CHALMERS

Personal Information Revelation and Privacy Mining

A Practice of Swedish Online Privacy Harvest

Master of Science Thesis in the Master Degree Programme Secure and Dependable Computer Systems

YANG YUAN JIN

Department of Computer Science and Engineering
Division of Networks and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010

Personal Information Revelation and Privacy Mining:

A Practice of Swedish Online Privacy Harvest

Master's Thesis in Secure and Dependable Computer System

AUTHOR: YUAN JIN YANG

Department of Computer Science and Engineering
Secure and Dependable Compu ter System
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2010

Personal Information Revelation and Privacy Mining
A Practice of Swedish Online Privacy Harvest
Master Thesis in the Master's programme in Secure and Dependable Computer Systems

YUAN JIN YANG

Examiner: Philippas Tsigas

Department of Computer Science and Technology
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover: The cover picture is from *www.istockphoto.com*
Göteborg, Sweden 2010

ABSTRACT

With the rapid development of social networking and micro blogging service, large quantities of users are uploading their personal information via website, social network application, or external application to establish the social connections with each other. However, it directly enables the personal information transparent to the persons who may abuse the information.

The purpose of this thesis is to provide a comprehensive research on perceived privacy and security concerns associated with online personal information in Sweden.

The thesis analyzes and investigates the privacy concerns on the basis of the social networks and various kinds of search engines and thereby reaches one kind of effective searching logic. This logic has been realized on an interface implemented by Python, JavaScript, HTML and social network API. The practical running and theoretic analyses prove that the logic of disclosing the personal information is effective and the information fetched by interface can be reliable in most of cases. The limitation of the interface is explained in detail and the thesis also provides the possible solutions to them in future work.

From the practical investigation, we can conclude that, there are many risks of leakage of personal information in social networks and other online service even though the users indeed enjoy and benefit a lot from them. Admittedly, there should never be a compromise between individual privacy and potential security.

Keywords:
Personal Privacy, Social Networks, Parsing, Text Processing, Regular Expression, Python

Preface

The thesis was derived from an idea of Philippas Tsigas and Magnus Almgren and conducted at Chalmers University of Technology in Division of Networks and Systems, at the Department of Computer Science and Engineering from March 2010 to September 2010.

In the process of concluding the thesis, I would like to thank my examiner Philippas Tsigas and supervisor Magnus Almgren, who initialled the idea of thesis and have supported of the whole work. I am particularly grateful to Magnus Almgren, since I have received a great support from him and he also enlightened me a lot to solve the problem. And besides above, I should thank the people who would like to help in developing interface even not directly involved, they have encouraged me to finish the thesis work.

1. Introduction

In this chapter, the background related to the selected research topic is introduced; the task description as well as the purpose and delimitation will be demonstrated.

1.1 Background

With the significant idea of Web 2.0 boomed over the internet, the networking application and service are experiencing a surprisingly change[1]. A group of new technologies, ranging from blogs, wikis, social networks, bring the whole society into the internet. The connected users on internet performs more socially with personal communication by publishing the personal profile as a brief description including real name or nickname, gender, photographs, living and work address, ethnicity, and favourites, with the intention to contact or being contacted by anyone surfing on the web, and in some extent, it has also evolved into a novel relationship maintenance way with friends[2]. This new relationship mode mainly depends on the updated user activities[3]. With "twitter" as an example of a micro blog, one can post a concerning sentence or terms, which makes the people learn one's current situation in real-time. By meaning of the most popular Social Networks (SNS – see Abbreviation in page 54) such as "Facebook", "MySpace", sharing pictures, achieving events and getting messages between friends turn fast and convenient.

Although one single pattern of the above personal information exposed on web can not directly generate the security and privacy concern, however, utilizing the systematically model gathering information in the largest extent and making logically judgement on truth of the information to categorize and store users' information can be a potential threat to the most of online Web 2.0 enjoyer. Unfortunately, an unprecedented trend[4] in today's society is rising: the collection and aggregation of personal identifiable information and other type of information, which can, when statistically analyzed in a larger context, reveal personal and sometimes sensitive information. This trend is fuelled by a few kinds of reasons, whereas all reasons share a common fact that the private information has special value for both publishers and receiver.

For example, from a business perspective, user information can offer valuable insight into one's customers, serving the customers better to meet the ever changing demand and feedback, which is considerably effective to embed target advertisement to the most focus area statistically concluding from the on line user information[5]. Reports from China real estate[6] can be the evidence for the commercial using of personal information: millions of proprietors' information is sold from the proprietors' name, telephone to bank account, finance situation and even the family's physical situation. According to this information, the real estate company can conclude an appropriate house price selling to these selected people, which can maximize the corporation profits and also target the customers with effective marketing promotion. For the personal user, the divulgence of data might be for convenience, a way to gain access to certain rebates or just to be able to connect to one's friend through a social network. Both of the ways mentioned above for personal use can become a potentially security issue, from publishing the email address resulting in spam, to leaking out some of the identity data important to a credit card application that could result in an identity theft.

To combat this, the supervision of private information may be handled by a third party or organizations with good reputation. For example, influential commercial organizations (i.e., banks) and the state have long had access to sensitive data from the users but done so in a responsible way. However, what has changed gradually is that more and more such information is shared openly and publicly over the Internet. In some situations, without need of the crack into database, the organizations already publish a lot of user data as white page or yellow page on web site. The information from these organizations shows a truer and correct side. The problem is that even trivial information can, when aggregated with other information from the user and her friends, give a detailed model of the user's interests. If this is then coupled with a statistical model, even more detailed information can be extracted. A very simple example is exactly the status update with a location -- anyone can then see that the user is at a certain location which is not their home. Then burglars can use it to break into the house. For example, Sweden, with a long tradition of open access to information coupled with a tight control of each person through the ubiquitous "personal number" is a very good example.

1.2 Related Work

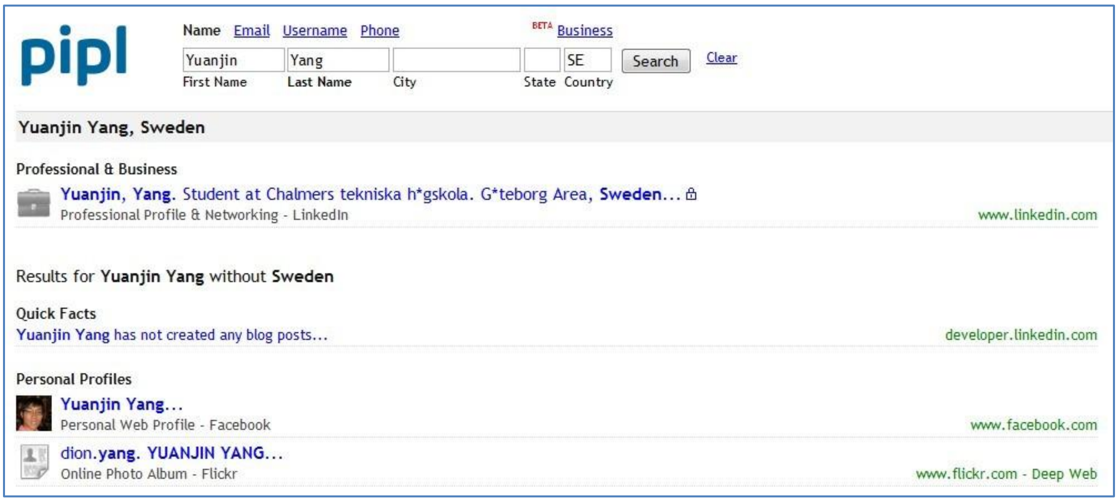
The online security concerns are never neglected and in fact, there are numerous resource and documents assisting one to explore the personal information on the internet.

As a concrete example to demonstrate the tools described here, I use my name (Yuanjin) as an example to discuss the identity.

Yahoo people search attempts to cover most scope of U.S, through referencing a variety of different sources, thus it can do a global search or locate to a state or city. By typing my name “Yuanjin”, it hits one result in Temecula, CA.

Yoname.com emphasizes on SN such as LinkedIn, Facebook, Friendster and so on. It display location, gender, age and business related information. The input of “Yuanjin” discovers various kinds of information, whereas none of them can really represent me and my online identity.

Pipl.com searches in a very smart approach that it firstly locates the IP and searches for information within specified geographic scope of the IP address.



First Name: Yuanjin
Last Name: Yang
City:
State:
Country: SE

Yuanjin Yang, Sweden

Professional & Business

Yuanjin, Yang. Student at Chalmers tekniska h*gskola. G*teborg Area, **Sweden...**
Professional Profile & Networking - LinkedIn
www.linkedin.com

Results for **Yuanjin Yang** without **Sweden**

Quick Facts

Yuanjin Yang has not created any blog posts...
developer.linkedin.com

Personal Profiles

Yuanjin Yang...
Personal Web Profile - Facebook
www.facebook.com

dion.**yang**. **YUANJIN YANG...**
Online Photo Album - Flickr
www.flickr.com - Deep Web

Figure 1.2: *pipl* search result of Yuanjin

All found result is consistent with my real situation online.

There is also some existing literature related to personal privacy and identification issues.

Liu and Maes[7] build the "InterestMap" for mining the social network profiles, which establishes the connection of interest and identities from analyzing the profiles and produce a recommendation. Chew, Balfanz and Laurie[8] point out three areas prone to divulge the user privacy: lack of control over streams, unwelcome linkage and deanonymization through merging of social graphs. Narayanan and Shmatikov[9] develop the framework to analyze the privacy and an algorithm for detecting the anonymized graphs. Alike from the above technique standpoint, Dourish[10] contributes to a cultural and social perspective to understand the cyber privacy issues.

There are still many approaches and tools assisting one to find the hidden information about a specific person online. The means or methods to discover the information vary, but it still shows some common features:

1. Searching among Social Networks is necessary
2. Public resource with high reliability is referenced
3. The telephone number contact to yellow page to reference one's contact information
4. Map is used in high frequency to locate one's address

However, among all these approaches and tools, none is effective for one to comprehensively and deeply search a target person in Sweden or coverage of Swedish web resources. Because of this, this thesis work can introduce some perspectives: "Swedish social networks' privacy problems".

1.3 The Task Description

As a first step, the search portal needs to be constructed where a user can input and formulate a searching query for example name, telephone, email, address, etc. The purpose of these is to discover how much information is available and related about them in public sources. By entering as much information as the user wants and feels comfortable with, the portal software then searches other available Internet services, fetches the suitable matching related information, and presents the final analyzed results to the user. The searching scope mainly covers Sweden and Swedish web resources, and about the validation of online personal information it is not in the coverage of this task. From the results of searching, the user can get a comprehensive view on the target person. By means of this experience, therefore, we achieve a better understanding on what their public data can tell a computer scientist / statistician.

1.4 Purpose and Delimitations

The main purpose of thesis is to alert cyber users to the privacy issues of them releasing information online, be it through social network memberships or other services. Users should be made aware of what they release and be in control of the process themselves. Moreover, the user can find the relatives, friends, and fellows in Sweden scope by means of the interface applied in this thesis to acquire more information about them rather than in daily life. With the assistance of the interface service, the related work about SNS privacy and identification of personal information can be conducted for further analyzing relationship in social networking and required data collection in the scope of Sweden.

The delimitation for this master thesis will be reflected in the percent of accuracy about the personal information fetching, which may also influence the final conclusion about the web trust issues. The reason causes the situation of incorrect will be explained in the discussion section. Besides of this, the query also needs to be specified to avoid vagueness and indistinctiveness due to the referenced search site unable to handle this and failing to return all possible results. The query may also raise other problems. For example, if the query is just the word “Johan”, which is simple and clear as one of most common names in Sweden. However, it will generate much waiting time for parser to analyze the content. The reason for that is the query designed too simple causes massive search results but also meets the search requirement results returned. Since the interface does not simply display the text of individual information, it involves web fetch operation, analyze operation and location comparison via Google Map, in some degree, the long delay time for simple query like “Johan” can be hardly prevented.

For the picture and graph fetched from Flickr or other SNS, the identification of object belonging is not effective owing to lack of the picture identification functional module to assist analyzing the person features. The Google search in this thesis is somewhat independent of the other search for its comprehensive and uncertainty of return results, the work related to this will be describe through in future work section.

2. Analysis and Methodology

The difficulty from the very beginning of the work is to determine a start point from a quantity of personal information, which is effective to locate on the certain target person, expanding the known information to every detail of the target. In order to address the first search pattern, the understanding of analyzing the process of searching people and the social web itself, to what extent it can disclose personal information is highly demanded.

2.1 Search Process Analysis and Social Network Privacy

The core idea of Web 2.0 “Individual production and User Generated Content” pushes internet users as a media[11], in order to broadcast the blog, pictures or messages, personal information in most cases is highly demanded. On this point, we can reach an agreement that: without the requirement for us to sneak into the police computer or tax office database, most of the web surfer themselves have provided sufficient information in public web resource for one to fetch.

In ordinary way to discover a pattern of available information of certain people, a search engine is one of the best effective tools to choose. Google is the most common use engine for a large amount of people, but due to its diversity, the search results returned are complex except when pre-defining a well designed search word. This method is workable only if the targeted search people or the piece of information is already known by the requestor, despite the search result in a wide range, the information still can be categorized. Nevertheless, in most cases, the search word and search result are both unpredictable. For example, to the index of name category or business category in white page[12], the situation turns obviously clear to be evidence. Since every letter in name category contains over hundreds people, the same name may occur many times even though it refers to different one in different location, but the possibility is also registered address information on web just for temporary login, in some extent, are junk data.

The correctness of information does not depend on the viewer, but count on register user. Even the search engine, what they can do, is also to filter and to fetch the information that already exists on the web. However, since we cannot prove all the information grabbed for the web as true, there are still some approaches that we can attempt to find the “correct” information to the largest extent. According to the research of Facebook [13], in Facebook file, 89% of all names are realistic and likely the true name for the user. 8% of names are obviously fake names, and 3% of people disclose their first name in profile. The social network turns to be an easily breakthrough to find the most real information. Besides the social network, the white page and yellow page are also the source that provides highly reliable information in names, telephones, address and short description. Thus, the solution of determining a breakthrough is definitely clear:

1. Implement search engines connected to the white page or yellow page as source to fetch the personal related information.
2. Name, telephone, address and etc. information can be considered as reliable, which can be used as index or information pattern to search in the social networking.

Based on the search engines and social networking, a great amount of personal related data can be integrated, and the rest of work moved to filter the true and reliable information out the present data. In this branch, the logic analysis and judgment on whether the capture information belongs to the certain person is necessary. The reference techniques include processing search result, parsing web content and fetched data, comparing the related piece of data, the logic filter process will be described in the next section.

2.2 Main Referenced Database and Comparison

In order to effectively exploit the online user information, a thorough knowledge of different referenced databases and their comparative analysis are almost necessary, which provides the fact basis for information correctness and information revelation.

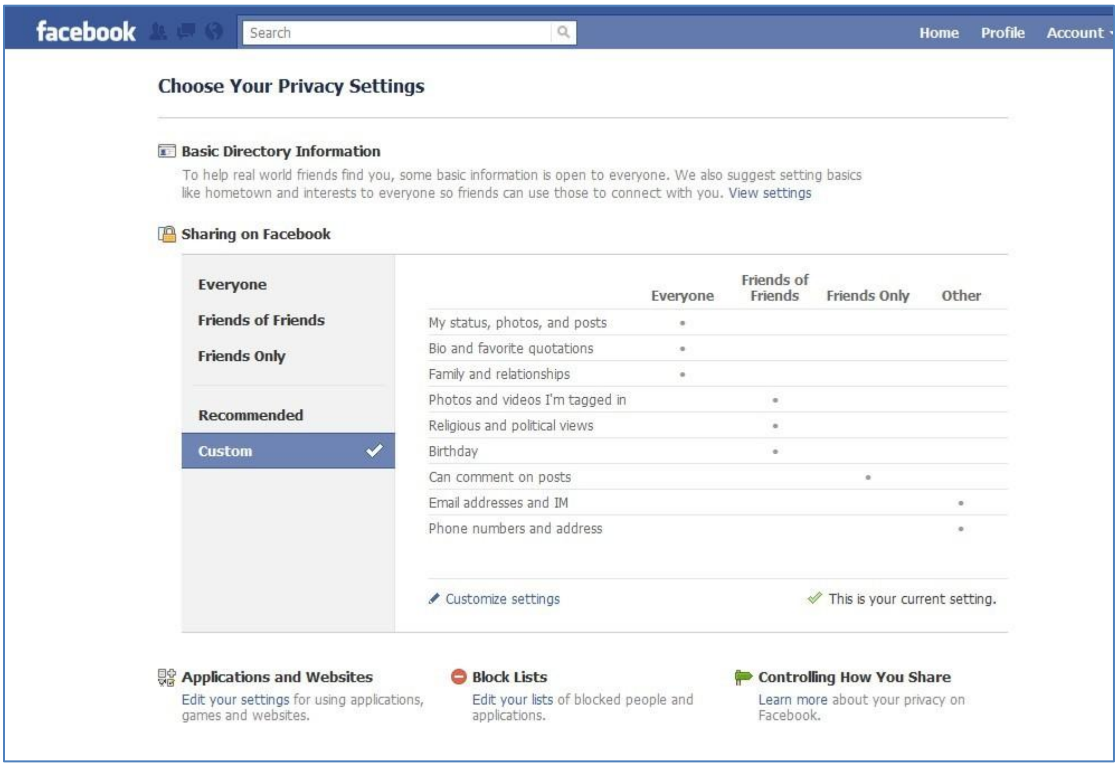
2.2.1 Social Network Accessibility and Privacy Control

The most popular site such as *Facebook*[14], *Twitter*[15], *MySpace*[16] and *Flickr*[17], these SNs maintains the attributes as “favourite”, “friend”, or “introduction” in a database, but each SN has a privacy control mechanism to protect the user information disclosure. In subsequence, the analysis of each SN privacy accessibility and relevant content is revealed, which also directs to search logic selection and shortcuts.

- Facebook

According to the Facebook privacy policy[18], sharing information on Facebook is classified into four levels. Name and Profile Picture is designed by Facebook open to everyone on internet, both Facebook search engine and public search engine can locate except when the user is setting privacy limits. The contact information and personal information like birth date and gender have flexible setting control from available to everyone to just someone.

The example of accessibility of Facebook can be seen in Figure 2.2.1:



Choose Your Privacy Settings

Basic Directory Information

To help real world friends find you, some basic information is open to everyone. We also suggest setting basics like hometown and interests to everyone so friends can use those to connect with you. View settings

Sharing on Facebook

	Everyone	Friends of Friends	Friends Only	Other
My status, photos, and posts	•			
Bio and favorite quotations	•			
Family and relationships	•			
Photos and videos I'm tagged in		•		
Religious and political views		•		
Birthday		•		
Can comment on posts			•	
Email addresses and IM				•
Phone numbers and address				•

Customize settings

✓ This is your current setting.

Applications and Websites

Edit your settings for using applications, games and websites.

Block Lists

Edit your lists of blocked people and applications.

Controlling How You Share

Learn more about your privacy on Facebook.

Figure 2.2.1: Facebook Privacy Setting

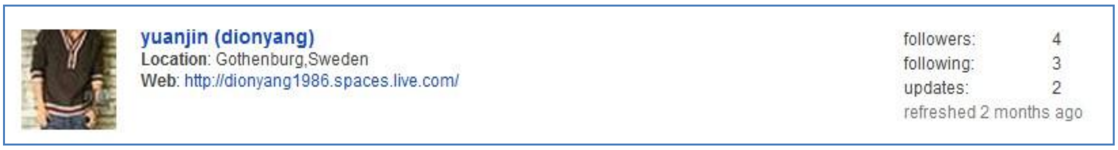
For the one who use Facebook, the default *Privacy Setting* is configured as Figure 2.2.1. The "Everyone" in *Privacy Setting* means completely open to public, which can be accessed by anyone even he/she do not log into the Facebook. The information normally available to "everyone" is status, photos, posts and relationships. The other information crucial to identify one's identification for instance phone number and address are usually protected by Privacy Control hidden to public. And some of is only revealed information after establishing the relationship with the person on SN.

In general, Facebook classifies information into different levels, and the accessibility of information depends on which level they belong to. In most cases, the users may select the recommended *Privacy Setting*, which makes the personal photos, name and relationships public, and these three attributes are the main recognized elements. Based on the research mentioned above, the name has nearly 90% reliability. Meanwhile, in Facebook Profile, 61% of image are identifiable (good quality to enable person recognize), and 19% shows semi-identifiable (due to some reason, the person is not directly to be recognized). Therefore, Name and Profile image are the primary elements for one to recognize unfamiliar or uncertain persons, whose accessibility is default available to all users.[19]

- Twitter

The privacy controls in Twitter do not function too much of information protection, as the sector in Twitter Privacy Policy states, "Our Services are primarily designed to help you share information with the world. Most of the information you provide to us is information you are asking us to make public."[20]

In Twitter, most of information is by default set to public, which includes name, username, tweets, and location information.
Figure 2.2.2 displays my name search results, and all privacy setting in my account is default set.



yuanjin (dionyang)
Location: Gothenburg,Sweden
Web: http://dionyang1986.spaces.live.com/

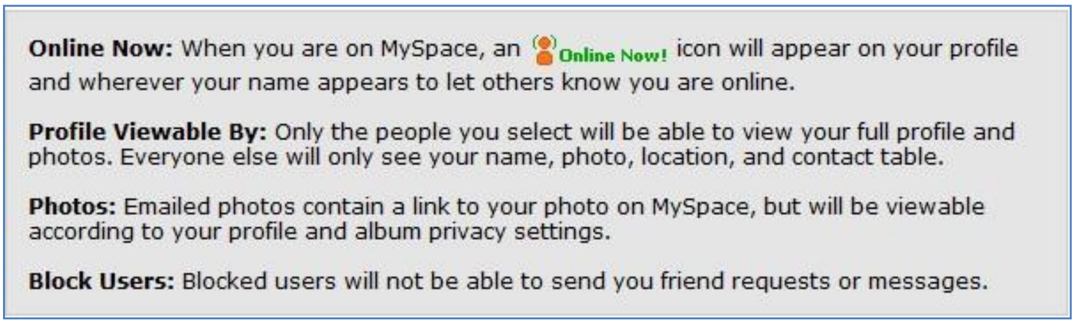
followers:	4
following:	3
updates:	2

refreshed 2 months ago

Figure 2.2.2: twitter search of “Yuanjin”

- MySpace

The Privacy in MySpace provides some options for user to determine, which manage whether online status, birthday is shown to anyone, whether comments, friends, and photos can be viewed by anyone or a specified group[21]. The Figure 2.3.3 explains the detail of accessibility of user.



Online Now: When you are on MySpace, an Online Now! icon will appear on your profile and wherever your name appears to let others know you are online.

Profile Viewable By: Only the people you select will be able to view your full profile and photos. Everyone else will only see your name, photo, location, and contact table.

