Aalto University
School of Science
Master's Programme in ICT Innovation

Siddhant Gupta

# Enhancing Search User Interface of an Assets Management Solution

Master's Thesis
Espoo, May 27, 2019

Supervisor: Professor Marko Nieminen
Advisor: Juha Lehtonen M.Sc.

Aalto University
School of Science
Master's Programme in ICT Innovation

ABSTRACT OF
MASTER'S THESIS

**Author:** Siddhant Gupta

**Title:**
Enhancing Search User Interface of an Assets Management Solution

**Date:** May 27, 2019 **Pages:** vii + 62

**Major:** Human-Computer Interaction and Design **Code:** SCI3020

**Supervisor:** Professor Marko Nieminen

**Advisor:** Juha Lehtonen M.Sc.

In today's time, information is of paramount importance and affects human life in conspicuous ways. Different companies, solutions and tools are developed to harness this information and retrieve is efficiently. One such tool is 'FA Platform', an Asset Management Solutions developed by FA Solutions Oy, which allows users to manage and search their assets and funds. This thesis contemplates over the issues faced by the search user interface of the 'FA Platform'. The thesis then proposes a new design implementation to improve the user experience of using the search functionality from the user's point of view.

This study explores the concepts of search user interface in-depth to understand the industry standards and research previously done in these fields that could assist this study. Interviews were then conducted with users of the software to identify problems with the current interface. To further identify the possibilities, client's search-usage history was collected and analysed for patterns. The problems identified were that the existing solution lacked features to add new search criteria, option to select multiple values, and the general user interface was cluttered and difficult to comprehend.

This thesis proposed a design prototype of the search user interface and tested ii with users against the existing implementation. Through testing, the new design prototype was found to be easy to use and provided a nice overall user-experience. The proposed design improved over the current implementation in three key areas: Redesigning the layout of search user interfaces to improve legibility; Adding an option to dynamically add more filter-fields as per requirement; and Providing an option to select multiple values for filter-fields.

Having a good search user interface design is an important factor while developing a database management tool as it will increase user's productivity, allowing them to find exactly what they want, how they want and when they want it. Even more so, a good design can also be very profitable for the company as it causes less distress to users.

**Keywords:** user interface, design, usability, search, information retrieval, user experience, management solution

**Language:** English

# Acknowledgements

It has been a different and interesting experience to write a thesis for the first time in my life. I learned quite a lot both about the academic achievements and the industrial applications of my topic. The user experience while searching information is quite important and has seemingly endless use-cases.

I have been extremely fateful to have a chance to pursue a masters degree in a discipline of my choosing from a reputed university in an developed country. None of this would have been possible without the constant support of parents who supported me both emotionally and financially throughout my education and life. My first thanks and eternal gratitude is towards my parents who actually made me capable of achieving my life goals.

It would wrong not to acknowledge the opportunity that is provided to me by EIT Digital Master School, Aalto University and KTH Royal Institute of Technology. I wanted to study HCI and design but have a background in mechanical engineering. These institutes nevertheless accepted me and allowed me pursue a degree in the discipline that I am most passionate about.

I want to thank my employer, FA Solutions Oy, for providing me with this change to experience their environment, to learn the workings of an internationally distributed software and allowing me write my master's thesis around it. Special thanks to Juha Lehtonen, my boss and advisor, for assigning me the task improve their software, for permitting me to use it as a topic for thesis and for advising me at every step of the way.

I wish to express my gratitude from the bottom of my heart to my supervisor, Dr. Marko Nieminen, who guided me throughout my thesis meticulously regardless of his other commitments. He was always patient, respectful and understanding of my situation. He saw potential in my work even when I did not see it myself. I thanks him for not giving up on me.

A shout-out to my friends who kept me going when things got tough.

Thank you all for making me what I am!

Espoo, May 27, 2019

Siddhant Gupta

# Abbreviations and Acronyms

| | |
|---|---|
| AMS | Assets Management Solution |
| API | Application Programming Interface |
| GUI | Graphical User Interface |
| HCI | Human-Computer Interaction |
| IPYNB | Jupyter Notebook File Extension [1] |
| PDF | Portable Document Format File Extension [2] |
| SaaS | Software as a Service |
| SQL | Structured Query Language |
| SUI | Search User Interface |
| UI | User Interface |
| UX | User Experience |
| XLS | Microsoft Excel Spreadsheet Binary File Extension [3] |

1. `https://fileinfo.com/extension/ipynb` accessed on May 5, 2019
2. `https://fileinfo.com/extension/pdf` accessed on May 5, 2019
3. `https://fileinfo.com/extension/xls` accessed on May 5, 2019

# Contents

# List of Figures

# Chapter 1

# Introduction

> Humans live in an age of information and are surrounded by it, whether someone is sharing it, selling it or even stealing it.

Irrespective of how it is perceived, life in today's time will be paralysed if the gigantic amount of information becomes unavailable. Furthermore, the sizeable information is futile if there is no way to retrieve it successfully and reliably. Almost all websites and applications which has a database and something to store in it, features an option to search through their database and retrieve required information. (Baeza-Yates and Ribeiro-Neto 1999)

The same is even more so true for web-applications that deal with managing data and information for users who are managing their real-time assets including money, properties, investments and more. These *Assets Management Solutions* are essentially a major information retrieval system wrapped in a user-friendly user interface. Hence, the ability to search and filter data in an AMS must be prioritised and the experience to use it should be made as effortless as possible. (Russell-rose and Tate 2013)

The software in this category are not build with user-centric design but are more data-centric and maybe even restricted by technology in order to sustain compatibility over various devices. This could cause software to ship with an search user-interface which may not be considerate of the user or their experience. (Moore and Rugaber 1993)

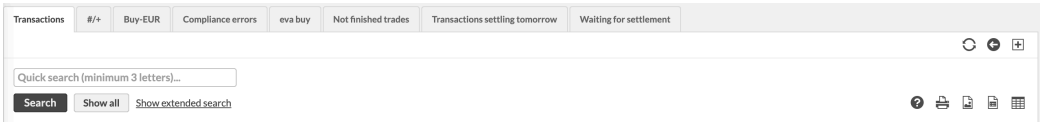
This thesis takes the challenge to understand the problems faced by an Asset Management Solution made by FA Solutions Oy specifically in regards of its Search User Experience. Furthermore, it would also make an attempt to provide solution in the form of a redesigned search user interface.

## 1.1 Motivation

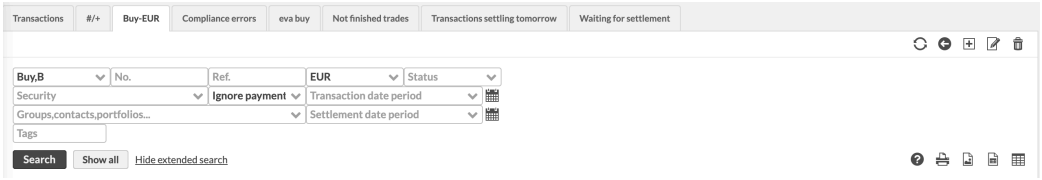
FA Solutions Oy[1] is a Finnish company based in the Kamppi region in the city of Helsinki. It was founded in 1999 and has expanded in Europe, Asia, Middle East and North America. The company specialises in providing an advanced Assets Management Solution. The solution is a Software platform as a service model and, is targeted to asset managers, fund managers, private banks, family-offices, individuals and other parties managing wealth or other assets.

The main offering of the company is called FA Platform, a web-based cross-platform software solution, which allows users to create, read, update and delete information about customers, portfolios, transactions, trade-orders and more. The solution is designed to serve needs of various clients through various modules and features which are built into one platform. FA deploys a separate environment or installation for each client to avoid cross-referencing and data-leaks. All environments have their own database which only contains information about respective client's environment. *FA Platform* is explained in-depth in section 2.2.

The platform deploys various options for searching and filtering the data. There are two types of search-views or search-layouts available in the platform. *Quick Search* that allows users to type and filter the database available in the specific view (as shown in fig. 1.1(a)). Alternatively, an option exists known as *Extended Search*, allowing users to minutely input filtration criteria for multiple fields at same time (as shown in fig. 1.1(b)).




(a) Quick Search




(b) Extended Search

Figure 1.1: Search Area in FA Platform (Current)

1. FA Solutions Oy, `www.fasolutions.com`

This is where the shortcomings begin to unravel. The *Quick Search* function is unclear regarding the process and the fields that are indexed. On the other hand, *Extended Search* is quite clear regarding which fields are indexed and affected but the user-interface is crowded and it is challenging to locate the exact field needed, if that even exists.

The company provided the author with a case to to examine their software, understand the problems faced by the users and propose new interface design to enhance the user experience.

## 1.2 Research Problem and Criteria

The *FA Platform* is a marvellous tool but is, nevertheless, riddled with some problems when the user-interface is concerned. The issue is most apparent in the search/filter area inside Views, as depicted in previous section (1.1).

The overall goal of the thesis is to produce a proposal for new user-interface for searching and querying data in the software developed by the FA Solutions Oy.

Therefore, the main Research problem is

> *How to improve to search user experience for the FA Solutions' Assets Management Solution?*

Sub-problems addressing user-centric usability are:

1. Whether filter fields that are not frequently should always be shown?
2. If an ability to choose fields that should be visible improve usability?
3. Can an option to select multiple values per field benefit the user in filtering?

The company established a criteria for measuring the success of this research. The thesis will be successful if the problems related to current SUI can be identified and implementable solution can be proposed which solves the apparent issues.

The scope of the study is quite limited since the case-company has compartmentalised development/improvement of the search-capabilities of the platform. The current study only deals with the issues related to the user-interface of the search and not dwell over the technical aspect of information retrieval.

# Chapter 2

# Background

This chapter digs deeper into technical and academic developments about search user interface and how can they be evaluated. This chapter should be a good read for some one who is not well versed with the topics covered in this thesis.

It is followed by details about the case software, *FA Platform*, and its functioning. The chapter will assist in understanding the current state of application.

## 2.1 Search User Interface

> "The role of the search user interface is to aid in the searcher's understanding and expression of their information needs, and to help users formulate their queries, select among available information sources, understand search results, and keep track of the progress of their search." - Hearst (2011)

This section aim to provide an overview of previous academic research in the field of search user interfaces and their design and evaluation. The reason to provide a contextual background is to present knowledge about the subject required to conduct this study and then evaluate the results.

### 2.1.1 Overview

Wilson (2011) in his book about Search User Interface Design introduces the importance of a SUI in order for humans to find the information that they are finding from a computer. The interface can be in the form of a command line, spoken, selection based, or even virtual reality. If the information is provided offline or online, it is a given that it would be searched by someone. Wilson also suggests there are six factors that contributes to the success of the SUI design. Ignoring the constraints caused by external factors, the six contributing factors are:

**Information Retrieval**. The availability and efficiency of IR algorithms affect the interface. The algorithms are responsible for searching and returning correct set of results in timely manner. In a research conducted in 2003, response time of search results was cited as being one of the most important factors that could affect the experience of using search engines (Liaw and Huang 2003).

**Graphic Design**. The graphical elements like layout and colour have shown to have profound impact on user's trust of a website (Cyr, Head, and Larios 2010). Zheng et al. (2009) showed that it takes people a fraction of a second to determine the professionalism and trustworthiness of website, based on purely aesthetic responses. Wilson, André, and Schraefel (2008) showed that simple visual cues can also impact the success of a SUI feature. These findings directs that aesthetic design decisions have a significant impact on usability of a SUI.

**User Experience**. User experience (UX) is a much broader term under which aesthetic or graphic design resides. According to Hassenzahl and Tractinsky (2006), there are three components of UX: emotional and effective factors, holistic and aesthetic factors, and temporal experience factors. With use of methods like personas, wire-frames, prototypes etc., UX designers try to simplify UI and make it more intuitive for a given set of users. Vermeeren et al. (2010) collected and analysed 96 methods that are being used by UX designers. Two of the methods mentioned are User Interviews and Task-based Usability Testing.

**Human-Computer Interaction**. Again, Human-Computer Interaction (HCI) is a broader term than UX and focuses on the methods through which users use the computer in order to complete tasks. The research in HCI assist in proving conceptually that a certain UI style could be better than others in a given scenario,

**Information Seeking**. Information Seeking (IS), also known as Information Behaviour, is a discipline dealing with how or why people search for certain information. There are multiple search situations, multiple stages

while searching, different feelings at different stage and the various tactics used when searching. Fisher, Erdelez, and McKechnie (2005) discuss about 72 of such information behavioural theories that further provides an overview regarding searching activities.

**Library & Information Science**. There exists people who create, organise, and curate sources of data and their metadata. Irrespective of type of collection, database, corpus of documents or generated information, the search feature can only show the results that have been correctly indexed or curated. Wilson (2011) gives an example, "searcher cannot filter a set of technical documents by the theme or subject, if thematic metadata has not been generated". Hence, the stored information and generation of metadata are interlinked.

All six of the factors mentioned are equally important for the success of a good Search User Interface. Considering the limitations of this thesis, many of the mentioned factors are out of the study's scope.

Prominently, the study focuses on the the aspect of *Graphic Design* as the author is allowed to modify that factor in order to improve the application. Furthermore, a basic understanding of *Information Retrieval* is required is order to completely grasp the utility of the proposed graphic or visual design.

### 2.1.2 Information Retrieval

As described by Baeza-Yates and Ribeiro-Neto (1999), the Information Retrieval (IR) is a research field that had aimed to get data out of computers since data was first stored in them. The retrieval has always been system focused, that is finding the most relevant results for the given query. In time, researchers realised that the interfaces should guide users to evolve or improve their searches. This was the basis of Interactive Information Retrieval (IIR) (Borlund 2003) where users could refine or improve a query to further narrow down the results.

A discussion about relevancy of results (Saracevic 1996) discovered that relevancy is improved if the task around the search or intention of the user is considered instead of just matching terms and keywords. Subsequently the search system's design did become more aware of the process of Information Seeking (Kuhlthau 1991). The process of Information Seeking is defined as the intention of a need for information (Schamber and Marchionini 1996), where the user recognises the need, formulate a query, examine the results, and accomplish the task.

Recently though, *Exploratory Search* (White and Roth 2009) has been utilised to represent situations where users are not able to perform a quick or

simple search, but relies on different searches, results analysis and evaluation (Marchionini and Gary 2006). The software provided by the case company implements a GUI form of *Exploratory Search* which enables users to search and filter the stored data without learning the method of SQL querying.

