



UIT

THE ARCTIC  
UNIVERSITY  
OF NORWAY

Faculty of Science and Technology

Department of Computer Science

## Creating a better Play Store for Cancer apps by using the Meta Data

*Is the metadata available enough to improve app finding for apps related to Cancer?*

---

**Håvard Hemmingsen Johansen**

*INF-3981 Master's Thesis in computing - December 2016*







## I. Abstract

The current app store is minimalistic and gives a minimum of functionality, there are in principle two options, a list of recommended apps and the search function. Where the search function is as good as the user is able to come up with search words. The question then is; is it possible to create a third party app that works as an overlay and give a more useful result. In order to make the problem more approachable and to take advantage of work done by others before this thesis focuses on cancer related apps [1, 2]

Unfortunately there is not enough data in the metadata in order to create such a system, with a meaningful improvement in result. Another big problem is there are really an extremely small number of users a system like this will be targeted at both this version, but also a general version.



## II. Acknowledgements

First of all I would like to express my gratitude to my supervisor through this study, Professor Randi Karlsen. She where able to handle my silences and annoyance with the result that where coming up.

Next I want to express my gratitude to my friends and my training buddy for getting me away from the computer.

Finally, I would also like to thank my family for their great support and guidance over the years. You are all much appreciated.



### III. Table of Contents

I.	Abstract.....	3
II.	Acknowledgements.....	5
III.	Table of Contents.....	7
IV.	List of Figures .....	9
V.	List of Tables.....	11
1	Introduction.....	13
1.1	Problem definition.....	13
1.2	Target Audience .....	13
1.2.1	Users skills .....	14
1.3	Methods and materials .....	16
1.3.1	Methodology applied for this thesis.....	16
2	Review of related literature.....	17
2.1	Apps for health.....	17
2.2	App overload.....	17
3	Review of related technologies.....	19
3.1	Related services.....	19
3.2	Web crawler.....	20
4	Design.....	23
4.1	Back End.....	23
4.1.1	SQL vs NoSQL.....	24
4.2	MongoDB.....	24
4.2.1	Meta data gathering.....	25
4.3	App rating .....	25
4.3.1	Example apps.....	25
4.3.2	User rating .....	27
4.3.3	Words and phrases.....	28
4.3.4	Narrowness .....	28
4.3.5	Security and privacy.....	29
4.4	Front End .....	30
5	Implementation.....	31
5.1	Web Crawler .....	31
5.1.1	Bugs .....	31
5.2	The Datasets.....	32
5.3	Available data on the apps.....	34

5.3.1	Limited categories .....	35
5.3.2	Incorrect information .....	35
5.4	Backend Data analyzer .....	36
6	Evaluation and result .....	39
6.1	Description .....	40
6.2	Developer data .....	41
6.3	Categories in the database.....	42
6.4	Removing Snake Oil .....	42
7	Future improvements and work.....	45
8	Concluding remarks.....	47
9	Bibliography.....	49



## IV. List of Figures

Figur 1 Finn.no top level categories .....	19
Figur 2 Main search page .....	20
Figur 3. The system architecture.....	23
Figur 4: Finn Android app.....	30
Figur 5 Cancer 101 Treatment a example of an app in the database.....	34
Figur 6 Example of Developer .....	35
Figur 7 Category voted for Harry Potter and the Sorcerer's Stone.....	42



## V. List of Tabell's

Tabell 1 Description of proficiency level in problem solving in technology-rich environments.....	14
Tabell 2 Backend command prompts.....	36
Tabell 3 General database statistics.....	39
Tabell 4 Words used in Descriptions, CancerOnlyDB .....	40
Tabell 5 Developer home page statistics.....	41



# 1 Introduction

## 1.1 Problem definition

This thesis is in many ways a continuation of the work done by Ruben Mæland [1]. In his work he focused on finding apps from app stores and gathering the apps metadata. One of Ruben's main motivations where the fact that the app stores have a limited search function where one have to choose between searching by name or by category and any more advanced search is close to impossible. This thesis presents the work done to use the metadata collected by Ruben's system to create a more advanced search function. Giving the user more control over what they are looking for and give an analyses of the relevance and dangers that might be associated with some apps [Ref to paper surveying over bad apps and what types of accesses they want]. It is also an attempt to create what Velsen, Beaujean and Gemert-Pinjnen [2] where looking for in order to handle the huge amount of apps that are out there. This program do not propose to give any user a definite list of the best apps out there because it have to rely on user feedback and surface indications in order to evaluate apps usefulness. In 2.1 there is a look at some apps that are in Google's app store and how hard it can be to say if apps are good or not. This program can give a recommendation, but users still might get bad apps or apps that where not exactly what they were looking for. However this program tries to give a better result than what is available in today's app store. This app is created with cancer apps as its area of expertise; this was to allow for a limitation on what the program has to do and getting better results. The program is simple to reprogram to change its target apps or expand it to more categories.

## 1.2 Target Audience

When considering the target audience for this app there are two things that have to be considered. First this is a specialty app, that is to say it have a subsection of the population that is interested in the product in this case people that are looking for apps related to cancer. The second is how complicated is the user interface. Before one can start with the design one have to decide how complicated one need the program to be. The more complicated the design is the more skills do the user need to have before using the application or the larger and better do the tutorial have to be [3]. If the program need a tutorial then one increase the time cost for a user before they can use the program thereby increasing the adoption cost. This again result in a situation where the program have to be a lot better before people are going to make the swap to using it. A simple example of this is the amount of work Microsoft<sup>1</sup> is having to do to try to make people and companies upgrade to the latest version of the operation system.

---

<sup>1</sup> <https://www.microsoft.com/>

### 1.2.1 Users skills

The person making the interface has tech knowledge far above the average person [3]. The fact is that the largest part of the population have little to no knowledge of using computers and there program. A 2016 study by OECD [3] researchers found that 29% of the population have no knowledge or are just able to start a program like E-mail open a mail and reading it, then responding (see Tabell 1). If they have to do anything more complex than going directly to have they need, it get too complex for a large segment of the population. The next 30% of the population is able to use more programs and familiarize themselves with programs. This group (Level 1 [3]) is the group this program is most likely to be the lower level of user on this program. The chance that people with less knowledge are going to install a program that gives a second level of complexity is quite unlikely. Even the Level 1 users might be hard to get to use this app if it requires too much work. So unless the program is extremely easy to use and gives an almost seamless interface to the Play Store it might be too much for these users. This means that realistically the people that might be interested in using an overlay like this front end is going to be is the remaining 31% of the population. Only real way to know is letting people test the program.

**Tabell 1 Description of proficiency level in problem solving in technology-rich environments<sup>2</sup>**

Level	Score range	Percentage of adults able to perform tasks at each level (average)	The types of tasks completed successfully at each level of proficiency
No computer experience	Not applicable	10.0%	Adults in this category reported having no prior computer experience; therefore, they did not take part in the computer-based assessment but took the paper-based version of the assessment, which did not include the problem solving in technology-rich environment domain.
Failed ICT core	Not applicable	4.7%	Adults in this category had prior computer experience but failed the ICT core test, which assesses the basic ICT skills, such as the capacity to use a mouse or scroll through a web page, needed to take the computer-based assessment. Therefore, they did not take part in the computer-based assessment, but took the paper-based version of the assessment, which did not include the problem solving in technology-rich environment domain.
“Opted out” of taking computer based assessment	Not applicable	9.6%	Adults in this category opted to take the paper-based assessment without first taking the ICT core assessment, even if they reported some prior experience with computers. They also did not take part in the computer-based assessment, but took the paper-based version of the assessment, which did not include the problem solving in technology rich environment domain.

