and then pass the result on to the web interface.

A simple RESTful interface is used for communication between the Request Handler and the presentation layer. It will support a GET method, which returns all data retrieved from the user identification component. Reasons why a RESTful communication interface was chosen, is because of its stateless nature. The system should receive requests, retrieve data and return the results in real time, meaning that the data should be returned immediately. Therefore there is no need to store any results, and the RESTful service can go down and restart without causing any damage to the system.

4.3.2 Identifying users

The identification component uses the username provided in the query from the Request Handler, and it will use Information Retrieval by issuing queries against the social media applications API in order to gather additional information about the tweets author. Information that is found may include the full name of the user, which along with the username is used to search for people on other social media. The identification component should find a list of possible candidates and try to narrow it down in order to find the correct person.

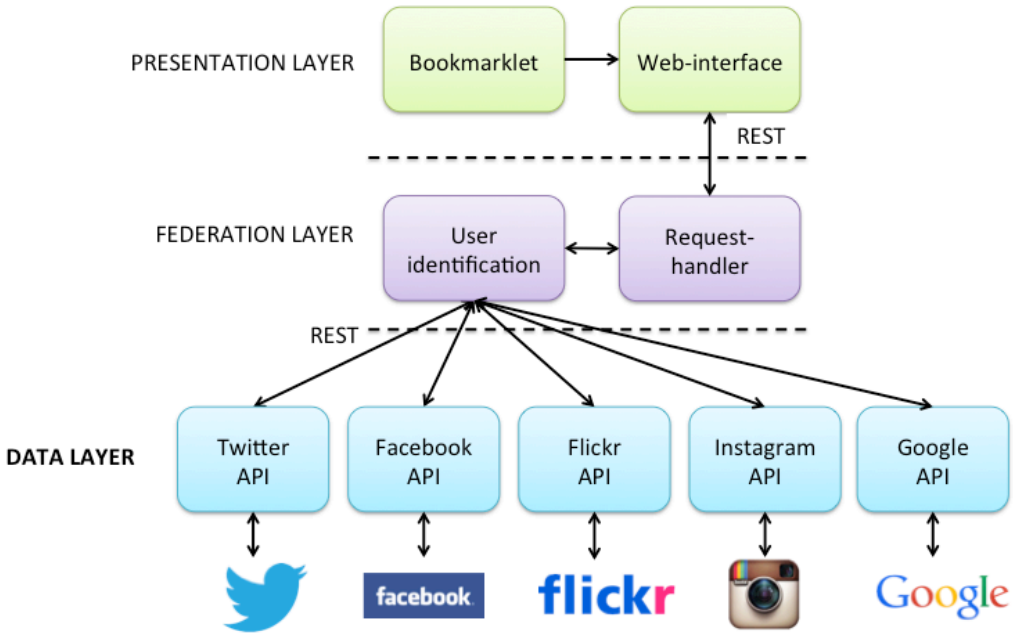


Figure 9: Data Layer

For the system to be able to gather information and identify users the user identification component should communicate with the data layer, which is illustrated in figure 9. The data layer consists of multiple social media applications public APIs, which communicate with the social media applications. In order to communicate with the social media applications APIs, the system must be authenticated by the different social media applications. Most of these applications use OAuth in order to authenticate third-party client applications. For the client application to be authenticated, it must send a request to the resource application through OAuth. Once the client application

is authenticated, there will be granted an access token that will be used when issuing queries against resource applications APIs.

Table 5: Profile attributes available/searchable through APIs.

	Twitter	Facebook	Flickr	Instagram
Name	Yes	Yes	Yes	Yes
Username	Yes	No	Yes	Yes
Email	No	No	Yes	No
Gender	No	No	No	No
Age	No	No	No	No
Location	Yes	No	Yes	No
Profile picture	Yes	Yes	Yes	Yes

The user identification component should use IR to gather as much information as possible about a user from each of the social media applications. Profile attributes that are available or can be searched through the APIs are listed in table 5.

Information available through the applications APIs vary from application to application and it might be tricky to compare information in order to identify a user. An example might be that you only have the name and picture from one application, and since many people may have a common name, without any other attributes it can be difficult to narrow it down to the right person.
Additional information available on Twitter, if the user provides it, includes a description the user has given about herself and URLs the user has posted to her profile. A description, if given by the user, is also available on Flickr and Instagram.

5 Implementation

This chapter explains the implementation details of the core functionalities of the presentation and federation layer of the prototype system.

5.1 Presentation Layer

The presentation layer of the system consists of a bookmarklet and a Graphical User interface (GUI) in the form of a web interface.

5.1.1 Bookmarklet

The bookmarklet is a JavaScript script block that runs it the browser. It has the ability to inject code into the web page, hence change the appearance and behavior of the site the user is currently on.

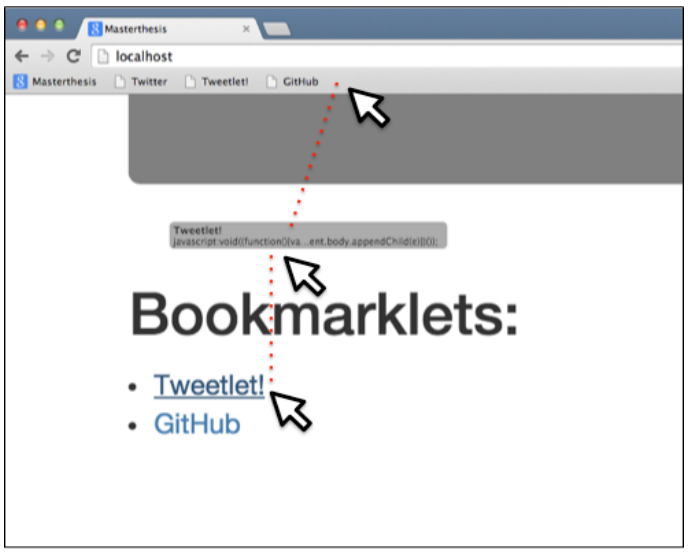


Figure 10: Installing the bookmarklet.

The bookmarklet has to be installed by the user, which is done simply by dragging it to the bookmarks bar in the browser, which is illustrated in figure 10.