### 2.1.3 Graphic Design

The graphic part of an SUI is commonly referred to as Interface Visual Design. According to Hearst, a design process begins with determining user's intended goals and then developing an interface design that could help the user achieve the intended goals by performing a series of tasks. A UI is commonly designed in iterations, where the tasks and goals are established by user research. Once the initial designs are created, either based on pre-existing designs or new ideas, the design are tested with prospective users. The designs are then evaluated and re-designed and the cycle repeats. Furthermore, the process of evaluating an interface is different than the process of evaluating an algorithm of information retrieval. Quality of an interface is determined by people's response to it. The response is subjective and qualitatively measured. People's choice may vary due to factors like speed, aesthetics, familiarity, perceived accuracy or some preferred features. "In search interfaces particularly, a new interface idea must be perceived as being qualitatively better than an old one before users will switch." (Hearst 2011)

### 2.1.4 Developing SUI

Berger (2011) describes how design methods formed via design thinking can assist SUI design. The paper then proposes a design thinking process for SUI design which includes six steps: (1) the problem and context must be understood by designer; (2) the users' problems are observed; (3) the gained knowledge in interpreted and analysed; (4) idea are thought of using uncommon or common techniques. (5) ideas are visualised and communicated through prototypes; (6)The prototypes are tested with users.

Pernice (2018) advocates for user-interviews as a research method for gauging problems face by users. This method is more intimate as interviews are generally one-on-one session in contrast to other methods like focus groups. Interviews tend to be easy and quick method to collect data about user and used quite often in agile and lean environments. The article mentions some worthy guidelines and tips to conduct a good user interviews, which have been followed in this thesis.

This thesis follows these six steps and they are further elaborated in chapter 3 (Methods and Data).

Nielsen (1993) suggests five basic attributes for usability testing: (1) ease of learning the functionality; (2) the efficiently of system to perform tasks; (3) ease of remembering the system functionality over time; (4) amount and type of errors made by user; and (5) user satisfaction in regards to the system.

> "Usability testing means gathering information about the usability of user interfaces or their prototypes from users who are not involved in the design of the products" - Holleran (1991)

Real user are involved as participants in usability testing. Although the number of participants depends on factors like user-roles to be covered, amount of time and money, but Nielsen (1994) shows that the first few participants provide the most data or information and any additional participants are most likely to reveal fewer and fewer problems that can categorised as severe. For an example, a study observed that 80 percent of the problems were identified with only four or five participants (Virzi 1992). Thus, Nielsen recommends having four plus-minus one users in a usability test.

Zabed Ahmed (2008) does mentions *Questionnaires and Surveys* as commonly used for accessing users' satisfaction with regards to the interface. Furthermore, the "Questionnaire for User Interface Satisfaction (QUIS)" had been practised in numerous usability experiments and proved quite useful (Chin, Diehl, and Norman 1988).

This thesis follows the similar implementation of questionnaire after performing task-based usability testing.

## 2.2 FA Platform

The main offering of the company is called FA Platform, a web-based cross-platform software solution, which allows users to create, read, update and delete information about customers, portfolios, transactions, trade-orders and more. The solution is designed to serve needs of various clients through various modules and features which are built into one platform. FA deploys a separate environment or installation for each client to avoid cross-referencing and data-leaks. All environments have their own database which only contains information about respective client's environment.

Fig 2.1 shows the *Overview* View of FA Platform. It is the first view an user see after logging in to the platform.

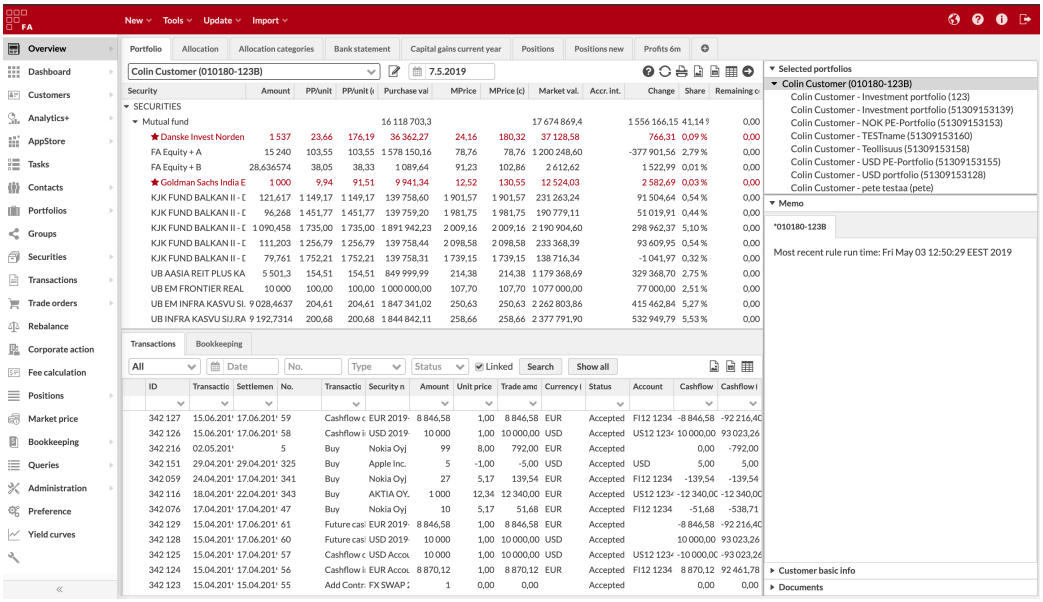



Figure 2.1: FA Platform - Overview View

### 2.2.1 Key Players

Within the ecosystem of the company, there are multiple stakeholders:

1. The Clients: The solution is targeted towards a multitude of clientele including asset managers, fund managers, private banks, family-offices and individuals. All clients are important to the company along with their satisfaction. They are the most prominent user of the company's services.

2. The End-Users of the Clients: Due to the possibilities presented by the company's solutions, the Clients can in-turn use this solution as service and direct it towards their end-users. This behaviour leads the company to always consider the possible end-users of the clients while developing the solution.

3. Internal Users: The company's offerings are also used internally by multiple teams of people for different purposes, namely, developing extensions, sales, marketing, providing professional services and providing customer services.

The professional and customer service providers are closest to the client as they are constantly conversing with clients about the issues and the problems.

### 2.2.2 Database Setup

The *FA Platform* uses a database to store almost all the data in an environment/installation. The database used is *MariaDB*[1]. Each environment has its own database; the database may be hosted on its own dedicated database server, or on the same server as the application itself depending on the size and bandwidth requirement of the database. By default, regular backups of the database are made every day during night to provide extra layer of reliability.

A considerable amount of information is stored in database including but not limited to

- All contacts, portfolios, transactions, trade orders, securities, tasks, processes, users, and all related objects (e.g. addresses, accounts, and profile data)
- A full version history of most of the aforementioned objects
- Some calculated values: purchase lots, daily positions and account balances, daily portfolio values (including indexed values), and transaction view objects
- Client-specific configurations

A common FA Platform's database contains around 200 tables. A table usually represents either some kind of *object* like contacts in the table named `pm2_contact` or a *linkage* between objects like signalling that which portfolio belongs to which contacts in a table name `pm2_portfolio_contact`. Linkage is required as it enables the possibility to link multiple objects of one type to a object of another type. A simple example would be to consider that a single contact can have multiple addresses or multiple contacts can exist in a single physical address.

A table is a combination of *columns* and *rows*. Columns are individual attributes of the object, e.g. contact's name is stored in the "name" column of table `pm2_contact`. Row represents a single object or linkage (depending on the table), e.g. every contact in the system has exactly one row in the table `pm2_contact`.

Every table as at least one column with unique values and that is generally called the *Database ID*. An object's "database ID" is the value of that object's "id" column. This id is a number, and is guaranteed to be unique within the table.

1. `https://mariadb.org/` accessed on May 6, 2019.

FA Platfrom uses a MariaDB database as it is an open-source alternative to MySQL[2] They both are a relational database. MariaDB has some performance improvements over My SQL while maintaining an almost identical query language/syntax. Fig. 2.2 shows how data is shown once retrieved from the database.

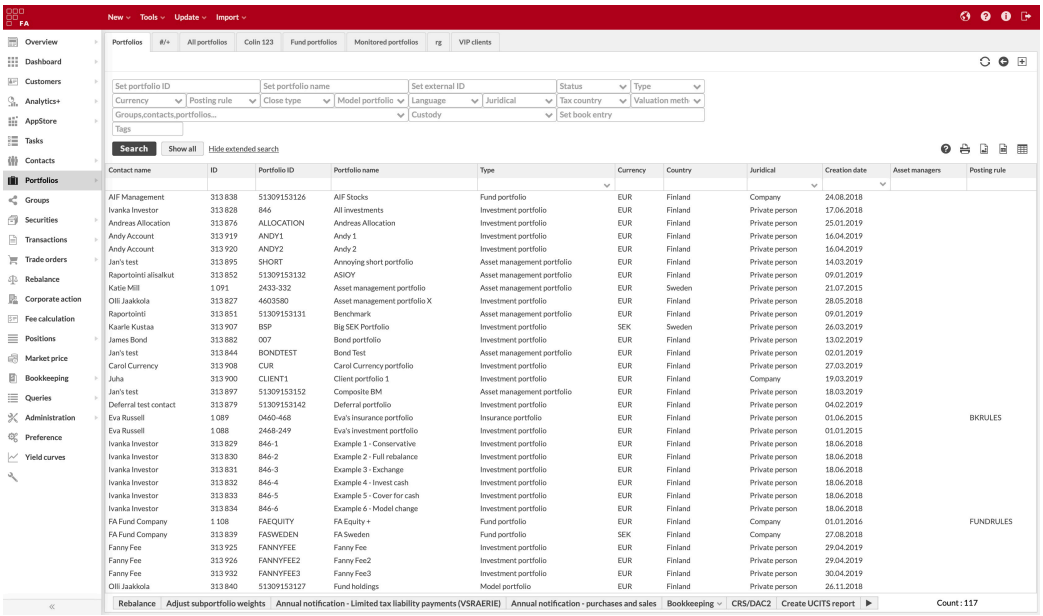



Figure 2.2: FA Platform - Portfolio View (Current)

### 2.2.3 Bookmarks

Every view in the platform contains a set of filter fields that allows users to filter the results to match their preferences. *FA Platform* offers a simple functionality to save and store these selected filters as a bookmark to be used at a later date. A bookmark does not store the records resulting from the search but only the search criteria. This implies that every time, a bookmark is accessed, a new search/query is placed and resulting records are displayed. It may happen that the resulting records differs due to change in date when bookmark is accessed or if the data is modified.

A bookmark can be saved with a custom name and shared with other users if needed. User can access these bookmarks from the tab-bar placed on the top of search-area. A set of saved bookmarks as tabs over the search-area

2. `https://www.mysql.com/` accessed on May 6, 2019.

can be seen in figure 1.1(b). In that figure, the "Buy-EUR" bookmark has saved the filters with some value.

# Chapter 3

# Methods and Data

This study follows a design method described by Berger (2011) and there, the author proposes a design thinking process for SUI design. There are six steps in the design process:

1. the problem and context must be understood by designer via methods like user-interviews, etc.;
2. the users' problems are observed, possible via information-collection.;
3. the gained knowledge in interpreted and analysed;
4. idea are thought of using uncommon or common techniques.
5. ideas are visualised and communicated through prototypes like paper prototypes or high-fidelity ones;
6. The prototypes are tested with users using methods like testing with users or usability testing.

Methods like *User Interviews* and *Collecting Search Usage Data* were applied to gather resourceful data and information about the application and its usage. The data collected would later be analysed to discover the results. Later, method of *Usability Testing* was applied to examine the validity of the results.

## 3.1 User Interviews

User interviews are generally considered a good practice for collecting user-feedback quickly and easily. They are focused on users' perception. In the authors personal opinion, an interview with user is quite informational and valuable is conducted properly.

### 3.1.1 Goal and Scope

The scope of the interview was decided to be limited in order to keep the goal of the interview as simple as possible. It is rare to witness a successful interview with a broad scope or a complex goal. (Pernice 2018)

The goal of the interview was to understand how a particular used the application and if faced any issues in her/his workflow. The scope was limited to searching and filtering information records inside the application, specifically in a single view.

### 3.1.2 Selecting Users

Since this was a very user-centred project, it implies that the most important stakeholders are the users of the application. Although, company's guidelines renders the direct interaction with the clients difficult. The primary clientele of the application are banks and funds managers, and setting an appointment with them was often cumbersome. The company also wished to avoid hinting the clients about potential improvements in application unless they are finalised, further preventing testing the designs with an actual client. This situation led to consider the next best user-base for the application which satisfied both of the following conditions: they use the application extensively and know it quite well; and are closely linked with clients of the company and have dealt with client's problems and issues.

Inside the company, several people who match the profile were contacted and engaged with. The subjects agreed to take part in the study voluntarily and provide valuable feedback and insights on different stages of the process. No rewards or incentives were provided to them as they were employees of the same company and this did not seem to affect the study in any way.

Six users were interviewed and analysed as a part of this study. The users were chosen from different teams in the company to reflect the occupational and ideological diversity. The users comprised of five males and one female. The average age of the users was 35 years. All users were born in Finland, and speak fluent Finnish and English. Furthermore, all users have completed at least one degree in higher education.

- Two user were from the *Professional Services* team
  The members of this team are closest to clients as this team was responsible for: Managing and executing on-boarding for new clients and other client projects; Ensuring the client's satisfaction and best possible solution to the clients during the on-boarding; and Developing the best practices and model to on-board different types of clients efficiently.

- Two users were from the *Product Development* team
  The subjects were the Head of Product Development and the Manager of Product Development. The members of this team are closest to understanding the product as this team was responsible for: Defining and managing the product (application) road-maps and versions; Maintaining product documentation for client's and internal reference; Specifying new features and functionalities for the application that may benefit the client; Making sure that the product works as specified; and Training employees from other teams regarding the product and its features.

- One user was from the *Customer Services* team
  The members of this team are also constantly communicating with clients and are responsible for: Developing and maintaining satisfaction levels of existing clients; and Supporting clients regarding troubleshooting and general guidance.

- One user was from the *Sales and Marketing* team
  The team is always meeting with prospective clients who do discuss about their preferences regarding the application. Furthermore, an improved feature can be included in a sales pitch. This team was responsible for: Acquiring new clients for the company by pitching current offerings; Building up market in the selected new segments; and Marketing, communicating, managing and developing online media and social channels of FA Solutions for attracting new clients, partners, new employees and interaction with existing clients.

### 3.1.3 Interview Questions

The interview was kept short and maximum time spent per interview was ten minutes. The interviews took place in an isolated meeting room at the company's premises. All questions were asked while an instance of application/software was running on the nearby computer for reference. User was allowed interact with the application in case the user was not completely familiar with a particular functionality of the application.

The following ten question were asked to all test-users. The interview was not strict as additional question were seldom asked to poke user for more information or clarification.

To the best efforts of the author, the questions were designed to be open-ended and not leading in any manner. Neither the author or the questions are intended to be biased or demeaning for the interviewee.