<sup>2</sup> Table is taken from page 53 of the OECD survey [3]



Below Level 1	Below 241 points	14.2%	Tasks are based on well-defined problems involving the use of only one function within a generic interface to meet one explicit criterion without any categorical or inferential reasoning, or transforming of information. Few steps are required and no sub-goal has to be generated.
Level 1	241 to less than 291 points	28.7%	At this level, tasks typically require the use of widely available and familiar technology applications, such as e-mail software or a web browser. There is little or no navigation required to access the information or commands required to solve the problem. The problem may be solved regardless of the respondent's awareness and use of specific tools and functions (e.g. a sort function). The tasks involve few steps and a minimal number of operators. At the cognitive level, the respondent can readily infer the goal from the task statement; problem resolution requires the respondent to apply explicit criteria; and there are few monitoring demands (e.g. the respondent does not have to check whether he or she has used the appropriate procedure or made progress towards the solution). Identifying content and operators can be done through simple match. Only simple forms of reasoning, such as assigning items to categories, are required; there is no need to contrast or integrate information.
Level 2	291 to less than 341 points	25.7%	At this level, tasks typically require the use of both generic and more specific technology applications. For instance, the respondent may have to make use of a novel online form. Some navigation across pages and applications is required to solve the problem. The use of tools (e.g. a sort function) can facilitate the resolution of the problem. The task may involve multiple steps and operators. The goal of the problem may have to be defined by the respondent, though the criteria to be met are explicit. There are higher monitoring demands. Some unexpected outcomes or impasses may appear. The task may require evaluating the relevance of a set of items to discard distractors. Some integration and inferential reasoning may be needed.
Level 3	Equal to or higher than 341 points	5.4%	At this level, tasks typically require the use of both generic and more specific technology applications. Some navigation across pages and applications is required to solve the problem. The use of tools (e.g. a sort function) is required to make progress towards the solution. The task may involve multiple steps and operators. The goal of the problem may have to be defined by the respondent, and the criteria to be met may or may not be explicit. There are typically high monitoring demands. Unexpected outcomes and impasses are likely to occur. The task may require evaluating the relevance and reliability of information in order to discard distractors. Integration and inferential reasoning may be needed to a large extent.

**Note:** The proportion of adults scoring at different levels of proficiency adds up to 100% when 1.9% of literacy-related non-respondents across countries/economies are taken into account. Adults in the missing category were not able to provide enough background information to impute proficiency scores because of language difficulties, or learning or mental disabilities.

## 1.3 Methods and materials

### 1.3.1 Methodology applied for this thesis

The science of computers is one of the youngest sciences, it has evolved over just 60 years, and it has been a fast and varied evolution. In 1989 the Task Force of the Core of Computer Science, formed by the ACM and the IEEE Computer Society; stipulated a definition of computer- science and engineering: "Computer science and engineering is the systematic study of algorithmic processes-their theory, analysis, design, efficiency, implementation, and application that describe and transform information..." [4]. This definition was conveyed in their final report that also forms the basis of computer science: theory, abstraction, and design.

Theory is an iterative process rooted in mathematics which is based on the idea of characterizing the objects of the study to create a definition and hypothesizing among their possible relationships to provide a theorem. The relationships provided in the theorem are thus analyzed to be proven or disproven and the results are evaluated.

Abstraction outlines an experimental scientific method aiming to use an iterative method. Forming hypotheses to construct models and make a prediction; designs an experiment and collect data to be further analyzed.

Design is the last one, it have it comes from engineering where system requirements and specification are defined. The systems are designed, implemented and teste, like the others it is an interactive and never ending process.

## 2 Review of related literature

### 2.1 Apps for health

“Apps for health Apps have also entered the medical field. In a recent review of articles discussing the development and evaluation of smartphone applications for health, Mosa, Yoo and Sheets [5] make a distinction between apps for healthcare professionals (including disease diagnosis apps, drug reference apps, and medical calculator apps), apps for medical and nursing students (including anatomy tools and electronic versions of medical books), and apps for patients (including chronic disease management apps and fall detection apps). For medical professionals, the use of mobile technology has been found to be beneficial, as it allows them to make decisions more rapidly and with a lower error rate, and to increase the quality of data management and data accessibility [6]. For patients, mobile technology improves patient education, self-management of chronic diseases and it greatly enhances the possibilities for remote monitoring of patients [5]. And these technologies are widely used. A recent study by the Pew Research Center pointed out that 31% of cellphone owners used it to access health information, while 19% of the smartphone owners have installed an app to manage their health [7]. A study among medical providers showed that 56% of them use apps in their clinical practice [8].”<sup>3</sup>

### 2.2 App overload

There will always be a lot of good apps out there or just apps that do exactly what a user needs but the user are never going to find the app because there are too many bad, or mediocre apps, or apps that just do not have what the user need that they need to look through first. There are many studies on how to improve search engines to give users what they want or at least guide them to what one believe they want.

This wide use of search engines like Google have resulted in a situation where people expect that if they write a word or two they get what they need. For google this more often than not work because google have so much information about the user’s behavior and what other users have been looking for [9] . This is great when one has the data, but the narrower the subject the less relevant data there is. However because of the amount of people using it (section 2.1), one can assume that Google have user behavior data that will help them in the app search. This program however has no opportunity to use such user data because it is not a part of the metadata. The big problem is still that there are just too many apps the algorithm might help, but it is not magical and cannot give a perfect result.

van Velsen and his team [2] did a study on this problem and the conclusion they came to is that the only real way of fixing this problem is creating third party apps that only give the good apps as a result. The second problem is that there are a lot of apps that are

---

<sup>3</sup> van Velsen, Lex, Desirée JMA Beaujean, and Julia EWC van Gemert-Pijnen [2] p 1-2

good but too narrow resulting in a situation where they do something very good, but they do not do enough to be worth having it on a separate app. Therefore it is also needed to create better apps that have access to more data and can do more things at the same time.

Abu Saleh Mohammad Mosa, Ilhoi Yoo and Lincoln Sheets [5] did a systematic study of different articles that again studied different medical apps, grouping them according to target users. They studied what the differences and similarities between these apps were in a step on the road to standardization of apps layout and what they do. As a small example of how hard it can be to find when one is looking for, when they started out they found 2894 articles that might talk about what they needed after skimming over most of them they were down to 114 and then after reading them down to 59. That is a lot of work and that's just in order to find articles about medical apps. Considering there are 2.2 million different apps on the Play<sup>4</sup> store alone it is in reality impossible to go through every single one and say if they are relevant and good at whatever they do. Therefore any database with only "good" apps is going to be incomplete, because it can only contain the apps that a human has taken the time to go over and analyze.

For gaming apps one can generally trust that when people rate an app as good the game is probably good too. That is not true when it comes to health-related apps, the main reason for this is that the regular person is not qualified to say if the information given by the app is correct or not. An example of an app that is easy to state as untrue for anyone that knows anything about the subject is "Cancer Curing Foods"<sup>5</sup> With this app red light starts coming up when one reads the name because of the fact that there are no cures for cancer at this point in time, there are many promising results for treatment of cancers [10] (better than going in with a scalpel and hoping one can remove everything). And as such any app that claims they can tell the user about how to cure their cancer is quite suspect. On the other hand putting in a flat statement that removes everything making this claim might one day in the future be a problem. Other than the word statement "cancer curing foods" there is not anything in the app's metadata that an algorithm can complain about. Almost all reviews are positive the only negative one is a person having trouble installing the program. It has a score of 4.4 among 43 users. Considering among the 700 000 apps that were cataloged in this thesis the average app has 4188 reviews this is a small number on the other hand if one uses the term Cancer one gets 140 apps whose average reviews is only 61.