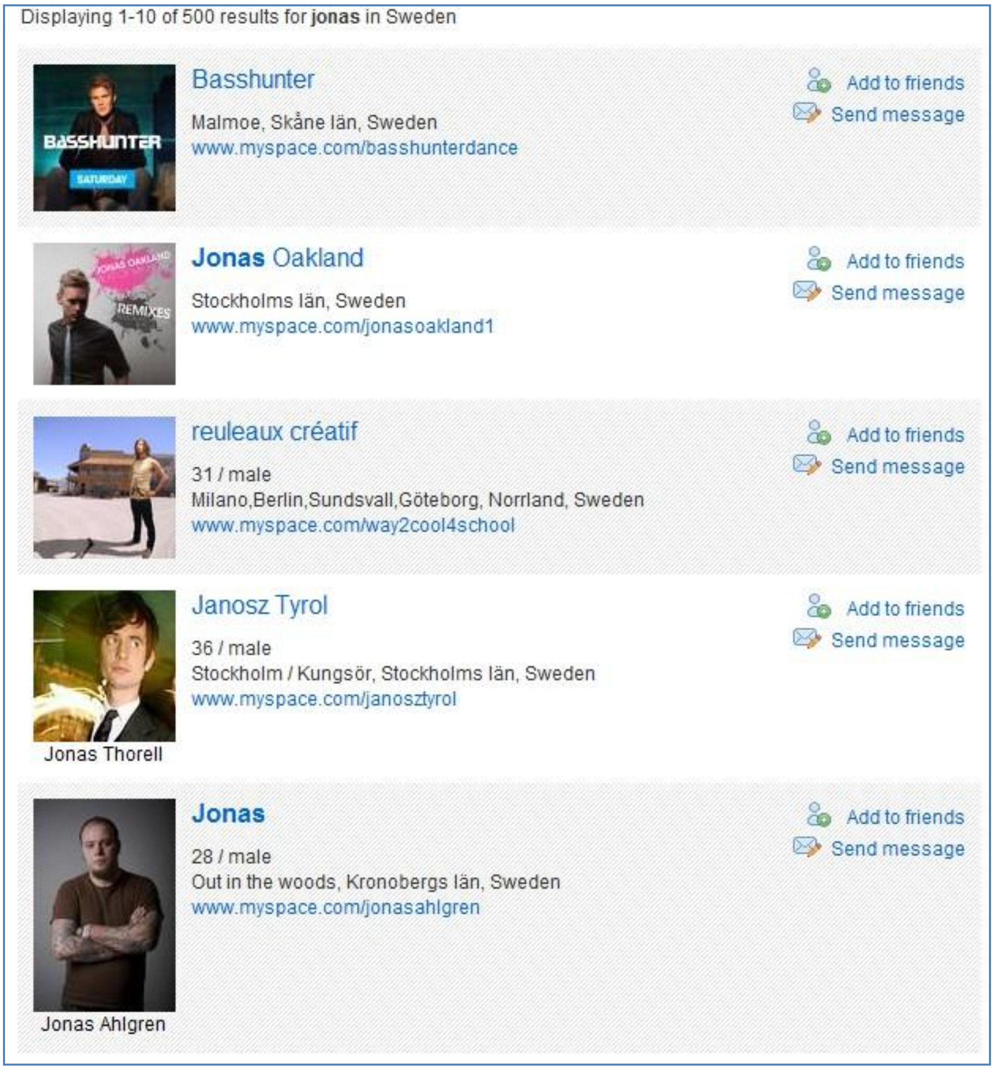
Photos: Emailed photos contain a link to your photo on MySpace, but will be viewable according to your profile and album privacy settings.

Block Users: Blocked users will not be able to send you friend requests or messages.

Figure 2.2.3: MySpace Profile privacy settings

According to the description, it is available for user to search via name, email, address except setting the privacy limits to the profile.

Figure 2.2.4 displays an example of search results as “Jonas” in scope of Sweden:



Displaying 1-10 of 500 results for **jonas** in Sweden

Basshunter
Malmoe, Skåne län, Sweden
www.myspace.com/basshunterdance

Jonas Oakland
Stockholms län, Sweden
www.myspace.com/jonasoakland1

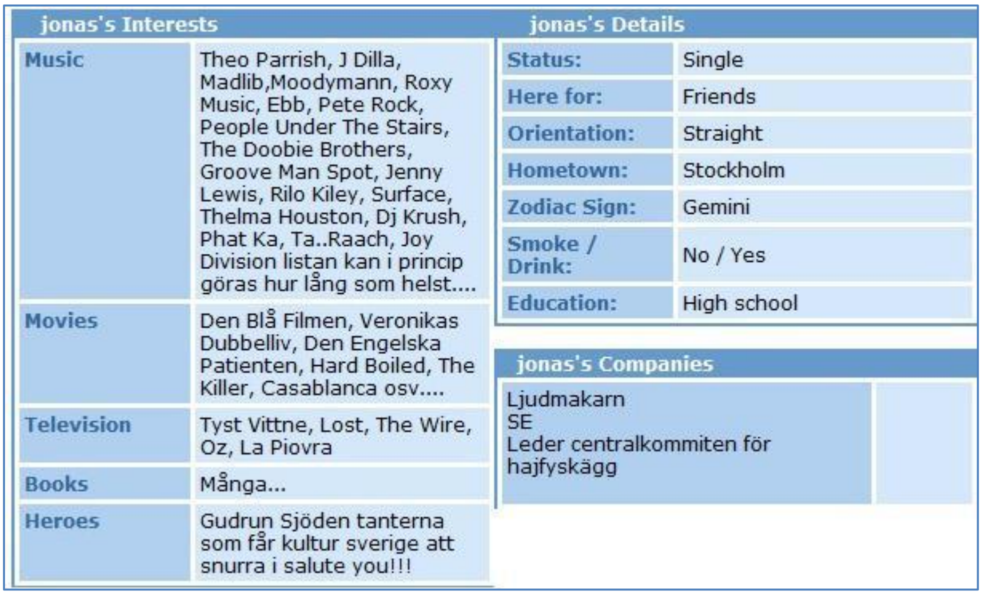
reuleaux créatif
31 / male
Milano,Berlin,Sundsvall,Göteborg, Norrland, Sweden
www.myspace.com/way2cool4school

Janosz Tyrol
36 / male
Stockholm / Kungsör, Stockholms län, Sweden
www.myspace.com/janosztyrol
Jonas Thorell

Jonas
28 / male
Out in the woods, Kronobergs län, Sweden
www.myspace.com/jonasahlgren
Jonas Ahlgren

Figure 2.2.4: MySpace search result of "Jonas"

From the format of results, the MySpace search returns results as name, age (if it is accessible), gender (if it is accessible), location, website address and profile photos. Besides basic profile, in website address, more information can be disclosed depends on privacy control in Figure 2.2.5:



jonas's Interests	
Music	Theo Parrish, J Dilla, Madlib,Moodymann, Roxy Music, Ebb, Pete Rock, People Under The Stairs, The Doobie Brothers, Groove Man Spot, Jenny Lewis, Rilo Kiley, Surface, Thelma Houston, Dj Krush, Phat Ka, Ta..Raach, Joy Division listan kan i princip göras hur lång som helst....
Movies	Den Blå Filmen, Veronikas Dubbelliv, Den Engelska Patienten, Hard Boiled, The Killer, Casablanca osv....
Television	Tyst Vittne, Lost, The Wire, Oz, La Piovra
Books	Många...
Heroes	Gudrun Sjöden tanterna som får kultur sverige att snurra i salute you!!!

jonas's Details	
Status:	Single
Here for:	Friends
Orientation:	Straight
Hometown:	Stockholm
Zodiac Sign:	Gemini
Smoke / Drink:	No / Yes
Education:	High school

jonas's Companies

Ljudmakarn
SE
Leder centralkommiten för hajfyskägg

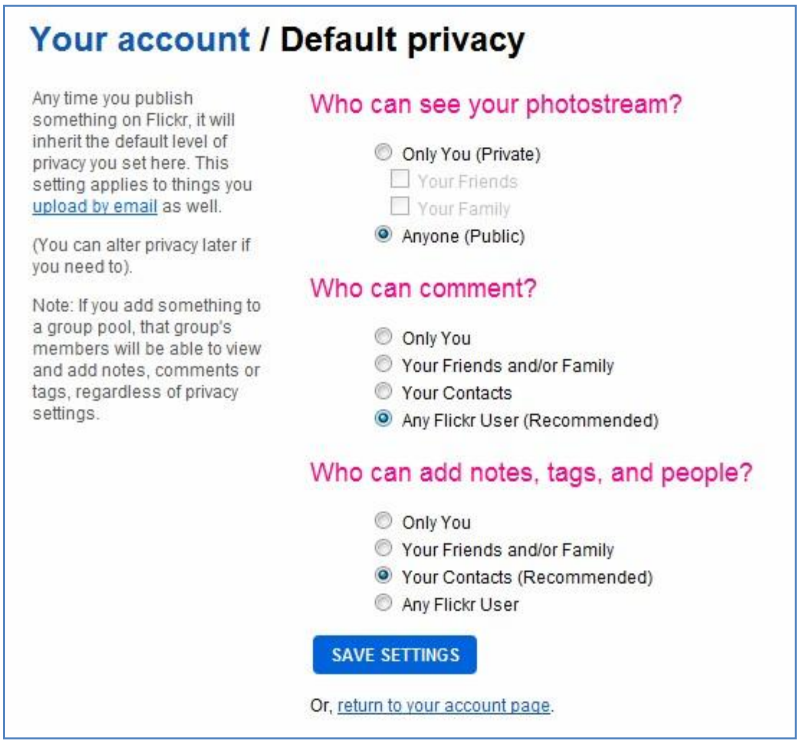
Figure 2.2.5: Jonas's website personal information

For MySpace, it reveals much personal information even if it has a privacy limit mechanism.

- Flickr

In Flickr, the privacy control is applied to each photo. The privacy level is mainly divided into two parts: private and public. In public, it is available for everyone online. In private, the level is limited progressively from friends, family to completely private[22].

The Figure 2.2.6 shows a detail of Flickr privacy setting:



Your account / Default privacy

Any time you publish something on Flickr, it will inherit the default level of privacy you set here. This setting applies to things you upload by email as well.

(You can alter privacy later if you need to).

Note: If you add something to a group pool, that group's members will be able to view and add notes, comments or tags, regardless of privacy settings.

Who can see your photostream?

○ Only You (Private)
□ Your Friends
□ Your Family
● Anyone (Public)

Who can comment?

○ Only You
○ Your Friends and/or Family
○ Your Contacts
● Any Flickr User (Recommended)

Who can add notes, tags, and people?

○ Only You
○ Your Friends and/or Family
● Your Contacts (Recommended)
○ Any Flickr User

Or, return to your account page.

Figure 2.2.6: Flickr privacy settings

From the collection of above SN analysis, for each SN, it could disclose partial if personal information in condition that privacy sets public. The Table 2.2.6 illustrates each SN accessibility and relevant information:

Social Network	Name(or username)	Location	Friends and Relationships	Personal Photos	Detail Information
Facebook	Public	Friends or Private	Public or Friends	Public	Friends or Private
Twitter	Public	Public	Public or Friends	Public	Friends or Private
MySpace	Public	Public	Public	Public	Public or Friends
Flickr	Public	N/A	N/A	Public	N/A

Table 2.2.7: the Matrix of SN accessibility and relevant information

In Table 2.2.7, Public means Public Searchable (anyone can search from the public search engines). Friends means Friends Searchable (After establishing relationship, some personal information can be viewed). Private is Private (the information can only be viewed by service provider and account owner). Based on the table, we can clearly know which kind of information can be extracted by each SN.

Note that Table 2.2.7 shows the default settings for the social networks. In many cases, these settings can be overridden by the user and be set differently.

2.2.2 Eniro and Hitta --Local Search Engine

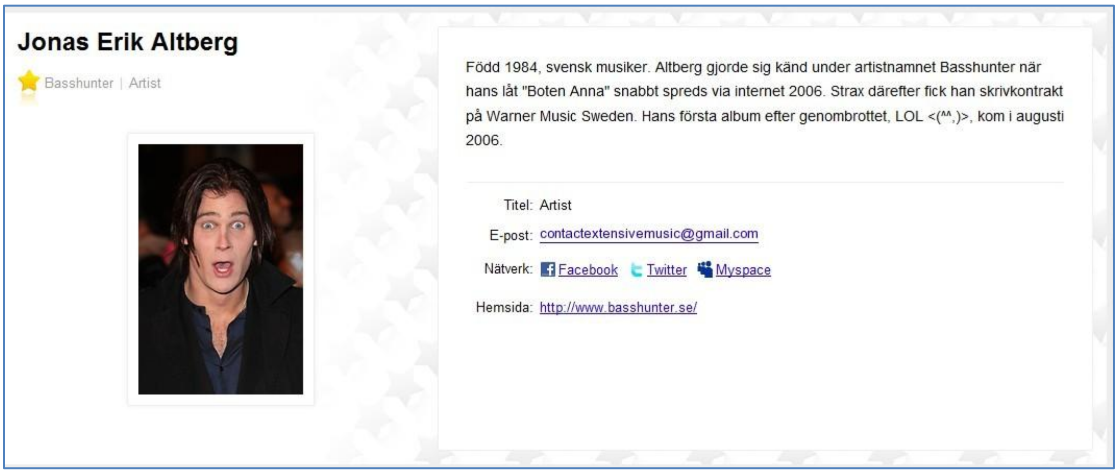
- Eniro

To the maximum extent searching the Swedish people, the local search engine Eniro[23] is introduced. Eniro provides a comprehensive service from business search to person target, the reason to be selected as a powerful tool is that the local database contains much more information consistent to the facts and its searching scope covers all Sweden.

From the Eniro search, the return may include name, telephone number, address, Facebook account, Twitter account, MySpace account and some other related description or messages about people. Among the list of personal information, not all returned results are available in Eniro database.

What should be noted here is that, in some cases, Eniro can also provide email address of the target people. Nevertheless, for security reasons, Eniro display it as a picture format (in .jpg) in HTML page, which make parser itself hard to read the content of picture and even save as an element to fetch more social network information. Because of this, name is considered as the primary element for this task. To understand the picture content afterwards read the related email address will be discussed in future work section.

Take “Jonas Erik Altberg” searching result in Figure 2.2.8 for example:



Jonas Erik Altberg

Basshunter | Artist

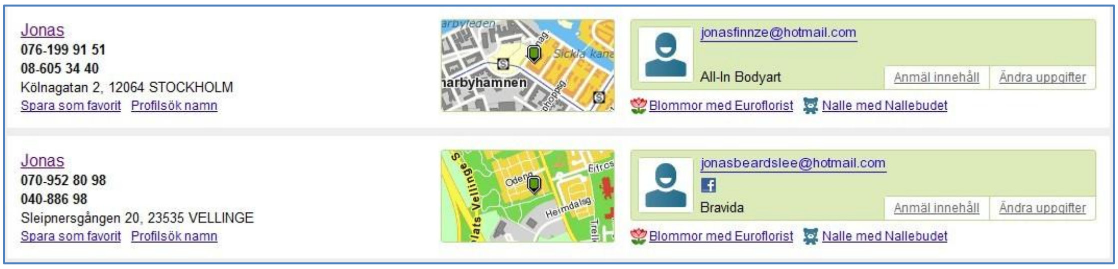
Född 1984, svensk musiker. Altberg gjorde sig känd under artistnamnet Basshunter när hans låt "Boten Anna" snabbt spreds via internet 2006. Strax därefter fick han skrivkontrakt på Warner Music Sweden. Hans första album efter genombrottet, LOL <(^^,)>, kom i augusti 2006.

Titel: Artist
E-post: contactextensivemusic@gmail.com
Nätverk: Facebook Twitter Myspace
Hemsida: http://www.basshunter.se/

Figure 2.2.8: Eniro search result of Jonas Erik Altberg

The search of “Jonas Erik Altberg” consists of full name, title, email (but in .jpg format), the account of Facebook, Twitter, MySpace, personal site and short description. Even if the result gives a lot of information about the “Jonas Erik Altberg”, it still needs to be complemented of address and telephone numbers.

Another vivid example can be showed in Figure 2.2.9 where the ordinary person is not as famous as “Jonas Erik Altberg”:



Jonas
076-199 91 51
08-605 34 40
Kölnagatan 2, 12064 STOCKHOLM
Spara som favorit Profilsök namn

jonasfinnze@hotmail.com
All-In Bodyart
Anmäl innehåll Ändra uppgifter
Blommor med Euroflorist Nalle med Nallebudet

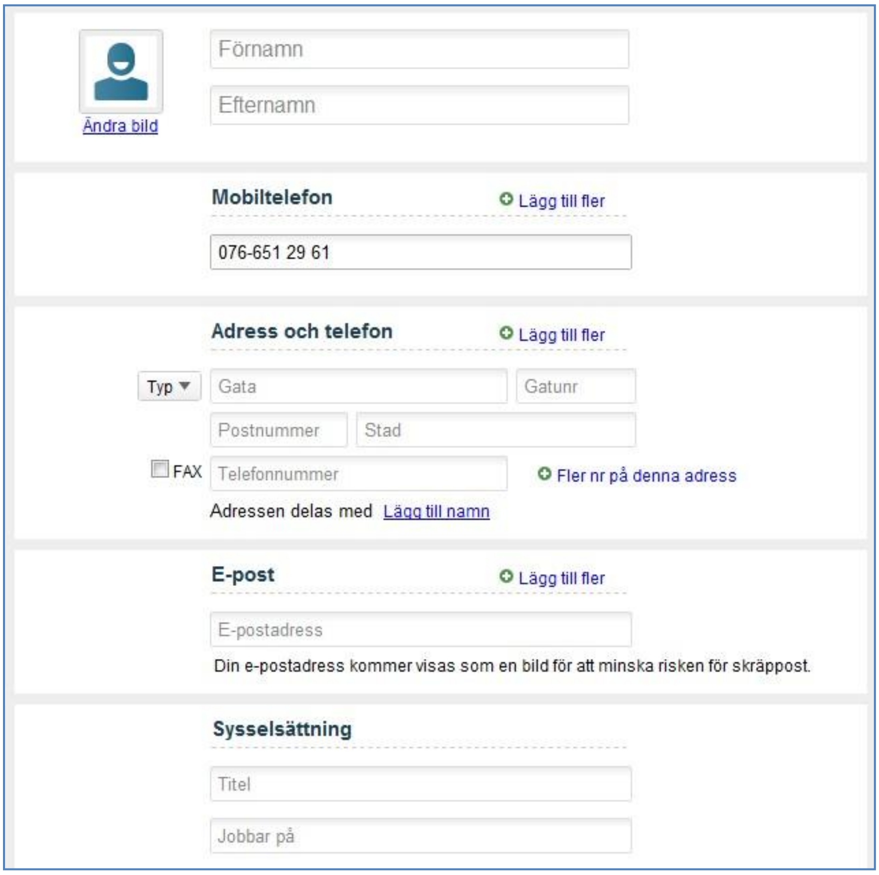
Jonas
070-952 80 98
040-886 98
Sleipnersgången 20, 23535 VELLINGE
Spara som favorit Profilsök namn

jonasbeardslee@hotmail.com
Bravida
Anmäl innehåll Ändra uppgifter
Blommor med Euroflorist Nalle med Nallebudet

Figure 2.2.9: Eniro search result of Jonas

Here, I search one of the most common Swedish names “Jonas”, it gives over 1000 results but I just display part of them. In this example, it is clear that, the subjects of name, telephone number, address and email are directly demonstrating a lack of the list of SN account (except the second Jonas’s Facebook).

The reason for this is that the Eniro database collects the personal information by the online users themselves who add the detail of personal information to Eniro as Figure 2.2.10:



Ändra bild

Förnamn

Efternamn

Mobiltelefon ⊕ Lägg till fler

076-651 29 61

Adress och telefon ⊕ Lägg till fler

Typ ▾ Gata Gatunr

Postnummer Stad

□ FAX Telefonnummer ⊕ Fler nr på denna adress

Adressen delas med Lägg till namn

E-post ⊕ Lägg till fler

E-postadress

Din e-postadress kommer visas som en bild för att minska risken för skräppost.

Sysselsättning

Titel

Jobbar på

Figure 2.2.10: Eniro information gathers

The update and change of information is verified and confirmed by telephone number or cell phone number, which can imply that the contact number in Eniro can be strongly considered as true. However, it does not presume that every search result contains contact number. The pages of famous people in Eniro are operated by Eniro itself, who substitute the address and contact number with description and SNs.

In order to face the searching information insufficient situation, it is essential to utilize the other web search to complement the lacking pattern which also guarantees the final results' completeness.

- Hitta

Hitta is another Swedish local search engine who offers contact number search service, addresses and map locating. Hitta shows the similar common features as Eniro:

1. The online user takes responsibility of updating and changing individual information.
2. The information is verified and confirmed via registered contact number.
3. The return individual information consist of name, address, contact numbers and SNs information if user has registered.

Besides of the similar feathers as Eniro, Hitta offers more precise map locating. For the main municipalities for example Stockholm, Göteborg, and Malmö, the street scene is continuously updated and improved, which leads to locating the address as the format in pictures like Figure 2.2.11:

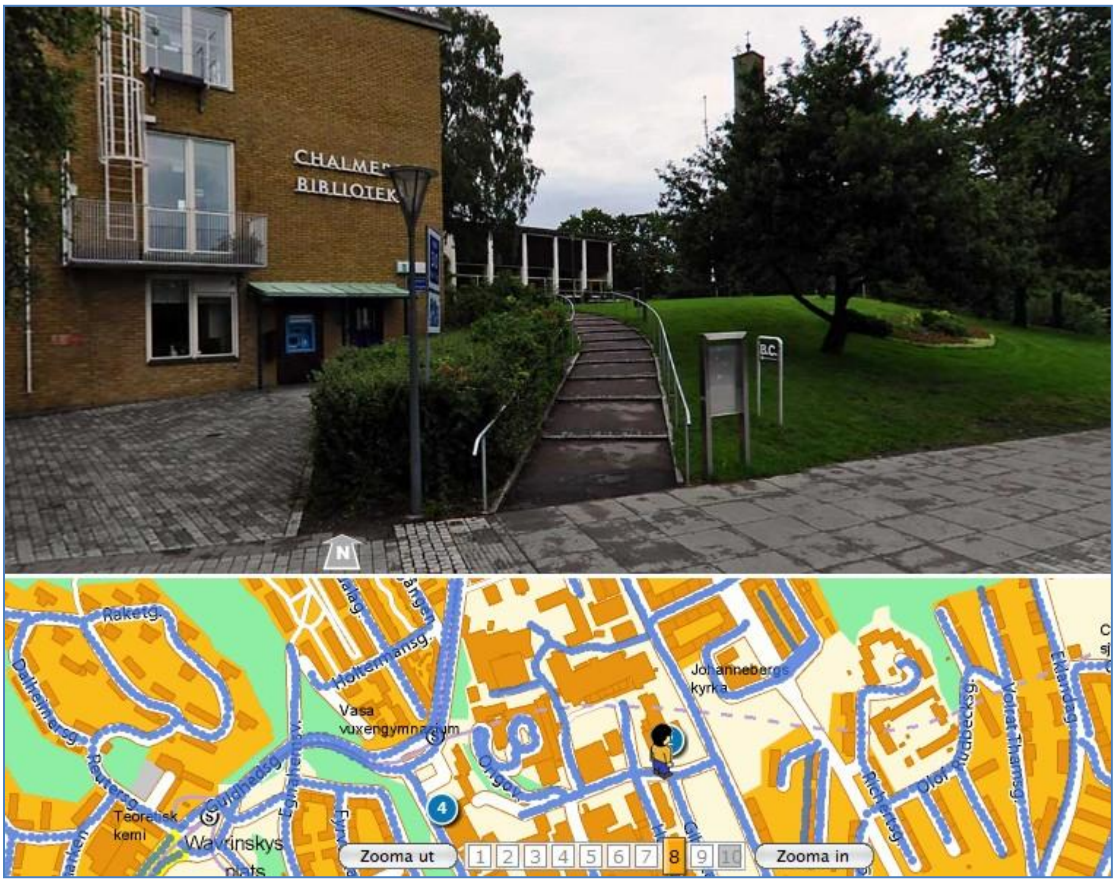


Figure 2.2.11: Street Scene of Chalmers Tvärgatan 1, Göteborg

The picture combined with the address locating makes the address more accurate and vivid.