1. Since how many years or months have you been using FA Platform?
2. Which view or Views do you use the most in FA Platform?
3. When using a view, do you use the feature to filter or search the records?
4. For searching, do you normally used the saved bookmarks or do you generally search with different criteria?
5. There are 2 types of search in a View, Quick Search and Extended Search. Which do you normally have to use to use to successfully search in the View?
6. While using Quick Search, is it clear to you what actually is searched or which fields are affected?
7. While using Extended Search, do you find it easy or simple to find the field you're looking for?
8. Is there any issue that you've found or was communicated to you by a client?
9. Any suggestion for improving the overall search experience?
10. Finally, if you were given an option to add or remove filter fields as per requirement instead of relying on a predefined set of fields, would that solve any of your previously faced issues or concerns?

All answers were recorded for further analysis.

### 3.1.4 Analysing Responses

The author followed the process defined by Taylor-Powell and Renner 2003 for analysing qualitative data. There are four steps in the process:

1. In this step the data and quality is evaluated. This revealed that the data collected was sufficient to support solving the research questions as the there were 6 users interviewed (Nielsen and Landauer 1993).

2. This step is about focusing on the analysis. Four key questions were formed to focus the analysis.

   (a) Is the Quick Search feature required?

   (b) Is the Extended Search feature required?

   (c) How does user interact with Search filters?

   (d) How the search process can be made better?

   The responses from participating users were organised using these questions in order to find differences and consistencies.

3. Next step, the data is categorised. Table 3.1 describes the categories identified per question. The emergent or frequent categories were used further in the study. Frequent categories were defined after understanding the data and spotting recurring issues and themes (Taylor-Powell and Renner 2003). The tool - TAMS Analyzer for Macintosh OS X [1] was used to conduct the categorisation.

4. The last step identifies patterns and connection between and within categories. The final results are presented in section 4.1.

| **Question** | **Categories** |
|---|---|
| Is the Quick Search feature required? | not required, example feature |
| Is the Extended Search feature required? | example feature, requirements change |
| How does user interact with Search filters? | structure changes, content changes |
| How the search process can be made better? | style changes, dynamic |

Table 3.1: Categorisation of Interview Responses

1. TAMS Analyzer for Macintosh OS X, `http://tamsys.sourceforge.net` accessed on May 12, 2019

## 3.2 Collecting Search Usage Data

In *FA Platform*, an user finds needed information by filtering and searching the correct criteria in respective views pertaining to *Portfolios*, *Contacts*, *Transactions*, *Trade-Orders* and more.

Once the user has successfully selected the correct filters and retrieved the needed information in a particular view, the application allows the user to save that filter-combination as a bookmark or saved-search tab. These bookmark tabs can later be accessed and will return updated results for the same filter-combination, allowing quick information retrieval without repetitive user-input. These bookmark tabs are recorded in application's database as the application requires to reload them every time the application is launched and they differ for different clients.

Clients of the company and users of the application use the bookmark feature to save the most frequently searched criteria and use it extensively on daily bases, as learned from user interviews.

By the method of *Direct Querying*, the saved-bookmarks can be accessed for each client, which can be further used to understand how the clients actually use the search feature and which are the most frequently use filter fields in each View respectively. This data can be used to map client's search usage patterns and design better visual interface for searching and filtering information.

### 3.2.1 Direct Querying

The FA Platform is feature-rich tool that allows information to be retrieved via multiple ways such as filtering in separate views (the main focus of this thesis) and directly querying the database. In this section, the method of directly querying the database has been used extensively to retrieve information about client's search usage patterns.

The method of directly querying the database is a an advanced feature of the solution and is generally only available to users with the administrator privileges to the software. It is restricted because users without proper knowledge may misuse the feature by making poor queries that could overload the system and result in shutdown of the system. Although, the restriction can be removed and access to this feature can be given to all users if the client wishes to do so. This feature is available for following reasons:

- There is an exact search that is needed but is not available otherwise via standard functionality
- The search results need to be further used for other purposes including

- Monitoring the results for any inconsistencies or matching criteria
- Using the queries as template to generate reports
- Accessing the results elsewhere in the application via reference to saved query
- Referring to results outside the application through an API call which would refer to the saved query

The feature is accessible through the *Queries View* in an instance of the *FA Platform* to users with the administrator privileges. This view allows users to inject an SQL query and retrieve information directly from database in tabular form without any further filtration from the software. The view prevents adding new data, modifying data or/and deleting data from the database.

The software's database (section 2.2.2) supports SQL query in format of MySQL (`https://www.mysql.com`). This support empowers users to create complex queries which otherwise would be very difficult to execute via standard methods of filtration in the application. Queries may be saved and shared, and can be installed from *FA AppStore*[2].

### 3.2.2 Forming Query

In order to directly query the application database, a meaningful SQL query must be formed. It shall follow the format of a MySQL query with SELECT syntax as mentioned in MySQL (2019). The `SELECT` is used to retrieve data selected from one or more tables which can be filtered further as per requirements.

```
SELECT
    [ALL | DISTINCT | DISTINCTROW ]
    select_expr [, select_expr ...]
    [FROM table_references
      [PARTITION partition_list]
    [WHERE where_condition]
    [GROUP BY {col_name | expr | position}, ... [WITH ROLLUP
   ]]
    [ORDER BY {col_name | expr | position}
      [ASC | DESC], ... [WITH ROLLUP]]
    [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

Listing 3.1: Abstracted MySQL SELECT syntax

2. A marketplace build into the FA Platform that hosts various scripts, templates, queries and applets which can be installed by user/client into their application environment.

1. *Database Table*: An important part of `SELECT` query is the `TABLE` from which the data is to be retrieved. As it is understood in section 2.2.2 about application's database, different types of data is stored in different tables referencing each other forming a relational database. In this scenario, all saved-search tabs / bookmarks are stored in table `pm2_view`.

2. *Selected Fields*: In this part of query, generally the name of the fields are defined which needs to be extracted from table(s). The available fields in table `pm2_view` are:

   ```
   id, base_view, init_params, shared, user_id, version,
   view_name, group_id, parent_view_id, view_type
   ```

   In this particular case, fields that should be retrieved are `base_view` and `init_params` in order to better understand the data and derive insights from it.

   `base_view` contains the name of the View in which the search is made and `init_params` contains the actual search parameters.

3. *Result Filtration and Sorting*: To filter the results from the table, `WHERE` command is used but in this case, there is no need of further filtration as all data is required. Although it would be easier is the results are sorted by their Views. That can be achieved by adding `ORDER BY base_view` to the query.

Hence, the formed SQL query which can be used in the application is shown in listing 3.2.

```
1  SELECT base_view, init_params
2  FROM pm2_view
3  ORDER BY base_view
```

Listing 3.2: Formed SQL Query

### 3.2.3 Accessing Client's Records

For the purpose of collecting records of saved-search data, the only possible method is to run the formed SQL query in the *Queries* View of the client's *FA Platform* installation. For each client of *FA Platform*, the user who wishes to run the query must have ADMIN access and needs to run the query one-by-one on each installation. The query will return data in tabular form with fields `base_view` and `init_params` [3]. A sample of returned query can be seen below in table 3.2. For simplification only one row of each View is shown, although there could be multiple rows for each View as per client's preference.

| base_view | init_params |
|---|---|
| analytics2View | dynamicPortfolioSelectFilter=C-540;tim... |
| analyticsView | dynamicPortfolioSelectFilter=C-314491;... |
| bookKeepingView | columnsFilter=id,contactName,contactCo... |
| contactsView | juridicalFilter=CO;columnsFilter=conta... |
| crmView | field=SELECT ld.limit_name AS 'Limit',... |
| customView | field=select * from pm2_user_login |
| dashBoardView | style=1;1=SalesReporting-TasksTimeline; |
| portfoliosView | securityFilter=FAEQUITY B;transactionD... |
| reportsView | reportDateFilter=0;languageFilter=null... |
| securitiesView | securityTypeFilter=FUND;columnsFilter=... |
| tradeOrdersView | tagsFilter=FromOnline,;columnsFilter=i... |
| transactionsView | transactionDatePeriodFilter=CALYEAR-0;... |

Table 3.2: Simplified Sample of Direct Query

The application has a nifty feature to export the results as an file in either a `.PDF` or a `.XLS` format. For the purposes of further data visualisation, it is imperative that the data must exported in `.XLS` format which then can be read by other tools and scripts.

On 4 April 2019, there are total 42 clients that are using *FA Platform* and thus, there must be 42 files in `.XLS` format with 2 columns each [4]. In total, there were 3719 data entries (rows) from all 42 data-files.

3. A complete export of queried result (20 pages, 0.1MB) can be accessed at `https://www.dropbox.com/s/1elutbrqbgll299/Sample_Result.pdf`

4. For privacy reasons, the data exported after directly querying each client cannot be shared publicly with personnel outside FA Solutions Oy.

## 3.3 Designing and Prototyping

Designing an user-interface which can be implemented in the application was a crucial part of this thesis. From the perspective of the case company, the a user-friendly and implementable design was the most important deliverable.

Designing an application is rarely simple and unfortunately, it is almost impossible to please all the users. Different people have different expectations from the same application.

The process of the developing the design was kept straight-forward. The author has had previous experience with and prefers a design and prototyping tool called *Sketch* [5]. The tool is developed by a dutch company by the name of *Bohemian B.V.* and it is available only for Apple® macOS® operating system [6]. The tool allows to create both low- and high-fidelity user interfaces while providing support for rapid-prototyping.

While working on the designs, few guidelines set by case company were adhered to. They are listed as follows:

- Since the platform is already follows a set of interface design guidelines, all the new elements must be designed in accordance and must not feel inharmonious or out of place.
- The filter fields must not be changed in function. New purpose must not be added to existing fields as it may confuse the existing clients.
- Irrespective of design, arrangement and function of the search area, other areas must not be altered to fit the new user-interface.
- The freedom of alteration is extended to all the buttons, labels, icons and fields inside the search area of a View.
- As a general guideline, avoid changing the interface drastically as it will lead to confused and possible angry clients.

Using the Sketch tool, a clickable prototype was created which replicated the look and feel of the existing application (FA Platform) which allowed adding or modifying elements to improve the search user-interface as the author seems appropriate while adhering to the guidelines.

---

5. Sketch - The digital design tool, `https://www.sketch.com/` accessed on May 11, 2019

6. macOS Mojave - Apple, `https://www.apple.com/macos/mojave/` accessed on May 11, 2019

## 3.4 Testing the Prototype

After the the development of the interface prototype, the prototype was tested with a limited number of users. The task-based usability testing was conducted by on the prototype with controlled conditions.

The limitations on selecting users largely remained intact from the the time interviews were conducted with users (in section 3.1.2), but in fact the limitations increased for the purpose of testing. Since the testing involved presenting a potential feature which may be included in the application in the future, the upper-management of the case-company decided to keep the results of implementation private. This decision directly implied that the prototype can only be tested with any user who is a part of the management in the company.

### 3.4.1 Scope

The scope was the testing was to determine how does the user perceive the new design while using the prototype. The scope was aimed to match the testing criteria set by the case company. Primarily the new interface must provide an improvement in the user experience while using the search functionality.

### 3.4.2 Test Subjects

From the six users chosen from user-interviews, due to company-imposed restrictions, only two could be further consulted for the purpose of testing. These two users were part of the *Product Development Team*, specifically the Head and the Manager of the team. In order to receive a different and unbiased perspective, another employee from the company's management (and the *Sales Team*) was engaged with. This employee did not participate during the user-interviews and was introduced to this ongoing research for the first time during the testing.

All subjects were properly trained users of the platform with at least five years of experience. Furthermore, two out of three subjects have an established history in professional financial assets management, and do understand the needs of the platform from a user's point-of-view.

### 3.4.3 Setup

The test was conducted in an isolated meeting room at the FA Solutions Oy office premises in Kamppi, Helsinki. There were two computers present in the room. One was running an instance of current application and the other presented a clickable prototype of the new design implementation. The author acted as the tester and was present in the same room as the user.

The user was first introduced to the the environment and then was given the tasks to perform. Each task had to be performed first on the current application and then on the new prototype. An user had in total four tasks to perform. The author tried to keep the task simple, straight-forward and relatable to the daily-usage of the application. The tasks were as follows:

1. In Portfolios View, search all the portfolios where type is equal to "Asset management portfolio".

2. In Portfolios View, search all the portfolios where tax country is equal to "Finland".

3. In Transactions View, search all the transactions where transaction period is equal to "Current month".

4. In Transactions View, search all the transactions where currency is equal to "EUR".

After the tasks, the users were asked to fill a short survey regarding their satisfaction. The survey questionnaire is available in Appendix D. The questionnaire contains four questions with five options as defined by Likert Scale (Albaum 1997) to select from as an answer. The options range to extreme positive to extreme negative on a linear scale. To the best efforts of the author, the questions were designed to be not leading in any manner.

The whole of testing took twenty minutes per user which included the introduction, task-based usability testing and user-satisfaction questionnaire.

# Chapter 4

# Results and Analysis

In this chapter, various results derived from different methods are presented. For each existing section about methodology, there is a section dedicated to displaying results of that particular method.

In regards to user-interviews, a data-set of 58 minutes of raw audio recordings was collected after interviewing six users for approximately 10 minutes per user.

While collecting data for finding search-usage patterns, 42 files in `.XLS` format with two columns each (view and parameters) were collected. In total, there were 3719 data entries (rows) from all 42 data-files which were then analysed.

For the usability testing, 22 minutes of video-screen-recording, where participant's screen (monitor) were recorded while they performed the assigned usability tasks. In continuation, the three satisfaction-questionnaires were also collected for the purpose of further analysis.

## 4.1 Interviews with Users

The answers collected from users during the interviews were compiled and analysed. This section will present the results derived from the interview in italics which then would be followed by an explanation or reasoning.

- *Search is important and used extensively*
  During the interview, it was clear that the search is an indispensable feature inside any View as the raw unfiltered data is seldom useful.

- *The feature of Extended Search is used more often than the Quick Search*
  All the users who were interviewed unanimously agreed that they use

the *Extended Search* feature over the *Quick Search* as the former allows to narrow the search more reliably and more precisely.

- *The Quick Search is slow and its function is unclear*
  Contrary to its name, the *Quick Search* feature is felt slow by four users than the alternative. Also, none of the users had a clear idea regarding which fields are actually included in the filtration process while using *Quick Search*. An user was quite perplexed to learn that not all fields are affected while using the *Quick Search* and exclaimed

  > "I had no idea that quick search didn't search everything. Shouldn't that be the default case!"

- *The Extended Search feature is difficult to comprehend and required fields are difficult to find*
  Four out of six users found the *Extended Search*'s current implementation to be crowded with less spacing. The required fields are difficult to find in the grid and the readability is reduced by the fact that the input-boxes do not have field-labels. Also, one user mentioned:

  > "I can't find the right field as the fields are not consistent between the views. I understand some fields can't be everywhere but at least the ones which are, should be at the same place."

  Although, not all users were confused by the layout of *Extended Search* as two users were quite satisfied and used-to the placement on fields. One of those user had this to say

  > "I have been using this platform for so long that I have mostly memorised everything."

- *Both users and clients want a feature to select multiple items per filter*
  Among other issues, the interviewees mentioned the lack of a feature to multi-select items in a filter as one of the most important requirement. The users find it quite hard to filter the records properly without being able to select more than one option for any filter. When asked for an example, one of the user mentioned

  > "I normally deal with Nordic market and it would be nice if I could see data from all the Nordic countries. Currently I have to select one country at a time and then export its results and then select another and so on."

- *An option to reset all filters*
  As an suggestion, two of the users mentioned a feature request to add a button to clear all filters or reset them to default as currently it requires manual labour.

- *An option to add more fields in Extended Search as current fields are not enough in some cases*
  Three users mentioned the lack of flexibility regarding which fields are available for filtration and sometimes more are required. Although users didn't care much about removing filters.

## 4.2 Search Usage Pattern Analysis

The client's search-usage patterns were gathered by using method mentioned in section 3.2.3 and then analysed. During the analysis, the data is visualised in tabular and graphical format. Then, the analysis is split in 2 iterations where the visualisations are studied and some conclusions are drawn from it.

The goal of this analysis to determine which filter fields are most frequently used by clients/users on average. The results will focus on determining a list of four filter-fields for each View, with which a new interface design can be developed. After gathering all the data-files exported from all client installations, the data was ready for analysis.

For the purpose of analysis, a script was written in Python language [1]. The script/code was written in a Jupyter Notebook [2] and saved as a `.IPYNB` file. The script is attached to this thesis in Appendix A.

All `.XLS` data-files must be kept inside a folder named `query_data`. As long as the files are ending with extension `.XLS`, the actual name of the file is inconsequential to the script.

That folder shall be placed in the same directory as this code. The file structure must match the following:

```
[~root]/
|-- query_analysis.ipynb
|-- query_data/
    |-- data1.xls
    |-- data2.xls
    |   ...
    |-- data42.xls
```

1. Welcome to Python.org, `https://www.python.org` accessed on May 5, 2019
2. Project Jupyter, `http://jupyter.org` accessed on May 5, 2019

### 4.2.1 Data Visualisation

Once all files are in place, the script was executed. Since the the script environment is Jupyter Notebook, the results exists alongside the script in the same file, and can be accessed in Appendix B.

The results contains data representation in both tabular and graphical format. Figure 4.1 shows a sample of graph generated by the script showing the frequency of usage of each filter field in the *Contacts* View.

The raw results are depicted in table 4.1.

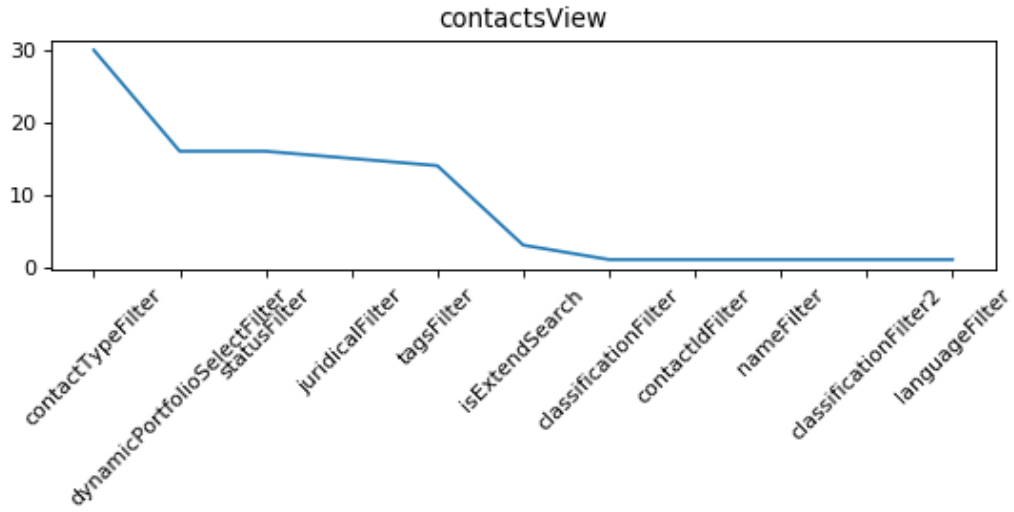



Figure 4.1: Usage of filter fields in Contacts View

### 4.2.2 Analysis Iteration 1

In this section, the thesis analysis the raw data and visualisation generated from the executing the script. The scope of analysis is limited and conclusions are made from understanding common practices in the company.

- From all the Views, the following doesn't require any further analysis as they are limited with their filtration capabilities.
  `analytics2View, marketView, positionsView` does not require further optimisation as they only contain 3 or 4 filter fields and all fields are important.
- Unfortunately `marketView` did not have any data to process and hence, cannot be included in the proposal.

| View | Most Frequently Used Filter Fields |
|---|---|
| analytics2View | dynamicPortfolioSelectFilter<br>groupByFilter<br>groupFilter<br>timePeriodFilter |
| bookKeepingView | dynamicPortfolioSelectFilter<br>statusFilter<br>accountNumberFilter<br>effectiveDatePeriodFilter |
| contactsView | contactTypeFilter<br>dynamicPortfolioSelectFilter<br>statusFilter<br>juridicalFilter |
| crmView | dynamicPortfolioSelectFilter<br>tagsFilter<br>juridicalFilter<br>statusFilter |
| portfoliosView | tagsFilter<br>dynamicPortfolioSelectFilter<br>statusFilter<br>typeFilter |
| positionsView | dynamicPortfolioSelect<br>securityFilter<br>transactionDateFilter |
| reportsView | languageFilter<br>purchaseValueTypeFilter<br>reportIdFilter<br>transactionListingFilter |
| securitiesView | tagsFilter<br>securityTypeFilter<br>nameFilter<br>statusFilter |
| tradeOrdersView | orderFilter<br>dynamicPortfolioSelectFilter<br>typeFilter<br>tagsFilter |
| transactionsView | dynamicPortfolioSelectFilter<br>tagsFilter<br>hasPaymentDateFilter<br>typeFilter |

Table 4.1: Most Frequently Used Filter Fields determined by Python Script

- The following Views can be omitted from the proposal as they behave differently that other views for different reasons:
  - `reportsView`: The data and bookmarks displayed in this view are saved reports query and that cannot be edited from this view.
  - `dashboardView`: The data displayed here is too complex to decode as it does not follow the same pattern as other views due to its different structure.
  - `queryView`: Similar to `dashboardView`, this View is different to other views as it stores the whole SQL query as parameters which are too complex to analyse and would not give any useful information in the scope of this thesis.
  - `analyticsView`: This View exists in database only for legacy-compatibility. The View is replaced by `analytics2View` in the application. Hence, it is irrelevant to propose design changes for it.

Thus, after a short analysis it is clear that few Views cannot be a part of this proposal and the focus should be kept on the remaining views:

```
transactionsView, tradeOrdersView, securitiesView,
portfoliosView, crmView, contactsView, bookKeepingView
```

### 4.2.3 Analysis Iteration 2

Now that the raw data is analysed in Iteration 1, Iteration 2 focuses on understanding the data further and what it implies. On a closer inspection, the findings are as follows:

1. It was revealed that `transactionView` already have some filters with previously filled information, causing that filter to appear in all queries irrespective of users' input. This view has a repeating field with value : `hasPaymentDateFilter="IGNORE"`.

   In `transactionView`, the field `hasPaymentDateFilter` is shown to be used 123 times (as mush as `tagsFilter`). It must checked if the value is ever changed by user and is so, how many times. If even after changing values, the field is used more that 25 times, then it must be included in the 4 most frequently used filter fields as the next most frequently used field is used 24 times (`transactionStartDateFilter`).

   It was observed that under the `transactionView` there are total 2187 data-entries and 123 data-entries contains the `hasPaymentDateFilter`.

Out of the 123 data-entries, all entries have the default value of `IGNORE` except 1 entry which contains `hasPaymentDateFilter=HAS`. It is apparent that this field is not actively used by users and must not be accounted when deciding the most frequently used filter fields.

Hence, `transactionStartDateFilter` should be considered in the top 4 most frequently used filter fields.

2. When considering DATE as a parameter while filtering the records, it is important to understand how the DATE parameter behaves and its impact while querying the database.

   The application considers DATE as a range instead of a static value. For an example if an user wishes to filter records for a single day (01.05.2019), then the application would consider it as range from `01.05.2019-00:00:00` to `02.05.2019-00:00:00`. This behaviour is a direct consequence of the way database interprets a date without associated time. By default time is considered 0 seconds. Hence, the above range will return records for single day and not two days.

   With that logic, it is not far-fetched to consider that date filters cannot perform adequately if there in only a start-date-filter or only an end-date-filter irrespective of analysis. Hence, if there is a start- or end-date filter in the most frequently used filters, its counterpart should also be included.

   Although Analysis clearly shows that a start-date-filter is used more than the end-date-filter. The reason for this phenomena is if user fails to input a value for a date filter, then the current-date is used as fallback. This leads to users only inputting start-date and allowing application to choose the end-date.

   Now that in `transactionsView`, unused field `hasPaymentDateFilter` has been replaced by `transactionStartDateFilter` field, it is essential that `transactionEndDateFilter` field is also placed alongside.

3. There is a clear repetition when considering fields across different views. Table 4.1 shows that:
   `dynamicPortfolioSelectFilter` field is used in following views:
   `bookKeepingView, contactsView, crmView, portfoliosView, tradeOrdersView, transactionsView`

   `tagsFilter` field is used in following views:
   `crmView, portfoliosView, securitiesView, tradeOrdersView, transactionsView`

Since a field is repeating in multiple views, it would be wiser to fix a position for them irrespective of their usage. For the purpose of this proposal, `dynamicPortfolioSelectFilter` will always be placed first in the View and `tagsFilter` will be placed last in the respective View.

After running analysis over different aspects of the results, there is better understanding of which fields should be considered for the four most frequently used filter fields for their respective views.

The final recommendation after analysis is recorded in Table 4.2.

| View | Most Frequently Used Filter Fields |
|---|---|
| bookKeepingView | dynamicPortfolioSelectFilter<br>statusFilter<br>accountNumberFilter<br>effectiveDatePeriodFilter |
| contactsView | dynamicPortfolioSelectFilter<br>contactTypeFilter<br>statusFilter<br>juridicalFilter |
| crmView | dynamicPortfolioSelectFilter<br>juridicalFilter<br>statusFilter<br>tagsFilter |
| portfoliosView | dynamicPortfolioSelectFilter<br>statusFilter<br>typeFilter<br>tagsFilter |
| securitiesView | securityTypeFilter<br>nameFilter<br>statusFilter<br>tagsFilter |
| tradeOrdersView | dynamicPortfolioSelectFilter<br>orderFilter<br>typeFilter<br>tagsFilter |
| transactionsView | dynamicPortfolioSelectFilter<br>hasPaymentDateFilter<br>typeFilter<br>tagsFilter |

Table 4.2: Most Frequently Used Filter Fields determined by analysis.

## 4.3 Proposed UI Implementation

At this point in thesis, it is imperative that all the ideas, analysis and solutions should be combined to create a visual representation which can assist users and clients in visualising the possibility.

From user interviews (section 4.1), it was gathered that the search is used extensively. The *quick search* is slow, unclear and not much used. The *extended search* is sometimes difficult to understand due layout of fields and the number of fields are sometimes too less to filter properly. Users can only select one option per filter and they cannot clear all filters efficiently.

After studying and analysing users' search patterns (section 4.2), it was discovered that few filters are used more frequently than others in respect to their Views. The start-date and end-date filter must be combined together as they are always used in unison. `dynamicPortfolioSelectFilterfield` field and `tagsFilter` field are used the most irrespective of the View in which they are present. With this information, the author developed an new interface for the search.

**Merging Quick and Extended Search**

To battle the the general confusion between the usability and functionality of *Quick-Search* and *Extended Search* layouts, both layouts were combined so the user doesn't have to switch between layouts.

After analysing the user's response and search-usage patterns, it was clear that users want to used specific fields for filtration as provided and there are some fields which more frequently used than others. Figure 4.2 presents a layout which acts a default view for a user accessing the *Portfolio View*. This layout features the top four most frequently used filter fields. It also makes sure that the common field like 'Portfolio' and 'Tags' are fixed to start and end respectively as these two fields will be repetitively visible in other Views too. All the fields have a field-label on the top of the input box for quick discoverability and all the fields are equal is width leading to a consistent and spacious grid. The layout otherwise contains two additional buttons: *Reset*, to reset all fields to their default values and *More fields*, to add more fields to the search-view layout.

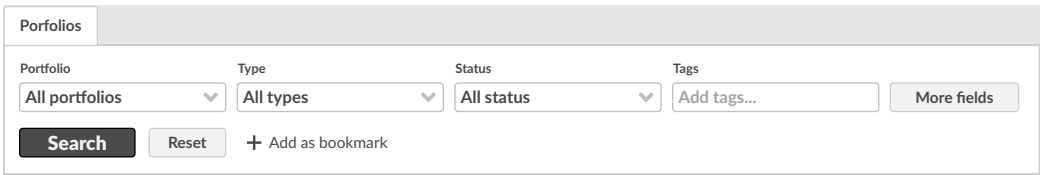



Figure 4.2: New unified search layout for Portfolio View.

### Adding Fields to the Layout

There is *More fields* button added at the end of the row in which all the default fields are placed. Clicking the button would render a drop-down modal that contains a list of all the fields available for that particular View. The list is in-turn filterable through an input field. User can select multiple fields at once by clicking or selecting the check-boxes alongside the field-name. The search layout will start to dynamically display all the filter-fields which are selected through the modal, in the order of selection.

The dynamically added filter will always feature an icon to remove the field if not required. Removing the field will remove its impact on the filtration of records. The field can also be removed by deselecting the field-name from the modal.

An user can add as many fields as required during the particular instance as shown in figure 4.3. Unless the user adds the filter-combination as a bookmark, the dynamically added fields will be removed when the page is reloaded.

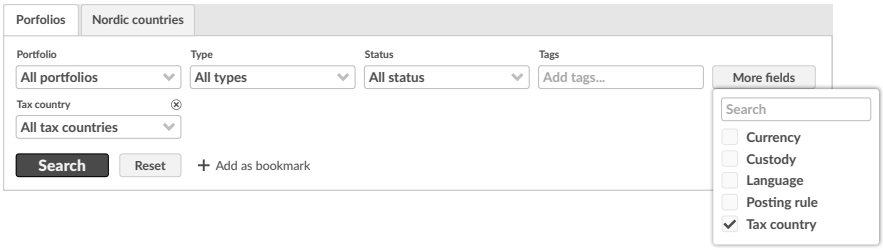



Figure 4.3: Additional field - "Tax country" added to the layout

### Selecting Multiple Items

Lastly, users should be able to choose more than one item or option per filter field. Figure 4.3 and 4.4 shows how an user can select multiple items.

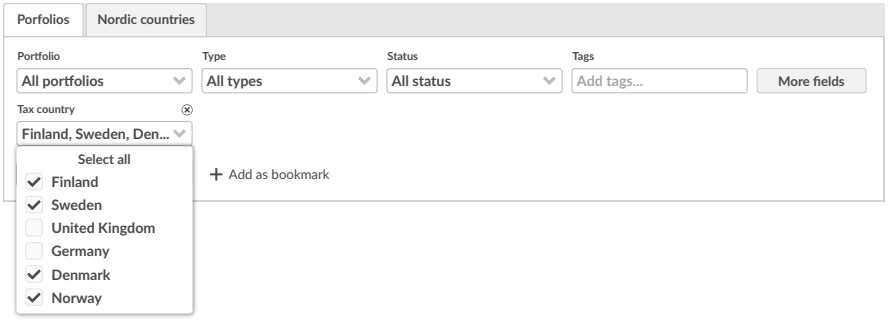



Figure 4.4: Selecting multiple items for a filter field

## 4.4 Test Results

The participants in the usability test were asked to evaluate the simplicity and ease of each task performed by them during the test. The users were asked to rate the ease in a range of 1-5. 1 indicated that the task was very difficult and 5 indicated that the task was very easy. The scores ware averaged are represented in figure 4.5. Light blue (light grey) bars refers to the scores received by the existing application whereas the dark blue (dark grey) bars refers to the scores received by the new design/prototype.

For reference, the tasks were: (1) Search all the portfolios where type is equal to 'Asset management'; (2) Search all the portfolios where tax country is equal to 'Finland'; (3) Search all the transactions where transaction period is 'Current month'; (4) Search all the transactions where currency is equal to 'EUR'.

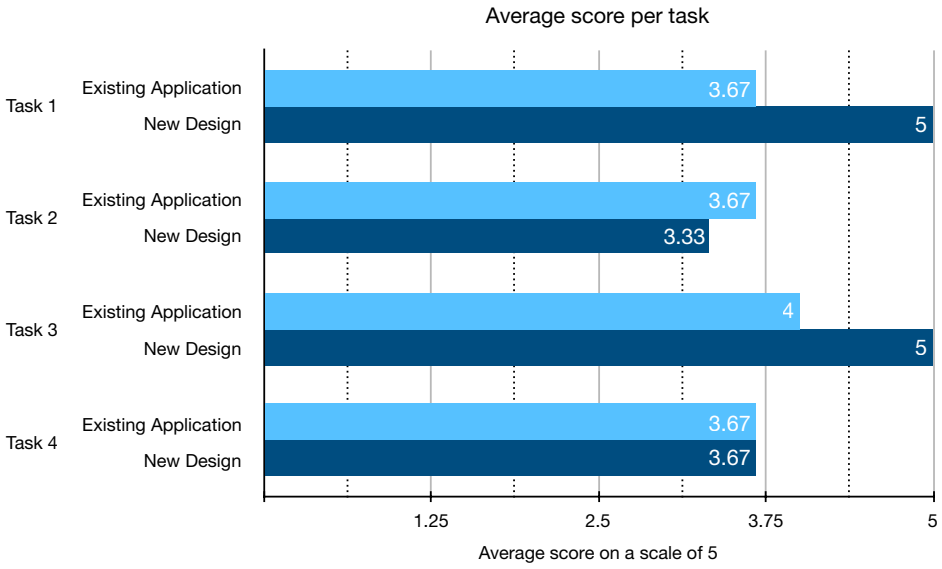



Figure 4.5: Average survey scores per task for all users

The ease of using existing application is mostly less than the ease of using the new design. Except, in two scenarios where the existing application either breaks even or even succeeds the new design. There is a pattern that can be seen here. The new design performs better when the required fields are displayed by default, and starts to lag when user has to add the field dynamically. Nevertheless, the new design doesn't lag behind too much in any task.

In regards to the second part of the questionnaire, the responses received by the open-ended are as follows:

- *Were the tasks assigned relatable to the daily-usage?*
  All participants agreed that the tasks were something that they would generally perform on the daily-basis.

- *Was it generally simple to perform the tasks in the new search view?*
  The general consensus was that the new design is simpler to use. Nevertheless, all the users are accustomed to the existing design and it will be little difficult to the change habits.

- *How was your experience when using the new search design?*
  The overall response was positive with an user saying:

  > "This looks so clean. It might take an extra click but everything is so organised."

- *Any other suggestions that you would like to add :*
  Two participants did provide suggestions:

  - "I wish there was an option to change the default fields if my workflow changes."
  - "Can this remember the fields I added last time and show them separately so I don't have to search the list every time."

# Chapter 5

# Discussion

In this chapter, the thesis will contemplate over the solutions that were found for the research problems. Furthermore, there will be a discussion about difficulties, limitations, retrospects and future possibilities.

## 5.1 Evaluation

**Problems faced by the Users**

The *user interviews* threw light upon the situation at hand. The interviews allowed the author to understand the environment, the limitations, the possibilities, and the expectations of the users. The results needs to be further evaluated and discussed upon their implications for the company and the *Platform*.

Search is important for the platform. This is one of those results which are sometimes considered as common sense. The *FA Platform* is essentially a data and information retrieval tool which heavily relies on the functionality of search. Users need the search to retrieve any format of data which is a bit more filtered than viewing literally everything.

A underrated revelation was that the feature of *Quick Search* is actually slower than the *Extended Search*. Users prefer the extended version of search over the quick one as the former provides more control and clearer understanding of the search functionality. This could mean that the users require more control over the fields that are provided for filtration. In an application, where anything can be searched, maintaining a single input field may only slow the system down.

Evaluating the criticism about the Extended Search functionality, it was revealed that although the feature is much used, a learning curve still exists which can be reduced. The experienced users have learnt or memorised the

functioning and the layout of the search-view but it is difficult to comprehend its complexity.

Moreover, the inability to select more that one item or option or value per filter-field is quite limiting for users and clients. Users sometimes wish to create complex inquiry but are limited by the applications' interface which currently doesn't allow to select or enter more than one value in a field. (Wroblewski 2007)

There is a basic functionality that is missing from the search view. The ability to reset fields to their initial state is quite common feature available in all forms which have more than a handful fields.

Finally, users are not content with amount of fields available in the interface to successfully filter the records. Currently, on average each View displays around 17 fields for user to work with but in reality the View boasts nearly a hundred fields which can be filtered. Hence, there exists some scenarios where user wishes that the records could be filtered more narrowly using an unavailable field. This seems to be a legitimate concern also shared by the author.

**Understanding Search-Usage**

There are some fields that are used significantly more than others. This indicates that not all fields have equal importance and the fields that aren't used as much could be hidden from the interface The more frequently used fields could be kept in focus. There are some Views which couldn't be analysed by the method followed by this thesis. Nevertheless, majority of Views were analysed and the script presented the usage patterns for those Views.

On additional analysis, it was apparent that two fields are most commonly used and furthermore, they are available in most of the views, making them the most frequently used fields in the whole Platform. The fields were `dynamicPortfolioSelectFilter` and `tagsFilter`. Since they are reoccurring almost every where, they must be positioned on similar places in all Views in which they appear.

**Validity of the Proposed Design**

The survey clearly showed that the new design was easier to use than the existing one even for the experienced users. Although it was not all smooth-sailing for the design upgrade. Since the tasks were defined in a manner that it can be achieved on both platforms, the tasks could not explore the full potential of the new design layout. The tasks involving user to add a field dynamically in new layout decreased the user-experience because those fields were already present in the existing application's layout. If there was a task to filter the records via a field which is not placed in current interface, the

new design will definitely seem like a better choice.

Nevertheless, the users appreciated the possibilities now available in proposed layout as well as the visual aspect of it. The proposed design improves over the current one by properly spacing the fields apart and placing field-labels with each input field for faster recognition.

## 5.2 Answering Research Questions

As mentioned in introduction section 1.2, the research question was:

> *How to improve to search user experience for the FA Solutions' Assets Management Solution?*

This thesis explores the problems and issues present in the platform and suggests improvements for it.

The suggestions are further evaluated and tested in order to answer the sub-questions of this thesis:

1. Whether filter fields that are not frequently should always be shown?
   The tests indicate that users can easily get confused if they are greeted with many fields. The problem worsen if majority of those fields are futile. Hiding them initially will improve user's experience. This case was only contradicted by long-time users who have memorised the layout and do not wish any changes to their workflow.

2. If an ability to choose fields that should be visible improve usability?
   Users in some cases need additional fields than those provided in current implementation to narrow their search. Providing an option to dynamically add more fields does improve the overall usability. This feature should be present irrespective of existing or current layout.

3. Can an option to select multiple values per field benefit the user in filtering?
   Interviews and tests have shown that users need a method to select multiple values for fields improve the usability and in-turn benefits users to better filter their data, This feature should be present irrespective of existing or current layout.

Overall, providing the aforementioned features does improve the user experience while using the search functionality.

## 5.3 Limitations

The research had its limitations both imposed by case company and otherwise imposed by time-restrictions. The research was conducted for one company and a very specific section.

Since the research was performed inside a company, there existed a few policies which restricted the methods used in the study. The author was not able to reach any actual clients of the software due to unavailability of channels to do so. The company had not previously hosted a researcher in the field of user-experience and this led to conundrum regarding the researcher meet or interview the clients directly. Furthermore, the company deals with financial industry and clients ranges from individual asset managers to established banks. The clients are mostly quite busy and seldom has time to participate is user-studies or interviews.

Other limitation of the platform is that it does not track user's search activity. The platform only stores the searches which are specifically saved as a bookmark for further use. The availability of daily-search usage would have provided a more in-depth insights regarding the searching patterns.

The author used method of usability testing for testing the designed prototype. The author could only gather three users to participate in the test due to both corporate restriction and time-induced restriction. Nevertheless, the biggest regret is not being able to do multiple iterations of designs and testing them with users multiple times.

The other limitation which author had to consider was the possibility of implementing the design. After working in the company for some-time, the author observed and learned the method of development of the Platform. Thus, the visual design and added features must be inline with company's development cycle and possible to code using the development tools already in use.

Finally, irrespective of the author's efforts, the design prototype is not the best object to conduct the usability testing with. The prototype does not emulate the real-world situation where information retrieval speed can be affected by factors like internet speed, server bandwidth, computer efficiency or the efficiency of the platform to display the data. To perform better testing, the design must be implemented inside the application and then tested with users.

## 5.4 Recommendations and Future Work

The primary recommendation to implement in future is to build a channel through which researchers or employees can reach the clients willing to participate in the research. Once, that is established the study must conducted again and the clients can participate in the interviews and usability testing.

Second, the company should implement tracking over client's live search-usage. To avoid privacy concerns, the platform can ignore the information entered by the client as long as the platform can save which fields were used to create the search query.

Third, since the top most frequently used fields are based on the data collected from all clients, it would change as clients perform more searches. The client's search-usage data must be collected and analysed at least bi-annually. If there is any change in frequent usage, then the interface must be updated to reflect that.

In addition to that, currently all clients were considered which defining the most frequently used search fields but that is an over-simplification as it ignores the different types of users and their use-cases. In order to better serve the customer, the company should differentiate the clients based on their roles and run separate analysis on each role. This should give a better understanding of how the platform is used by different markets/user-base and then new user-oriented interfaces can be created just for them.

As recommended by users during the testing, there are few features that could be considered as well. An option to change the recommended and default fields in every View in case the client some specific or different needs. Also a feature which could sort hundreds of fields in the "More fields" list by analysing the client's search patterns. The tracking of search-field usage will come handy in here.

This means more design iterations could be and should be performed on design and the new design should be tested with more users till the general consensus is positive.

Finally, the design needs to be coded and implemented into the platform. Once the changes have been made into the platform, further testing should be done to improve user experience.

# Chapter 6

# Conclusions

Information and data are building blocks of today's society. The methods to retrieve that information will always deemed important. In this study, a software solution called "FA Platform" was analysed for the case-company - FA Solutions Oy. The software is an Asset Management Solution and depends heavily on methods of searching and filtering to reliably retrieve requested information for the gigantic database. The software is used by asset managers, fund managers, private banks, family-offices and individuals.

The case company expected this study to identify the problems plaguing their current implementation of search user interface and suggest improvements for the same. The goal of the thesis was to improve user's experience while using the search functionality of software.

Interviews were conducted with users of the software to identify problems with the current interface. Client's search-usage history was collected and analysed for patterns. The findings were as follows:

- Search is extensively used
- Extended search is used more often than Quick Search
- Quick Search is slow and its function is unclear
- Extended search is difficult to comprehend
- Required an option to select multiple values in a field
- Option to add more fields when required.
- Option to reset all the fields to initial state.

As a consequence of this thesis, a design prototype of the search user interface was created and tested in trials with users against the existing

implementation. Through testing, the new design prototype was found to be easy to use and provided a nice overall user-experience. The proposed design improved over the current implementation in three key areas:

1. Redesigning the layout of search user interface where the feature of Extended and Quick Search should be merged to increase legibility and recognition of the available fields. Is it better to hide the filter-fields which are not used frequently to provide a cleaner interface. Each filter-field should have a field-label accompanying the input box for clarity.

2. Adding an option to dynamically add more filter-fields as per requirement

3. Providing an option to select multiple values for filter-fields that has more than one pre-defined options.

It is now imperative that the proposed design should be implemented in the software and tested with real-data and actual clients.

This study was conducted under some established limitations and guidelines, so there is quite much that can be improved by further research. The research does use a software solution made to manage assets or investments as the base platform for understanding problems related to search user-interface.

Nevertheless, the methods used in this study can be applied to applications or software in other fields or industry sectors, as the SUI in most applications offer quite similar functionality. The author believes and research shows that improving the graphical or visual aspect of a feature does play an important role is improving the user's experience. The case with the search feature is quite similar as an user may not enjoy or even like to use a fast or efficient search functionality if the user-interface is not designed with user's need in-mind.

Having a good search user interface design is an important factor while developing a database management tool as it will increase user's productivity, allowing them to find exactly what they want, how they want and when they want it. Even more so, a good design can also be very profitable for the company as it causes less distress to users. This leads to a satisfied customer who understands the software better and has less complaints overall. The design ideas as well as suggestions proposed in this thesis can be applied in mostly any solution which deals with data and require a method of retrieving it.

# References


Albaum, Gerald. 1997. "The Likert Scale Revisited." *Market Research Society. Journal.* 39, no. 2 (March): 1–21. ISSN: 0025-3618. doi:10.1177/147078539703900202. http://journals.sagepub.com/doi/10.1177/147078539703900202.

Baeza-Yates, R. (Ricardo), and Berthier Ribeiro-Neto. 1999. *Modern information retrieval.* 513. ACM Press. ISBN: 020139829X. https://dl.acm.org/citation.cfm?id=553876.

Berger, Arne. 2011. "Design Thinking Search User Interfaces." *Proceedings of the 1st European Workshop on Human-Computer Interaction with Information Retrieval (EuroHCIR'11):* 1–4. http://ceur-ws.org/Vol-763/posterB.pdf.

Borlund, Pia. 2003. "The IIR evaluation model." *Information Research. an International Electronic Journal* 8 (3). ISSN: 09598928. https://www.forskningsdatabasen.dk/en/catalog/2398381363.

Chin, John, Virginia Diehl, and Kent Norman. 1988. "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface." *ACM CHIi'.*

Cyr, Dianne, Milena Head, and Hector Larios. 2010. "Colour appeal in website design within and across cultures: A multi-method evaluation." *International Journal of Human-Computer Studies* 68, nos. 1-2 (January): 1–21. ISSN: 1071-5819. doi:10.1016/J.IJHCS.2009.08.005. https://www.sciencedirect.com/science/article/abs/pii/S1071581909001116?via%3Dihub.

Fisher, Karen E., Sanda Erdelez, and Lynne. McKechnie. 2005. *Theories of information behavior.* 431. Published for the American Society for Information Science / Technology by Information Today. ISBN: 157387230X.

Hassenzahl, Marc, and Noam Tractinsky. 2006. "User experience - a research agenda." *Behaviour & Information Technology* 25, no. 2 (March): 91–97. ISSN: 0144-929X. doi:10.1080/01449290500330331. http://www.tandfonline.com/doi/abs/10.1080/01449290500330331.

Hearst, Marti A. 2011. "User Interfaces for Search." *Modern Information Retrieval: The Concepts and Technology behind Search Engines:* 21–56.

Holleran, Patrick A. 1991. "A methodological note on pitfalls in usability testing." *Behaviour & Information Technology* 10, no. 5 (September): 345–357. ISSN: 0144-929X. doi:10.1080/01449299108924295. http://www.tandfonline.com/doi/abs/10.1080/01449299108924295.

Kuhlthau, Carol C. 1991. "Inside the search process: Information seeking from the user's perspective." *J. Am. Soc. Inf. Sci.* 42:361–371. doi:10.1002/(SICI)1097-4571(199106)42:5<361::AID-ASI6>3.0.CO;2-{\#}. https://onlinelibrary.wiley.com/action/showCitFormats?doi=10.1002%2F%28SICI%291097-4571%28199106%2942%3A5%3C361%3A%3AAID-ASI6%3E3.0.CO%3B2-%23.

Liaw, Shu-Sheng, and Hsiu-Mei Huang. 2003. "An investigation of user attitudes toward search engines as an information retrieval tool." *Computers in Human Behavior* 19, no. 6 (November): 751–765. ISSN: 0747-5632. doi:10.1016/S0747-5632(03)00009-8. https://www.sciencedirect.com/science/article/pii/S0747563203000098?via%3Dihub.

Marchionini, Gary, and Gary. 2006. "Exploratory search." *Communications of the ACM* 49, no. 4 (April): 41. ISSN: 00010782. doi:10.1145/1121949.1121979. http://portal.acm.org/citation.cfm?doid=1121949.1121979.

Moore, Melody, and Spencer Rugaber. 1993. "Issues in user interface migration." In *Proceedings of the Third Software Engineering Research Forum.* ftp://coffeetalk.cc.gatech.edu/pub/groups/reverse/repository/serf.ps.

MySQL. 2019. *MySQL :: MySQL 8.0 Reference Manual :: 13.2.10 SELECT Syntax.* https://dev.mysql.com/doc/refman/8.0/en/select.html.

Nielsen, Jakob. 1993. *Usability Engineering.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 0125184050.

———. 1994. "Usability Inspection Methods." Chap. Heuristic, edited by Jakob Nielsen and Robert L Mack, 25–62. New York, NY, USA: John Wiley & Sons, Inc. ISBN: 0-471-01877-5. http://dl.acm.org/citation.cfm?id=189200.189209.

Nielsen, Jakob, and Thomak K. Landauer. 1993. "A Mathematical Model of the Finding of Usability Problems." In *CHI '93 Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems,* 206–213. ISBN: 0-89791-575-5. doi:10.1145/169059.169166.

Pernice, Kara. 2018. *User Interviews: How, When, and Why to Conduct Them.* https://www.nngroup.com/articles/user-interviews/.

Russell-rose, Tony, and Tyler Tate. 2013. "Designing the Search Experience: The Information Architecture of Discovery." *Designing the Search Experience: The Information Architecture of Discovery,* no. January 2013: 23–45.

Saracevic, Tefko. 1996. "Relevance reconsidered." In *Proceedings of the second conference on conceptions of library and information science (CoLIS 2),* 201–218. ACM New York.

Schamber, Linda, and Gary Marchionini. 1996. "Information Seeking in Electronic Environments." *Journal of Education for Library and Information Science* 37 (1): 81. ISSN: 07485786. doi:10.2307/40324289. https://www.jstor.org/stable/10.2307/40324289?origin=crossref.

Taylor-Powell, E., and M. Renner. 2003. "Analyzing qualitative data." *University of Wisconsin-Extension* 76 (July): 459–463. doi:10.1259/bjr/16316438.

Vermeeren, Arnold P O S, Effie Lai-Chong Law, Virpi Roto, Marianna Obrist, Jettie Hoonhout, and Kaisa Väänänen-Vainio-Mattila. 2010. "User Experience Evaluation Methods: Current State and Development Needs." In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries,* 521–530. NordiCHI '10. New York, NY, USA: ACM. ISBN: 978-1-60558-934-3. doi:10.1145/1868914.1868973. http://doi.acm.org/10.1145/1868914.1868973.

Virzi, Robert A. 1992. "Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough?" *Human Factors: The Journal of the Human Factors and Ergonomics Society* 34, no. 4 (August): 457–468. ISSN: 0018-7208. doi:10.1177/001872089203400407. http://journals.sagepub.com/doi/10.1177/001872089203400407.

White, Ryen W., and Resa A. Roth. 2009. "Exploratory Search: Beyond the Query-Response Paradigm." *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1, no. 1 (January): 1–98. ISSN: 1947-945X. doi:10.2200/S00174ED1V01Y200901ICR003. http://www.morganclaypool.com/doi/abs/10.2200/S00174ED1V01Y200901ICR003.

Wilson, Max L. 2011. *Search User Interface Design,* vol. 3. Morgan & Claypool Publishers. ISBN: 9781608456895. doi:10.2200/s00371ed1v01y201111icr020.

Wilson, Max L., Paul André, and M. C. Schraefel. 2008. "Backward highlighting." In *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08,* 235. New York, New York, USA: ACM Press. ISBN: 9781595939753. doi:10.1145/1449715.1449754. http://portal.acm.org/citation.cfm?doid=1449715.1449754.

Wroblewski, Luke. 2007. *Primary & Secondary Actions in Web Forms.* https://www.lukew.com/ff/entry.asp?571.

Zabed Ahmed, S.M. 2008. "A comparison of usability techniques for evaluating information retrieval system interfaces." Edited by Alison Scammell. *Performance Measurement and Metrics* (Abingdon, UK) 9, no. 1 (March): 48–58. ISSN: 1467-8047. doi:10.1108/14678040810869422. https://www.taylorfrancis.com/books/9780203403914%20http://www.emeraldinsight.com/doi/10.1108/14678040810869422.

Zheng, Xianjun Sam, Ishani Chakraborty, James Jeng-Weei Lin, and Robert Rauschenberger. 2009. "Correlating low-level image statistics with users - rapid aesthetic and affective judgments of web pages." In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09,* 1. New York, New York, USA: ACM Press. ISBN: 9781605582467. doi:10.1145/1518701.1518703. http://dl.acm.org/citation.cfm?doid=1518701.1518703.

# Appendix A

# Python Script