Table 6: Edited Twitter HTML.

```
<div 1.class="stream-item-header">
      <a class="account-group js-account-group js-action-profile js-
      user-profile-link     js-nav"     href="/2sporten"     data-user-
      id="20976857">
            <img        class="avatar        js-action-profile-avatar
            hoverZoomLink"
            src="https://pbs.twimg.com/profile_images/3281726687/c1cb
            5c2b44665111adc9f7e5809c30cc_bigger.png" alt="">
            <strong   class="fullname   js-action-profile-name   show-
            popup-with-id"        data-aria-label-part="">TV        2
            Sporten</strong>
            <span></span><span   2.class="username   js-action-
            profile-name"                          data-aria-label-
            part=""><s>@</s><b>2sporten</b></span>
      </a>
      <small class="time">
            <a             href="/2sporten/status/595841703220801537"
            class="tweet-timestamp     js-permalink     js-nav     js-
            tooltip"><span  class="_timestamp  js-short-timestamp  js-
            relative-timestamp"   data-time="1430894706"   data-time-
            ms="1430894706000"       data-long-form="true"       aria-
            hidden="true">43m</span><span     class="u-hiddenVisually"
            data-aria-label-part="last">43 minutes ago</span></a>
      </small>
      <button    type="button"    class="btn-link    js-translate-tweet
      translate-button expand-stream-item" data-nav="translate_tweet">
      <span    class="translate-label">View    translation</span><span
      class="Icon Icon--translator"></span></button>
      3.<a href=
"http://localhost/result.html?username=2sporten&amp;source=twi
tter"> Click me!</a>
</div>
```

Table 6 shows the edited HTML of a tweets header containing all HTML elements of the header including their IDs. The bookmarklet will single out each different post on the site by fetching the HTML elements by class ID. It then adds a link to the HTML elements and updates the site. It also fetches the HTML element containing the username of the posts author and store it temporarily. There are some highlighted and numbered sections in table 6:

1. The header class ID that is used to find the area where the username is, and where the link should be placed.
2. Class ID of the username that is used to retrieve the username and send it as an embedded query string in the URL.
3. HTML that is added to the site by the bookmarklet in order to create the link that redirects the user to the systems web interface.

Figure 11: Tweet with highlighted header.

Figure 12: Tweet with link added to header.

Figure 11 shows an example tweet where the header is highlighted, and figure 12 shows the same tweet with the link added to the header. The user can click the link and is redirected to the applications result page of the web interface where she is presented with the results of the search.

5.1.2 Web Interface

The structure of the web interface is implemented using HyperText Markup Language (HTML), while the visual layout of the page is implemented using Bootstraps Cascading Style Sheets (CSS) framework.

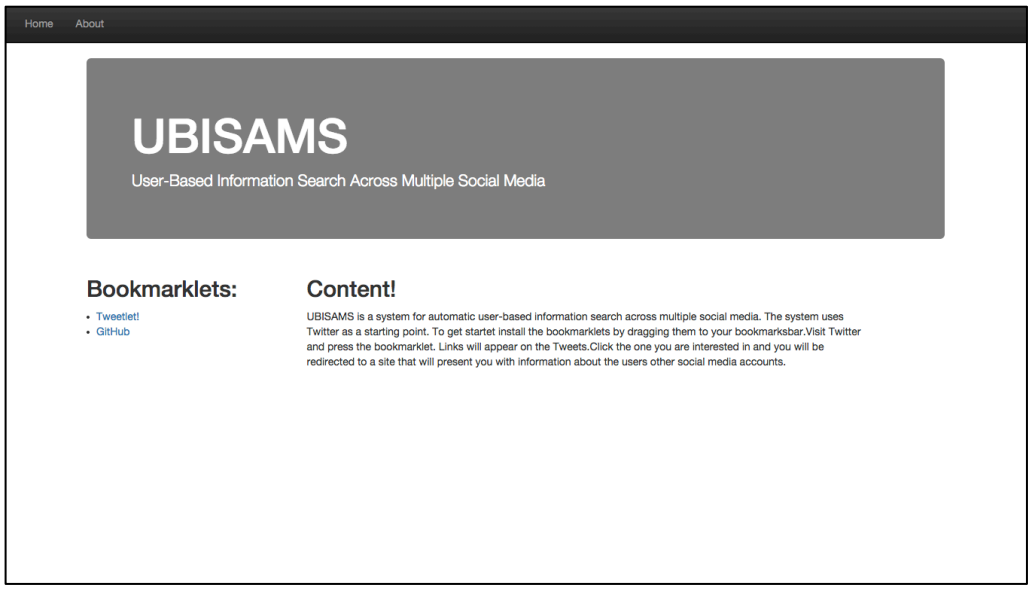


Figure 13: Screenshot of the applications homepage.

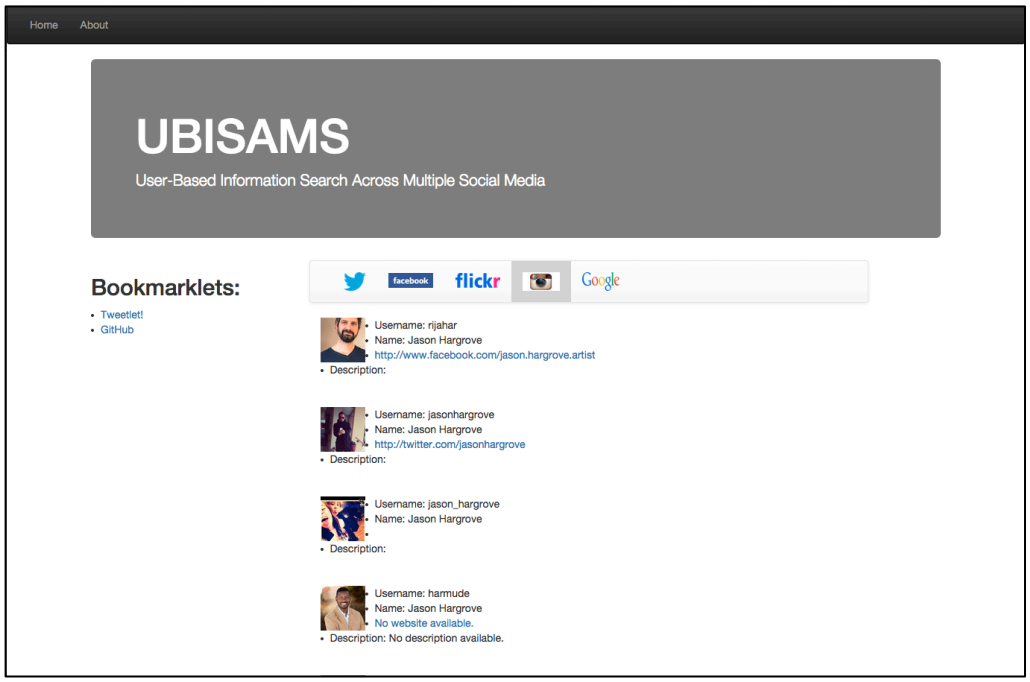


Figure 14: Screenshot of the applications result page

Figure 13 is a screenshot of the applications homepage, where the HTML divides the page into different sections. These sections are styled using Bootstrap CSS. The first section is the header that contains the navigation bar at the top of the page, and the rest of the page is placed within a container, which consists of three different sections:

1. The Jumbotron that contains the application name. It is placed at the top of the container, and it scales to fit the size of the screen.
2. The sidebar column that contains the bookmarklet. It is set to be 3 spaces wide and fixed to the left side of the screen.
3. The content column that contains the content of the site. It is placed next to the sidebar column, and is set to be 8 spaces wide.