From the search function aspect, Hitta performs similar to Eniro, but the databases are independent of each other and the data in each database are not exactly the same. Therefore, it is crucial that Hitta synergize with Eniro, existing as the complementary relation.

2.2.3 Ratsit information

Ratsit[24] provides quick, easy and inexpensive access to tax and credit information on individuals and businesses. Searching for people has become easier with more selection such as marital status, age, gender and corporate engagement.

In the light of the Ratsit claims, all the data disclosed on Ratsit are from the authorities, the public records or tax offices. The data cannot be changed by individuals or even Ratsit, and most individuals' information can be searchable except the one which does not have public registration.

2.2.4 Merits and Drawbacks

In view of the above analysis, each database can disclose more or less information from the personal basic information of details as interests or martial situation. In order to indicate the degree of reliability; I give the benchmark from 0 to 5 to measure the data reliability:

0	The data are totally false and easy to be identified to be unreal.
1	The data are false except some pattern may be true.
2	The data seem true but it contains fake which can be easily identified.
3	Most of the data are true but may contain some unreliable pattern.
4	The data are true but hard to prove.
5	The data are true and able to be proved.

Table 2.2.13: benchmark of reliability

In addition, for the degree of data leverage, it measures as L (light), M (medium) and S (serious):

L	Only disclose some information, but hard to ascertain the people in real life.
M	Disclose much enough information but need to combined with public search engines or other resource to determine the identity of the person in real life.
S	It can directly determine the person in real life.

Table 2.2.14: benchmark of availability

Table 2.2.15 shows the results of the analysis on reliability and availability of the data of each database.

	Reliability	Leverage
Facebook	3	M
Twitter	3	L
MySpace	3	M

Eniro	4	M
Hitta	4	M
Ratsit	5	S

Table 2.2.15: the benchmark of each database in Reliability and Leverage

What should be noted in SNs is that since these three subjects' information can be relied on to some extent but different type of SN discloses in various degree. Facebook contains a lot of personal information but due to its privacy setting, utilizing the public search is hard to inspect all the related information except name, friends and head portraits if it is default privacy set, hence it ranks as M.

Even the Twitter is designed to be open, but it has limited information to exploit. Unlike the other SNs, it contains little details about the one which causes difficulties to identify one in real life except following one's Tweets for some time. From tweets, they may imply something related to one's daily life. So it ranks as L.

MySpace is similar to Facebook. Because in most cases, it reveals a lot of personal information, and due to its default privacy setting, many people have no awareness that their information is public. From the profile, it can infer one's interest, ethnic and other information. Therefore, it ranks as M.

2.3 Search Logic

On the basis of the above analysis, Ratsit exploits more explicit information than any other web sources. Hence, it should be considered as the core search resource. Nevertheless, Ratsit can only disclose personal information as name, age, gender, birthday, martial situation and address, about the telephone number, cell phone number and SN accounts, no more information is displayed.

A feasible solution to exploiting individual data in maximum extend is to synergize the Ratsit with other databases, and contrast different data streams in order to seek the common features, and then refine the search result to most optimum.

From the functional side, the logic process involves two main parts: *Fetch* and *Analyze*.

In order to find one target person, the basic information belongs to the one is required. In this thesis, the starting point of information I set is the ***name***.

The Logic diagram mainly consists of two logical components: *Fetch* and *Analyze*.

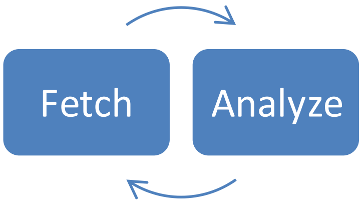


Figure 2.3.1: Functional components of *Fetch* and *Analyze*

The *Fetch* component is responsible for grabbing the intended content from the web. The intended contents require name, telephone number, home and personal SN account. The *Analyze* component mainly focuses on verifying the fetched contents whether it belongs to the target person.

The work flow starts from the fetching, afterwards, the *Analyze* component parse the content to match the possible result and make a judgement according to the new search return. The work flow circuit will be jumped out when the *Analyze* component considers the fetched information belonging to the target person.

From the procedure side, the logic starts from the Ratsit search, after Ratsit has successfully grabbed name, age, birthday .etc. Since the information is not complete, the rest of work to find the contact number is carried out by Eniro and Hitta. SN is fetched by each SN search engines as long as relevant information can be searchable.

The logic of procedure displays as below in Figure 2.3.4:

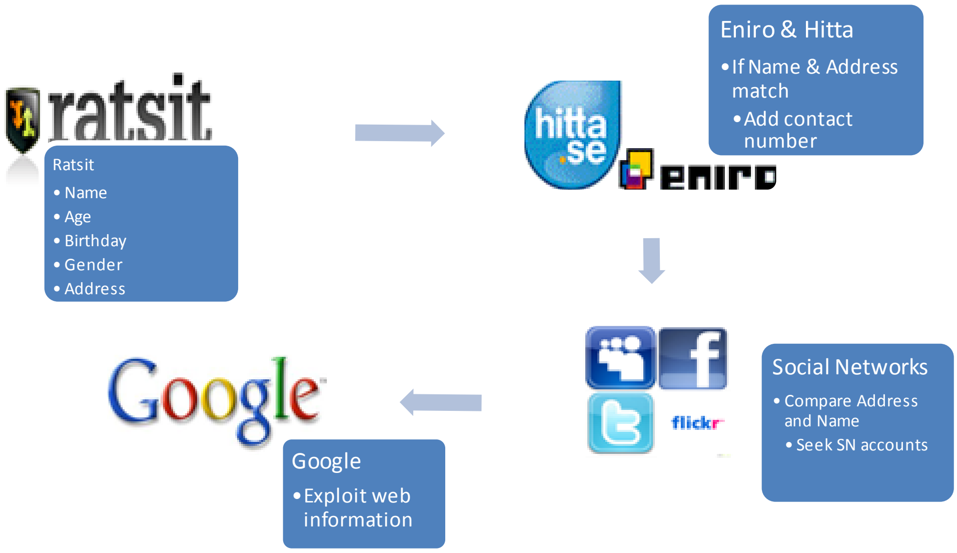


Figure 1.3.4: Procedure Logic

Since Ratsit has disclosed the personal data which can be considered as reliable, the data themselves (like address) can be used as judgment condition to verify whether the Eniro and Hitta search result belongs to one entry from the Ratsit result. Eniro and Hitta attempt to search name and if the return name and address match with the result from Ratsit, it will add the corresponding contact number to personal information, but if nothing is found in neither Eniro nor Hitta, it will be just skipped and the next steps continued. Since Flickr search is quite different from the other search. It has not much attributes to make the person identified except the photos discerning. In that way, it may reference the function of photos distinguishing. In this thesis, it will not be discussed but probably in future work.

The incomplete of SN account is supplemented by Facebook search, Twitter search and MySpace search. The description and photos are supported by Google and Flickr to assure personal information completeness.

Even there are enough support tools to provide a large amount of data, a suitable data filtering and judging mechanism are still wanted to refine the information more close to the reality of the target person.

The verifying process depends on the verifying typical attributes. To verify the SN account, a group of <address, location> is introduced. The address is the subject token from Ratsit, and the location is from SN. In SNS, most of the personal profile requires specifying location attributes, and some of the locations are particular in details, which can be marked as the essential element to distinguish the most likely results among multiple SN accounts.

Based on the location to distinguish may generate another tricky situation, for example, the address is empty even after searching over the ratsit or SN fail to reveal any information related to address. In order to deal with this situation, I define that, in the group of <address, location>, if either of it is empty, the program will default take it as passing and keep the fetched SN account. The idea behind it to keep possible junk account is that, as the location point cannot give any support to make the right judgement, it is better to avoid filtering the potential matched one.

Because this filtering mechanism mainly counts on location attributes, it is very hard to absolutely filter the junk data or target the specified one. Practically, in order to perfect the functionality of filtering analyze mechanism, more attributes need to be introduced as the elements determining the information belongings, whereas it will complex the parser component a lot, not only just extending the text length, but also improving the intelligent selection of key word. Therefore, in this thesis, only the location is set as key element to make a judgment.

2.4 Program Requirement

- **Platform and Programming Language Choice**

In this thesis, the Eclipse is chosen to be as the primary platform of coding due to its powerful support to project management and multiple environment programming.

As described in Core Python Programming[25], Python is a simple but powerful programming language. The advantages make it taken into consideration are:

1. Simplicity

Python represents the idea of simplicity, which shows a great advantage due to its source code like pseudo code. A great effort can be paid to settle down the problem rather than understanding the language itself.

2. Enough Libraries to Get Support

First of all, the Python code is freely available for everyone, which means I can read the source code and modify part of it into a function. Therefore, in this work I can save a lot of energy avoiding constructing every module in the beginning.

Furthermore, even my code is not sufficient to maintain the whole functionality and requires enhancement, there are still a lot of developers' codes and similar task description being public to community, which enlighten me to solve the problem.

3. Portability

Python can be run on multiple systems as Windows, Linux, and Mac .etc. As soon as the program once has been completed, the change of code to transfer to the other platform is minimized.

- **Main Techniques Description**

1. Regular Expression

Regular expressions are powerful and standardized ways of searching, replacing, parsing text with complex patterns of character[26]. In Python, the function is built in RE module. In this thesis work, the relevant regular expression works include parsing URL, Swedish Phone Number, Swedish Address, Facebook, Twitter, and MySpace URLs.

2. HTML Processing

The responsible module is `urllib` and `urllib2` which support downloading data from the URL address, and combining with URL encoding and decoding. In addition, the module named Beautiful Soup[27]also plays an important role in HTML parsing, which provides simple and effective methods for a programmer to search and modify the parsing tree. The specific technical description of this part will follow in next chapter of code analysis.

3. Program and Framework

This chapter describes the details of the program structure and the realisation of key techniques in the whole program.

3.1 Program Structure

The structure of program implements the program logic described in section 2.3. The program first uses Ratsit (see Figure 3.1 ①) and the main steps involving the Ratsit Search are to fill in the client form. Then the branch splits into two parts: if the result returns successfully, it enters into the filter module. The filter module filters results based on the age ranges and location ranges. Otherwise, it goes to the error page with an option back to the start page.

The Eniro (see Figure 3.1 ②) search starts if there are results from the filter module. In this step, the Eniro search module submits the name and address fetched by Ratsit to the Eniro database, attempting to find the telephone number. In many cases, if one registers the information with Eniro, it has a match address in Ratsit. If Eniro cannot provide telephone number, the search goes to Hitta (see Figure 3.1 ③) and repeats the same process as with Eniro, and skips if it still finds nothing.

After the Eniro and Hitta modules have been finished, the SN search module (see Figure 3.1 ④) runs for Facebook, Twitter, MySpace, and Flickr. SN search module search requires filter module to match name and address with result from previous found information. As noted before, it may keep many unproved SN accounts due to SN privacy setting, but it covers all the possible results.

Finally, the Google search module (see Figure 3.1 ⑤) is the supplement of all information found, which scans web content for the targeted person.

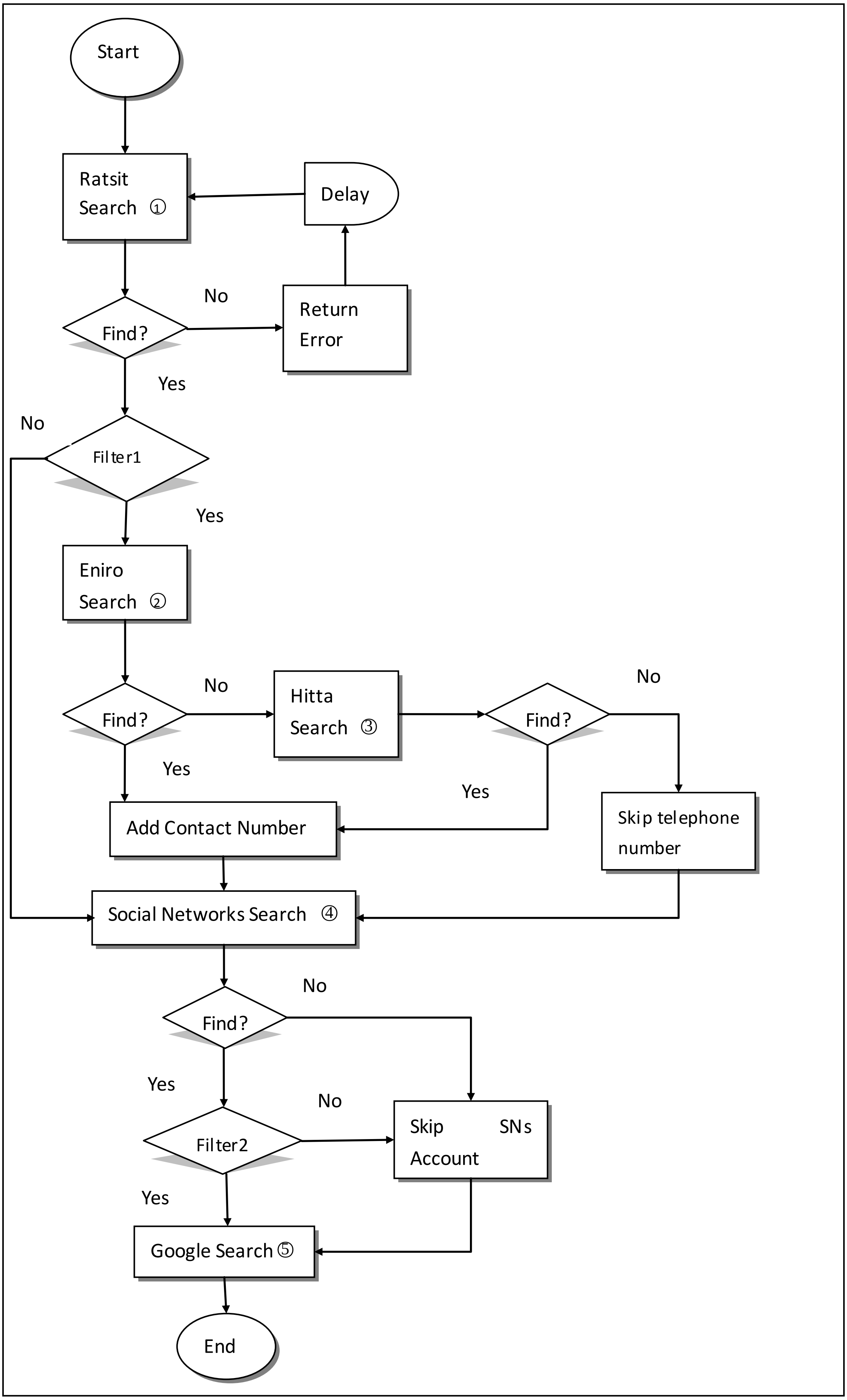

Start
Ratsit Search ①
Delay
No
Return Error
Find?
Yes
No
Filter1
Yes
Eniro Search ②
No
Hitta Search ③
Find?
No
Find?
Yes
Yes
Add Contact Number
Skip telephone number
Social Networks Search ④
No
Find?
Yes
No
Filter2
Skip SNs Account
Yes
Google Search ⑤
End

Figure 3.1: The Structure of Program

The filter module in the structure of program performs different function. The filter1 is mainly designed for users' specifications. In specifications, the user can limit the search in age ranges and give a location with coverage of distance in kilometres. If the person is not found regarding to the specification, it goes directly to the SNS module to proceed. Otherwise it goes to Eniro.

The filter2 is primarily to remove the irrelevant SNS account. If the SNS can be accessed and declares the location attributes, the filter2 will be based on the location to match the fetched account and remove the inconsistent one.

3.2 Python Programming and Regular expression

In this thesis, Python is introduced to implement submitting and fetching content on the website. In referenced Python library are urllib, urllib2, re, ClientForm, cookielib, BeautifulSoup and some package from *Python.org*[28]. Each Python library functions differently regarding web requesting, client form filling in, regular expressions and web content parsing.

3.2.1 Client Form in Python

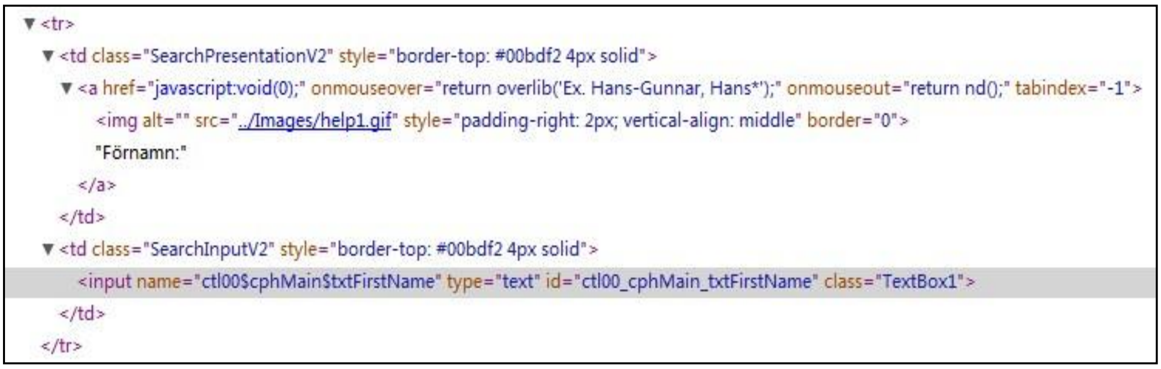
In order to submit a query to the server, the process firstly has to parse HTML forms filling on the client side, and then fills in the value which can be name or location. The final step is delivering the filled form to the server.

The handling library for the whole process is urllib, urllib2, ClientForm and cookielib.

The urllib and urllib2 are both modules reading server data from the URLs, the task for which is to establish connections and retrieve the data from the server web page[29][30] . In order to submit the query and fetch data from the response page, this part of the code actually behave as a browser, which means appropriate cookie handling is indispensable.

The cookielib takes responsible for handling cookies. A cookie contains bits of information sent by the server combing with the web page, and then return back to the server in next HTTP request[31]. Since the next step involved is fetching web page, it is safety to make sure the page keeps constant and the server identifies the session. The cookielib enable all cookies in the session appropriately handled.

Before submitting query to form field, it benefits a lot to know the structure of the form as HTML code:



```
▼ <tr>
  ▼ <td class="SearchPresentationV2" style="border-top: #00bdf2 4px solid">
    ▼ <a href="javascript:void(0);" onmouseover="return overlib('Ex. Hans-Gunnar, Hans*');" onmouseout="return nd();" tabindex="-1">
        <img alt="" src="../Images/help1.gif" style="padding-right: 2px; vertical-align: middle" border="0">
        "Förnamn:"
      </a>
    </td>
  ▼ <td class="SearchInputV2" style="border-top: #00bdf2 4px solid">
      <input name="ctl00$cphMain$txtFirstName" type="text" id="ctl00_cphMain_txtFirstName" class="TextBox1">
    </td>
  </tr>
```

Figure 3.2.1: part of sample code of ratsit form

The relevant line is the `input`, which is the field to fill in the web form. Rather than filtering the unusable code by programmer, the `ClientForm` can automatically locate the form field in web page using class `ParseResponse`, and the form is extracted as:

```
<aspnetForm POST http://www.ratsit.se/BC/Search.aspx
application/x-www-form-urlencoded
  <HiddenControl(ctl00_RadScriptManager1_TSM=) (readonly)>
  <HiddenControl(__VIEWSTATE=/wEP…8XogdvcM=) (readonly)>
  <TextControl(ctl00$cphMain$txtFirstName=)>
  <TextControl(ctl00$cphMain$txtLastName=)>
  <TextControl(ctl00$cphMain$txtBirthDate=)>
  <TextControl(ctl00$cphMain$txtAddress=)>
  <TextControl(ctl00$cphMain$txtZipCode=)>
  <TextControl(ctl00$cphMain$txtCity=)>
  <TextControl(ctl00$cphMain$txtKommun=)>
  <CheckboxControl(ctl00$cphMain$chkExaktStavning=[on])>
  <ImageControl(ctl00$cphMain$cmdButton=)>>
```

The Ratsit form gives you many search option for one to customize the query, but the value needed to be passed to the server is the first name and second name. For returning the value to server, the '*control*' text instance in extracted form is the place to be filled in the value. The submit process is to click the '`ctl00$cphMain$cmdButton`' and after the value passed to the server, the result page is returned.

The returned page is processed by `urllib2.urlopen ()`. After reading the page into memory, the parser of web page is next job of the whole process. In next sector, the related parsing procedure and referenced library will be explicitly explained.

3.2.2 Relevant Content Parsing

The web content parsing is the crucial point of this thesis, the main parser function in this thesis work relates to Eniro parser, Ratsit parser and Hitta parser.

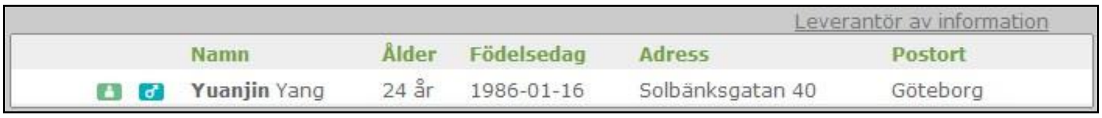
The library responsible for processing returned HTTP is `BeautifulSoup`. As a HTML parser, `BeautifulSoup` offers a powerful interface to parse the web data structure with service of searching and navigation of tags and HTML entity.[32]

1. Ratsit parser

Since the form is submitted via transmit the value of '`ctl00$cphMain$cmdButton`' to the server, the returned content is read and stored as HTML code in the cache. The HTML code can be processed by transferring the HTML code into string, and then using `string.find ()` functions to find the needed values, or dealing with the whole content by regular expression library `re`. In `re`, there are methods named `re.search()` and `re.findall()` to scan all string with the purpose of matching patterns.