```
### query_analysis.ipynb - Siddhant Gupta - 04.04.2019

## Import Required Libraries
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from collections import Counter
import glob

## Define Variables
VIEWS = ['transactionsView', 'tradeOrdersView', '
    securitiesView', 'reportsView', 'positionsView', '
    portfoliosView', 'marketsView', 'crmView', 'contactsView',
     'bookKeepingView', 'analytics2View'] # Views to Analyse
FILES = "query_data/*.xls" # Files to Analyse
N_FIELDS = 4 # Number of Top Fields to recommend
GRAPH = True # Show Graphs
TABULAR = True # Show Table

## Read Files and Generate Data
DATA = pd.DataFrame()
for f in glob.glob(FILES):
    df = pd.read_excel(f)
    DATA = DATA.append(df,ignore_index=True, sort=True)

## Auxiliary functions
def graph (X, Y, width, height, title):
    plt.figure(num=None, figsize=(width, height), dpi=80,
    facecolor='w', edgecolor='k')
    plt.plot(X,Y)
    plt.xticks(rotation=45)
    plt.title(title)
    plt.show()

```

```
## Analyse Data
freqfields = []
for view in VIEWS:
    data = DATA[DATA['base_view'].str.contains(view)]
    data = data['init_params'].str.split(',', expand=True).rename(columns=lambda x: f"string_{x+1}")
    data = data['string_1'].str.split(';', expand=True).rename(columns=lambda x: f"field{x+1}")
    flist = []
    for i in range(0, data.shape[1]):
        flist.extend(data[f'field{i+1}'].str.split('=', 1, expand=True)[0])
    fields = list(Counter(flist).keys())
    freq = list(Counter(flist).values())
    d = {'fields': fields, 'freq': freq}
    frame = pd.DataFrame(d)
    frame = frame.dropna()
    frame = frame[frame.fields != '']
    frame = frame[frame.fields != 'columnsFilter']
    frame = frame.sort_values(by=['freq'], ascending=False)
    frame = frame.reset_index()
    frame = frame.drop(columns=['index'])
    frame.index += 1
    if frame.shape[0] > 0:
        print('-\033[1m', view , '\033[0m-'*(60 - len(view) - 3))
        freqfields.append({view: ', '.join(frame[:N_FIELDS].fields)})
        if TABULAR:
            display(frame)
        if GRAPH:
            graph(frame.fields, frame.freq,8,2,view)
    else:
        print('\n\tNot enough data!\n')
print('-' * 60)