When the screen is made smaller and the two columns do not fit next to each other, the content column is placed under the sidebar column. Figure 14 is a screenshot of the result page. The result page contains the same sections as the homepage, but there has been added a navigation bar to the content column. This navigation bar allows the user to choose which results to display.

When implementing the Web interface there was also done some client-side scripting for facilitating the interaction with the web interface. Some of the scripting is embedded within the HTML documents, but there is also an external script that the HTML documents references. The language used for the client-side scripting is JavaScript. JavaScript's jQuery library is used in the external script to simplify the client-side scripting of the HTML. The script retrieves the query string from the URL, and stores it in a variable. It also contains functionality for the navigation bar on the result page. Asynchronous JavaScript and XML (AJAX) is also used in the script for exchanging data with the REST server. This data is the GET request containing the username that was retrieved from the URL. Once there is a response from the server, AJAX is used for updating parts of the web page to present the results, without having to reload the whole page.

In this thesis the embedded scripting is used to create a function that adds the link to the bookmarklet on the web page. It contains functionality for dragging it to the bookmarks bar, and a reference to the actual bookmarklet script that is

ran when the bookmarklet is clicked. The script finds all the tweets by their IDs, and it adds a new “href” object at the end of the tweets header, which is the link that redirects to the result page.

5.2 Federation and Data Layer

The federation layer is the server-side of the system that handles the requests from the presentation layer. For the implementation of the federation layer, Python was chosen as the programming language. The reason that Python was chosen as the programming language is that it has a simple and clear syntax, and features that make it easier and faster to write programs. It is a high-level programming language that offers multiple libraries for functionality. The libraries used in this thesis are listed in table 7.

Table 7: Python Libraries

Library	Functionality
sys	Provides access to the interpreter
json	Support for JSON (encode/decode)
twython	Wrapper for the Twitter API
requests	Sending HTTP requests
web	Web framework

5.2.1 Request Handler

The RESTful Request Handler is implemented using web.py, which is a lightweight web framework for Python. The Request Handler supports a single HTTP GET method. The presentation layer web application will issue a request containing a parameter, which is a person’s username. The Request Handler parses the data and passes the username on to the user-identification component, which executes the request. Once the Request Handler receives the data from the user identification component, it sends a response back to the presentation layer.

5.2.2 User Identification

The user identification component is invoked by the Request Handler, and has the username as an input parameter. In order to communicate with the different APIs there was requested an access token for each of the APIs. The application was granted basic access rights from each of the APIs, and the lifetime of each token was long enough, so that they just had to be requested once during this thesis.

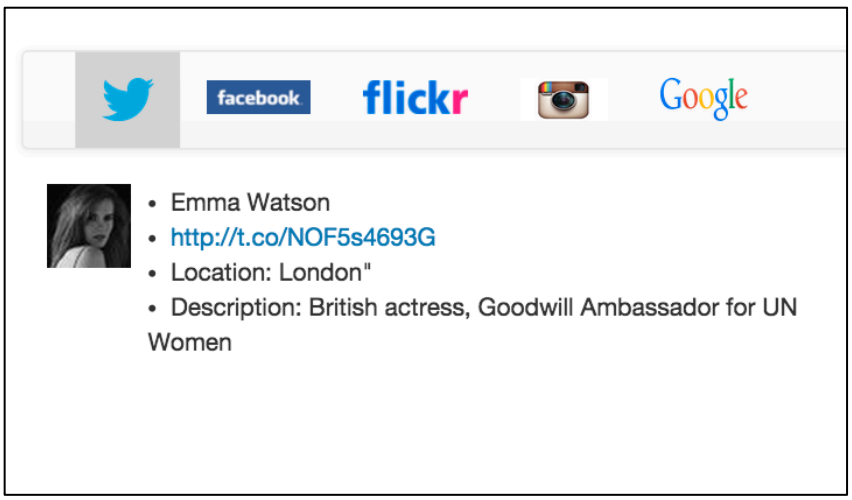


Figure 15: Screenshot of Twitter results.

The User Identification component starts by sending a request to Twitter in order to obtain more information about the user.

Communication with Twitters API is done using Twython, which is an open source Python framework that works as a wrapper for the Twitter API. Twythons feature for querying data for user information is utilized for retrieving user information from twitter. Information that is retrieved from twitter is the users full name, location, description and other URLs if the user has included it in the user profile. This information is stored in a json dictionary, and is sent back to the Request Handler.

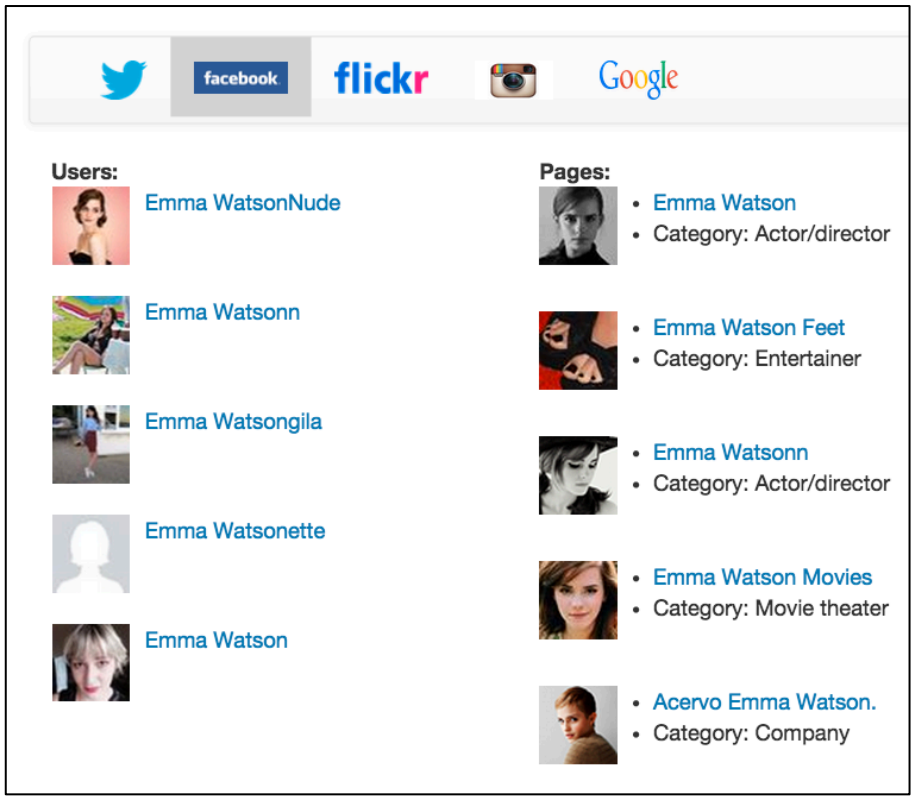


Figure 16: Screenshot of Facebook results.