---

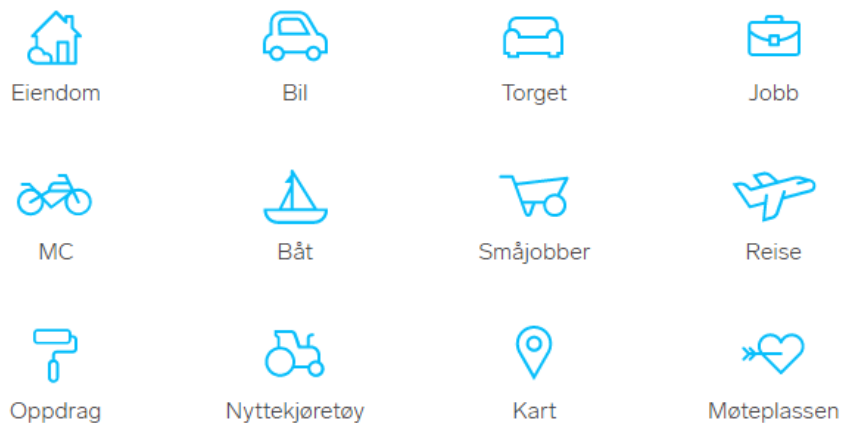
<sup>4</sup> <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>  
25.11.2016

<sup>5</sup> <https://play.google.com/store/apps/details?id=com.proven.cancercure.AOUJZCYXQQQEVGMK>  
10.12.2016

### 3 Review of related technologies

#### 3.1 Related services

There are web pages out there that also have a large amount of data that they have to present to the users, some of them are worse, but others are more users friendly than the one in the play store. One of the best examples of this is Finn<sup>6</sup> this page an everything one might be interested in page. Figur 1 shows what subject one can pick between, if a person wants a job, a car or a new house one can find this on this web page.



Figur 1 Finn.no top level categories

If one were to select job then one are presented with the option to limit the search to only part time, supervisor or all jobs. Next is the main search page (see Figur 2) on the left one can limit the search more and on the right one can see the different jobs that come up in the given search. This allows the user to use search words, and category limiting in order to find exactly what they are looking for.

---

<sup>6</sup> <http://m.finn.no>





**Stilling**

- Butikksjef (28)
- Forretningsutvikling og strategi (22)
- Franchise (16)
- Ingeniør (41)
- IT drift og vedlikehold (18)
- IT utvikling (15)
- Kvalitetssikring (13)
- Ledelse (278)
- Mat og servering (17)
- Prosjektledelse (37)
- Rådgivning (15)
- Salg (41)
- Salgsledelse (29)
- Undervisning og pedagogikk (27)
- Økonomi og regnskap (39)

[Vis alle](#)

**Bransje**

- Barn, skole og undervisning (35)
- Butikk og varehandel (68)
- Bygg og anlegg (69)
- Drift og vedlikeholdstjenester (21)
- Eiendom (20)
- Forskning, utdanning og vitenskap (35)
- Helse og omsorg (36)

**Betalt plassering** Blomsterdalen

Spennende nytt konsept på Bergen Lufthavn Flesland!

**Kjøpmann/Daglig leder søkes til NORTHLAND**

NORTHLAND  
1 stilling

Ny i dag Arendal

**Logistikk sjef**

Maritim Båtutstyr  
1 stilling

Ny i dag Kristiansand S

**Senioringeniør**

Hodejeger Åstveit  
1 stilling

12 timer siden Oslo

**Leder nasjonalt salg**

Relacom  
1 stilling

Figur 2 Main search page

### 3.2 Web crawler

A web crawler or web robot or we spider as it is also knows as is an automatic program that download web pages. The program starts on a web list or a list of web pages; it then takes all the URLs on that web page and adds it to its list of URLs. When it finishes with a page it picks a new URL from its list and continues working [11]. In order to create a web crawler there are two problems that need to be solved one is relatively simple the other is harder. The first problem is reading web sites. For most web sites this is simple because after all web browsers have to be able to read them so that users can get to them. There are always going to be some sites that are not meant to be read by humans. So if one want the crawler to understand that one need to do a bit more work. But in general web browsers can do it there for a crawler can to. It might be a lot of work but its doable. The big problem with web crawling is that the web is so large, if one want to crawl one billion pages in one month one have to visit 400 pages every second. This means that the crawler both needs to handle large amounts of data in a short amount of time. But also most web sites do not like DDoS attacks and a web crawler and a DDoS attack look a lot like one another if one gets 400 requests every second from a single



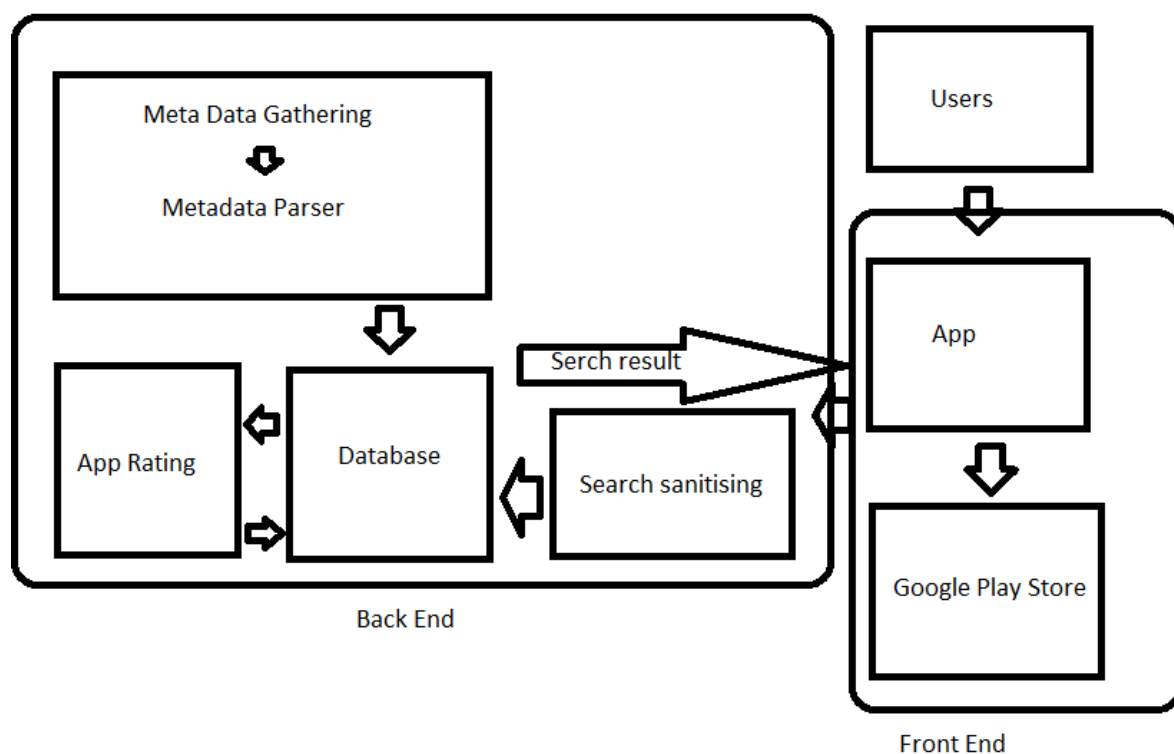
server. This means that in order to not be shut down the crawler have to spread itself around so that it talks to different servers and not a single server. And make sure it waits a period between each time to talks to a given server.



## 4 Design

This program uses the back end created in Ruben Mæland thesis [1]. This is to say it uses his program for the app metadata retrieval, parse this metadata to find apps about cancer and create a database out of these. The part this thesis will look into is rating these apps and looking into their relevance. This amounts to adding more metadata to the database because even if the algorithms used here where to find an app extremely unlikely to be useful it should still show up if the user really wants it.

On the front end there is an Android app that allows the user to specify advanced search parameters inside the category of cancer where the search results are listed after either preset criteria or by the user's override.