## Printing Results
print("\033[1m{:<20} {:<15}\033[0m".format('View','Most Frequently Used Search Fields'))
print('-' * 60)
for index in range(len(freqfields)):
    for key in freqfields[index]:
        print("\033[1m{:<20}\033[0m {:<15}".format(key, freqfields[index][key]))
```

# Appendix B

# Analysis Report - 05.04.2019

The script `query_analysis.IPYNB` is executed with following parameters:

Listing B.1: query_analysis.ipynb - Variables

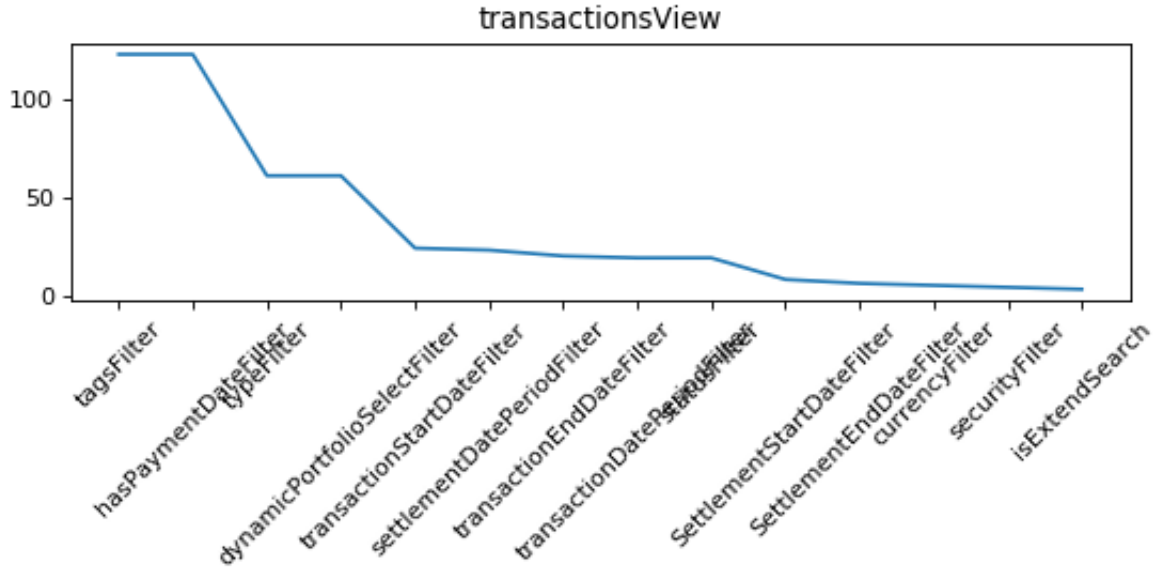
```
## Define Variables
# Views to Analyse
VIEWS = ['transactionsView', 'tradeOrdersView', '
    securitiesView', 'reportsView', 'positionsView', '
    portfoliosView', 'marketsView', 'crmView', 'contactsView',
     'bookKeepingView', 'analytics2View']
# Files to Analyse
FILES = "query_data/*.xls"
# Number of Top Fields to recommend
N_FIELDS = 4
# Data Output
GRAPH = True # Show Graphs
TABULAR = True # Show Table
```

The analysis report is generated for 11 Views and all data-files (42) in `query_data` directory. The data is visualised in both TABULAR and GRAPHICAL format. In the result, 4 most frequently used filter fields are listed across each View name as a recommendation.

# Data Visualisation

**- transactionsView ----------**

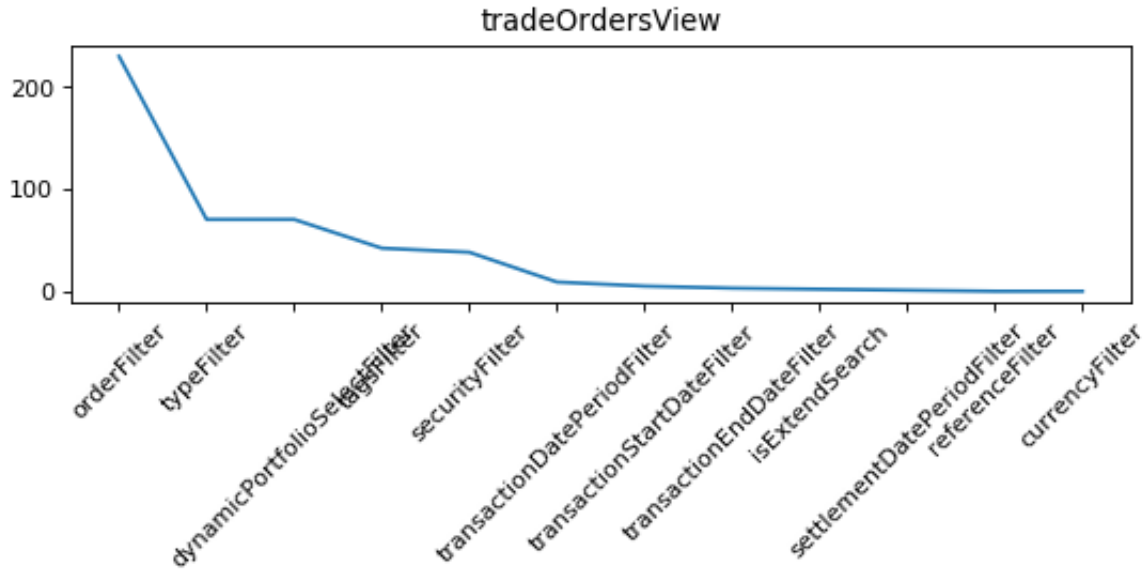
| | fields | freq |
|---|---|---|
| **1** | tagsFilter | 123 |
| **2** | hasPaymentDateFilter | 123 |
| **3** | typeFilter | 61 |
| **4** | dynamicPortfolioSelectFilter | 61 |
| **5** | transactionStartDateFilter | 24 |
| **6** | settlementDatePeriodFilter | 23 |
| **7** | transactionEndDateFilter | 20 |
| **8** | transactionDatePeriodFilter | 19 |
| **9** | statusFilter | 19 |
| **10** | SettlementStartDateFilter | 8 |
| **11** | SettlementEndDateFilter | 6 |
| **12** | currencyFilter | 5 |
| **13** | securityFilter | 4 |
| **14** | isExtendSearch | 3 |




**- tradeOrdersView ----------**

| | fields | freq |
|---|---|---|
| **1** | orderFilter | 230 |

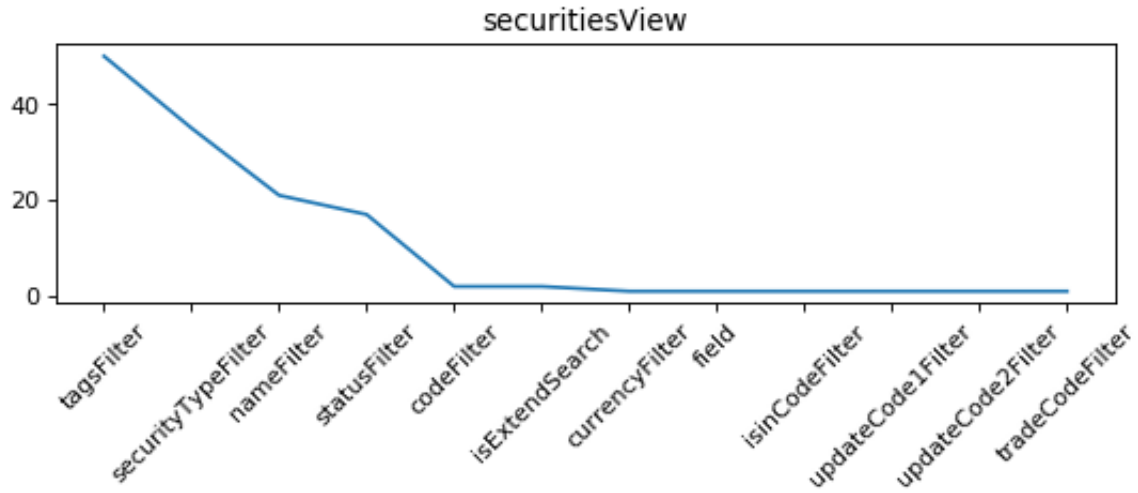
| | | |
|---|---|---|
| **2** | typeFilter | 71 |
| **3** | dynamicPortfolioSelectFilter | 71 |
| **4** | tagsFilter | 43 |
| **5** | securityFilter | 39 |
| **6** | transactionDatePeriodFilter | 10 |
| **7** | transactionStartDateFilter | 6 |
| **8** | transactionEndDateFilter | 4 |
| **9** | isExtendSearch | 3 |
| **10** | settlementDatePeriodFilter | 2 |
| **11** | referenceFilter | 1 |
| **12** | currencyFilter | 1 |




**- securitiesView ----------**

| | fields | freq |
|---|---|---|
| **1** | tagsFilter | 50 |
| **2** | securityTypeFilter | 35 |
| **3** | nameFilter | 21 |
| **4** | statusFilter | 17 |
| **5** | codeFilter | 2 |
| **6** | isExtendSearch | 2 |
| **7** | currencyFilter | 1 |
| **8** | field | 1 |
| **9** | isinCodeFilter | 1 |

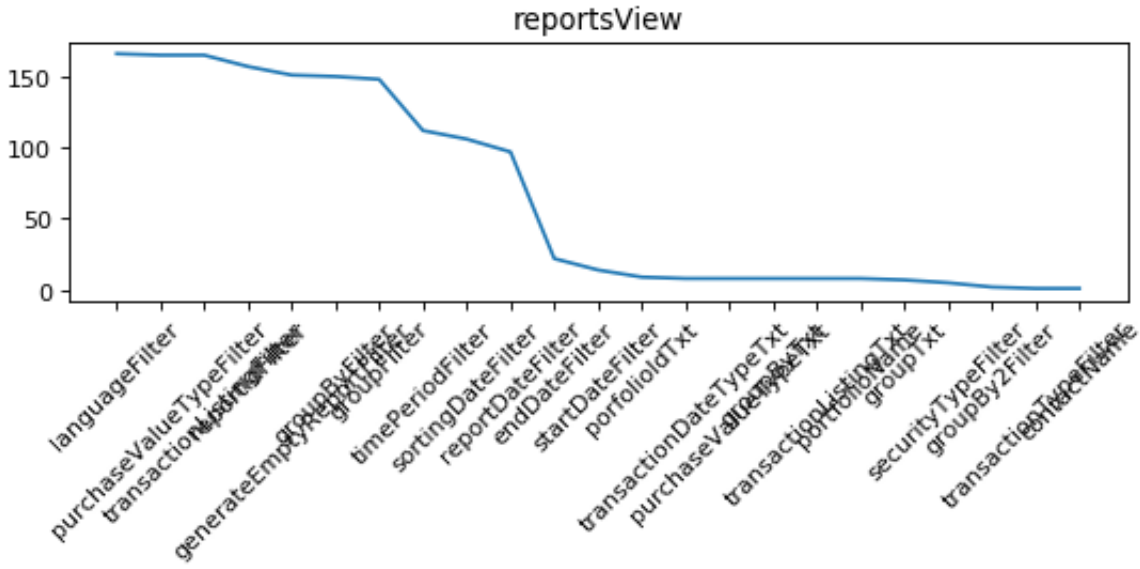
| | | |
|---|---|---|
| **10** | updateCode1Filter | 1 |
| **11** | updateCode2Filter | 1 |
| **12** | tradeCodeFilter | 1 |




**- reportsView ----------**

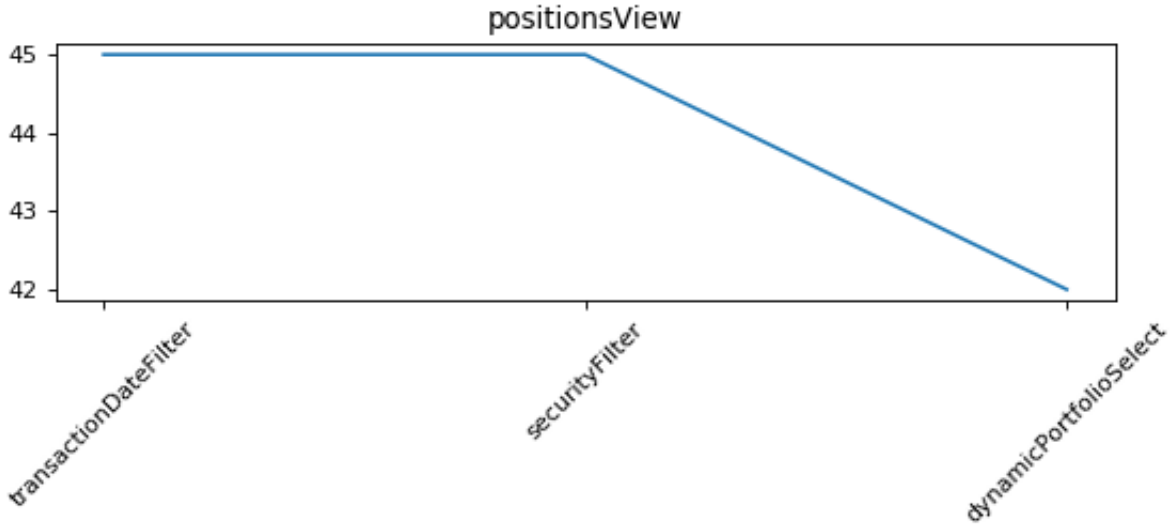
| | fields | freq |
|---|---|---|
| **1** | languageFilter | 166 |
| **2** | purchaseValueTypeFilter | 165 |
| **3** | transactionListingFilter | 165 |
| **4** | reportIdFilter | 157 |
| **5** | generateEmptyReportFilter | 151 |
| **6** | groupByFilter | 150 |
| **7** | groupFilter | 148 |
| **8** | timePeriodFilter | 112 |
| **9** | sortingDateFilter | 106 |
| **10** | reportDateFilter | 97 |
| **11** | endDateFilter | 22 |
| **12** | startDateFilter | 14 |
| **13** | porfolioIdTxt | 9 |
| **14** | transactionDateTypeTxt | 8 |
| **15** | purchaseValueTypeTxt | 8 |
| **16** | groupByTxt | 8 |
| **17** | transactionListingTxt | 8 |
| **18** | portfolioName | 8 |
| **19** | groupTxt | 7 |

| | | |
|---|---|---|
| **20** | securityTypeFilter | 5 |
| **21** | groupBy2Filter | 2 |
| **22** | transactionTypeFilter | 1 |
| **23** | contactName | 1 |




**- positionsView ----------**

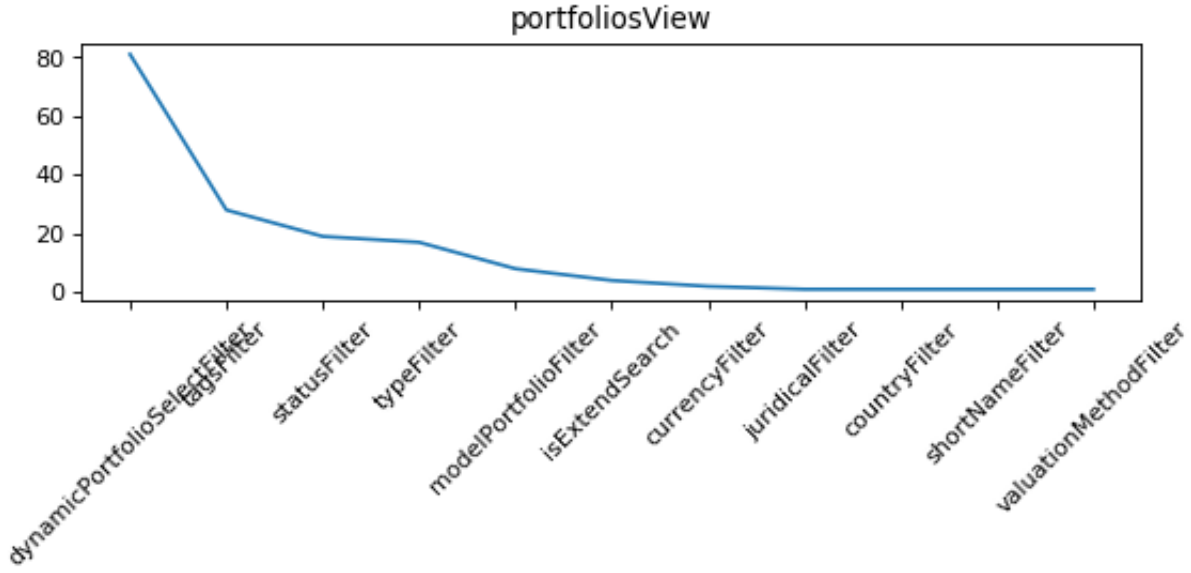
| | fields | freq |
|---|---|---|
| **1** | transactionDateFilter | 45 |