The user identification component proceeds by issuing search queries to Facebook using the full name of the user that was retrieved from Twitter.

Communication with Facebook Graph API is performed using Pythons request library. All queries issued to the Facebook Graph API are sent using HTTP requests. The queries to Facebook are for both users and pages. If the user is a public figure she might not have a user profile on Facebook, but an official Facebook page instead. A response from Facebook can contain multiple hits on the search queries that were issued. There is chosen a set of the top results of users and pages that placed in two different lists and sent to the Request Handler. A result about the users contains the full name and the profile picture. The result from the pages contains the name of the page, profile picture and description of the page.

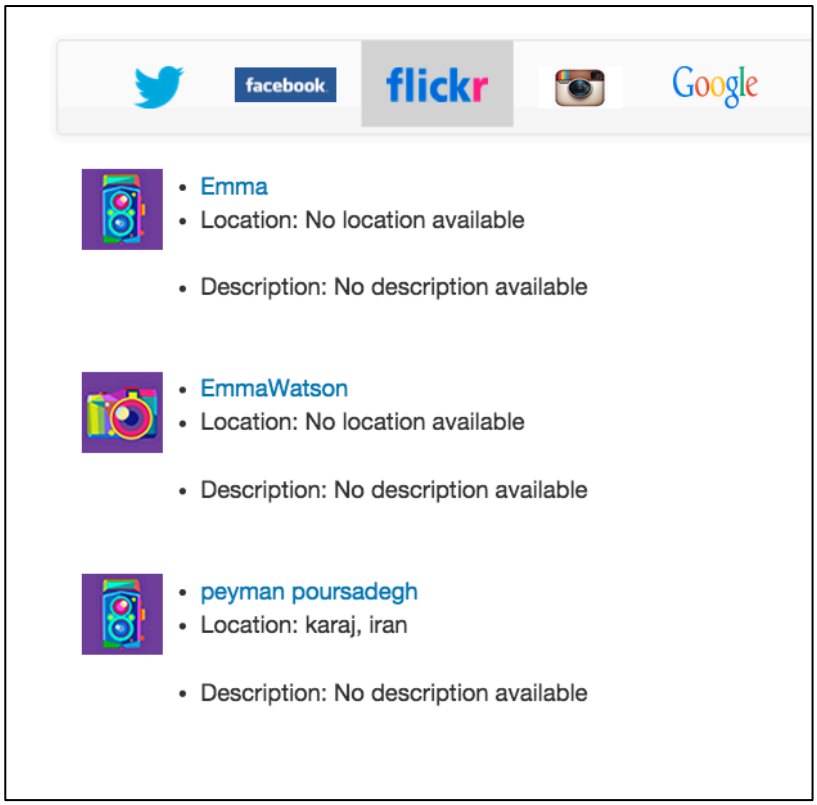


Figure 17: Screenshot of Flickr results.

The step that follows is to issue a request to Flickr. Communication with Flickr API is also performed through HTTP requests. All queries issued to the Flickr API are sent using HTTP requests. Flicker only return a result if the username is an exact match, and hence there is issued two requests: one with the username and another with the full name of the user. If there is a match, the result contains the name of the user, profile picture, location and description if included in the user profile.

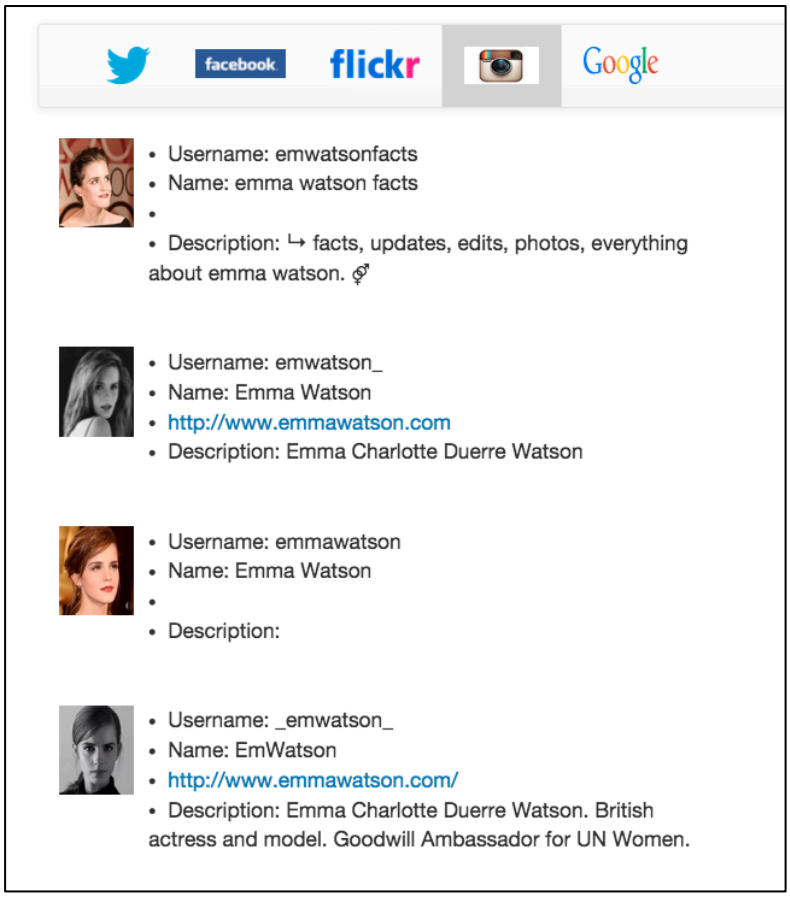


Figure 18: Screenshot of Instagram results.

The user identification component then proceeds by issuing search queries to Instagram using both the username and the full name of the user that was retrieved from Twitter. Communication with the Instagram API is done through HTTP requests to its public REST API endpoint. A response from Instagram may contain multiple hits on the issued queries. These are placed in a list and sent to the Request Handler. The result from the users contains the username, full name, other URLs, description and profile picture if included in the user profile.