However, both of the methods above involve complex coding in line with the web page data structure. Besides, dealing with the special character and building loop methods to compute may inevitably generate a lot of long and ineffective codes.

Before `BeautifulSoup` parsing the code, it is good to show the returned HTML page and code:

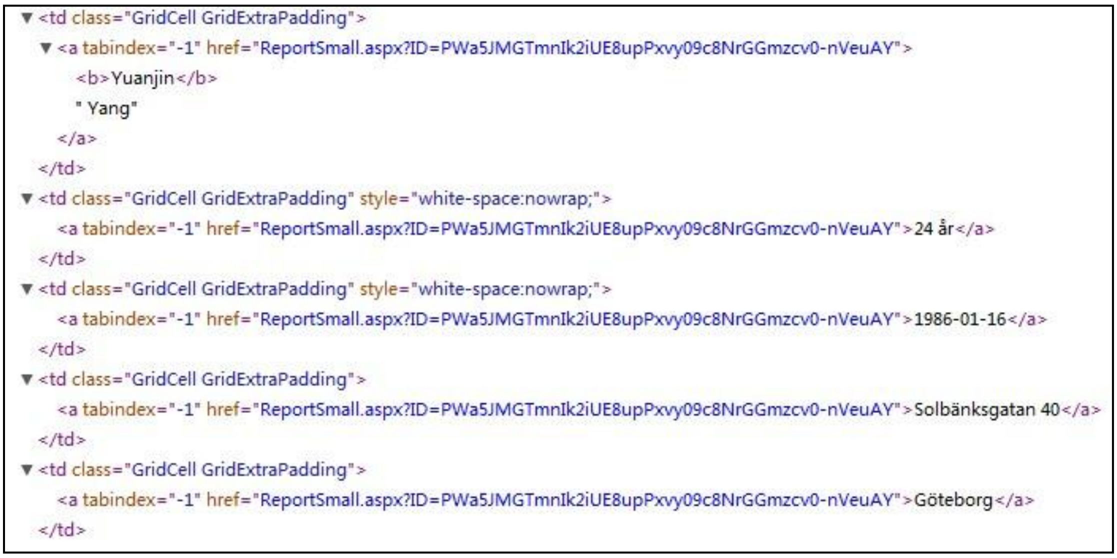


Leverantör av information

	Namn	Ålder	Födelsedag	Adress	Postort
	Yuanjin Yang	24 år	1986-01-16	Solbänksgatan 40	Göteborg

Figure 3.2.2: Ratsit returned page with query "yuanjin"

The result is displayed in a table and below there is the relevant source code:



```
▼<td class="GridCell GridExtraPadding">
  ▼<a tabindex="-1" href="ReportSmall.aspx?ID=PWa5JMGTmnIk2iUE8upPxvy09c8NrGGmzcv0-nVeuAY">
      <b>Yuanjin</b>
      " Yang"
    </a>
  </td>
▼<td class="GridCell GridExtraPadding" style="white-space:nowrap;">
    <a tabindex="-1" href="ReportSmall.aspx?ID=PWa5JMGTmnIk2iUE8upPxvy09c8NrGGmzcv0-nVeuAY">24 år</a>
  </td>
▼<td class="GridCell GridExtraPadding" style="white-space:nowrap;">
    <a tabindex="-1" href="ReportSmall.aspx?ID=PWa5JMGTmnIk2iUE8upPxvy09c8NrGGmzcv0-nVeuAY">1986-01-16</a>
  </td>
▼<td class="GridCell GridExtraPadding">
    <a tabindex="-1" href="ReportSmall.aspx?ID=PWa5JMGTmnIk2iUE8upPxvy09c8NrGGmzcv0-nVeuAY">Solbänksgatan 40</a>
  </td>
▼<td class="GridCell GridExtraPadding">
    <a tabindex="-1" href="ReportSmall.aspx?ID=PWa5JMGTmnIk2iUE8upPxvy09c8NrGGmzcv0-nVeuAY">Göteborg</a>
  </td>
```

Figure 3.2.3: an excerpt of source code of Ratsit search "yuanjin"

From the source code, it is obvious that the returned information is under `class` of "`GridCell GridExtraPadding`". The most effective way to parse the related information is to get all class of `GridCell GridExtraPadding` out from the page and then remove the tags.

`BeautifulSoup` can meet all the demands in the above analysis. The steps are:

```
page = urllib2.urlopen(p).read()
soup = BeautifulSoup(page)
form = soup.findAll('td',attrs={"class":"GridCell GridExtraPadding"})
```

The type of `form` is actually the class of `BeautifulSoup.ResultSet`. The format is set to be as a list, in which contains all the tag starts in "td" and with class of "`GridCell GridExtraPadding`".

One problem here about the HTML code is the character encoding problem. Since Swedish alphabet contains three more small and capital different letters: ö, ä, å, Ö, Ä and Å. Unless the web pages specify the encoding format as "ISO8859-1" or "Latin-1"[33], otherwise HTML entity replace function has to be introduced. In some situations, `BeautifulSoup` cannot recognize the encoding or the web page omit the encoding declaration. Instead, it will encode the character as 'UTF-8', which causes the terminal display and parser unable to discern the relevant content. The cases not only exist in the Ratsit parser but also in the other parsers, and sometimes the situation turns more complex than Ratsit. When it comes to Eniro parser, we will look deep into encoding issues.

In Ratsit, the tag removing function is done by self-defined function `html_to_text`. The main job of `html_to_text` is to remove all the tags and replace special character in HTML entity format into relevant symbols.

The tags to remove requires regular expression `re.sub (r'<.*?>', '', s)`. This expression means replacing all the tags into space if there is possibility to match. The special character in web page in a normal case is expressed as HTML entity, but the problem occurs for example: the city "Göteborg" in source code displays as "G&ouml;teborg", if using expression "Göteborg" to match, it will return none due to the character is not transformed. Hence, character replace to some extent determine whether the matching can be successful.

Finally, when finishing the decoding and removing part, the return contents have been parsed into a list named "`ratsit_info`". Since all the information is stored in one list, which causes trouble in displaying on the pages, grouping and classifying the content can make things easier.

As all the information about one person in Ratsit is shown in five columns: name, age, birthday, address, and city, it is simple to construct a function to split the list with every five elements in a group.

The code is displayed below:

```python
def split_list(L, n):
    assert type(L) is list, "L is not a list"
    for i in range(0, len(L), n):
        yield L[i:i+n]
for i in split_list(ratsit_info,5):
    ratsit_personinfo.append(i)
```

The key word `yield` in Python performs like a return but the function is not exactly the same. Instead, it will return a generator other than just one value. The reason for that is if return is used, only one value can be transmitted. The yield overcome these drawbacks and will generate multiple values unless given a condition to jump out.

Having grouped the elements of every five on a new list, each list can then be added into `ratsit_personinfo`. The result of a list contains one or more lists and each represents one possible hit search result. The limitation of an overfull return result is implemented by inputting more information in location field, age ranges rather than just giving a name.

2. Eniro parser

The Eniro search is primary searching the contact number on the basis of Ratsit search. For every item, after input the name and address from Ratsit, Eniro returns the found person information. In this process, the Eniro parser is prerequisite for the rest of the work.

In Eniro search, a query string is part of URL when you press the submit button, which is flexible to retrieve data from Eniro database. The first step, design a URL which contains the query contributing a lot to parser.

It is simple to attempt some search examples to find out how Eniro URL is constructed. And the format for Eniro URL is

```
http://personer.eniro.se/query?what=wp&lang=&ax=&search_word=geo_area
=.
```

Obviously, `search_word` is the place for entering the search name, and `geo_area` is the location field limitation.

Since some character cannot be recognized by URL and some of them for URL have its own meaning, the search word should be properly sanitized replacing the space character with "+" and selecting the encoding format.

The HTML reading and saving process is done by `urllib.urlopen ()` and `BeautifulSoup ()` as Ratsit:

```
content = urllib.urlopen(URL.encode('UTF-8')).read()
content = re.sub(r"document.write.*?pt>'.*;", '', content)
soup = BeautifulSoup(content)
```

What should be noted is that the Eniro page contains a special JavaScript code:

```
document.write('<scr' + 'ipt type="text/javascript"
src="http://eas8.emediate.eu/eas?cre=mu;js=y;encoding=utf-8;cat1=REG0
0;EASInclude=;EASInclude2=mona%20sahlin;sw2=mona%20sahlin;cu=1504;sw=
;target=_blank;EASTpecresp=' + adptpecresp + ';misc=' + (new
Date()).getTime() + '"></scri' + 'pt>');
```

In this code, the tag `<script>` is split for its own purpose of use. However, it leads to `BeautifulSoup` unable to handle that because of the splitting tag. For this reason, `content` has to be properly managed to remove the harmful and useless code by regular expression function `re.sub()`.

Eniro search result divides people as normal and "kändis" (means "famous" in English). For normal item, which contains name, telephone number, address and SN account (if one register). For "kändis", it only gives SN accounts and a short description.

Even contents are different, in HTML source code, they both have the tag which has `title="Visa mer"`. As in Ratsit, `BeautifulSoup` provides `findall()` to handle this situation. The information needed for next parsing is the name and URL redirecting to the personal information page.

The URL is extracted by the third party module `htmldata.py`. Since the URL is extracted from the web page, it can be redirected to the personal information page to fetch the relevant information. The parser for personal page is defined by the function `eniro_parser ()`.

The name in web page is in tag:

```
<span class="given-name">given_name</span>
<span class="family-name">family_name</span>
```

It can use `find` function setting the attributes as *"class":"given-name"* or *"class":"given-name"* to parse first name and second name, respectively. And then add two strings into one.

The telephone number in source page has in tag "`<span class="tel">`". Parsing as name field, set the attributes and add the strings into the list.

Besides the similar steps in name and contact number, the address parsing requires a modifying part of the letter encoding. Due to `BeautifulSoup` encodes the HTML code as UTF-8 to process[34], when displaying, some Swedish letters are mistakenly transformed into other characters.

When the name, contact number and address are appropriately handled, the Eniro parser is actually finishing the task.

3. Hitta parser

Hitta is serving as a complement to Eniro search for telephone number found. Besides, in map exporting, Hitta contributes a lot to giving a living example of parsed address. In next chapter, some pictures will reveal the related function.

In most lines of the code, Hitta behaves similar to Eniro, using `BeautifulSoup` to submit the query to the server and then parse the relevant field. Except parsing the contact number utilizes regular expression.

3.2.3 Regular expression

The contact number in Sweden contains area code and numbers. The area code varies from two to five digits, and the rest may have five to eight digits separated by a dash line.

Here are some examples from the parser:

```
Anna-Mona Nysten Telefon: 0580-20526 Mobil: 070-2345384
Mona Mona Mobil: 070-6566429
Hans Jonas Magnusson Telefon: 0122-16175 Mobil: 070-7708699
Erik Axel Jonas Lander Telefon: 0565-15135 Mobil: 070-6003915
Carl-Jonas Berggren Telefon: 0322-51405
Bernt Jonas Olsson Telefon: 08-962759 Mobil: 070-6962759
```

From the examples, we can see that the digit occurs more than one time before the dash line, and after the dash line, the digit also repeats many times. Hence, the regular expression can be deducted: `'(\d+\-\d+)'`. But in some situations, one has more than one contact number, in order to face that, we can simply make the existing expression repeatedly to match as `'(\d+\-\d+)+'`.

3.2.4 Social Networks Search and Google Search

The Social Network search in this thesis involves Facebook search, Twitter search Flickr search and MySpace search. Twitter, Flickr and MySpace are parsed via a module `websearch.py` (a module has been improved from web_search.py by Connelly Barnes),[35] which is a module consisting of several search engines with the construction of regular expression, character parsing and URL encoding.

Apart from the other SN search, Facebook is implemented by an independent third-party Facebook API application.

1. Facebook Search

Facebook Search implementation requires authentication as a user to exploit web information. The purpose for this is to protect the privacy of users. The auth utilizes a OAuth 2.0 protocol to maintain the security[36]. Using OAuth 2.0, it will generate an access token for users to link to Facebook. Only through the access token distributed by OAuth 2.0, the programmer can initiate an authorized request with the access token in URL.

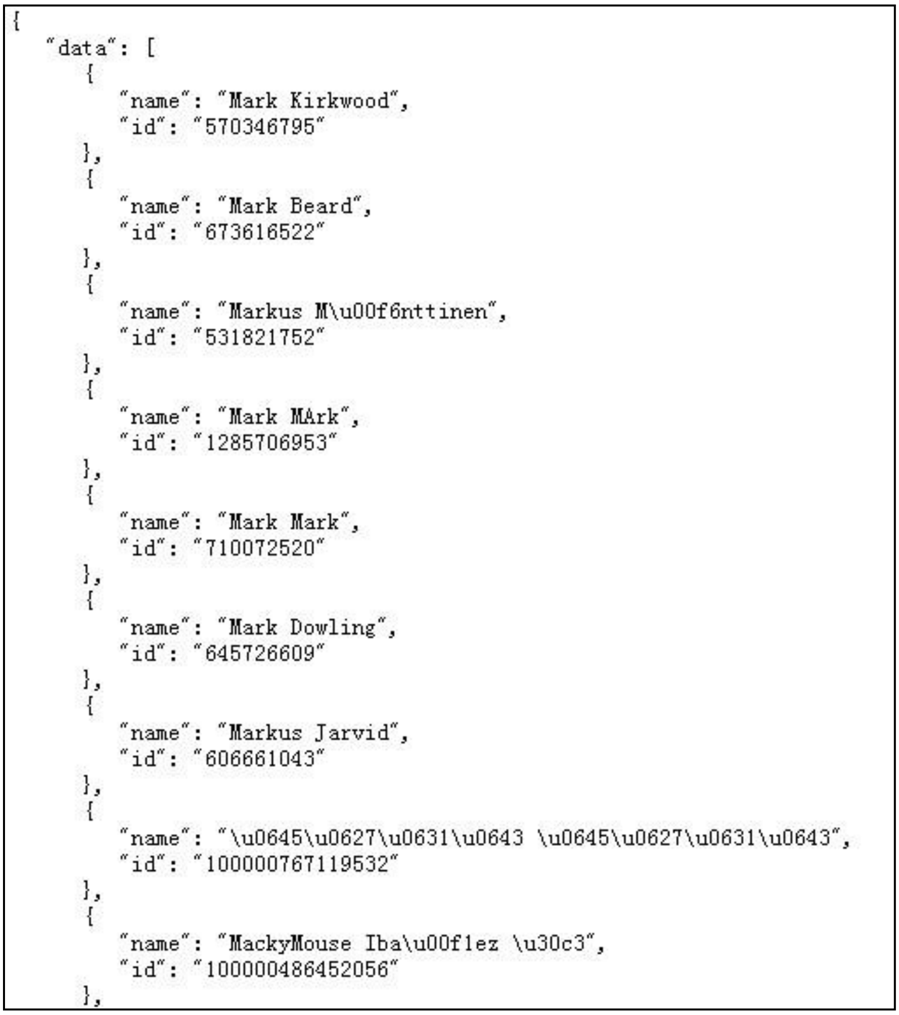
In order to have an access token, the developer has to register the application to get an app ID and secret, the authorizing process is via redirecting "`https://graph.facebook.com/oauth/authorize`" and passes the values to get the arguments.

From the analysis, we can obtain the URL format to search people in Facebook:

```
"https://graph.facebook.com/search?q="+Search_Name+"&type=user&access
_token="+facebook_token.
```

In URL, `Search_Name` is the value query needed to pass to the Facebook server and `facebook_token` is the access token.

After passing the `Search_Name` to Facebook server, a list of names will be returned, and here we give a section of name of the "mark" search result:



```
{
   "data": [
      {
         "name": "Mark Kirkwood",
         "id": "570346795"
      },
      {
         "name": "Mark Beard",
         "id": "673616522"
      },
      {
         "name": "Markus M\u00f6nttinen",
         "id": "531821752"
      },
      {
         "name": "Mark MArk",
         "id": "1285706953"
      },
      {
         "name": "Mark Mark",
         "id": "710072520"
      },
      {
         "name": "Mark Dowling",
         "id": "645726609"
      },
      {
         "name": "Markus Jarvid",
         "id": "606661043"
      },
      {
         "name": "\u0645\u0627\u0631\u0643 \u0645\u0627\u0631\u0643",
         "id": "100000767119532"
      },
      {
         "name": "MackyMouse Iba\u00f1ez \u30c3",
         "id": "100000486452056"
      },
```

Figure 3.2.4: Facebook search result for "mark"

The data exploit for name and id respectively, and with the `id`, a URL to one's page can easily be constructed as "`http://www.facebook.com/profile.php?id=`". Besides redirecting to the page, personal pictures as head portrait can also be referenced by `id`.

Since Facebook privacy setting in most cases blocks the other information disclosing to public, Facebook parser has its own limitation to fetch enough information for identifying person.

2. Twitter, MySpace and Flickr Search

Twitter, MySpace and Flickr search are accomplished as search engines in `websearch.py`. When it finds the content, it returns as a tuple consisting of name, URL and description. Besides the normal function via browser search, it can restrict the quantity of return results, which avoid excessive results shown in pages and saving much of process time.

In websearch.py, all search engines are functioned by a key function -- `make_searcher()`.

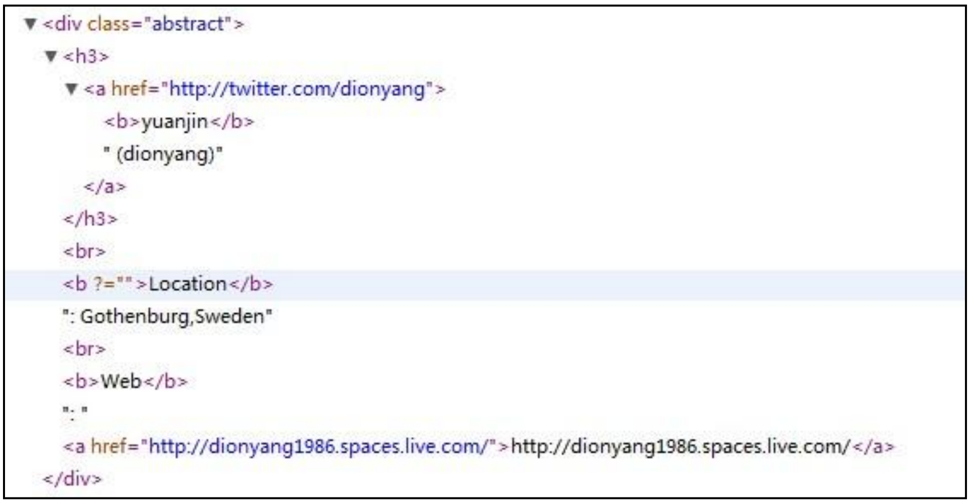
`make_searcher()` contains seven arguments : `query_url, results_per_page, page_url, page_mode, begin, end,` and `link_re`.

query_url is the initial search URL
results_per_page is returned results shown in every page.
page_url is the URL for the rest of the pages.
page_mode controls value of the page "number"
If begin is not None, then only text after the first occurrence of begin will be used in the search results page.
If end is not None, then only text before the first occurrence of end will be used.
link_re is the regular expression string trying to match "name", "url" and "desc".
In general, begin and end is set to None, and results_per_page gives ten as default use.

Twitter search references "http://tweepz.com/" as the query_url. In URL, we can set the search scope to Sweden and intended search name as query string:

```
'http://www.tweepz.com/search?q=loc:(Sweden)+name:%(q)s'.
```

The regular expression requires to web source page:



```
▼ <div class="abstract">
  ▼ <h3>
    ▼ <a href="http://twitter.com/dionyang">
        <b>yuanjin</b>
        " (dionyang)"
      </a>
    </h3>
    <br>
    <b ?="">Location</b>
    ": Gothenburg,Sweden"
    <br>
    <b>Web</b>
    ": "
    <a href="http://dionyang1986.spaces.live.com/">http://dionyang1986.spaces.live.com/</a>
  </div>
```

Figure 3.2.5: source code of twitter search of "yuanjin"

From the source code, URL is in the tag <a> and name is in its subset tag <b> , Location and Web are in <b> that we take them as description; hence, we can construct the expression as:

```
"r'<h3><a href="(?P<url>.*?)".*?>(?P<name>.*?)</a>' +
r'(?P<desc>.*?)</div>')".
```

Someone who registers Twitter may only use first name, in thesis code, I set if search for full name is none then continue to search first name, the process stops until first name still cannot be matched.
MySpace and Flickr perform similar to Twitter search except that the query_url is set to different sites and regular expression is designed based on the web source code. What need to be specified are Flickr's name, URL and description. The name and

description of Flickr is omitted for the reason that in final demonstration they are unimportant than picture itself. And URL is the web address of each picture, which can be directly embedded into web page.

3. Google Search

The Google Search is implemented almost the same as SN search engines. In the whole search process, Google is served as a comprehensive search for the target people disclosing information on the cyber world. The drawback of Google search is that I can do little effort combing Google search data flow with the other search data flow, which directly makes Google search a little bit of independent of the other search, and limits the power of Google search capability.

3.2.5 Filtering Mechanism

The filter function in the whole process involves specification for age ranges and the location with coverage of distance in kilometres. The age ranges can be simply used as `if` statement to handle:

```
if age <= int(to_age) and age >= int(from_age):
```

However, the distance specification requires `GoogleMaps` to calculate the exact the length of distance and make a judgement to filter. The `GoogleMaps` demands API key for the length calculation. In order to request API key, one should register on Google, since the key is private, in code it will not display to public. The process of calculation is:

```
gmaps = GoogleMaps(api_key)
directions = gmaps.directions(start,end)
meters = directions['Directions']['Distance']['meters']
```

What should be noted here is the exception of `GoogleMaps error 602`. The reason causing the problem occurred is that, the address cannot be recognized by `GoogleMaps API` and `GoogleMaps API` prefers a more complete address containing street name, city, which might work in some address but according to Google Group online discussion, there are still many addresses arising the errors. Therefore, for some of addresses, there is no general way to handle this. In the thesis, if the distance can be calculated, it adds the country behind the string, while if it still fails to find, this personal information is printed out as output anyway except printing a note:

```
Note that this address can't be processed by GoogleMap, Please check the
address.
```

The SNS filtering is based on the location from the user specification and fetched content. If there is no address showed in the SNS, the filter just skips the account. Otherwise it removes the unqualified the account.

4. Windows Implementation

The fourth chapter mainly explains the CGI implementation and the demonstration of relevant search result. Moreover, it illustrates the function of JavaScript and CSS working on HTML frames.