Figur 3. The system architecture

### 4.1 Back End

The back bone in the back end is the database that holds the metadata about any given app. This is where the web crawler is to deposit the data it finds, where the user's searches are ultimately to be handled everything goes around the database. This back end can be considered a black box<sup>7</sup>. That is to say one put data into it, take it out change it and put it back in again. How the data is handled inside these black boxes is of no real

<sup>7</sup> [https://en.wikipedia.org/wiki/Black\\_box](https://en.wikipedia.org/wiki/Black_box) 15.11.2016

interest to this program. The database is the most important point in the system, and as such is noting there is any point in reinventing the database.

Because of this the question is what system to use; first one is the SQL or NoSQL and then what implementation to use.

#### 4.1.1 SQL vs NoSQL

Each system that uses the databases just interface with the mongo database, but never with one another. That is to say the web crawler has no idea that there is an app rating system or an app that also uses the database. Same for the app it just knows that the data it wants is in the database (or the lack of data if the is the case).

There is a drawback from considering the database as a black box and that is that sometimes it might have been smarter to let programs talk directly with one another. The flip side of this potential increase in efficiency is that it allows the system to be modular. If one look at Figur 3 the system has 4 distinct components.

- Data gatherer

Gather in the data from the Google play store and store in in the database.

- Database

The database itself that holds the data and makes sure nothing gets lost.

- Data evaluator

Generate ratings for apps and evaluate the database to get statistics from it

- User interface

A front end app that acts as the access point for any user wanting to find the app they are looking for. The app is intended to send requests to the database in order to give the user the information they are interested in.

#### 4.2 MongoDB

This system uses the MongoDB one could list the virtues back and forth but in the end the simple answer is that is the system the web crawler uses and there were no compelling reasons to change it.

### 4.2.1 Meta data gathering

In order to have any data to work on, one needs to get the data. The simplest way of doing this in the case of the Google Play is using a web crawler on the web portal for the Play store<sup>8</sup>.

## 4.3 App rating

In order to design an rating system for the apps

### 4.3.1 Example apps

#### 4.3.1.1 Pink Ribbon Breast Cancer

This is a simple app in the awareness category, so in general should not be a problem. When one search for “Pink Ribbon Breast Cancer” one get a lot of results. None of the results gives a real clue what app the researchers where studying so let’s take a look at some of the top results.

Most of this are breast cancer wallpapers in different languages costing about 1\$ each<sup>9</sup>. Great looking apps all of them, but interestingly enough if one try to visit the developers website one are informed that it does not exist<sup>10</sup>. If one use the WayBackMachine<sup>11</sup> one finds that at least at some points there is a redirect link to a different site belonging to “The Breast Cancer Library's Blog”<sup>12</sup>, only problem on this site is that there are a few post from 2010, then one post about the apps in 2013 and that is all. In short all these apps cost 1\$ each does not give any confidence in that the money goes to support breast cancer programs<sup>13</sup>. Considering this is the point with the pink ribbon all these apps have to be considered suspect.

Just studying the Meta data this is hard to find out. A Program can try the link to the developer, find it not working, great the app is suspect, but as the WayBackMachine shows sometimes it works. In this case the apps does not have any user rating so one can use that to get rid of all of them, but if they did have a good rating what then?

The third most liked app<sup>14</sup> have 3 rating of 1,3 and 5 stars, so it can be tossed out because of too few ratings, because the program plans to only present the “good” apps this have an up and down because the newest app might be the best app this app came

---

<sup>8</sup> <https://play.google.com/store/apps?hl=en>

<sup>9</sup> <https://play.google.com/store/apps/details?id=com.ebook.wallpaperlatvian> 11.11.2016

<sup>10</sup> <http://www.thebreastcancerlibrary.com> 11.11.2016

<sup>11</sup> <https://archive.org/web/> 11.11.2016

<sup>12</sup> <https://thebreastcancerlibrary.wordpress.com> 11.11.2016

<sup>13</sup> <http://thinkbeforeyoupink.org/resources/before-you-buy/> 11.11.2016

<sup>14</sup> <https://play.google.com/store/apps/details?id=com.staffordsigns.ribbonwallpapers> 11.11.2016

out 16.07.2016 according to the meta data, so many it is just so new people have not spotted it yet? So if the algorithm automatically tosses it out it is never going to show up on any list, therefore the algorithm have to reduce its score not toss it out. Next is the fact that all comments on the app complains about the fact that the app does not work. There are two options for handling this; one is to use keywords, like “bugged, “refund” and the like.

This app has 10-50 installations and 3 ratings. An interesting statistic to check out, as in how large the ratio of people trying to people rating. It might be problematic because of the size of the range. The thing that once more makes the app questionable is the web site listed as the home site of the developer. The site belongs to a company making custom drum decals<sup>15</sup>, what that has to do with “Pink Ribbon” app is unknown. Again an app that has a questionable developer braced on the home site, and again hard for an algorithm to find, but this time a bit easier. Considering there is nothing on the site about the app or any apps.

The second app on the recommendation list is “Breast Cancer Ribbon doo-dad”<sup>16</sup> this one looks promising, it have a 4.3 star rating. With more than 200 ratings and 10 000-50 000 downloads, and only a few of the comments on the apps are negative. The home page of the developer is an interesting view, but looks to be legit.

Then the most popular app is “Ribbons - Breast Cancer Icons”<sup>17</sup> this one to have more than 200 ratings 10 000-50 000 thousand downloads and 4.5 star rating. Almost all written reviews are positive and the home page of the developer looks legit.

#### **4.3.1.2 *The Ride to Conquer Cancer***<sup>18</sup>

Next app that is still in the store is one that is in the gray area, the home site is a legit site, it have mixed reviews both writhen and score, with 28 people that have rated the app and with people complaining about bugs and problems with others saying it works perfectly.

The problem with this app is that it is impossible to know if the complains is because people are complaining on the app not doing what it should do when it should do it. The purpose of the app is taking how far people are bicycling during a two day window. The longer the users travel the more money is raised for charity. So it is natural that they are unable to raise money outside of this two day period. This might be an annual even hard

---

<sup>15</sup> <http://www.staffordsigns.com> 11.11.2016

<sup>16</sup> <https://play.google.com/store/apps/details?id=com.dml.ribbon.breastcancer> 11.11.2016

<sup>17</sup> <https://play.google.com/store/apps/details?id=com.jayrod.ribbons> 11.11.2016

<sup>18</sup> <https://play.google.com/store/apps/details?id=com.conquer.canada> 11.11.2016



to say from the app. However it is the app can be great at what it does but because it only “works” two days out of 365 users might get frustrated by it.

This app shows where one has to decide on a divide between an algorithm encompassing enough to include this app or narrow enough not to include it. If the algorithm does not include this app then people is most likely going to get annoyed that they are not finding what they were looking for. On the other hand if they find it they might be happy, because they got what they wanted, or they might be unhappy that they found a bad app and thereby reducing the credibility of the algorithm.

#### **4.3.1.3 Cancer.Net Mobile**

Another app with good reviews, good score and 150 ratings. The developers website is completely valid and it have 10 000-50 000 downloads. So from just the metadata the app is perfect. However there is a danger sign in the app all the bad written reviews are from after the newest update and are complaining about the update. This might just be people that want the old app back same as “everyone” complains when Facebook<sup>19</sup> update their layout<sup>20</sup>.

#### **4.3.2 User rating**

With the examples above to consider in general the user rating of apps have to have a good priority, with a lower limit somewhere in re region of 200-100 votes. This will result in a situation where new apps are in trouble. Some experimentation is required to find the right balance so this number is just a starting point.