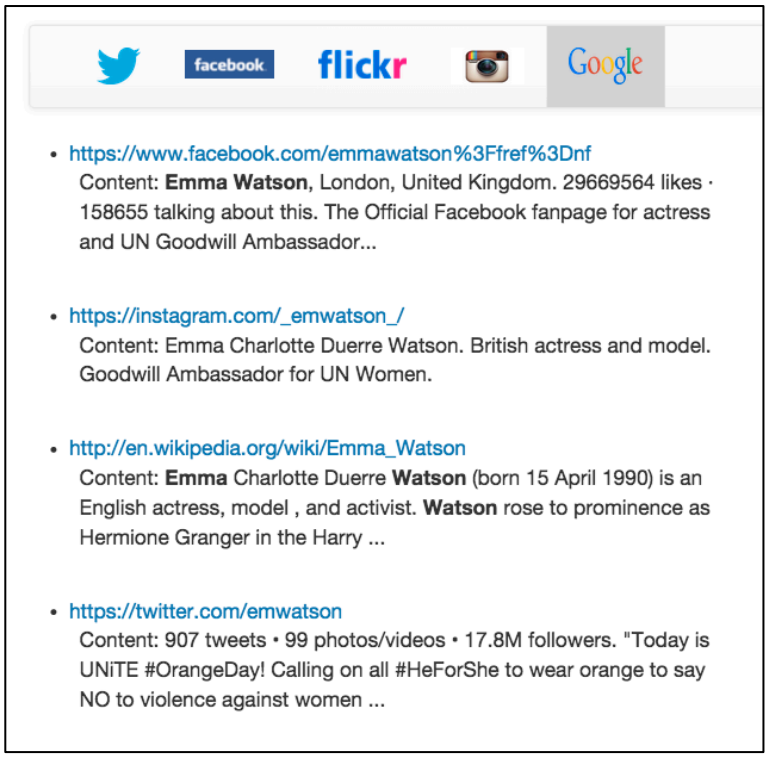


Figure 19: Screenshot of Google results.

The next step is to issue queries to Google. Communication with the Google API is done through HTTP REST queries to the public Google REST API endpoint. There are issued two queries to Google, one containing the username and the other containing the full name of the user. The top 5 hits of both queries are chosen, duplicates are removed and the results are placed in a list and sent to the Request Handler.

6 Tests and Results

In order to test the application an apache web server were set up to run the web interface on localhost. Twitter refuses to load scripts that are stored locally, they only allows code run from a location using secure communication over a computer network (https). As GitHub already supports referencing JavaScript code over https, we chose to upload the script there to be able to load it on Twitter.

There was hand picked a set of 100 Twitter users to enter into and test the application. These users were selected so that most of the users also have profiles on Facebook, Flickr and Instagram in order to measure if the application can find their user profiles on these sites. However some of these may not have profiles on Facebook, Flickr or Instagram. 14 of the 100 users are known public figures.

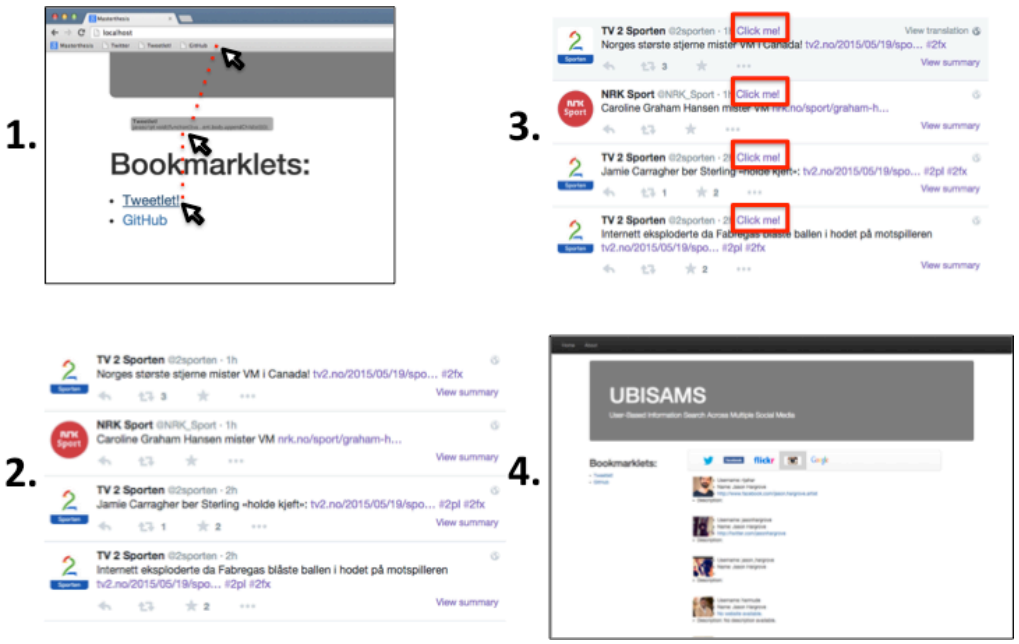


Figure 20: Installation and testing of the bookmarklet.

The first thing that was done when testing the application was to check that the bookmarklet functioned as intended. Figure 20 illustrates the 4 steps of testing the bookmarklet. Step one was to install the bookmarklet from the applications home page of the web interface. This includes dragging the bookmarklet onto the bookmarks bar in the browser (1). Step two was to open Twitter in the

browser, where (2) illustrates a standard unmodified Twitter feed. Step three was to click the bookmarklet; hence the links are inserted into the regular feed by modifying the underlying HTML code (3). Step four was to press one of the links, redirecting the user to the applications result page with the username of the tweets author embedded as a query string in the redirect URL (4). Information in this query string was further used to issue a request to the federation layer, retrieving the results from the different social networks to present for the user.

Once it was confirmed that the bookmarklet worked, the rest of the users were tested by manually exchanging the username in the embedded query of the URL that redirects to the applications result page. This is equivalent to looking up and clicking a tweet for each of the 100 users. The username of a Tweets author is usually gathered from Twitters HTML and embedded in the URL by the bookmarklet as discussed in section 5.1.1. By exchanging the username in the URL manually, it saves time and eliminates the need to go to Twitter, find a tweet and pressing the bookmarklet for each single user from the set that was selected for testing the application.

Results collected for each user includes whether the user was found on Facebook, Flickr or Instagram. It also includes whether the user was found in links provided by the user on the Twitter profile, as well as other results found on Google including homepage, blogs, other social media profiles and articles. All the data that was found were manually checked to determine if the data about the user were correct. This was checked by manually visiting all profiles and links that were found. If a social media profile belonged to the user, the links contained data about the user, or content produced by the user, they were considered correct. When determining if the data was correct, all profile data were matched against each other. This includes profile picture, age, location, description and other data that could help in identifying the user. If sufficient profile attributes overlapped such as profile picture and description, the profiles were considered to belong to the same user. The results was registered in an excel spreadsheet to easily be able to pivot on the data. Table 8 lists the

number of user found by the application, and they are illustrated in a bar chart in figure 21.

Table 8: number of users found by the application.