4.1 CGI

4.1.1 CGI Introduction

CGI is the Common Gateway Interface on which the web server interacts dynamically with the clients. The process of CGI working starts when the web server receives a request from client side (GET or POST). The web server will invoke on application to handle the request and then wait for the respond of HTML frame. Once the application finishes, it returns the dynamic pages to server, which are by means of the server, resent to the client. The Figure 4.1.1 illustrates the CGI process:

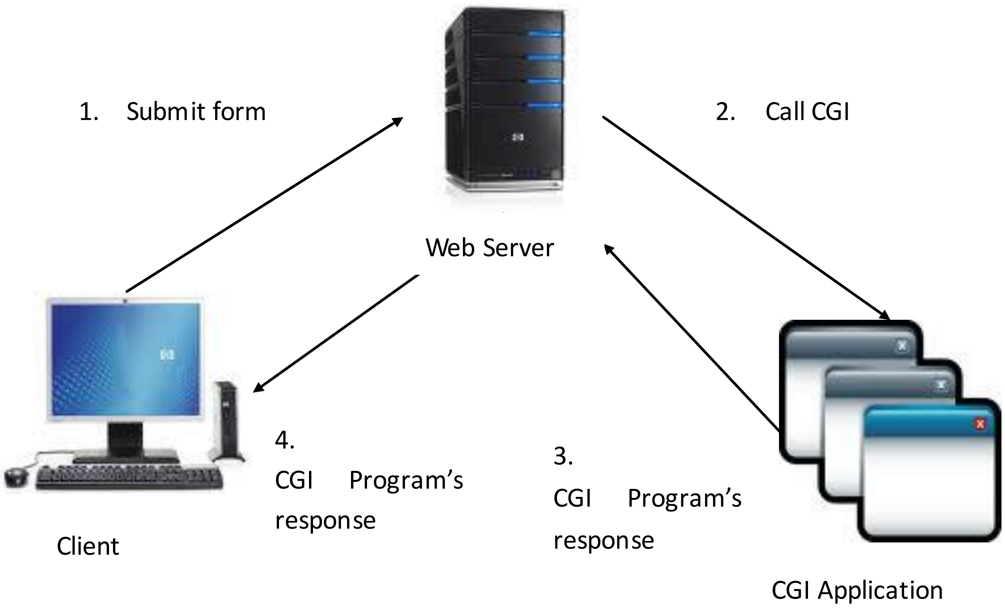


Figure 4.1.1: CGI working process, transferring information between server and CGI program.

Then the CGI program executing consumes server process time and memory space. If millions of requests simultaneously are sent by clients, the server can be overloaded. CGI is more suitable for developing small scale web.

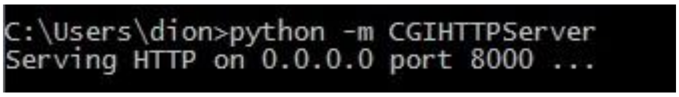
In next sector, I will concentrate on the assistance of module `cgi`, and answer how python can build a `cgi` for this thesis.

4.1.2 CGI Module

The Python module `cgi` supports the CGI programming and implementation. The main class in cgi module can archive most of the functionality. When CGI script is invoked, this class is instantiated, reads the related information from the web client and contains an object like a Python dictionary with the key and corresponding value.

And the key is the name of form items, the value is the data.

Since the purpose of this thesis is to build a small scale web apt to serve people's search, Python itself already has the modules to support related functionality. In this task, Python command can be directly input in the terminal, and a server based on the web is built.



```
C:\Users\dion>python -m CGIHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Figure 4.1.2: configure web server

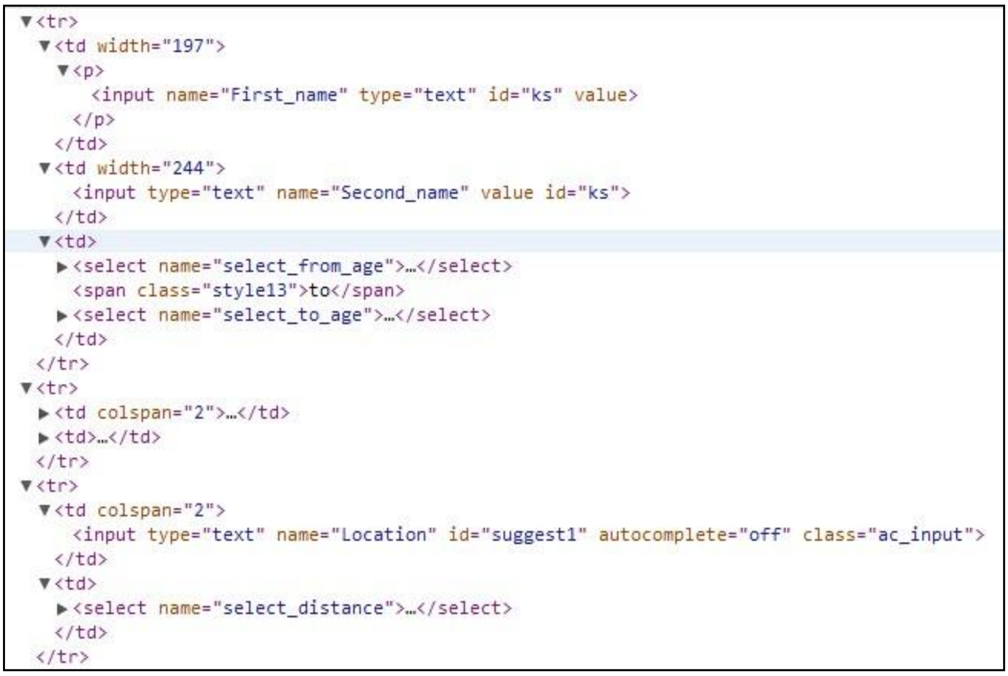
After the command is issued, it has built a web server with port 8000 in the current category. In order to initial the service, a file in the category named `cgi-bin` needs to be produced for CGI script and HTML documents.

To visit the web site, just input the following address:
http://localhost:8000/Desktop/web_portal/Main.htm
http://localhost:8000/Desktop/web_portal/FAQ.html

4.1.3 CGI Programming and HTML Implementation

1. Create Form

The `Main.html` is needed to be generated for CGI script. As shown in the code below from `Main.html`, the frame includes several variables: `First_name`, `Second_name`, `Select_from_age`, `Select_to_age`, `Location` and `Distance`. These values are transmitted to CGI script when it is executed. The `First_name` and `Second_name` are intended to submit the name of search query, meanwhile, `Select_from_age` and `Select_to_age` are specified for age ranges for searching. `Location` is the municipality selection for Sweden area with a search limitation of `distance` range.



```
▼<tr>
  ▼<td width="197">
    ▼<p>
        <input name="First_name" type="text" id="ks" value>
      </p>
    </td>
  ▼<td width="244">
      <input type="text" name="Second_name" value id="ks">
    </td>
  ▼<td>
    ▶<select name="select_from_age">…</select>
      <span class="style13">to</span>
    ▶<select name="select_to_age">…</select>
    </td>
  </tr>
▼<tr>
  ▶<td colspan="2">…</td>
  ▶<td>…</td>
  </tr>
▼<tr>
  ▼<td colspan="2">
      <input type="text" name="Location" id="suggest1" autocomplete="off" class="ac_input">
    </td>
  ▼<td>
    ▶<select name="select_distance">…</select>
    </td>
  </tr>
```

Figure 4.1.3: part HTML code from Main.html

The CGI script initializes default in the host named `cgi-bin` file, and what should be noted here is the method `POST` for submitting the HTML forms. There are two main reasons for using `POST` in `Main.html`:

- The query might contain non-ASCII character, which makes `METHOD = "GET"` inapplicable theoretically even if it works when query is submitted. The alternative of `POST` also reduces the risk of the failure of server processing.
- Another constraint to avoid using `GET` is due to the long length of URLs. `GET` could rise practical problem when many systems are implemented as certain length of URL to handle. Even through the total size of the query URL itself is not over 1 KB but it may be encoded resulting in oversize.

Then as the Figure 4.1.4 shows the form representing in Windows system, Chrome browser:

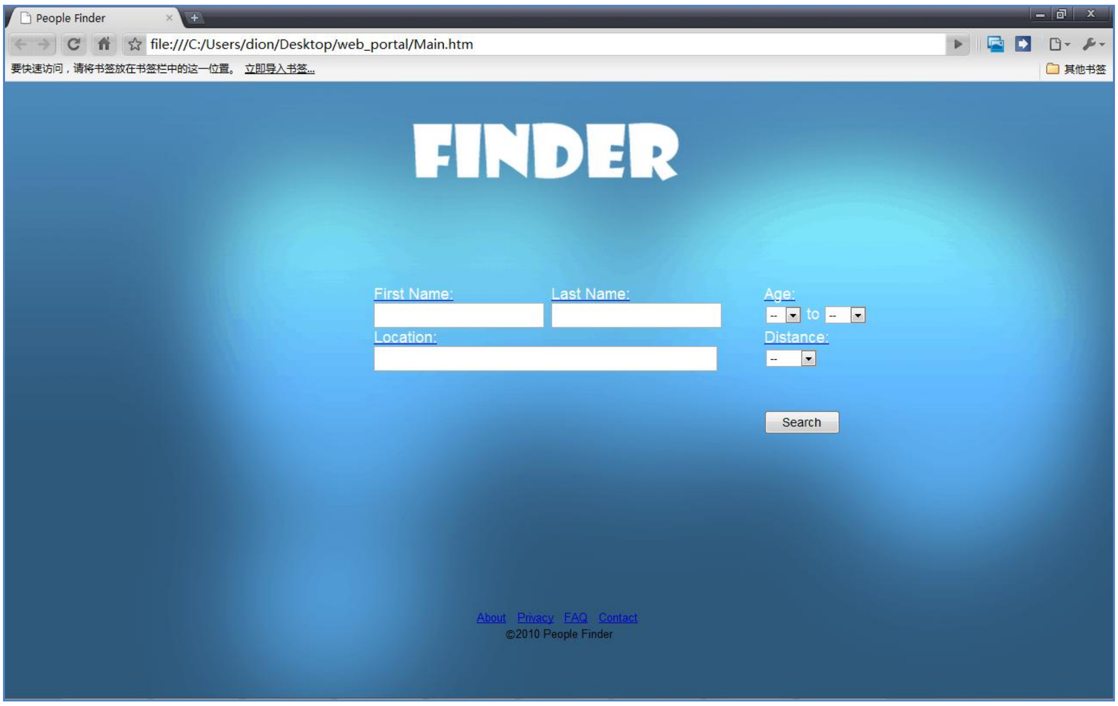
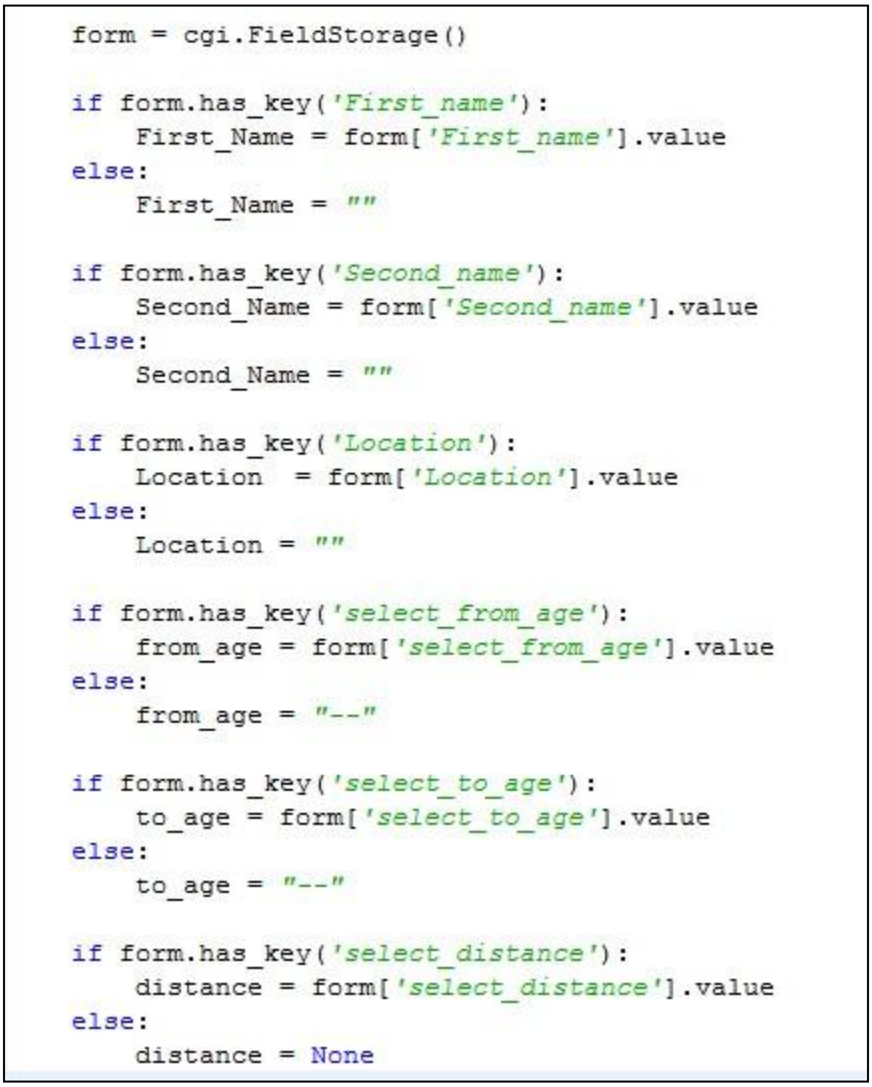

People Finder
file:///C:/Users/dion/Desktop/web_portal/Main.htm
要快速访问，请将书签放在书签栏中的这一位置。 立即导入书签...
其他书签
FINDER
First Name:
Last Name:
Age:
-- to --
Location:
Distance:
--
Search
About Privacy FAQ Contact
©2010 People Finder

Figure 4.1.4: Form Main.html in Windows System, Chrome Browser.

2. Process and Output

The process of submitting is completed by user when pressing the button of "Search". The CGI script `get.py` comprises all the programming functions, reading in and processing form submitting, together with returning the result to user. In next lines, the implementation of Python code is described in detail.

The code for reading in the form values:

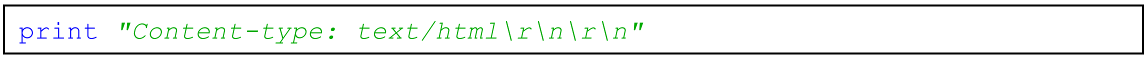


```python
form = cgi.FieldStorage()

if form.has_key('First_name'):
    First_Name = form['First_name'].value
else:
    First_Name = ""

if form.has_key('Second_name'):
    Second_Name = form['Second_name'].value
else:
    Second_Name = ""

if form.has_key('Location'):
    Location  = form['Location'].value
else:
    Location = ""

if form.has_key('select_from_age'):
    from_age = form['select_from_age'].value
else:
    from_age = "--"

if form.has_key('select_to_age'):
    to_age = form['select_to_age'].value
else:
    to_age = "--"

if form.has_key('select_distance'):
    distance = form['select_distance'].value
else:
    distance = None
```

Figure 4.1.5: Python code for reading in the Form values.

The form variable is the instance of `FieldStorage`, which includes `First_name`, `Second_name`, `Location`, `select_from_age`, `select_to_age` and `select_distance`. Since the `FieldStorage` instance performs like a dictionary, the dictionary method `has_keys()` can get the variable in form and the `value` attributes of the instance save the value to the local variable.

After processing the data read from the form, the next step needed to comply is to send the feedback to the client's browser. Since HTML header is separated from HTML, therefore, when CGI script is called back, a HTML header is required to be specified including the blank line:



```python
print "Content-type: text/html\r\n\r\n"
```

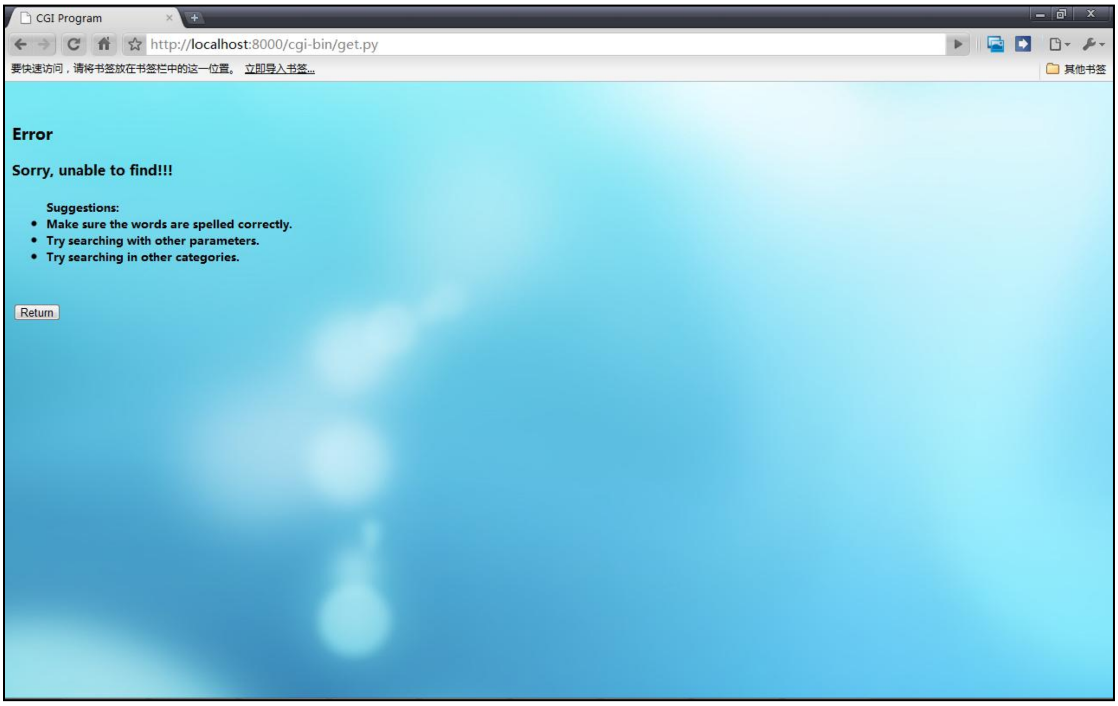
And then print the normal HTML code including JavaScript and CSS if it is necessary.

Another situation here required to be exhibited is to handle the error result, in other words, returning the error page. If the user does not enter any text into the page or gives the totally wrong information to pages in query, in this case, showError () function will return an error page to the user. The display of error page applies JavaScript “Return” button. As we just simply return back to the previous page of browser history, the action is not difficult. The script can only support for one type of error, but I still use the global variable error to continue developing the script, detecting more errors in the future.

The error process code:

```
header = 'Content-type:text/html\r\n\r\n'
errhtml = '''
          <HTML>
              <HEAD>
                  <TITLE>CGI Program</TITLE>
              </HEAD>
          <BODY>
              <h3>Error</h3>
              <b>%s</b><p>
             <FORM><input type="button" value="Return"
ONCLICK="javascript:window.history.back(-1);">
             </FORM>
          </BODY>
          </HTML>
          '''
def showError(error_str):
    print header+errhtml%(error_str)
```

The HTML page demonstration of the error in condition no string is entering into the page:


CGI Program
http://localhost:8000/cgi-bin/get.py
要快速访问，请将书签放在书签栏中的这一位置。 立即导入书签...
其他书签

Error

Sorry, unable to find!!!

Suggestions:

- Make sure the words are spelled correctly.
- Try searching with other parameters.
- Try searching in other categories.

Figure 4.1.6: the error page in Windows System, Chrome Browser.

4.2 JavaScript and CSS

The purpose of JavaScript implemented in the thesis work is primarily to enhance user-friendly features.

The function to prompt the example of form complied by HTML event `onmouseover="return overlib();″`. When mouse point moves over the object, it displays relevant prompting message.

Below figure is the part consequence of the implementation:

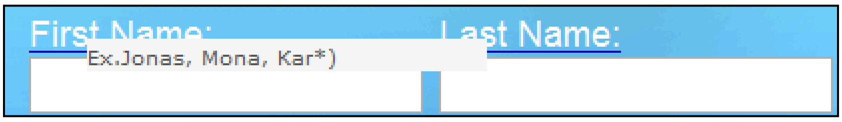
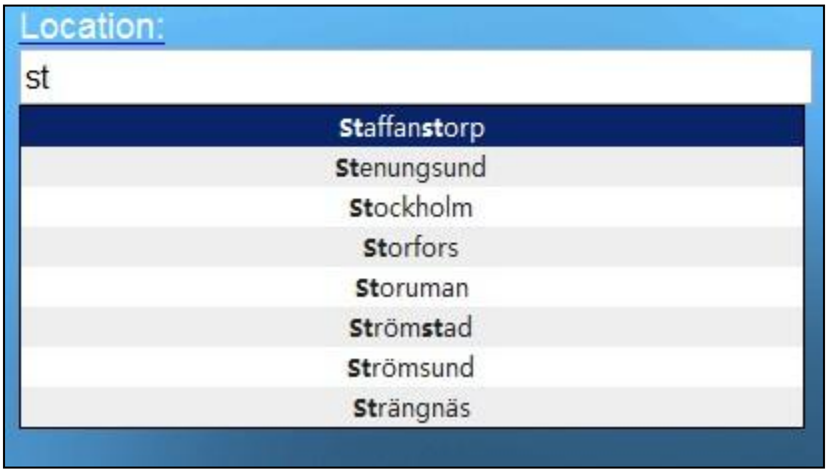


Figure 4.2.1: prompting message of object

The other object shows a different example of prompting message but principally calls the same function and reflects the identical character.

In addition to this feature, a local cache of location suggestion is applied which provide a set of Swedish municipality. In `Main.html`, JavaScript function is embedded when entering the words into grids.

We can see the demonstration of result as Figure 4.2.2:



Location: st

Figure 4.2.2: the demonstration of Location suggestion

When entering the Latin letters, it automatically displays the possible matching location.

CSS in the home page Main.html effects on appropriately and efficiently defining the styles for displaying HTML elements. Besides the home page, the other page can also directly make use of the styles stored in the CSS file.

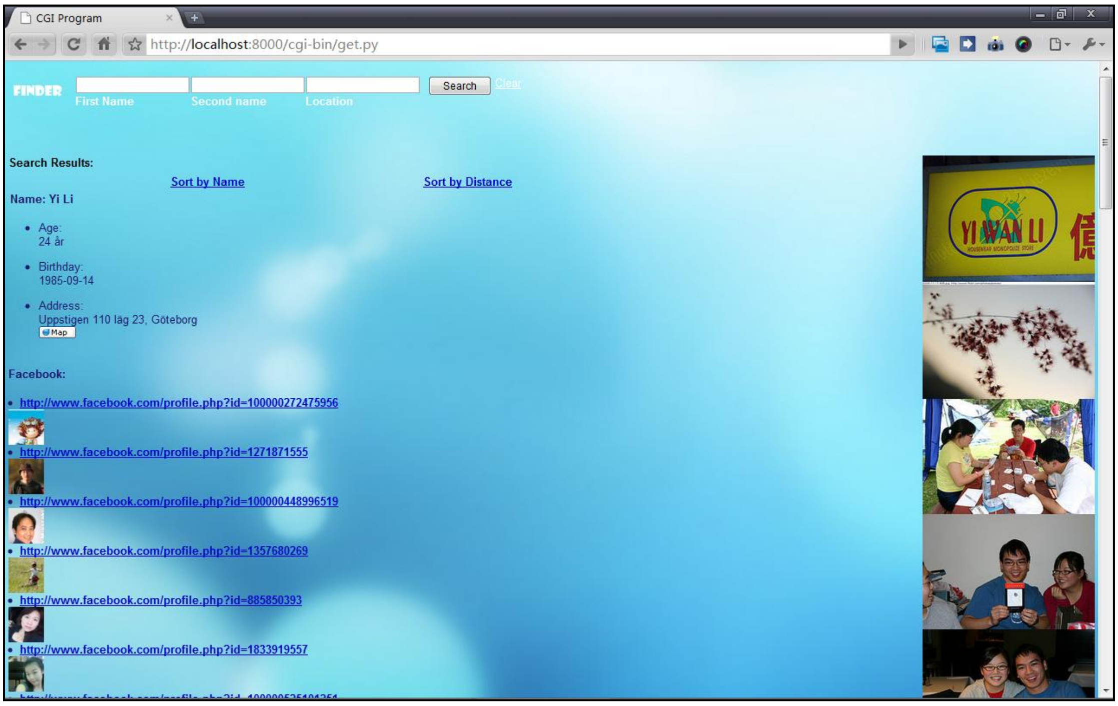
5. Data Demonstration and Analysis

The fifth chapter concentrates on the presentation of the searching results collected from the implemented interface, and analyzes the collected data comparing to the equal condition but utilizing each single search to reveal personal information and degree of disclosure. The relevant evaluation is discussed in the next chapter.

5.1 Case One

Name	*Yi Li*
Age	*About 25*
Email	*yili@student.chalmers.se*
Telephone	*0737701312*
Location	*Göteborg*

The first person I select as the example is one of my friends studying in Chalmers. However, I am not aware of the detail information belonging to him. For example, the exact currently address in Göteborg, the birthday and implied age. In order to search him, I manipulate the query as "Yi Li" for the first name and last name, respectively. And I also give the general location as "Göteborg", but not specify the exact distance. Part of results shows below:



Search Results:

Sort by Name | Sort by Distance

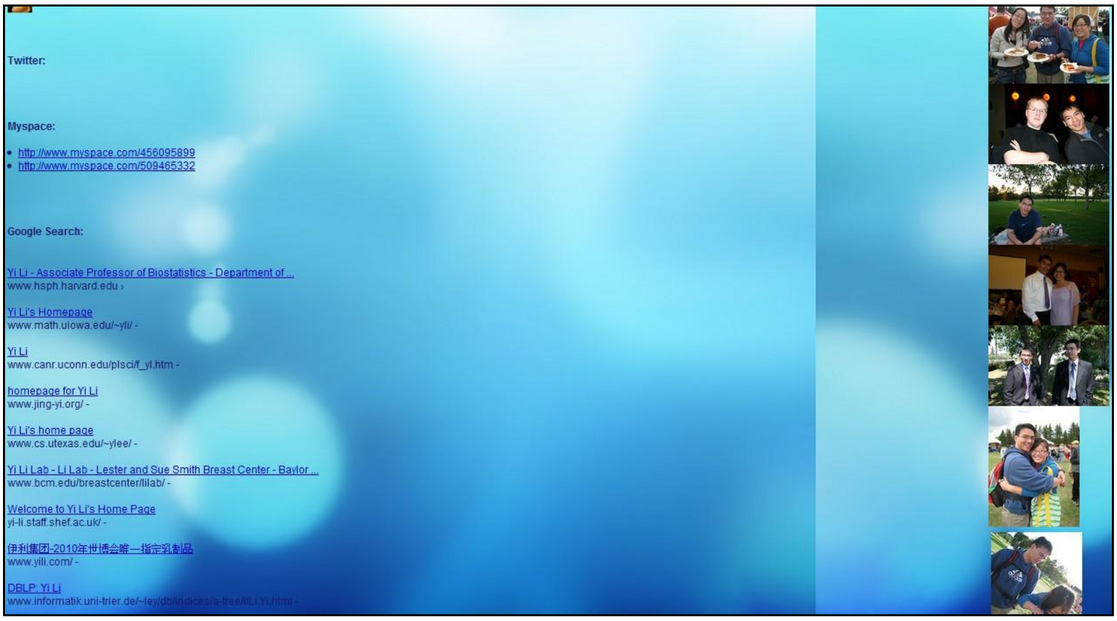
Name: Yi Li

- Age:
  24 år
- Birthday:
  1985-09-14
- Address:
  Uppstigen 110 läg 23, Göteborg

Facebook:

- http://www.facebook.com/profile.php?id=100000272475956
- http://www.facebook.com/profile.php?id=1271871555
- http://www.facebook.com/profile.php?id=100000448996519
- http://www.facebook.com/profile.php?id=1357680269
- http://www.facebook.com/profile.php?id=885850393
- http://www.facebook.com/profile.php?id=1833919557

Figure 5.1.1: part 1 of search result of "Yi Li"



Twitter:

Myspace:

- http://www.myspace.com/456095899
- http://www.myspace.com/509465332

Google Search:

Yi Li - Associate Professor of Biostatistics - Department of ...
www.hsph.harvard.edu ›

Yi Li's Homepage
www.math.uiowa.edu/~yli/ -

Yi Li
www.canr.uconn.edu/plsci/f_yl.htm -

homepage for Yi Li
www.jing-yi.org/ -

Yi Li's home page
www.cs.utexas.edu/~ylee/ -

Yi Li Lab - Li Lab - Lester and Sue Smith Breast Center - Baylor ...
www.bcm.edu/breastcenter/lilab/ -

Welcome to Yi Li's Home Page
yi-li.staff.shef.ac.uk/ -

伊利集团-2010年世博会唯一指定乳制品
www.yili.com/ -

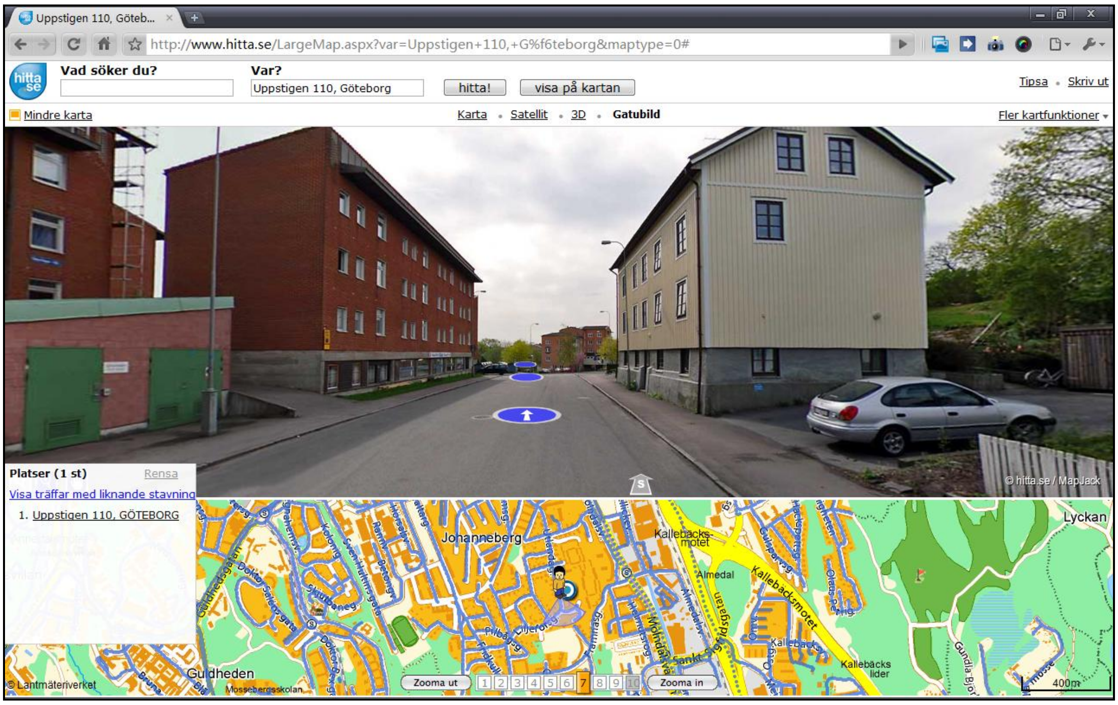
DBLP: Yi Li
www.informatik.uni-trier.de/~ley/db/indices/a-tree/l/Li:Yi.html -

Figure 5.1.2: part 2 of search result of "Yi Li"

From the page returned, it is clear that there is only one match in personal information but serveral possible Facebook account, Myspace site address, and the picture "suspect" to be "Yi Li".

The basic information of "Yi Li" is the age, birthday (which can imply the personal ID as YYMMDD-XXXX) and current address. Since he is one of my friends, I easily confirm with him if the information I fetched is all true.

I have no idea about the street name where he lives but from the hitta service, I can see the real image of his living condition outside:



Uppstigen 110, Göteb...

http://www.hitta.se/LargeMap.aspx?var=Uppstigen+110,+G%f6teborg&maptype=0#

Vad söker du?

Var? Uppstigen 110, Göteborg

Platser (1 st)

Visa träffar med liknande stavning

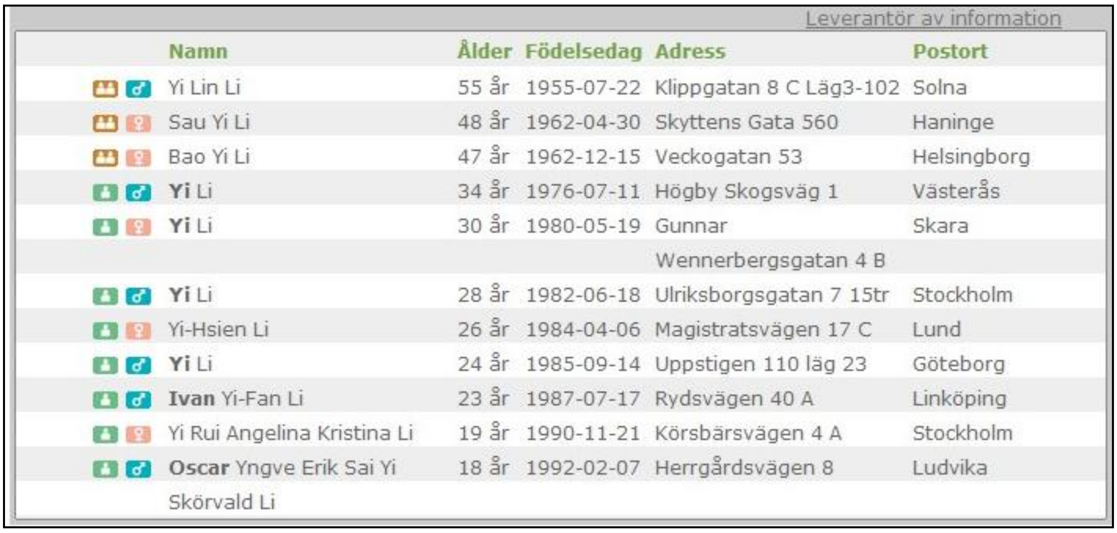
1. Uppstigen 110, GÖTEBORG

Figure 5.1.3: the real image of Li's address in Göteborg

The SN information found by the interface consists of many Facebook accounts having the same name as "Yi Li" (but due to authority limits, it is hard to filter) and two MySpace site addresses. One is easily wiped off because of the head portrait and another might be his account but cannot be proved owing to less pieces of information.

Unfortunately, the Flickr did not find the picture belonging to “Yi Li” because he is not registered to Flickr (Being verified by a phone call with him.)

Then I also try to search him by single search engines one by one. The Ratsit gives over 10 results:



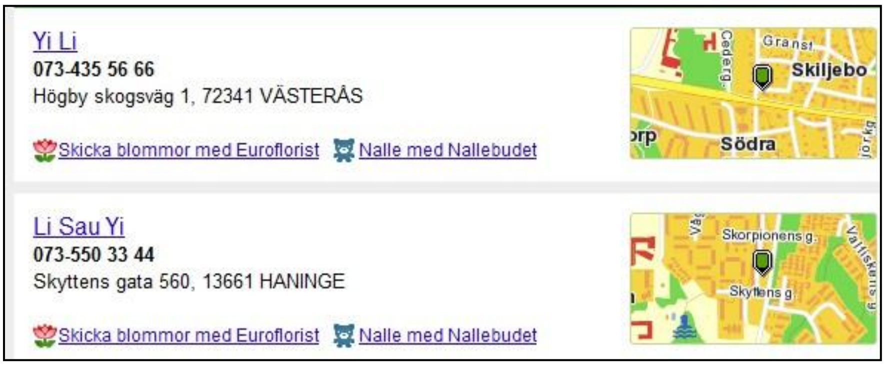
Leverantör av information

	Namn	Ålder	Födelsedag	Adress	Postort
♂	Yi Lin Li	55 år	1955-07-22	Klippgatan 8 C Läg3-102	Solna
♀	Sau Yi Li	48 år	1962-04-30	Skyttens Gata 560	Haninge
♀	Bao Yi Li	47 år	1962-12-15	Veckogatan 53	Helsingborg
♂	**Yi** Li	34 år	1976-07-11	Högby Skogsväg 1	Västerås
♀	**Yi** Li	30 år	1980-05-19	Gunnar Wennerbergsgatan 4 B	Skara
♂	**Yi** Li	28 år	1982-06-18	Ulriksborgsgatan 7 15tr	Stockholm
♀	Yi-Hsien Li	26 år	1984-04-06	Magistratsvägen 17 C	Lund
♂	**Yi** Li	24 år	1985-09-14	Uppstigen 110 läg 23	Göteborg
♂	**Ivan** Yi-Fan Li	23 år	1987-07-17	Rydsvägen 40 A	Linköping
♀	Yi Rui Angelina Kristina Li	19 år	1990-11-21	Körsbärsvägen 4 A	Stockholm
♂	**Oscar** Yngve Erik Sai Yi Skörvald Li	18 år	1992-02-07	Herrgårdsvägen 8	Ludvika

Figure 5.1.4: single search of “Yi Li” by ratsit

What should be noted here is that Ratsit only support adding birthday or birth year for filtering instead of giving an age range for user to search. It is inconvenient for user who wants to search via ratsit.

Our needed information is in the eighth, but it is the same to the basic information I found by interface. And then I also search the name via Eniro and Hitta attempting to find the telephone number, Eniro matches two results:

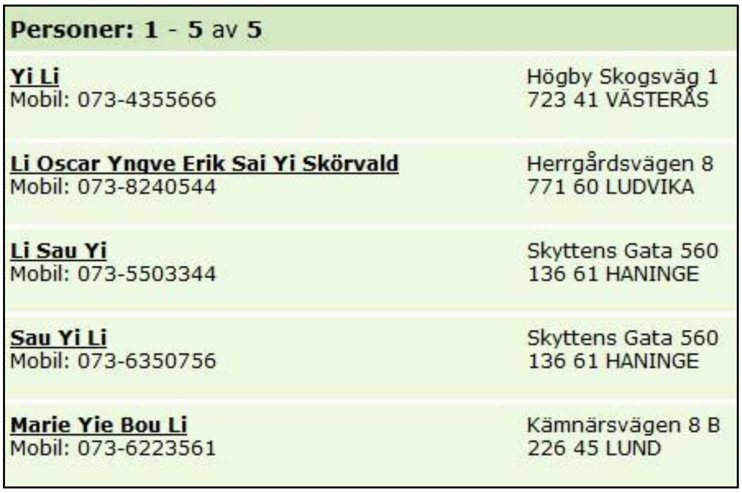


Yi Li
073-435 56 66
Högby skogsväg 1, 72341 VÄSTERÅS
Skicka blommor med Euroflorist Nalle med Nallebudet

Li Sau Yi
073-550 33 44
Skyttens gata 560, 13661 HANINGE
Skicka blommor med Euroflorist Nalle med Nallebudet

Figure 5.1.5: single search of “Yi Li” by Eniro

While Hitta finds five results containing the name:

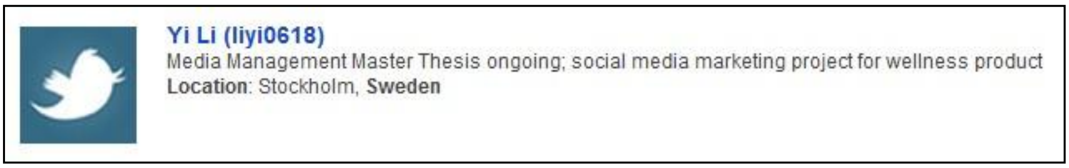


Personer: 1 - 5 av 5

Yi Li Mobil: 073-4355666	Högby Skogsväg 1 723 41 VÄSTERÅS
Li Oscar Yngve Erik Sai Yi Skörvald Mobil: 073-8240544	Herrgårdsvägen 8 771 60 LUDVIKA
Li Sau Yi Mobil: 073-5503344	Skyttens Gata 560 136 61 HANINGE
Sau Yi Li Mobil: 073-6350756	Skyttens Gata 560 136 61 HANINGE
Marie Yie Bou Li Mobil: 073-6223561	Kämnärsvägen 8 B 226 45 LUND

Figure 5.1.6: single search of "Yi Li" by Hitta

From these two searches, it can prove that basic information via interface is giving the right result because no one lives in Göteborg and a telephone number on the page. The Facebook search performs the identical result to the interface while Twitter search shows different and matches one:



Yi Li (liyi0618)
Media Management Master Thesis ongoing; social media marketing project for wellness product
Location: Stockholm, **Sweden**

Figure 5.1.7: single search of "Yi Li" by Twitter

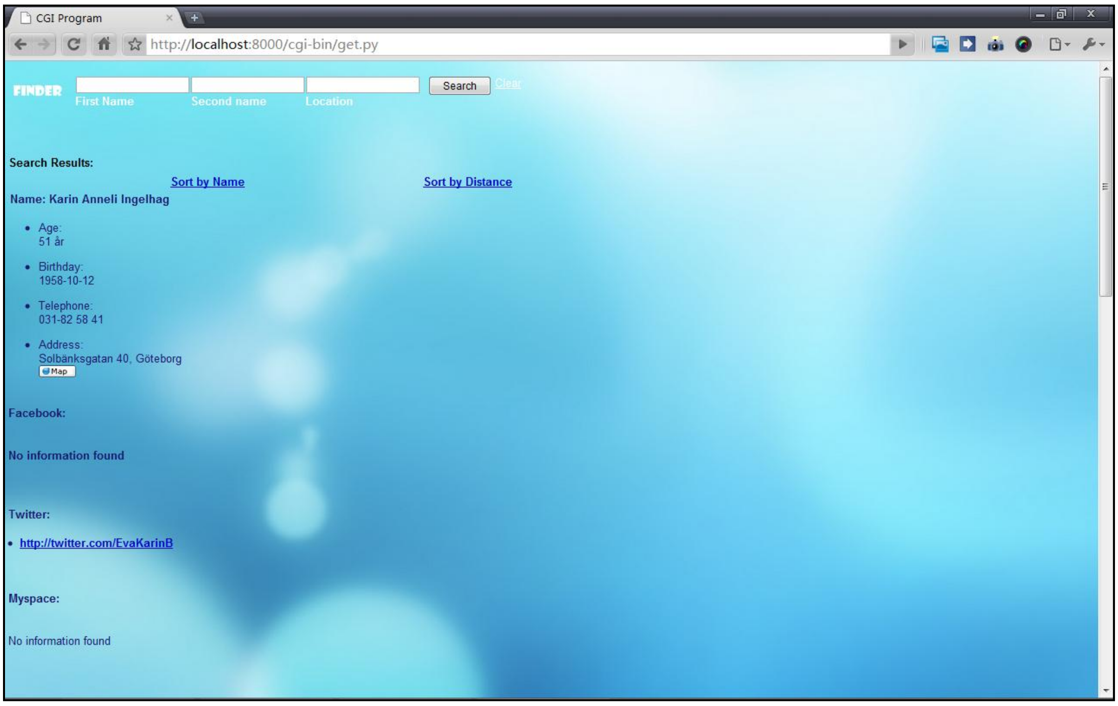
Since the owner of this account specifies the location in Stockholm, I can conclude that it does not belong to the one "Yi Li" I want to find.

5.2 Case Two

Name	*Karin Ingelhag*
Age	*35-60*
Email	*k.ingelhag@gmail.com*
Telephone	*Unknown*
Location	*Göteborg*

The second example as the target person is the land lady of my house. I only know her name, Email and living location in Göteborg area. The query submitted is her first and last name with location in Göteborg but limiting the coverage distance of ten kilometers.

The result includes three persons' basic information:



Search Results:

Sort by Name Sort by Distance

Name: Karin Anneli Ingelhag

- Age:
  51 år
- Birthday:
  1958-10-12
- Telephone:
  031-82 58 41
- Address:
  Solbänksgatan 40, Göteborg

Facebook:

No information found

Twitter:

- http://twitter.com/EvaKarinB

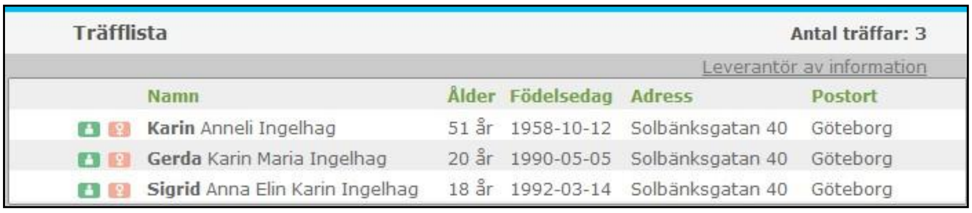
Myspace:

No information found

Figure 5.2.1: information result of search "Karin Ingelhag"

The search result of Karin Ingelhag gives age, birthday, telephone number and address. After confirming with her, all the information found by interface is true. The Twitter account does not exist, probably the junk account.

Comparing with the single search ratsit, which gives three matched pieces of information, two of them are her daughters' information. Her daughters' contain the same family name as their mother:



Träfflista Antal träffar: 3

Leverantör av information

Namn	Ålder	Födelsedag	Adress	Postort
Karin Anneli Ingelhag	51 år	1958-10-12	Solbänksgatan 40	Göteborg
Gerda Karin Maria Ingelhag	20 år	1990-05-05	Solbänksgatan 40	Göteborg
Sigrid Anna Elin Karin Ingelhag	18 år	1992-03-14	Solbänksgatan 40	Göteborg

Figure 5.2.2: ratsit search of Karin Ingelhag

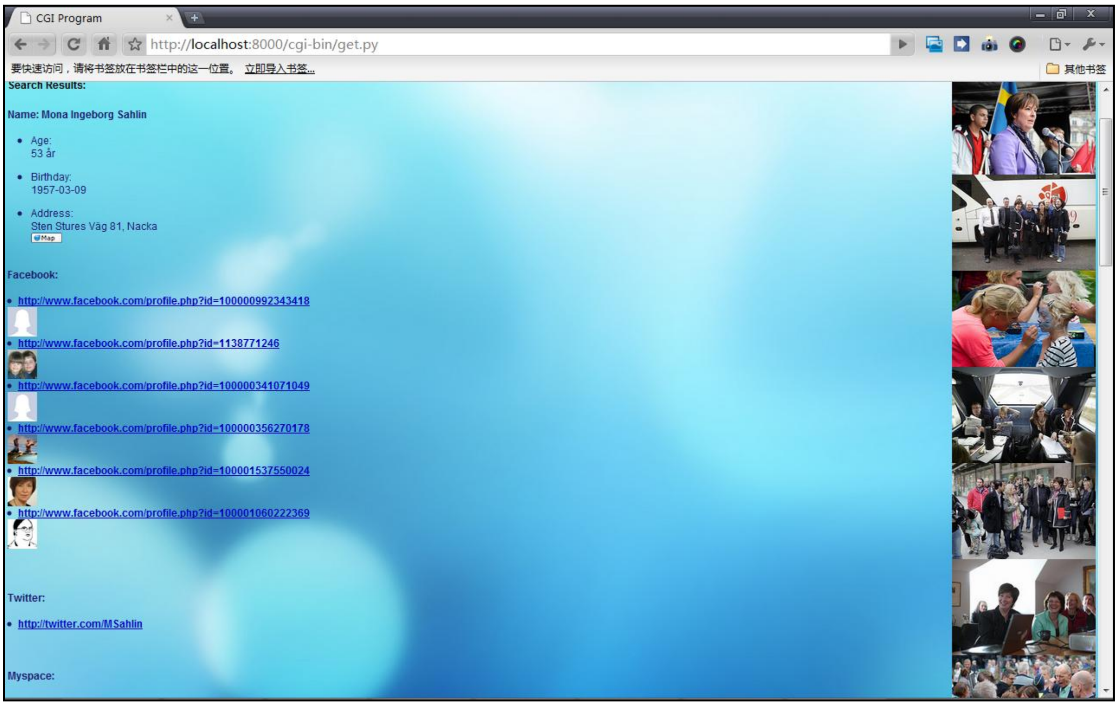
The interface in this way successfully filters the information by age elements. The telephone number is consistent with the Eniro search as well as the address. Except the junk account of Twitter, SN search does not find any information related to target person. However, Karin has a Facebook account but shortening the name as "Karin Ing", which causes the Facebook unable to target the information. The possible solution for handling this kind of situation is making a list of all possible shortening names. If one can find a full name, then try the shortening name. However, it may cause another problem: how can we gather the list of names and the relevant possible abbreviation? In this thesis, the search will stop if it does not match the provided name.

5.3 Case Three

Name	*Mona Sahlin*
Age	*53*
Email	*Unknown*
Telephone	*Unknown*
Location	*Stockholm area*

The third case introduces searching Swedish famous people like "Mona Sahlin". The only information I can be sure of is the birthday and age of her due to the mass media, which is 53 years old and born in 17 March 1957. The query to search specifies the age and location as Stockholm.

The return page finds one match result in basic information:


CGI Program
http://localhost:8000/cgi-bin/get.py
要快速访问，请将书签放在书签栏中的这一位置。 立即导入书签...
其他书签
Map

Search Results:

Name: Mona Ingeborg Sahlin

- Age:
  53 år
- Birthday:
  1957-03-09
- Address:
  Sten Stures Väg 81, Nacka

Facebook:

- http://www.facebook.com/profile.php?id=100000992343418
- http://www.facebook.com/profile.php?id=1138771246
- http://www.facebook.com/profile.php?id=100000341071049
- http://www.facebook.com/profile.php?id=100000356270178
- http://www.facebook.com/profile.php?id=100001537550024
- http://www.facebook.com/profile.php?id=100001060222369