Then one also has to consider how one is going to weight the reviews that are written. Are the newer ones going to gain more weight or do one weight them the same? There are ups and down to both, if one put more focus on the new ones a developer can pay someone to give a good reviews in order to boost the apps score [12]. There is nothing one can do to avoid this when one are using the metadata. On the other hand one can miss out on sudden drops in score if bugs or the like comes up. However this method also got the problem that it is weak against manipulation. If a group of people decide they do not like something they can go in and give a lot of bad reviews and that could drop the score drastically. For more on this read the work by Mao Chen and Jaswinder Pal Singh [13]

---

<sup>19</sup> [www.facebook.com](http://www.facebook.com) 11.11.2016

<sup>20</sup>

[http://www.slate.com/articles/technology/technology/2009/03/stop\\_whining\\_about\\_facebooks\\_redesign.html](http://www.slate.com/articles/technology/technology/2009/03/stop_whining_about_facebooks_redesign.html) 11.11.2016, article about how people dislike the new Facebook look and getting over it.

### 4.3.3 Words and phrases

For most people this is the most important thing. However users are not the most intelligent people out there, something that the general contempt the medical community has for homeopathy and the contrast to the popularity it have among some part of the public. This might result in a situation where apps proclaiming the virtues of different homeopathy cures for different types of cancer might be highly rated but still not be recommended. Therefore the rating program looks for words and phrases that should not be part of a good app and words that often is a part of a good app. Because the author of this thesis have no education in the medical field this list is quite stunted as and a is mostly based upon statistical analyses then looking at the apps using words that started ringing bells if none of them looked good to the author the word where flagged. In short the word list used her is limited and might be totally wrong.

### 4.3.4 Narrowness

This one is somewhat hard to quantify, but in general when a program in this case app tries to do everything at once they have a tendency to get a lower quality. That is not to say an app that does a lot of different things might not be perfect for all of those things. However an app that is great at one thing might be a lot simpler and easier to learn to use. Because of this the more complex something is the more valuable do each part has to be. The problem is that the more new features one adds the harder it is for new users to use a program. A solution to this is hiding it away in advanced settings or menus so if someone knows how to find it they can use it if not they can just use the basic features. An example of this is Google Search<sup>21</sup> there are a lot of advanced search options but for those that do not want much there is basic search, then there are search in categories (all, images, videos...). Most of the time one does not need more than this, but if one where to need more then these options is there. They do not make learning to use the service harder, but it gives the features. The tradeoff is that it is harder to find these features and the user has to invest in finding them.

The next thing is that Google Search is only a search engine, if one is after something else one has to open a different page (application). Google Search is narrow it does one thing and it does thins one thing expertly.

---

<sup>21</sup> <https://www.google.com>

#### 4.3.5 Security and privacy

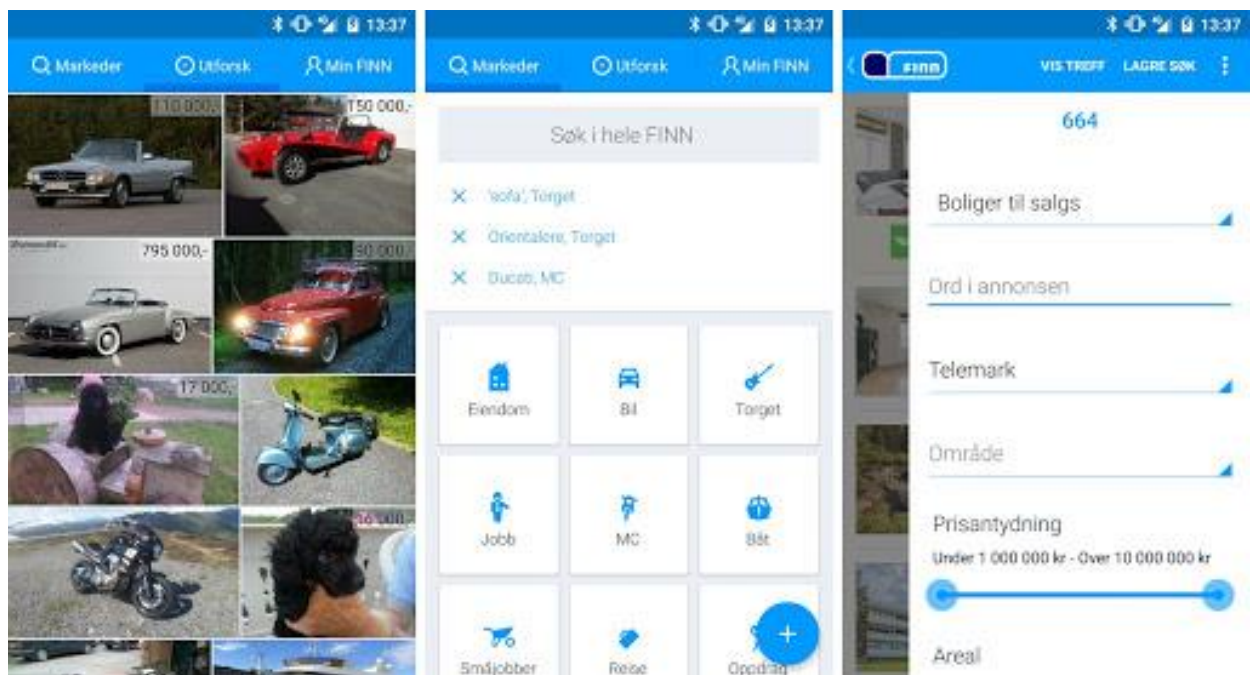
A problem that always exists and always will exist is ensuring the security and privacy of the users for this program there are two things that have to be considered. The first is the app itself how much data does it need about the user. The second is what impact the permission request is going to have on the rating of a given app.

One of the nice things about creating apps is that a lot of this is taken care of as long as one uses the built in functionality and do not try to work outside it. Future more this program has no need for any information about the users. Therefore the only permissions it is going to need are what are required to run it and install it. The program has no need to know anything about the user.

The second and more important is the app that is rating, what permissions they need, Mario Frank, Ben Dong, Adrienne Porter Felt and Dawn Song [14] found a pattern between the rating of an application and how many permissions they were requesting. It is then a good assumption that one can find potentially harmful applications by looking at what permissions they are after. One of the limitations on this is the number of categories that are available in the metadata; this gives a potentially large amount of apps doing wildly different things in the same datasets. So being able to reduce the datasets and create links between apps in order to find those that do the same things so that one can do a realistic comparison of permissions can be hard. However creating warnings for “obviously” dangerous applications should be easy. It does not need to be much just a warning that this application is asking for an unusual amount of permissions and that the user should double check that the program really need them all. For this algorithm marking them for human checkup with a request for clarification from the developer might be a good idea.

## 4.4 Front End

This section when somewhat out the window because of the result from the backend implementation turn up that there was no real point in implementing it. However the plan where to implement something close to the excising Play Store, with added features from the Finn<sup>22</sup> implementation. This allows the user to do advanced search if they find that the results they first find is not what they were looking for or there where to many results for their liking. An example of an advanced setting is allowing user to only look for apps that have been downloaded more then 10 000 times, thereby only getting apps that have had a lot of users testing the app.



Figur 4: Finn Android app<sup>23</sup>

<sup>22</sup> <https://play.google.com/store/apps/details?id=no.finn.android> 2.12.2016

<sup>23</sup>

<http://lh3.googleusercontent.com/tyOdMYBqcJpGxtWrPYqV73W8kOeJkNFQ7qSwX6ff3xh1Z4HawfbiwLbYW5gKkk-FNsBmcbgcANTVOBqqZ8Y> 24.10.2016

## 5 Implementation

### 5.1 Web Crawler

This part was originally meant to use Ruben Mælands program for more information read his Thesis [1].

Some interesting results started turning out from the web crawler, according to the crawler there were no reviews of the apps. The decision where made to try a different web crawler to see if it comes up with different results. The main reason for doing it this way is the fact that there is a good web crawler on Github<sup>24</sup> called Crawlerplay<sup>25</sup>. Unfortunately also this one gave the same results of finding no reviews. This is to say everyone that have rated this app have just left the rating and no comment on the apps. There is no guarantee that there might be some comments left, they might have been purged by google or the developers themselves but at that point it was impossible to know.