Social Media	Total Number Found
Facebook	46 / 100
Flickr	71 / 100
Instagram	63 / 100

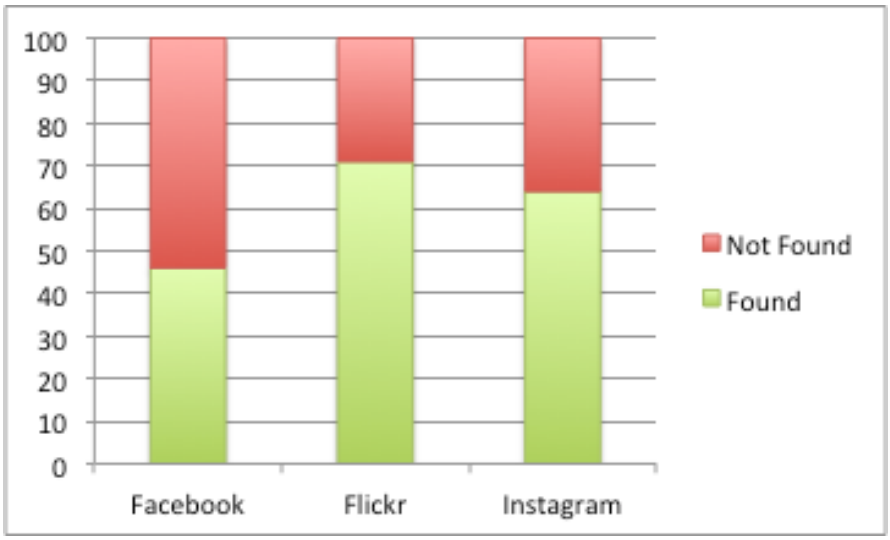


Figure 21: Number of users found by the application.

The application was able to identify 46 of the 100 users on Facebook, while 71 users on Flickr and 63 on Instagram were identified. As Facebook only provide search by full names, a reason that the fewest users were found on Facebook could be because some of the users have common names such as "John Smith". Another reason is the limited profile attributes available through Facebook's API, which make it difficult to compare to other social media profiles and find the correct person, which is furthered discussed in chapter 7.1.1. Out of the 56 users not found on Facebook, it is difficult to determine if a person has a profile on Facebook or not as some may have private profiles where no one can view any profile information unless they are friends. Others might have reserved themselves and will not be visible in any search results. Due to the challenges in manually tracking down users that may or may not be present on Facebook, we can only calculate a precision that is at least as good as:

$$\frac{46}{100} = 0{,}46$$

To Our knowledge 9 out of the 29 users that were not found on Flickr by the application, are registered and have a user profile on Flickr, while 3 of them are confirmed to not have a user profile on Flickr. The 9 users that are known to have a Flickr profile was found through postings on Twitter or search on Flickr's website. The 3 people that are confirmed not to have Twitter includes the author of this thesis and public figures that have stated in the media that profiles with their name on Flickr are fake, and that they do not have an official account on Flickr. This makes it difficult to calculate any precision since one cannot be certain whether the remaining 17 users have a profile or not. By removing the 3 persons that are confirmed not to have a Flickr profile, the precision is at least as good as:

$$\frac{71}{97} = 0{,}73$$

For Instagram the same applies as for the others, it is difficult to calculate any precision, as it is not certain that the users that were not found have profiles on Instagram or not. Therefore the precision for Instagram is at least as good as:

$$\frac{63}{100} = 0{,}63$$

Table 9: Number of users found by Google, and through Twitter profile.

Social Media	Found on Google	Found on Twitter-profile
Facebook	10	5
Flickr	34	25
Instagram	9	1

The Google results that are presented in the application contain a list of the top 4 results from a search on the username and full name of the user, where duplicates are removed.

Table 9 lists the total number of users that were found by Google and through links on the users Twitter profile out of the 100 original users. These numbers shows that the application finds more users for Facebook, Flickr and Instagram than what is available through links on Twitter and top results found on Google through an unspecified search containing a persons name or username.

One reason that so few people choose to link to their other social media profiles on Twitter might be that most people have closed account on other social media sites. It is natural to suspect that to keep these accounts private, users will avoid linking them from a publicly opened social media network such as Twitter. In most of the other networks, users will have to send friend requests to view information in people's private profiles. Linking private profiles through Twitter will probably lead to unwanted friend requests from strangers. This is the case for both Facebook and Instagram where people can choose to either have an open or closed profile. Flickr also allows people to have closed profiles, but most choose to leave them open, this might be the reason that Flickr is the social media application that is most linked through user profiles on Twitter.

As only the top results from a Google search is presented to the user, the results found by the application might or might not appear further down the list of a traditional search result on Google. It is also possible to add specific Google searches to improve the results by for instance adding keywords such as "Facebook", "Flickr" or "Instagram" to the query. However the numbers listed in table 9 shows that the application finds more information about users other social media profiles than the top search results found by Google when using username or full name as a keyword.

Table 10: Users found only by Google and through Twitter profile.

Social Media	Only Found on Google	Only Found on Twitter-profile
Facebook	**2**/46	**1**/46
Flickr	**11**/71	**13**/71
Instagram	**0**/63	**0**/63

Table 10 shows the number of users that were found only through Google and only found through links on the users Twitter profile, and not the applications APIs. An observation made about Instagram users is that Google and links found on Twitter profile do not find any users that is not already found by the

application through the Instagram API. This shows that the application works very well for finding users on Instagram. However a specified Google search might find users that the application missed. But since this has not been tested, one can only speculate. An observation made about the numbers presented in table 10, is that Google complements Flickr very well as the numbers shows that many users that were not found through the use of Flickr's APIs were found through Google.

Table 11: Number of users found on other social media on Google or through links in Twitter profile.

Social media	Number of users found
Pinterest	6
LinkedIn	11
Tumblr	11
YouTube	4
Vimeo	3
SlideShare	3
Google+	4

Other social media where the users are registered that where found include Pinterest, LinkedIn, Tumblr, YouTube, Vimeo, SlideShare and Google+. The number of users found on these sites is listed in table 11. In addition the application also found the blogs and homepages of 58 of the users. These were either listed at the Twitter profile or found by Google. For public figures, the application found Wikipedia articles about 14 of the users, and IMDB articles about 4 of the users. Out of the 100 users that were tested by the application, there was only 1 user that the application could not find any additional information about, but for 8 of the user the application were only able to find additional information through their Twitter profile.

To calculate the precision score for the Google links that were found for each user, the total number of links, as well as how many of these that was correct were manually counted. Links that contained data about the user or content

produced by the user were considered correct (l_C). The precision p for the individual users were calculated by taking the numbers of correct links divided by the total number of links (l_T).

$$p = \frac{l_C}{l_T}$$

The average overall precision (p_O) was calculated by summing up the precision of all the individual users, and dividing it by the total number of users, which gave an average of:

$$p_O = \frac{1}{100}\sum_{i=1}^{100} p_i = 0{,}54$$

One observation that was made while going through the results is that the application struggled in finding people on Facebook when the users full name was not listed in their Twitter profile. The reason for this is because the Facebook API searches for users by their name, and do not allow other parameters to be sent with the search query, so if a full name is not provided, the user is hard to find.

7 Discussion

This section discusses limitation of the application including identification of users and the possibility of searching for blogs. It also discusses the possibility for searching through social media aggregation application, and privacy concerns. Throughout this section there will also be a focus on future work.