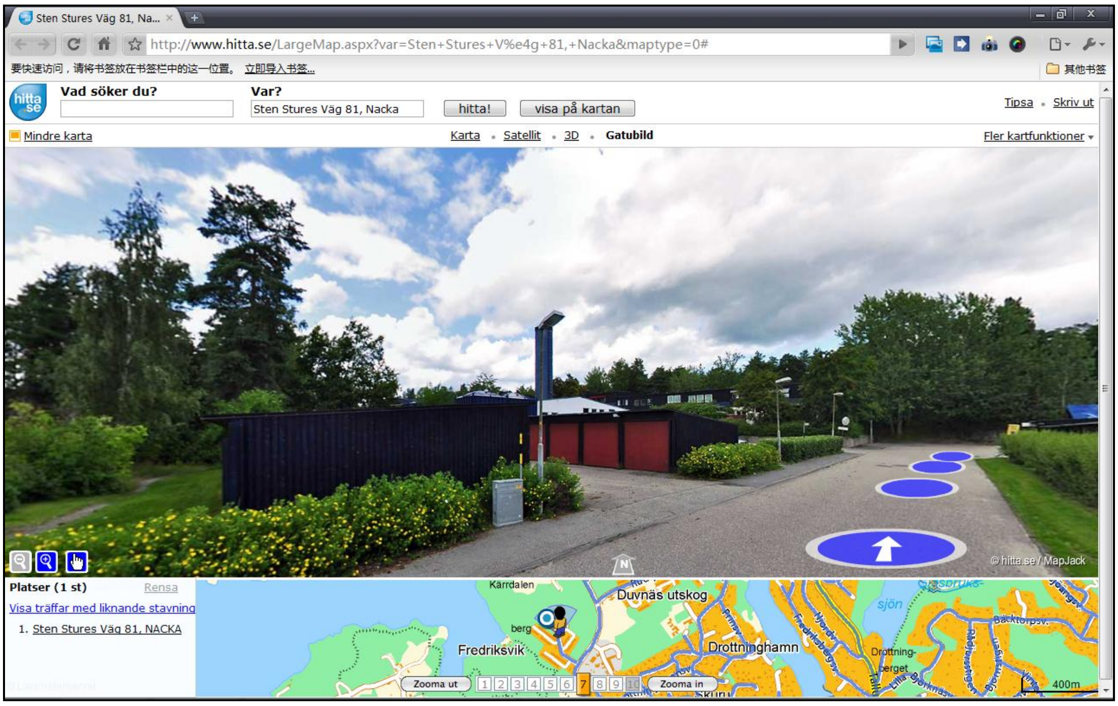
Twitter:

- http://twitter.com/MSahlin

Myspace:

Figure 5.3.1: Search result of Mona Sahlin

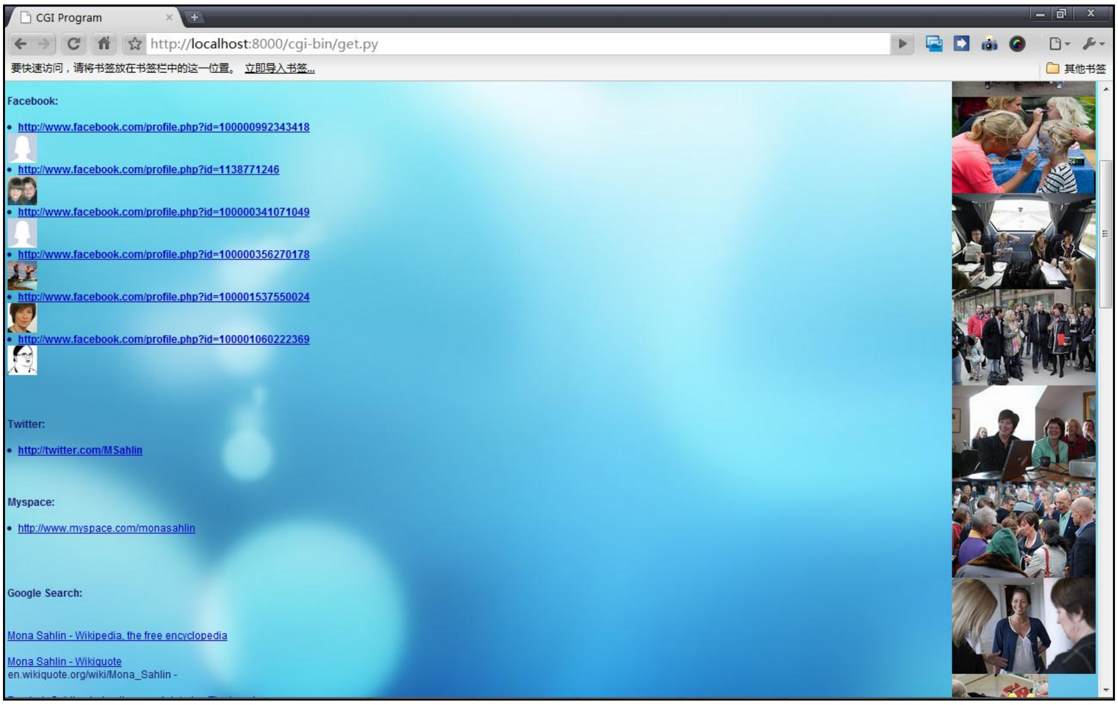
Since the age and birthday is identical to the mass media publishing, I can conclude that this "Mona Sahlin" is the target person I want to find. By using Hitta map (the button in address), the real image of Mona address can be seen:



Sten Stures Väg 81, Na...
http://www.hitta.se/LargeMap.aspx?var=Sten+Stures+V%e4g+81,+Nacka&maptype=0#
要快速访问，请将书签放在书签栏中的这一位置。 立即导入书签...
其他书签
Vad söker du?
Var?
Sten Stures Väg 81, Nacka
hitta!
visa på kartan
Tipsa
Skriv ut
Mindre karta
Karta
Satellit
3D
Gatubild
Fler kartfunktioner
© hitta.se / MapJack
Platser (1 st)
Rensa
Visa träffar med liknande stavning
1. Sten Stures Väg 81, NACKA
Kärrdalen
Duvnäs utskog
Fredriksvik
Drottninghamn
sjön
Zooma ut
Zooma in
400m

Figure 5.3.2: the image of Mona Sahlin's address from Hitta

Furthermore, the interface also finds SN account of Mona Sahlin:



CGI Program

http://localhost:8000/cgi-bin/get.py

要快速访问，请将书签放在书签栏中的这一位置。 立即导入书签...

Facebook:

- http://www.facebook.com/profile.php?id=100000992343418
- http://www.facebook.com/profile.php?id=1138771246
- http://www.facebook.com/profile.php?id=100000341071049
- http://www.facebook.com/profile.php?id=100000356270178
- http://www.facebook.com/profile.php?id=100001537550024
- http://www.facebook.com/profile.php?id=100001060222369

Twitter:

- http://twitter.com/MSahlin

Myspace:

- http://www.myspace.com/monasahlin

Google Search:

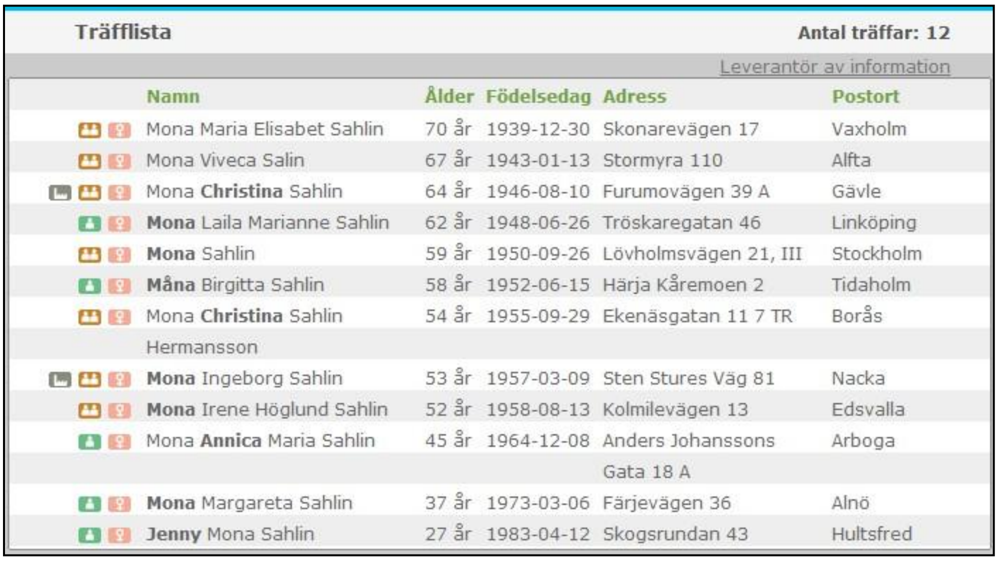
Mona Sahlin - Wikipedia, the free encyclopedia

Mona Sahlin - Wikiquote
en.wikiquote.org/wiki/Mona_Sahlin -

Figure 5.3.3: SN account of Mona Sahlin found by interface

From the head portrait, it is easily to infer that Facebook account http://www.facebook.com/profile.php?id=100001537550024 is her account. And Twitter http://twitter.com/MSahlin and MySpace http://www.myspace.com/monasahlin can be confirmed as true after linking to the address. Flickr works fine in this case fetching a lot of pictures with Mona Sahlin.

The ratsit gives 12 results while some of them are also in Stockholm area but not in the specified age range:



Träfflista

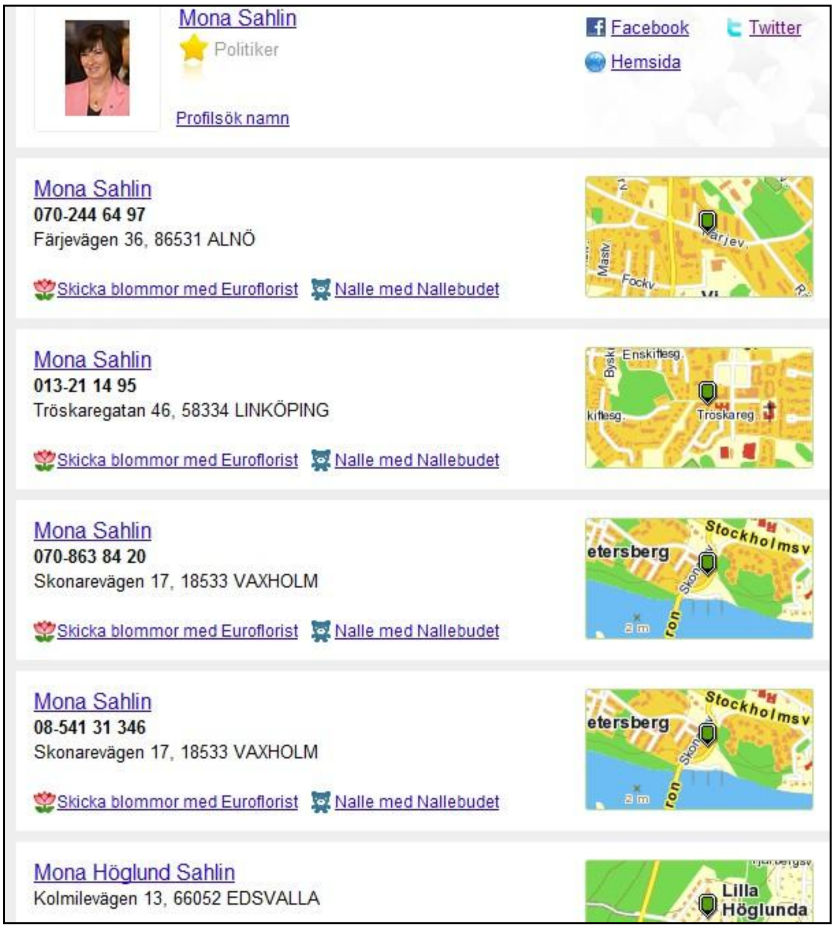
Antal träffar: 12

Leverantör av information

Namn	Ålder	Födelsedag	Adress	Postort
Mona Maria Elisabet Sahlin	70 år	1939-12-30	Skonarevägen 17	Vaxholm
Mona Viveca Salin	67 år	1943-01-13	Stormyra 110	Alfta
Mona **Christina** Sahlin	64 år	1946-08-10	Furumovägen 39 A	Gävle
Mona Laila Marianne Sahlin	62 år	1948-06-26	Tröskaregatan 46	Linköping
Mona Sahlin	59 år	1950-09-26	Lövholmsvägen 21, III	Stockholm
Måna Birgitta Sahlin	58 år	1952-06-15	Härja Kåremoen 2	Tidaholm
Mona **Christina** Sahlin Hermansson	54 år	1955-09-29	Ekenäsgatan 11 7 TR	Borås
Mona Ingeborg Sahlin	53 år	1957-03-09	Sten Stures Väg 81	Nacka
Mona Irene Höglund Sahlin	52 år	1958-08-13	Kolmilevägen 13	Edsvalla
Mona **Annica** Maria Sahlin	45 år	1964-12-08	Anders Johanssons Gata 18 A	Arboga
Mona Margareta Sahlin	37 år	1973-03-06	Färjevägen 36	Alnö
Jenny Mona Sahlin	27 år	1983-04-12	Skogsrundan 43	Hultsfred

Figure 5.3.4: ratsit search result of Mona Sahlin

The interface filter the unqualified match information while Eniro and Hitta do not find the telephone number belonging to Mona Sahlin even if there are many other "Mona Sahlin":



Mona Sahlin
Politiker
Profilsök namn

Facebook
Twitter
Hemsida

Mona Sahlin
070-244 64 97
Färjevägen 36, 86531 ALNÖ
Skicka blommor med Euroflorist Nalle med Nallebudet

Mona Sahlin
013-21 14 95
Tröskaregatan 46, 58334 LINKÖPING
Skicka blommor med Euroflorist Nalle med Nallebudet

Mona Sahlin
070-863 84 20
Skonarevägen 17, 18533 VAXHOLM
Skicka blommor med Euroflorist Nalle med Nallebudet

Mona Sahlin
08-541 31 346
Skonarevägen 17, 18533 VAXHOLM
Skicka blommor med Euroflorist Nalle med Nallebudet

Mona Höglund Sahlin
Kolmilevägen 13, 66052 EDSVALLA

Figure 5.3.5: Eniro search of Mona Sahlin

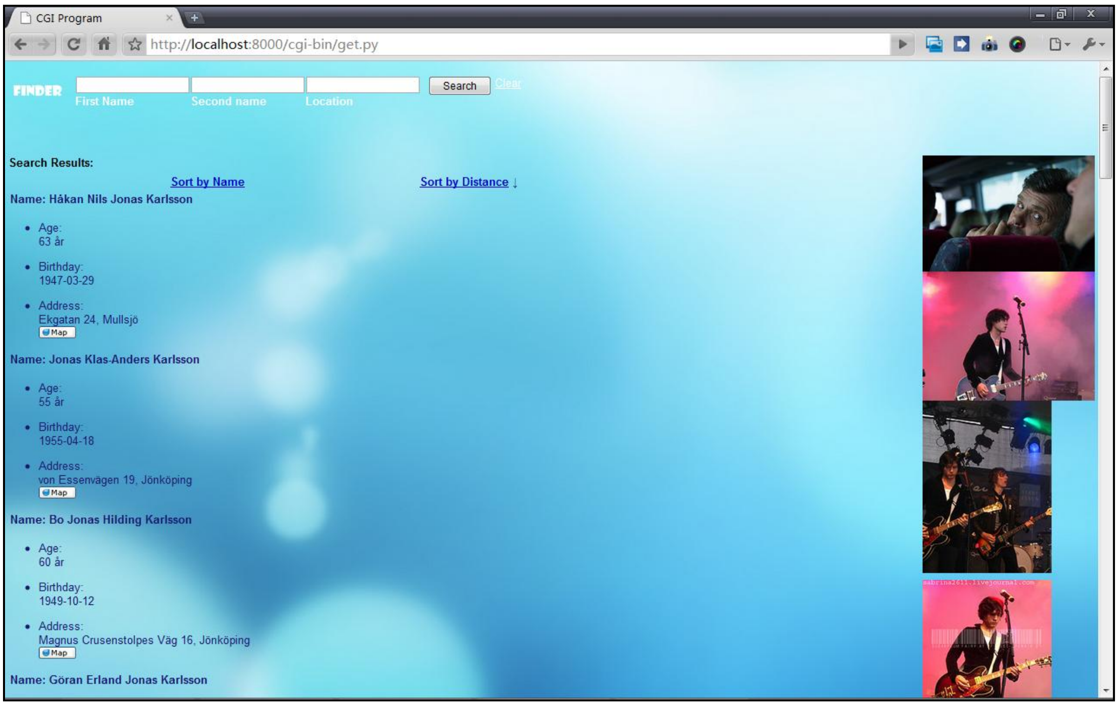
The SNS search presents the equal result as interface function.

5.4 Case Four

Name	*Jonas Carlson*
Age	*Unknown*
Email	*Unknown*
Telephone	*Unknown*
Location	*Jönköping area*

In the forth case, I am intended to exploit a person I totally have no idea about. The query designed for this case is to search a common Swedish name "Jonas Carlson" in Jönköping area within the distance of thirty kilometres. And an additional explanation is, in fact, I am unaware of this person's existence.

The interface returns the page as:



Search Results:

Name: Håkan Nils Jonas Karlsson

- Age:
  63 år
- Birthday:
  1947-03-29
- Address:
  Ekgatan 24, Mullsjö

Name: Jonas Klas-Anders Karlsson

- Age:
  55 år
- Birthday:
  1955-04-18
- Address:
  von Essenvägen 19, Jönköping

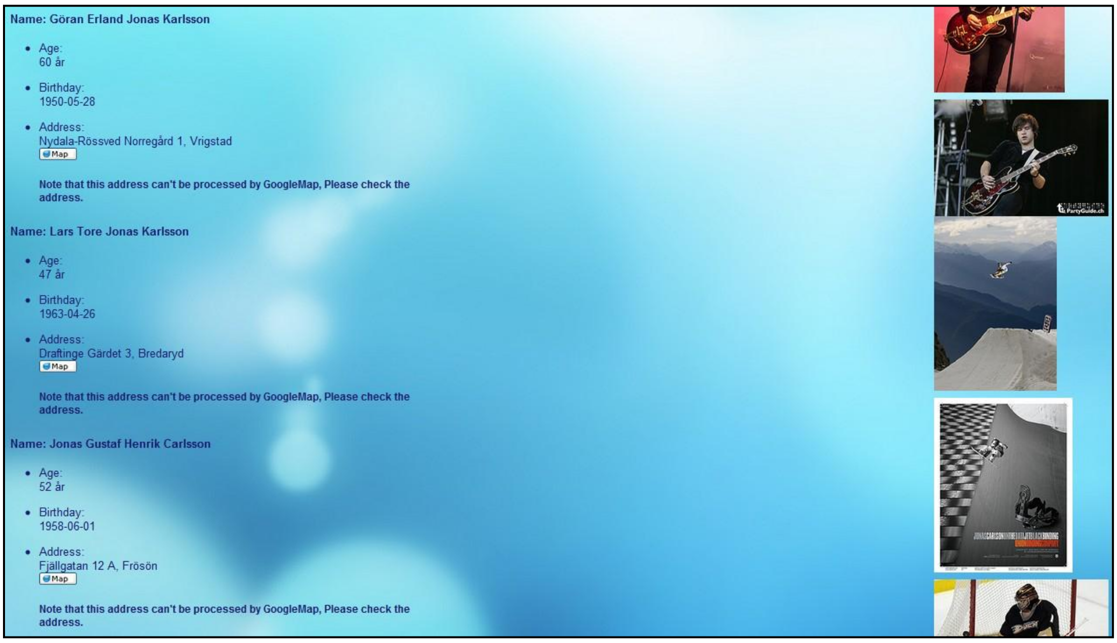
Name: Bo Jonas Hilding Karlsson

- Age:
  60 år
- Birthday:
  1949-10-12
- Address:
  Magnus Crusenstolpes Väg 16, Jönköping

Name: Göran Erland Jonas Karlsson

Figure 5.4.1: Search results of "Jonas Carlson" around Jönköping

The results are sorted by the distance from the farthest to closest in the city. However, it should be noted that some Swedish addresses are hardly parsed by Google Maps interface which is implemented to calculate the distance. The reason for that is mainly because some of the addresses have not been registered on Google Map or Google Map itself unable to handle the text (see details in section 3.2.5). In this situation, the search interface alarms the user to check the address manually. As an example:



Name: Göran Erland Jonas Karlsson

- Age:
  60 år
- Birthday:
  1950-05-28
- Address:
  Nydala-Rössved Norregård 1, Vrigstad

Note that this address can't be processed by GoogleMap, Please check the address.

Name: Lars Tore Jonas Karlsson

- Age:
  47 år
- Birthday:
  1963-04-26
- Address:
  Draftinge Gärdet 3, Bredaryd

Note that this address can't be processed by GoogleMap, Please check the address.

Name: Jonas Gustaf Henrik Carlsson

- Age:
  52 år
- Birthday:
  1958-06-01
- Address:
  Fjällgatan 12 A, Frösön

Note that this address can't be processed by GoogleMap, Please check the address.

Figure 5.4.2: the alarm information about address of search Jonas Carlson around Jönköping

Since the query is on a purpose to search a stranger, the validation of information is barely possible.

SNS searches several Facebook accounts, one Twitter address and two MySpace address which they suspect to be Jonas's. Like the basic information, Facebook is hard to prove that it really belongs to somebody lives in Jönköping. Twitter hits one result marking location as Sweden, in this condition, the filter consider it as true result. What can be excluded of MySpace finding result is one of them indicating the age as twenty six, it is inconsistent with any of found subjects.

The single search of "Jonas Carlson" returns massive results for example Ratsit found over one thousand results in Sweden, and Eniro matches 82 in scope of Jönköping. Since Ratsit and Eniro cannot filter distance within the specified location, the returned results contain much information about "Jonas Carlson" in other location. And therefore, the results contain irrelevant search result.

From the filter processing aspect, the interface filters a lot of information not related to this case, even in MySpace, it failed to handle one account.

In general, the interface function of filtering and processing the true information performs well.

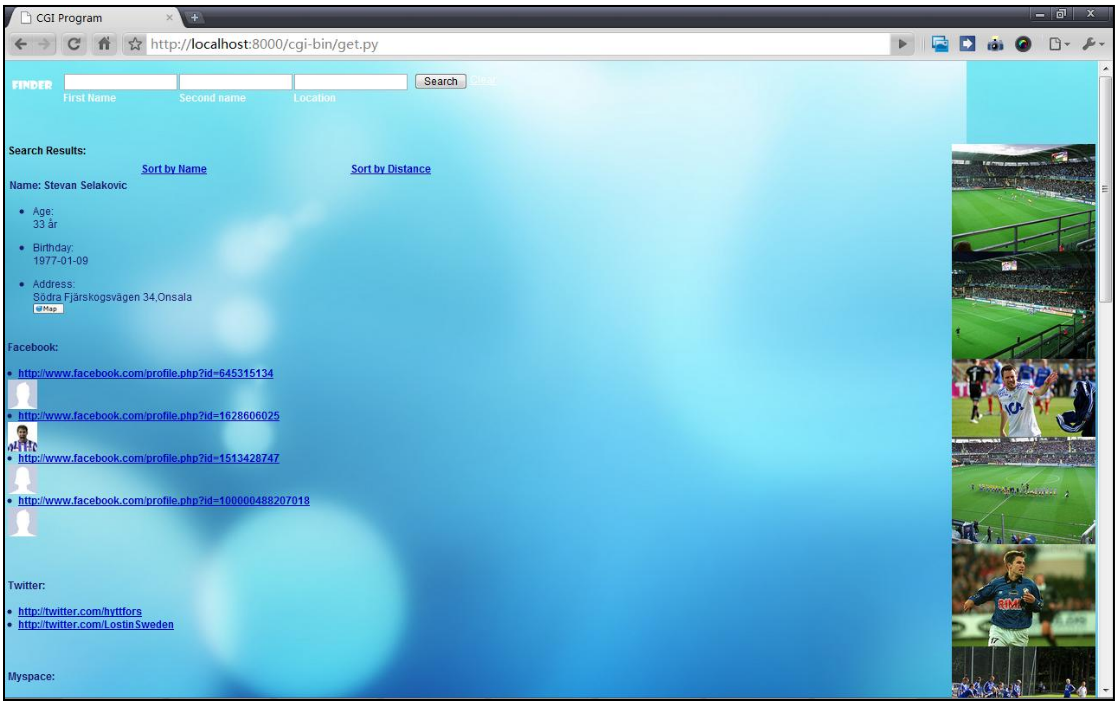
5.5 Case Five

Name	*Stenfan Selakovic*
Age	*Over 30*
Email	*Unknown*
Telephone	*Unknown*
Location	*Göteborg*

The person as the last example to show is a Swedish footballer who plays for IFK

Göteborg as a forward, Stenfan Selakovic, whose information is on the IFK Göteborg official web site[37]. The query submitted to interface is the name and specified age ranges as over thirty.

The result of page is:



Search Results:

Name: Stevan Selakovic

- Age:
  33 år
- Birthday:
  1977-01-09
- Address:
  Södra Fjärskogsvägen 34,Onsala

Facebook:

- http://www.facebook.com/profile.php?id=645315134
- http://www.facebook.com/profile.php?id=1628606025
- http://www.facebook.com/profile.php?id=1513428747
- http://www.facebook.com/profile.php?id=100000488207018

Twitter:

- http://twitter.com/hyttfors
- http://twitter.com/LostinSweden

Myspace:

Figure 5.4.1: search result of Stenfan Selakovic

The found information about Stenfan is in accordance with the official information on Wikipedia[38]. Furthermore, the interface also digs out the living address, pictures and Facebook account.

6. Discussion

This chapter discusses the function efficiency of implemented interface on the basis of data from the fifth chapter. In order to reasonably evaluate the efficiency and deficiency of interface, I search the name of a person attempting to answer the following questions:

1. Whether all the information I know from the daily life about the target person is identical with the fetched information from web?
2. Is there any other information I have no chance to touch daily life but can learn from the web fetched information?
3. Comparing with single search of name from ratsit.se to Flickr, is the interface effective to give the results of target person?
4. How true can the information be when searching a person who we already know some part of the information?
5. Is there any important information not found?

The questions are answered based on data presentation in five cases in the fifth chapter.

6.1 Case Analysis

After experiencing the search process and data analysis, the question proposed in the beginning now can have the answer for each case:

- Case One

1.	The basic information found about “Yi Li” is identical to the one I know in the daily life, and the interface gives more precise result than the speculation.
2.	The more information I found in the first case about “Yi Li” is the birthday, age from official records, and the real image of his living address from outside look.
3.	The interface is more effective because it really did filter some co-name but not target personal information.
4.	The basic information of name, age, birthday and address has been proved as true, while the Facebook fetched a lot but failed to find him may be due to the authority issues. And he has no Twitter, Flickr and MySpace, the interface performs well in this part.
5.	The important information not found is email, telephone number and pictures. (Yi Li does not put them into public web.)

- Case Two

1.	The basic information found about “Karin Ingelhag” is identical to the one I know after confirming with her, and the interface gives more precise result than the speculation as well.
2.	The more information I found in the second case about Karin is the birthday, age, and telephone number.

3.	The interface is effective in some way because it filters information by age.
4.	The basic information of name, age, birthday and address has been proved as true, while the Facebook fails to find due to her name display. And she has no Twitter, Flickr and MySpace, the interface also performs well in this part.
5.	The important information not found is email and pictures.

- Case Three

1.	The basic information of "Mona Sahlin" is identical to the one I know from the mass media, and the interface gives more precise result than the speculation as well.
2.	The more information I found in the second case about Mona Sahlin is the address and SN account.
3.	The interface is effective in some way because it filters information by age and location.
4.	The information of name, age and birthday can be proved as true from the mass media and also address can be proved as true. In this case, interface successfully fetches the SN account and pictures online.
5.	The important information not found is Email and telephone number.

- Case Four

1.	Since the target person is a stranger, it is hard to prove all the information as same as in real life
2.	Since I do not have much information about the stranger, all the information found is new.