After some weeks the reviews started turning up again on the Google Plays Store, hard to say when because it probably took some time for me to notice it. Unfortunately it turned out that the structure of where the reviews are placed on the web side had changed. Because of this the web crawler had to be changed to handle the new cite structure.

Because of all of this the decision where made to change from Ruben Mælands web crawler to the Crawlerplay crawler.

#### 5.1.1 Bugs

These are the bugs that have been found in the Crawlplay implementation.

##### 5.1.1.1 Extra Permissions

For some reason the web crawler sometimes get an extra field in the Permissions, this bug where discovered when it results in a situation where the dataset claims there are many thousands of different permissions. Among 500 000 apps the dataset clamed there were 25 000 thousand different permission's. This is a completely impossible number to work with. However after looking at some of the permission's all those that looked good used the word "Allows" first in its description if one where to do a search for permissions that start there description with "Allows" one end up with 293 different permissions. Still a large number, but one it is possible to work with. So until the bug can be found and fixed a patch on this is running "fixPerm" (see section 5.4) that runs through and removes all permissions that do not start with "Allows" there is a chance

---

<sup>24</sup> <https://github.com/>

<sup>25</sup> <https://github.com/crawlerplay/GooglePlayAppsCrawler> 25.10.2016

that it might remove valid permissions, but if it does so it do not have any impact on this thesis.

#### **5.1.1.2 Reviewers**

In the metadata the value Reviewers and Score.Count both denote the number of reviews an app have gotten. Bothe values are stored as Doubles in the database, not sure why, considering it should always be a whole number, but that it is a decision made by the creator of Crawlerplay. They are not always the same value however; there are two cases of them differentiating.

- If there are no reviews Reviewers have the value of -1
- If Score.Count is larger than 1000 then reviewers is 1

Not that in the second case if count is 3840 then Reviewers is 3.84. The most probable cause is that the crawler is not handling the thousands separator correctly<sup>26</sup>. In English one often use ',' to separate large numbers for easy reading e.g. 100,000,000.5. This is probably the cause of the error, because of this if one want to use the Reviewers number use the Score.Count value.

#### **5.1.1.3 Unable to crawl reviews**

The Crawlerplay implementation of review crawling is a bacik implementation.. This implementation where crated before there where made a change to the Play Store, because of this change the crawler are unable to find the written reviews from the users. In order to get these the Crawlerplay have to be updated. This can be done relatively easily, but how valuable it is for this project is questionable. For the same reason that the description is of questionable value to the rating algorithm (see section 6.1).

## **5.2 The Datasets**

The system ended up with 3 main datasets first is a dataset made up of 140 apps that are the search results from the query "cancer"<sup>27</sup>. This dataset is easy to work with, but is not very interesting for any users because it is too limited. The next set is the previous 140 apps and then every app that is related to these apps. These apps are apps that Google play stores algorithm consider related to the first given apps. This dataset consist of 1802 different apps, this dataset is a lot more interesting in terms of working with and give a comprehensive result to the users.

---

<sup>26</sup> <https://docs.oracle.com/cd/E19455-01/806-0169/overview-9/index.html> 11.12.2016

<sup>27</sup> Result of a search on the google play store last updated 15.11.2016



The last and in one way most interesting but also least interesting is a dataset that consist of every app that is on the google play store. The dataset that is gathered so far is not complete. At the time of writing it consist of 700 000 different apps, but there are 400 000 more apps waiting to be added to the databases. On top of any app related to those 400 000 apps, so in short by continuing to run the web crawler this dataset is going to go up by a lot.

This last dataset is too big to be really interesting for the narrowed down subset that this is interested in, this means that in order to use the dataset one have to create a new subset. There are two way of doing this, one is by creating a generic interface where for all intent and purpose the we are only working on a subcategory called Cancer. The other one is go use a back end algorithm to search through the entire dataset and by using some requirements create a smaller database that only contains the interesting apps. The benefits of this is that users get faster responses because in place of having to search through hundreds of thousands of apps. They might only have to search through thousands. If there are interesting widening the number of apps the users might be interested in, all one need to do is change the backend algorithm and run it anew. The users will not notice anything except from one search to another they might get more apps or more search options. They might also get less option if it is discovered that some options are never used and are not interesting for the client therefor they are only and distraction.

### 5.3 Available data on the apps

When Crawlplay adds an app to the database the document containing the information about an app has 36 fields. Fig 5 show all the metadata on the app named Cancer 101 Treatment.

_id	ObjectId("58272e37e88b97c36978d54d")	ObjectId
ReferenceDate	2016-11-12 14:59:02.902Z	Date
Url	https://play.google.com/store/apps/details?id=revolxa.inc.cancertreatment	String
Appld	revolxa.inc.cancertreatment	String
RelatedUrls	[ 32 elements ]	Array
Name	Cancer 101 Treatment	String
Developer	Revolxa Inc	String
IsTopDeveloper	false	Boolean
DeveloperURL	/store/apps/developer?id=Revolxa+Inc	String
DeveloperNormalizedDomain	null	Null
PublicationDate	2014-12-27 23:00:00.000Z	Date
Category	MEDICAL	String
IsFree	true	Boolean
Price	0.0	Double
Reviewers	32.0	Double
CoverImgUrl	//lh3.ggpht.com/q4pRS8GjFPW7PrDWkipLhIPKkr8kkQN7R1shl8m6yPOHFwE39Vz7Nq8UtV4ImO...	String
Screenshots	[ 8 elements ]	Array
Description	This app will shows the information about cancer treatment. Best comprehensive overview cover...	String
WhatsNew	V 1.0 Cancer 101 Treatment Update new SDK for support android devices Complied with new Go...	String
Score	{ 7 fields }	Object
Total	0.0	Double
Count	32.0	Double
FiveStars	20.0	Double
FourStars	5.0	Double
ThreeStars	3.0	Double
TwoStars	2.0	Double
OneStars	2.0	Double
LastUpdateDate	2014-12-27 23:00:00.000Z	Date
AppSize	0.0	Double
Instalations	5,000 - 10,000	String
CurrentVersion	1.1	String
MinimumOSVersion	2.3.3 and up	String
ContentRating	Unrated	String
HavelnAppPurchases	false	Boolean
DeveloperEmail	Revolxa.inc@gmail.com	String
DeveloperWebsite	null	Null
DeveloperPrivacyPolicy	null	Null
PhysicalAddress		String
InteractiveElements	null	Null
ReviewsStatus	Visiting	String
Reviews	null	Null
Permissions	[ 4 elements ]	Array
PermissionDescriptions	[ 4 elements ]	Array

Figur 5 Cancer 101 Treatment a example of an app in the database

Out of this information one are able to extract information on the developer of the apps, unfortunately many of the apps are missing information about the developer. 36%<sup>28</sup> of developers do not have a homepage.

<sup>28</sup> 48,603 out of 133,138 developers, that where checked.

<input type="checkbox"/> _id	ObjectId("582a5261366f4531287e2fd9")	ObjectId
<input type="checkbox"/> DeveloperPrivacyPolicy	null	Null
<input type="checkbox"/> DeveloperNormalizedDomain	null	Null
<input type="checkbox"/> DeveloperWebsite	null	Null
<input type="checkbox"/> Name	Revolxa Inc	String
<input type="checkbox"/> IsTopDeveloper	false	Boolean
<input checked="" type="checkbox"/> Apps	[ 4 elements ]	Array
<input type="checkbox"/> [0]	ObjectId("5825de58c20c85678a443b03")	ObjectId
<input type="checkbox"/> [1]	ObjectId("5825de6ac20c85678a443b1e")	ObjectId
<input type="checkbox"/> [2]	ObjectId("5825de7cc20c85678a443b39")	ObjectId
<input type="checkbox"/> [3]	ObjectId("5825de93c20c85678a443b5b")	ObjectId
<input type="checkbox"/> DeveloperEmail	Revolxa.inc@gmail.com	String
<input type="checkbox"/> DeveloperURL	/store/apps/developer?id=Revolxa+Inc	String

Figur 6 Example of Developer

This information is missing on the Play Store and is not an error in the web crawler. As note in 4.3.1 some developers that have a home site have one that is questionable. So what about those that do not have a home site? Some of this might be students or people that just want to make an app that do something helpful. These apps can be great, but with 36% all developers do not have a home site. Because of this large number if one where to disqualify all apps based on this one are going to lose a large number of them. While if one does not then one lose an important point in the evaluation of if an app is actually valuable.