7.1 Limitations

7.1.1 Identifying Users

Given a users online identity on one social media site, in order to correctly identify the user on a different social media site, the most common approach is to collect profile attributes and compare them. In order to do so, there must be requested access tokens to the application APIs. Different social media applications have different permissions for what is available through the API as listed in table 5 (chapter 4.3.2). In this thesis the application was only granted basic permissions on Facebook, meaning that the only profile attributes available was the users name, profile link and profile picture. With these limited attributes, identifying the user proved difficult. To solve this, the user was presented with a set of potential users discovered by the application, including a profile picture and a link to the person's profiles to enable the user in making a decision and finding the correct user. This way the users can visit the different profiles themselves and determine if one of the suggested users are a match.

The reason that the application only has basic permissions is that it is only a prototype system running on localhost. If sometime in the future the system is launched as a full-fledged application, there can be requested a new access token with other permissions from Facebook. If these permissions were granted they could help in retrieving other profile attributes such as gender, age and location, which would help in identifying the user.

Besides the username, the only other common profile attribute available through the other applications APIs is the users location. If Facebook had

provided the users location the application might have had a better chance at identifying the user.

One limitation with the approach of gathering profile attributes is that the user might not provide any helpful information on their profile. Besides the name or username, most profile attributes are optional, and the user might choose not to provide any additional information.

The social media applications used in this thesis allows retrieval of profile images through their APIs. For future work the system could implement support for comparison of profile pictures, as many people use the same profile picture in several places. It would also be possible to do a facial comparison of the images, but in might be a very expensive operation that potentially will take quite a long time.

7.1.2 Blogs

In addition to a Twitter users profiles on other social media sites, a second idea was to search for blogs belonging to that user. It proved difficult to search for blogs because of the numerous blog-providing services, and most of these do not provide any API that can be utilized in order to search for specific blogs. However a users blogs might turn up among the results if the user provides a link to their homepage or blog on Twitter, or on one of the other social media sites. It might also turn up in the complementary Google search. As mentioned in chapter 6, there were found homepages or blogs for 58 out of the 100 users that were tested by the application. For future work it might be interesting to complement results with a Google search specifically for blogs where the keyword “blog” is added to the search query.

7.1.3 Bookmarklet

One limitation with the bookmarklet is that it only works on tweets on Twitter. I made an attempt to get it to work for posts on Facebook as well, but the problem with this was that Facebook refused to run the bookmarklet script. The reason Facebook refused to run the script was because it violated some of their content security policy directives. Because of this problem, I made no further

attempts in trying to get it to work for Facebook, but for future work it might be interesting to look into the problem and make the bookmarklet work for different other social media sites. This would enable users to use the application on other social media sites, and the application will potentially reach out to a wider group of social media users.

7.2 Social Media Aggregation Applications

Social network aggregation is a process where data is collected from multiple social networking services and is combined to provide one single presentation of the data. There are several social network aggregators that perform this task in slightly different ways [21] [22]. Information can be combined and presented in a single location, or they may help users to combine their different social networking profiles into one. A common goal and the main purpose of all these applications is to substitute the need to visit one social network application after another, and make it easier for the users to combine and get information from their social media sites. They do so by allowing their users to search across multiple social networks, read RSS feeds for multiple social networks, combine bookmarks, track friends, consolidate messages, accessing their various profiles from one location and getting updates when their name is mentioned or they are tagged in a photo. Social media aggregators handle all data from the users social networks, but some allow the user to filter the information from the different sources in order to only display the information the user is interested in. Users of these services have to register and choose which of their social media applications they want the service to aggregate for them. [22] [21]

For future work it might be interesting to see if any of these applications provide an API that can be utilized to find a users profile on different social media. This might eliminate some of the needs of gathering profile attributes in order to identify the users since the users themself provide information on which sites they use.

7.3 Privacy Concerns

As the number of social networking site has grown, and people create personal profiles on multiple sites, there is a concern for the users privacy [23]. Through social networking sites people post their personal information on the Internet without knowing who has access to it, and what they can do with it. Social networks are known for keeping track of users interactions and for storing the information for later use [24] [25].

There are multiple issues concerning privacy. One of the more serious issues is Cyberstalking [26], which is the act of using the Internet to stalk and harass individuals, groups or organizations. People that Cyberstalkes monitor their victims online and they can threaten, steal identities, vandalize and make false accusations against their victim. The victim might be a person the stalker knows, or it can be a total stranger. The stalker may hide their identity online, which might make them hard to find.

Another issue is that some social media applications including Facebook, use "Places" where people can check in and tell people where they are. By revealing their location it makes it easier for others to track a persons whereabouts. This makes it easier for robbers to know when a person is not home, or for Cyber stalkers to further harass their victims.

Social profiling and third party disclosure has also become an issue. Social profiling is used by social networking sites to cluster their users into groups for directing advertisement to different groups of people. This might include age groups, gender groups, interest groups and different ethnicities. This allows advertisers to "buy" a group of people. Another concern is third party disclosure by data aggregators. Since data aggregators hold large amount of private information about individuals, one concern is that it is possibly only a matter of time before the bubble bursts and private information is leaked. Another concern is how the data is being used, for what is it used, and who should use it. Without people knowing it; marketers, insurance companies, credit companies and others may use this information to judge or make

decisions about people. This means that people are being prejudged and do not have a say in the matter, and won't be able to correct information that possibly might be false or inaccurate.

There are also concerns about the government's use of social networking sites in investigations. In some countries the government can access and use information from social media without securing a search warrant. They use social media as a part of their surveillance in order to maintain social control, recognize and monitor threats, investigate criminals, and criminal activities. It is also used to find and prevent potential terrorists and terrorist threats.

Many Gmail users do not know that Google routinely scans the content of their email messages. However they do state in their terms of service that "*Our automated systems analyze your content (including emails) to provide you personally relevant product features, such as customized search results, tailored advertising, and spam and malware detection. This analysis occurs as the content is sent, received, and when it is stored."*[6] But what they don't say is that by doing so, they are monitoring private conversations among individuals. Google have received positive feedback for assisting in getting a convicted sex offender arrester after finding images of child abuse in his emails. They use a scanning technology that can pick up on content that might suggest criminality. Employees of Google are not able to manually view every email that is sent, but they do view some, as the software is imprecise in distinguishing between innocent pictures of kids bathing and genuine abuse. [27]

During the development of the system in this thesis, privacy concerns have been considered at all times. Therefore the application only uses information that is publicly available through the social media applications APIs. Another aspect of the system is that it does not store any data. The system allows real-time search, and present the results to the user. When the browser page is

[6] http://www.google.com/intl/policies/policies/terms/

closed, all information is lost and the system does not store any of the information.