3.	The interface is effective in some way because it filters information by age and location and provides sorting service for results.
4.	Since I do not know much information about the target person, information is hard to prove.
5.	The important information not found is Email and telephone number.

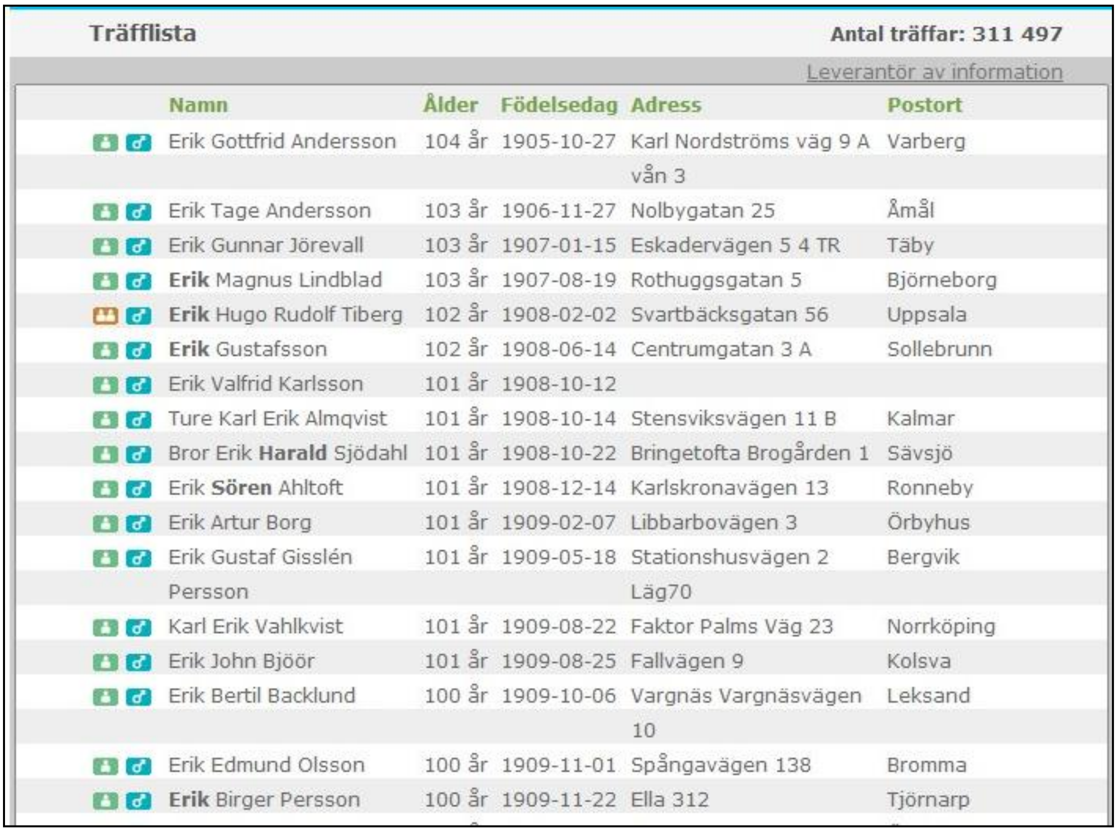
- Case Five

1.	The information is identical with the official records as well as in real life.
2.	The address, SNs and pictures are other information different from real life known.
3.	The interface is effective when just few people have same name.
4.	All information found by interface can be proved as true.
5.	The important information not found is email and telephone number.

In general, the interface works well in data filtering and data correctness. Furthermore, it can not only find some information which is not disclosed on daily life but also perform most effective other than single search one by one.

6.2 Problems in Processing

The first problem caused by the interface feasible to fail finding the target person is the name. In some situations, the query is simple such as just inputting “Erik”, which probably returns the false result or no strings marching. The reason for that is if the name is common and information provided is not enough, the search engines will receive massive matched result but only display first one hundred:



Träfflista **Antal träffar: 311 497**

Leverantör av information

Namn	Ålder	Födelsedag	Adress	Postort
Erik Gottfrid Andersson	104 år	1905-10-27	Karl Nordströms väg 9 A vån 3	Varberg
Erik Tage Andersson	103 år	1906-11-27	Nolbygatan 25	Åmål
Erik Gunnar Jörevall	103 år	1907-01-15	Eskadervägen 5 4 TR	Täby
Erik Magnus Lindblad	103 år	1907-08-19	Rothuggsgatan 5	Björneborg
Erik Hugo Rudolf Tiberg	102 år	1908-02-02	Svartbäcksgatan 56	Uppsala
Erik Gustafsson	102 år	1908-06-14	Centrumgatan 3 A	Sollebrunn
Erik Valfrid Karlsson	101 år	1908-10-12		
Ture Karl Erik Almqvist	101 år	1908-10-14	Stensviksvägen 11 B	Kalmar
Bror Erik **Harald** Sjödahl	101 år	1908-10-22	Bringetofta Brogården 1	Sävsjö
Erik **Sören** Ahltoft	101 år	1908-12-14	Karlskronavägen 13	Ronneby
Erik Artur Borg	101 år	1909-02-07	Libbarbovägen 3	Örbyhus
Erik Gustaf Gisslén Persson	101 år	1909-05-18	Stationshusvägen 2 Läg70	Bergvik
Karl Erik Vahlkvist	101 år	1909-08-22	Faktor Palms Väg 23	Norrköping
Erik John Bjöör	101 år	1909-08-25	Fallvägen 9	Kolsva
Erik Bertil Backlund	100 år	1909-10-06	Vargnäs Vargnäsvägen 10	Leksand
Erik Edmund Olsson	100 år	1909-11-01	Spångavägen 138	Bromma
Erik Birger Persson	100 år	1909-11-22	Ella 312	Tjörnarp

Figure 6.2.1: Erik search example by ratsit

In Figure 6.2.1, the matching objects is 311,497 while the Ratsit only displays the first one hundred order by the age. The setting of Ratsit leads to the interface incapable to handle such simple but insufficient query.

Besides Ratsit, Eniro and Hitta also limit the search result number in one page, since the parser itself is unaware to link to the next page (if so, it generates long waiting time). In fact, the rest of pages are not in process.

The solution to solve this problem concentrates on the query and the implication of the query determinate the whether the relevant data can be disclosed reliably and truly.

Another issue troubling the interface parsing is the speed. In normally case, the delay time of parsing ten subjects are 5 seconds to 10 seconds, while if the result increases to over 20, the delay time surges sharply. Each subject found by interface needs at least four operations: open URL, fetch, compare and print out. The possible solution for speed issues consists of two aspects:

1. Utilizing multi-thread programming to handle each subject thereby to achieve fast parallel processing.

2. Reforming the regular expression to avoid the multi-cycle in matching particularly non-greedy matching.

7. Conclusion

The cyber world contains relationships and personal information reflecting to the real world. People sharing their name, gender, life photos, home or work address and families deliberately or accidentally, will risk everything corresponding to the real life. Some small pieces of information may not damage the privacy concerns but if assembling all the information or giving a clear head portrait can results in a potential serious crime. The interest data to criminals are birthday, age, telephone and home address. The birthday and age infers the personal number which can lead to fake real-life identities abused by the criminal purpose. Since obtaining the telephone number, it is possible to track the call history then get a list of relationship. The home address considered to be the privacy place. As you publish the house address online, it will lose its privacy and may cause the risk of becoming a victim of all kinds of harassment. More seriously, combined with the map function, home has no secret to the strangers.

The thesis has reviewed the related work concerns on online privacy in theoretical approach, and tested my name to see possible results with the online existing tools. Afterwards, I have derived the idea from them and analyzed the existing popular social networks and local search engines including Facebook, Twitter, MySpace, Flickr, Eniro, Hitta, Ratsit and Google. The degree of disclosure and data reliability is also discussed each in the section of analysis.

On the basis of the analysis, I conduct one kind of logic for effectively mining the personal information in the scope of Sweden. The logic goes through the search engines and social networks one by one and filters the irrelevant information out of the users' specifications. The techniques involved to realize the logic are: Python, JavaScript, HTML and CSS. The script gains the data from the front end web portal, and then sends to the back end to proceed. Firstly, it goes to ratsit to check the identity if the person exists then fetches the telephone number in Eniro or Hitta. The social networks search is based on location to filter the irrelevant information. But if the user setting of privacy is limited, location is hidden. In that situation, the SNS search will still keep the account as the possible result. Google and Flickr are implemented as the supplement to the whole process, which can make the information mining more ideal.

In addition to presenting the logic and implementing the code, five cases from the real life are investigated for the purpose of testing the implemented interface of personal information mining and the situation of privacy disclosure of Sweden.

It has been proved by five cases that the implemented tool performs more advantage than the single search, and it can give more clean and trustable results. The drawback of the tool shows in searching the email (Because of the format of file in .jpg discussed in chapter two). It fails to find the email of victims. And the Flickr and Google are somewhat of independent of the other search, but the solution may be provided in future work.

In general, the thesis achieves the goal to reveal the problem and practical circumstances of online privacy concerns in Sweden. According to the actual proof of the data obtained from the search, a large quantity of people indeed disclose the personal information on web, and not many of them pay attention to that. Social networks and online communities are vulnerable to become a source of personal information leakage.

8. Future work

There are still some rooms for improvement on the interface in this thesis making the prospect more attractive. Besides the solution discussed in chapter six for speed could enhance the capability of search, two new features can greatly strengthen the power of search:

1. Google search

The Google search in this thesis to some extent is parallel to the other search; the interaction of information process with the other search is less involved. In future, by means of bringing Google information to participate could give a more precise and comprehensive result.

2. Picture reading and recognising

The picture reading and recognising will enable the email parsing into the whole process, while picture recognising will directly search result more vivid and truthful.

Abbreviations

SN	**-**	**Social Network**
SNS	**-**	**Social Networks**
HTTP	**-**	**Hypertext Transfer Protocol**
HTML	**-**	**Hypertext Markup Language**
CSS	**-**	**Cascading Style Sheets**
JavaScript	**-**	**ECMAScript, an object-oriented scripting language**
API	**-**	**application programming interface**
CGI	**-**	**Common Gateway Interface**
ASCII	**-**	**American Standard Code for Information Interchange**

Reference

[1] <What is Web 2.0? Ideas, technologies and implications for education> by Paul Anderson, JISC Technology and Standards Watch, Feb. 2007

[2] <the Next Digital Divide: Online Social Network Privacy> by Avner Levin, Mary Foster, Bettina West, Mary Jo Nicholson, Tony Hernandez, Wendy Cukier, Ryerson University Ted Rogers School of Management, Privacy and Cyber Crime Institute, March 2008.

[3] <Link Privacy in Social Networking> by Aleksandra Korolova, Rajeev Motwani, Shubha U. Nabar, Ying Xu, Computer Science Department, Stanford University.

[4] <Social Networking: A quantitative and qualitative research report into attitudes, behaviors and use> from Ofcom Research Document, Publication date: 2 April 2008

[5] <Determining Internet Users' Values for Private Information> by Buffett, S., Fleming, M.W., Richter, M.M., Scott, N., and Spencer, B. published in The Second Annual Conference on Privacy, Security and Trust (PST'04). Fredericton, New Brunswick, Canada. October 14-15, 2004. pp. 79-88. NRC 47408.

[6] http://finance.sina.com.cn/blank/bwzlxl.shtml

[7] <InterestMap: Harvesting Social Network Profiles for Recommendations> by Hugo Liu and Pattie Maes, In Proceedings of the Beyond Personalization 2005 Workshop, 2005.

[8] < (Under) mining Privacy in Social Networks> by Monica Chew, Dirk Balfanz, and Ben Laurie. http://w2spconf.com/2008/papers/s3p2.pdf.

[9] < De-anonymizing Social Networks> by Arvind Narayanan and Vitaly Shmatikov. http://userweb.cs.utexas.edu/~shmat/shmat_oak09.pdf

[10] <Collective information practice: exploring privacy and security as social and cultural phenomena> by Paul Dourish and Ken Anderson. Human-Computer. Interaction, volume 21, 2006, pp. 319-342.

[11] <Social Networking Media > by Department of Administration Service. http://www.oregon.gov/DAS/EISPD/EGOV/BOARD/docs/social_networking_guide_v1.pdf?ga=t

[12] http://www.whitepages.com/person

[13] <Information Revelation and Privacy in Online Social Networks (The Facebook case), Pre-proceedings version> by Ralph Gross, Alessandro Acquisti. ACM Workshop on Privacy in the Electronic Society (WPES), 2005

[14] http://www.facebook.com/

[15] http://twitter.com/

[16] http://www.myspace.com/

[17] http://www.flickr.com/

[18] http://www.facebook.com/policy.php

[19] <Social Networking Apps Pose Surprising Security Challenges> By Anthony Bettini, McAfee Labs.

[20] http://twitter.com/privacy

[21] http://www.myspace.com/index.cfm?fuseaction=misc.privacy

[22] http://www.flickr.com/help/privacy/

[23] http://www.eniro.se/

[24] http://www.ratsit.se/

[25] <Core Python Programming>, by Wesley J. Chun, ISBN: 0132269937 .2nd ed.

[26] <Dive Into Python>, Mark Pilgrim (mailto:mark@diveintopython.org), Version 1.1, March

2000 Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111−1307 USA

[27] http://www.crummy.com/software/BeautifulSoup/

[28] http://python.org/

[29] http://docs.python.org/library/urllib.html

[30] http://docs.python.org/library/urllib2.html

[31] http://docs.python.org/library/cookielib.html

[32] http://www.crummy.com/software/BeautifulSoup/documentation.html

[33] HTML ISO-8859-1 Reference, http://www.w3schools.com/tags/ref_entities.asp.

[34] http://www.crummy.com/software/BeautifulSoup/documentation.html

[35] Python Module web_search.py by Connelly Barnes, 2005-2007. http://www.connellybarnes.com/code/web_search/web_search-1.0.2

[36] The OAuth 2.0 Protocol, http://tools.ietf.org/html/draft-ietf-oauth-v2-10

[37] http://www.ifkgoteborg.se/

[38] http://en.wikipedia.org/wiki/Stefan_Selakovic