### 5.3.1 Limited categories

Her one comes to the point where design and reality crashes together and gives problems. There are a lot of data on each app, but at the same time there is not much data on the apps.

### 5.3.2 Incorrect information

One problem that is hard to do anything with is the fact that the information the developer has put on the app is not true or at the very least is a twist on the truth. A very good example of this is "Davis's Lab & Diagnostic Tests"<sup>29</sup> it is listed as a free app, but the truth is that it cost money if one wants to use it. It is free to download and looking at the interface, but anything more one have to pay for it. Unbound Medicine, Inc<sup>30</sup> response to one of the customers complaining on the false advertisement: "Hi Elizabeth, our applications are listed as a free app because we allow our users the option to preview the content before purchasing. Please feel free to call into our support team with any questions." This is a good argument because a person might not like the interface, and finding that out before getting the app is a good thing. But there are no listings of price on the play store. They state that it is a free preview, but they do not list what it actually cost. This means that for the user to find out what it cost hey have to

<sup>29</sup> <https://play.google.com/store/apps/details?id=com.unbound.android.cqdtl> 28.11.2016

<sup>30</sup> <http://www.unboundmedicine.com> 28.11.2016

download and install it. This also means that it is impossible for this program to find the actual cost. Unbound Medicine, Inc have more than 50 apps on the Play store 10 of these have a listed price, how many of the once that are listed as free actually are free is unknown.

#### 5.4 Backend Data analyzer

The back end data analyzer is intended to generate data so that when a search is done the data do not need to be generated on the fly. The main point is to generate the rating of the program so that the client knows what order to place the apps in when presenting them to the user. It is also used to generate information about the database

**Tabell 2 Backend command prompts**

Command	Description	Input values
initDB	Run all functions needed to create all the collections	DBName (DB <sup>31</sup> )
minDB	Removes all collection that can be calculated from the data left Not recommended!	DB
addField	Copy collection from one database to another	Collection, fromDB, toDB
addDev	Add a new Boolean field to all documents in a collection:	Collection, DB, Field, value
devStats	Prints statistics on developers	DB
nullDevs	The number of developers with no data on home site	DB
premColl	Find all permissions used and add then to the Permissions collection	DB
getPrems	Prints all permissions used in the database	DB
avgPrem	Prints average number of permissions in apps:	DB
avgPremC	Prints average number of permissions in apps in a category	DB, Category
getCats	Prints all categories in the database	DB
catColl	Add any new categories to the database	DB
upWord	Update the word collection, Not for large datasets	DB
lsWord	Get the least used words, not for large datasets	DB, NumberOfWords
fixPerm	Does a fix on the database to remove all permissions that do not start with "ALLOWS"	DB
toFile	Print app data to a file using XML, output goes to the outDir	DB, CollectionName, fileName
avgReviw	Print the average number of reviews an app is getting	DB

---

<sup>31</sup> Name of the database

Most of these functions are meant to test that everything is working and test the running speed of the operation before it can then be combined to create the evaluation algorithm. Or they are just there to get interesting data about the database.



## 6 Evaluation and result

The biggest problem is finding what apps to use as a “this is a good app” benchmark. There are a lot of web pages talking about what one should and should not do when creating an app. The interface is important, but this is just one aspect, an app also need so have content of value. In the context of this program this means that in order for this app to have a value it needs to deliver a product that the user wants. The truth is that in order for this program to be of any use it need to not only give a result, it also needs to give result that is worth more than the original search engine can give.

Tabell 3 General database statistics

Database	Apps	Developers	Categories	Permissions	QApps
cancerDB	698,431	133138 <sup>32</sup>	34	275 <sup>33</sup>	428,976
CancerOnlyDB	140	110	12	50	0
CancerDBrelated	1802	527	24	134	0

The data above is information about the 3 different datasets that are gathered and used. CancerDB is somewhat of misnamed dataset because it is developing to get all apps in the Google Play Store. At the time the data gathering where stopped qApps have a lot of apps that have not been added to the list and those again have more apps to be added. With more than 2 million apps on the app store it is a lot more crawling to do before it is finished.

The interesting point about the permissions is that there are quite a few of them but not so many that it is impossible to create a rating for each of them so a future project can create a rating system based only on the permissions that apps need, to be more precise a warning system for when an app ask for a bit more than they probably need. Just be warned that such a system is probably going to have the same problem this thesis has found in this work.

---

<sup>32</sup> This number is not up to date, there are apps added after this where last updated.

<sup>33</sup> This is after removing the bugged permissions before that it was 24 987

## 6.1 Description

Tabell 4 Words used in Descriptions, CancerOnlyDB

Word	Count	Word	Count	Word	Count
cancer	940	features	31	liver	21
app	279	body	30	consult	21
information	129	family	30	healthy	21
treatment	89	research	29	different	21
application	89	search	29	available	21
help	82	useful	28	home	21
breast	77	support	28	cell	20
medical	74	diseases	28	version	20
doctor	73	best	27	note	20
symptoms	70	treatments	27	prevention	20
prostate	69	effects	27	today	19
free	59	people	26	purposes	19
skin	59	way	26	great	19
colon	58	important	26	tnm	19
cervical	57	complete	26	families	19
risk	53	want	26	day	19
care	53	provides	25	community	19
make	48	tumor	25	things	19
health	48	game	25	hope	19
like	46	facts	24	learning	19
learn	45	latest	24	imaging	19
use	44	does	24	oncology	18
download	42	developed	23	knowledge	18
share	41	signs	23	colorectal	18
patients	40	tools	23	read	18
know	40	patient	23	stomach	18
easy	39	lung	23	used	18
horoscope	39	email	23	android	18
cancers	38	sign	23	events	18
staging	38	factors	23	world	18
surgery	37	live	23	problems	18
cells	36	causes	22	test	18
types	35	questions	22	contains	17
need	35	stage	22	drugs	17
just	34	life	22	daily	17
access	34	data	22	based	17
new	33	content	22	http	17
friends	33	rate	21	lifestyle	17
disease	32	foods	21	spread	17
diagnosis	31	time	21	mesothelioma	17



There are some words that are used a lot in these 140 app descriptions the problem is that there are no true way of using the words that are used in the description in order to find out what the app is about. If one decide that one word is a good word and another is a bad one. The result is going to be the same as one have in spam filters [15]. A constant battle between the app creator and the algorithm creator, in order to prevent apps that use a word bingo to get through the algorithm. Even if this battle is possible to win it will require constant work and just for this reason it is not feasible. The next alternative is having the algorithm read the description and “understand it”. Unfortunately this is not even less feasible, as a very good example look on how hard it is for Google Translate<sup>34</sup> to understand text. In short it is going to be an extremely huge algorithm that might or might not work, and is far too much for this app or this thesis. Therefor the description is useless for evaluating the apps quality.

## 6.2 Developer data

Tabell 5 Developer home page statistics

Database	Developers	No home page	
CancerDB	133138	48603	36.5%
CancerDBrelated	527	164	31.1%
CancerOnlyDB	110	28	25.5%

With a lot of app missing data and the fact that when we looked at few apps earlier some of them had home pages that where suspect, so it is likely that a substantial amount of the remaining developers have suspect home pages. But finding that out can be hard, and such an algorithm is going to have to be large. Because of this the only real way of using the data about the developer is using the number of apps a developer have made and how good rating those apps have gotten. When an app supports multiple languages this increases the size of the app, because of this one get app like the pink ribbon app where there are 20+ apps just in different languages. This increases the amount of apps, but might also result in many of them having bad rating or no rating because there are no users using it in a given language. This will also hurt the download rate of the app, if it was in one langue it might have gotten 10 000 downloads, but because its 20 different apps, it only have 500 each.