As no information is stored, there is no personal data about any user that can be sold to third parties such as marketers. The application is merely a tool for finding a users social media profiles, and do not present any data that is not already exposed to the public through the users profiles on the different social media applications.

8 Conclusion

In this thesis a system has been developed and tested, which automatically use information from a posting on a social media site to find a specific user on multiple other social media. The system is integrated with Twitter by the use of a bookmarklet, where the bookmarklet changes the appearance of the site by adding links on the tweets. It also gathers username from the tweet, which it embeds as a query sting in a URL that redirects the end-user to the applications web interface. The application uses the username when issuing search queries to multiple social media applications including Facebook, Flickr and Instagram. The system also issues queries to Google to complement the search results. Profile attributes are gathered by the system and presented to the end-user to make it easier to identify a specific user. The results are presented to the end-user through a web interface.

The application was tested with tweets from a set of 100 handpicked Twitter users, where most of them have profiles on Facebook, Flickr and Instagram. 14 of the 100 users are known public figures. These users were run through the system and the application was able to identify 46 users on Facebook, 71 users on Flickr and 63 users on Instagram. As I do not have information on which social media the users are registered to, I can conclude the precision to be least as good as 0,46 for Facebook, at least as good as 0,73 for Flickr and at least as good as 0,63 for Instagram. Since the complementary Google search did not find any Instagram user that were not already found through the Instagram API, I suspect that the system found most of the users that were possible to find, which allows me to conclude that the application works very well for finding users on Instagram. Through the Google search, the application were able to find 11 users on Flickr that it could not find through the Flickr API, meaning that Google complements Flickr very well. In addition to Facebook, Flickr and Instagram the application were able to find users on other social media application including Pinterest, LinkedIn, Tumblr, YouTube, Vimeo, SlideShare and Google+. For public figures the application were able to find Wikipedia articles for 14 of the users and IMDB articles for 4 of them. The

application also found blogs and homepages for 58 of the users. There were only 1 person the application could not find any information about, but for 8 of the users, additional information were only found through their Twitter profile.

Future work includes launching the application as a full-fledged application and requesting a new access token for Facebook that contains permissions to retrieve profile attributes such as gender, age and location, which might help in identifying users. In the identification process future work also includes image and facial comparison, and the use of aggregated profiles where the users themselves provide information on which social media they use. Other future works involves searching for blogs by adding support for specified Google search where the keyword “blog” is added to the search query. It might also be interesting to make the bookmarklet work for different social media, not only Twitter. This enables users to use the bookmarklet on other social media sites, and the application will potentially reach out to a wider group of social media users.

9 Bibliography

[1] Douglas E. Comer et al., "Computing as a dicipline," *Communications of the ACM*, vol. 32, no. 1, pp. 9-23, 1989.

[2] Michael Haenlein Andreas M. Kaplan, "Users of the world, unite! The challenges and opportunities of Social Media," *Business Horizons*, 2010.

[3] Twitter. [Online]. https://about.twitter.com

[4] (2015) Facebook. [Online]. https://www.facebook.com/facebook/info?tab=page_info

[5] Flickr. (2015) About Flickr. [Online]. https://www.flickr.com/about/

[6] Instagram. (2015, May) About Us. [Online]. http://instagram.com/about/us

[7] w3schools. HTML Introduction. [Online]. http://www.w3schools.com/html/html_intro.asp

[8] w3schools. CSS Introduction. [Online]. http://www.w3schools.com/css/css_intro.asp

[9] w3schools. JavaScript Introduction. [Online]. http://www.w3schools.com/js/js_intro.asp

[10] Bookmarklets | Bookmarklet Search Engine. [Online]. www.marklets.com

[11] Bootstrap. [Online]. http://getbootstrap.com/

[12] C. D. Manning et al, *Introduction to information retrieval*. Cambridge, UK: Cambridge University Press, 2008.

[13] Andrew Sears and Julie A. Jcko (Eds.)., *Human-computer interaction: design issues, solutions, and applications.*: CRC Press, 2009.

[14] Peter Morville. (2004, June) Semantic Studios. [Online]. http://semanticstudios.com/user_experience_design/

[15] Mari Motoyama and George Varghese, "I Seek You: Searching and Matching Individuals In Social Networks," in *Web information and data management*, Hong Kong, 2009, pp. 67-75.

[16] Paridhi Jain and Ponnurangam Kumaraguru and Anupam Joshi, "@I seek 'fb.me': Identifying Users across Multiple Online Social Networks," in

Proceedings of the 22nd international conference on World Wide Web companion, Rio de Janeiro, 2013, pp. 1259-1268.

[17] Cheng-Ta Chung, Chia-Jui Lin, Chih-Hung Lim, and Pu-Jen Cheng, "Person Identification between Different Online Social Networks," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 01, 2014, pp. 94-101.

[18] Reza Zafarani and Huan Liu, "Connecting Users across Social Media Sites: A Behavioral-Modeling Approach," in *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, 2013, pp. 41-49.

[19] Tereza Iofciu and Peter Fankhauser and Fabian Abel and Kerstin Bischoff, "Identifying Users Across Social Tagging Systems," in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[20] Yuanping Nie and Jiuming Huang and Aiping Li and Bin Zhou., "Identifying Users Based on Behavioral-Modeling across Social Media Sites," *Web Technologies and Applications*, vol. 8709, pp. 48-55, 2014.

[21] Stan Schroeder. (2007, July) 20 Ways To Aggregate Your Social Networking Profiles. [Online]. http://mashable.com/2007/07/17/social-network-aggregators/

[22] Rachael King. (2007, June) When Your Social Sites Need Networking. [Online]. http://www.bloomberg.com/bw/stories/2007-06-18/when-your-social-sites-need-networkingbusinessweek-business-news-stock-market-and-financial-advice

[23] Ralph Gross and Alessandro Acquisti, "Information Revelation and Privacy in Online Social Networks (The Facebook case)," in *ACM Workshop on Privacy in the Electronic Society*, 2005.

[24] Eric Bangeman. (2010, May) Report: Facebook caught sharing secret data wth advertisers. [Online]. http://arstechnica.com/tech-policy/2010/05/latest-facebook-blunder-secret-data-sharing-with-advertisers/

[25] Keith Gladdis. (2012, Feb.) Twitter secrets for sale: Privacy row as every

tweet for last two years is bought up by data firm. [Online]. http://www.dailymail.co.uk/sciencetech/article-2107693/Twitter-sells-years-everyones-old-vanished-Tweets-online-marketing-companies.html

[26] Brian H. Spitzberg and Gregory Hoobler, "Cyberstalking and the technologies of interpersonal terrorism," *New Media & Society*, vol. 4, pp. 67-88, 2002.

[27] Irish Examiner. (2014, Aug.) Google scans content of your Gmail messages. [Online]. http://www.irishexaminer.com/ireland/google-scans-content-of-your-gmail-messages-278470.html