---

<sup>34</sup> <https://translate.google.com>

### 6.3 Categories in the database

An interesting aspect of the Play Store is that there are only 34 different categories if one assumes that GAME\_ARCADE is a subcategory of game, and because of this there are 17 subcategories of game. Turns out 1 in 3 categories in the Play Store is a type of game.

This means that there is some sorting by category, but not detailed enough to create a larger tree of categories. So in order to create this tree the program need to crate these subcategories. And in reality without the developer declaring the categories themselves it is almost impossible to do. The best solution is the one used by websites like GoodRead<sup>35</sup> where users vote for what categories a book is a part of. Overt time a list of categories the books belong to starts forming.

---

Fantasy	43,080 users
Young Adult	13,926 users
Fiction	12,027 users

Figur 7 Category voted for Harry Potter and the Sorcerer's Stone

However crating such a system requires a user base and it not something that helps getting a baseline quality. In short it will help in getting people to continue using a program but it does not help with getting them to start using it in the first place.

### 6.4 Removing Snake Oil

There biggest problem is finding a of what is good apps, as noted again and again, the only real way of doing this is opening up the app and test it. It is possible to find a good app if one chose to compromises on something. One can do it the way the Play Store does already and compromise on the truthfulness of the content of the app. That is to say the Play Sore assumes that the users know what a good app is and what is not. Problem is that in the case of cancer there is the problem that a lot of people do not know what is based on science and what only contains Snake oil<sup>36</sup>. Because of this it is impossible to trust the rating of the app when one cannot compromise on the quality of the information. In order for an the app this thesis wanted to create to be useful it have to be able to give apps with valid information without this it is just a reimplementaion of the Play Store itself.

---

<sup>35</sup> [www.goodreads.com](http://www.goodreads.com)

<sup>36</sup> [https://en.wikipedia.org/wiki/Snake\\_oil](https://en.wikipedia.org/wiki/Snake_oil) 08.12.2016

One can use the rating in order to reduce the chance of finding snake oil apps the problem is this is only an assumption because in order to see the numbers one has to find out what is truly a snake oil app and what apps have valid information. Using only the metadata it is not possible to see what the app contains.



## 7 Future improvements and work

With the unfortunate conclusion that using data algorithms to only find the “good” apps is not going to work. There are always going to be some apps that are not great that are going to fall through the system and one end up with a reimplementing of the regular app store. Because of this there are really only two ways of going forward.

The first one is to implement a better version of the Play Store, but not try to call it anything else this can only realistically be done by Google them because getting people to move to any other app store that just links back to the regular one is hard to impossible.

The second is to use humans to analyze apps and validate them, this is something google is to an extent already doing with their Top Developer program, this will then be a specialized version of this where one have people with knowledge about the information presented in the apps validating them so that when the apps give a statement users can trust that the statement is true.



## 8 Concluding remarks

When this project started the plan where to create an app that should work as a frontend for a database giving of apps rated and verified, giving the best possible responses to users looking for apps about cancer. The problem with this ended up being that the algorithm have to choose between giving a very limited list that can only be expended by having experts look over the apps and verify them by hand. Or the program had to basically give the same result as the already excising app stores. In witch case it will end up as an attempt to create a better user interface. With no shame can the author admit that he is not capable of doing that.

With the limited information that the metadata are providing about the apps. The fact that any analyses of this data do not really say anything about the app itself because if nothing else over time app creators are going to learn what any such automated system think is a good app and adopt there apps metadata to fit this criteria's. Creating a fight to constantly update the criteria's or let the users remove them after they find them unfitting resulting in the situation that already existing on the app store. The implementation might be able to get rid of a lot of bad apps, but not enough to be convince users that it worth changing away from the general app store to a specialty app.

So to conclude, perfection in finding the best apps requires a hands on approach one can reduce the number of apps that have to be verified. However one cannot remove the need for it. There might be possible with the help of more data that that is not available through the metadata. With only the available metadata there is not much one can get out of the data and be able to guarantee a better result.

It turns out that the algorithm the google play store uses to order the apps are quite good at finding the best app and presenting them first in its lists. The only place it struggles is in terms of helping the user find what they are looking for when the user is unsure about what is out there. The best way of improving this is probably by implementing a user interface closer to the one that Finn is using. In order to do this one need more detail in the categorization of apps. This is not possible to do with any precision just from the metadata.





## 9 Bibliography

- [1] R. Mæland, "Retrieval of Health Related Mobile Applications," UiT - the arctic university of Norway, Tromsø, 2016.
- [2] L. D. J. B. a. J. E. v. G.-P. van Velsen, "Why mobile health app overload drives us crazy, and how to restore the sanity," *BMC medical informatics and decision making*, vol. 13, no. 1, p. 1, 2013.
- [3] OECD, "Skills Matter: Further Results from the Survey of Adult Skills, OECD Skills Studies," OECD Publishing, Paris, 2016.
- [4] J. S. B. a. B. C. Pierce, "What is a file synchronizer," in *Proceedings of the 4th annual ACM/IEEE international conference*, pages 98–108. ACM, 1998..
- [5] Y. I. S. L. Mosa ASM, "A systematic review of healthcare applications," *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, p. 1, 2012.
- [6] "The impact of mobile handheld," *Journal of the American Medical Informatics Association*, vol. 16, no. 6, pp. 792-801, 2009.
- [7] S. a. M. D. Fox, "Mobile health 2012," Pew Internet & American Life Project, Washington, DC, 2012.
- [8] O. I. a. T. F. T. Franko, "Smartphone app use among medical providers in," *Journal of medical systems*, vol. 36, no. 5, pp. 3135-3139, 2012.
- [9] E. B. S. D. Eugene Agichtein, "Improving Web Search Ranking by Incorporating User Behavior Information," in *SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval Pages 19-26*, Seattle, Washington, USA, 2006.
- [10] C. E. e. a. DeSantis, "Cancer treatment and survivorship statistics, 2014," *CA: a cancer journal for clinicians*, vol. 64, no. 4, pp. 252-271, 2014.
- [11] A. H. Marc Najork, "High-Performance Web Crawling," in *Handbook of Massive Data Sets*, Springer US, 2002, pp. 25-45.
- [12] E. e. a. De Cristofaro, "Paying for likes?: Understanding facebook like fraud using honeypots.," in *Proceedings of the 2014 Conference on Internet Measurement Conference. ACM*, 2014.
- [13] C. a. U. R. f. I. Ratings, "Mao Chen, Jaswinder Pal Singh," in *EC '01 Proceedings of the 3rd ACM conference on Electronic Commerce, pages 154-162*, Tampa, Florida, USA, 2001.
- [14] B. D. A. P. F. D. S. Mario Frank, "Mining Permission Request Patterns from Android and Facebook Applications," in *2012 IEEE 12th International Conference on Data Mining*, (pp. 870-875). IEEE., 2012.
- [15] J. G. V. C. a. D. H. Goodman, "Spam and the ongoing battle for the inbox," *Communications of the ACM*, vol. 50, no. 2, pp. 24-33, 2007.
- [16] Y. Y. a. N. A. Pern Hui Chia, "Is this app safe?: a large scale study on application permissions and risk signals," *Proceedings of the 21st international conference on World Wide Web*, pp. 311-320, 2012.
- [17] B. J. et.al, "A lot of action, but not in the right direction: systematic review and content analysis of smartphone applications for the prevention, detection, and

management of cancer, J Med Internet Res 2013;15(12):e287".