| **2** | securityFilter | 45 |
| **3** | dynamicPortfolioSelect | 42 |




**- portfoliosView ----------**

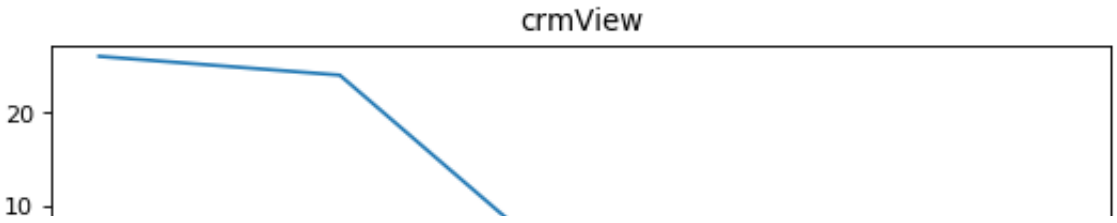
| | fields | freq |
|---|---|---|

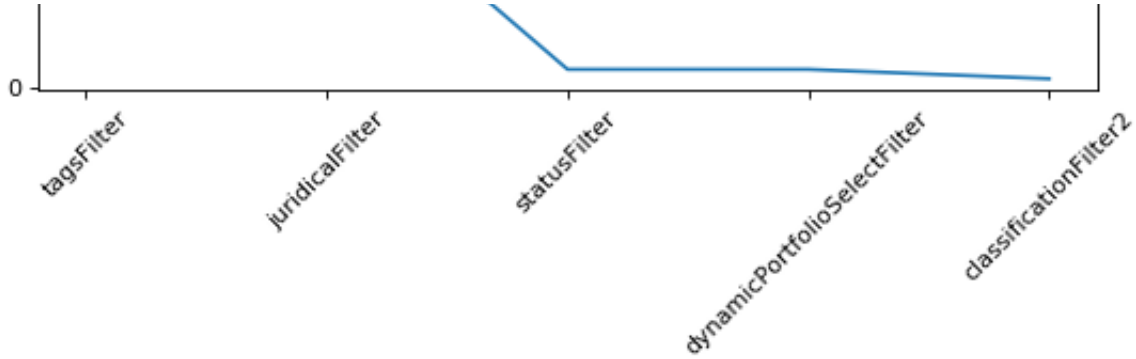
| | | |
|---|---|---|
| **1** | dynamicPortfolioSelectFilter | 81 |
| **2** | tagsFilter | 28 |
| **3** | statusFilter | 19 |
| **4** | typeFilter | 17 |
| **5** | modelPortfolioFilter | 8 |
| **6** | isExtendSearch | 4 |
| **7** | currencyFilter | 2 |
| **8** | juridicalFilter | 1 |
| **9** | countryFilter | 1 |
| **10** | shortNameFilter | 1 |
| **11** | valuationMethodFilter | 1 |




**- crmView ----------**

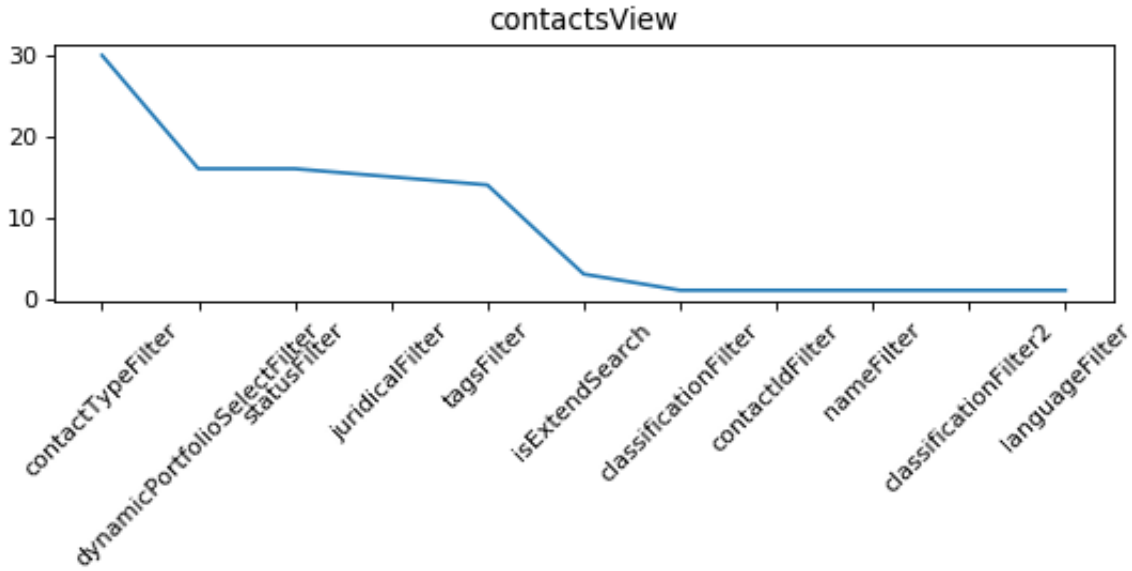
| | fields | freq |
|---|---|---|
| **1** | tagsFilter | 26 |
| **2** | juridicalFilter | 24 |
| **3** | statusFilter | 2 |
| **4** | dynamicPortfolioSelectFilter | 2 |
| **5** | classificationFilter2 | 1 |





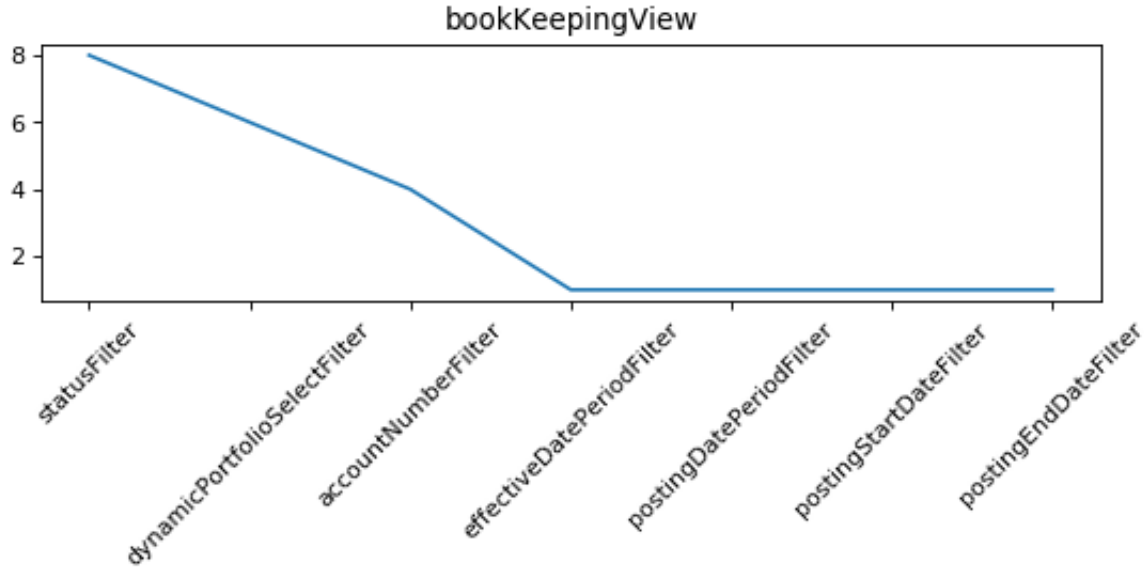

**- contactsView ----------**

| | fields | freq |
|---|---|---|
| **1** | contactTypeFilter | 30 |
| **2** | dynamicPortfolioSelectFilter | 16 |
| **3** | statusFilter | 16 |
| **4** | juridicalFilter | 15 |
| **5** | tagsFilter | 14 |
| **6** | isExtendSearch | 3 |
| **7** | classificationFilter | 1 |
| **8** | contactIdFilter | 1 |
| **9** | nameFilter | 1 |
| **10** | classificationFilter2 | 1 |
| **11** | languageFilter | 1 |



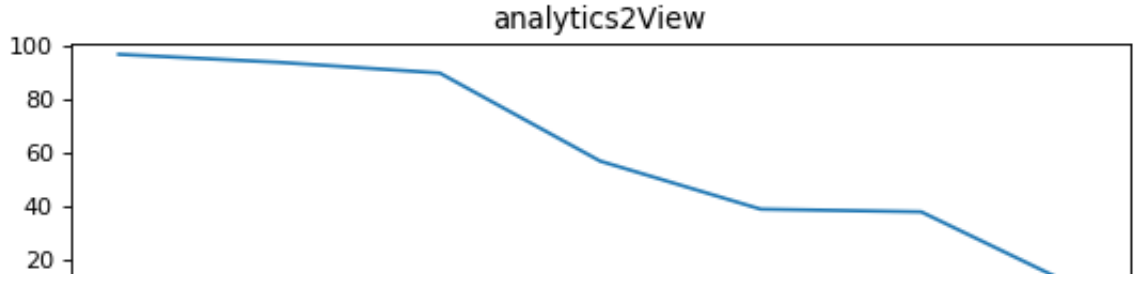

**- bookKeepingView ----------**

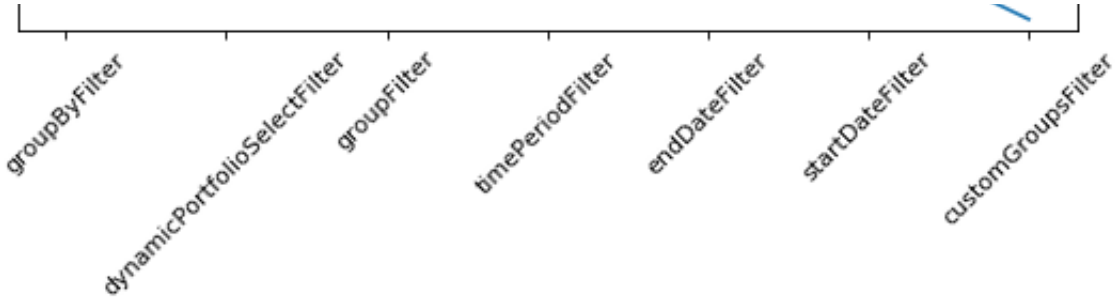
| | fields | freq |
|---|---|---|
| **1** | statusFilter | 8 |
| **2** | dynamicPortfolioSelectFilter | 6 |
| **3** | accountNumberFilter | 4 |
| **4** | effectiveDatePeriodFilter | 1 |
| **5** | postingDatePeriodFilter | 1 |
| **6** | postingStartDateFilter | 1 |
| **7** | postingEndDateFilter | 1 |




**- analytics2View ----------**

| | fields | freq |
|---|---|---|
| **1** | groupByFilter | 97 |
| **2** | dynamicPortfolioSelectFilter | 94 |
| **3** | groupFilter | 90 |
| **4** | timePeriodFilter | 57 |
| **5** | endDateFilter | 39 |
| **6** | startDateFilter | 38 |
| **7** | customGroupsFilter | 9 |






---

# Results

The following are the most frequently used search fields in each of the view

| View | Most Frequently Used Search Fields |
|---|---|
| **transactionsView** | tagsFilter, hasPaymentDateFilter, typeFilter, dynamicPortfolioSelectFilter |
| **tradeOrdersView** | orderFilter, typeFilter, dynamicPortfolioSelectFilter, tagsFilter |
| **securitiesView** | tagsFilter, securityTypeFilter, nameFilter, statusFilter |
| **reportsView** | languageFilter, purchaseValueTypeFilter, transactionListingFilter, reportIdFilter |
| **positionsView** | transactionDateFilter, securityFilter, dynamicPortfolioSelect |
| **portfoliosView** | dynamicPortfolioSelectFilter, tagsFilter, statusFilter, typeFilter |
| **crmView** | tagsFilter, juridicalFilter, statusFilter, dynamicPortfolioSelectFilter |
| **contactsView** | contactTypeFilter, dynamicPortfolioSelectFilter, statusFilter, juridicalFilter |
| **bookKeepingView** | statusFilter, dynamicPortfolioSelectFilter, accountNumberFilter, effectiveDatePeriodFilter |
| **analytics2View** | groupByFilter, dynamicPortfolioSelectFilter, groupFilter, timePeriodFilter |

# Appendix C

# Proposed SUI Design

On the basis of previous analysis, the following are the Search User Interface implementation for eight views present in the *FA Platform*. The views are in alphabetical order.

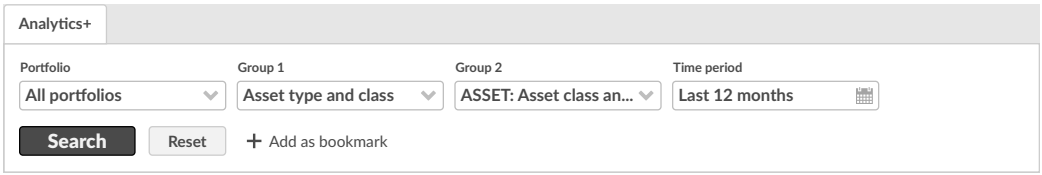



Figure C.1: Proposed SUI for Analytics+ View

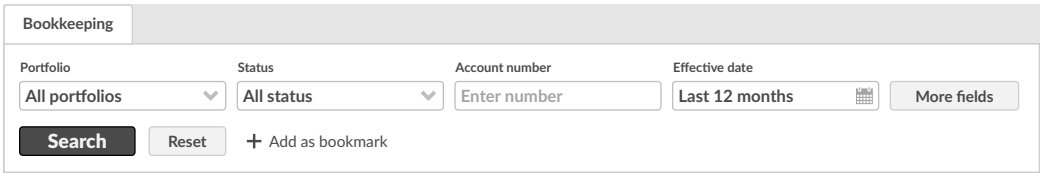



Figure C.2: Proposed SUI for Bookkeeping View

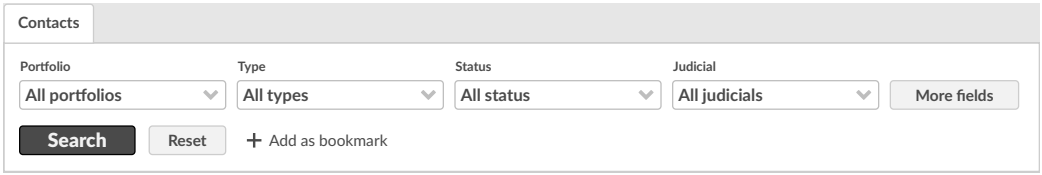



Figure C.3: Proposed SUI for Contacts View

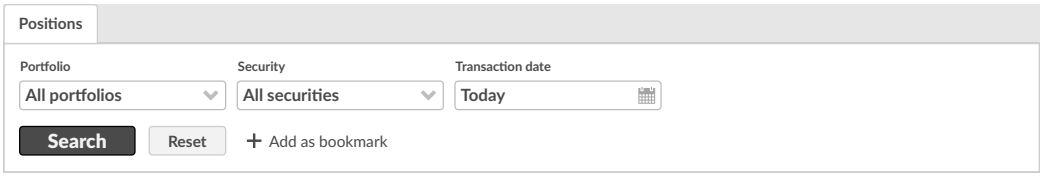



Figure C.4: Proposed SUI for Positions View

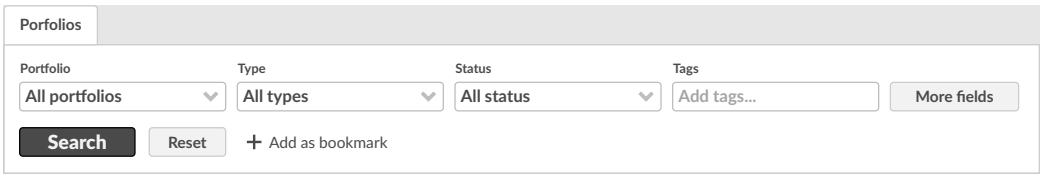



Figure C.5: Proposed SUI for Portfolio View

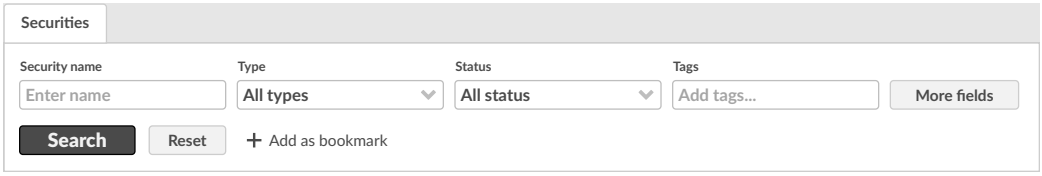



Figure C.6: Proposed SUI for Securities View

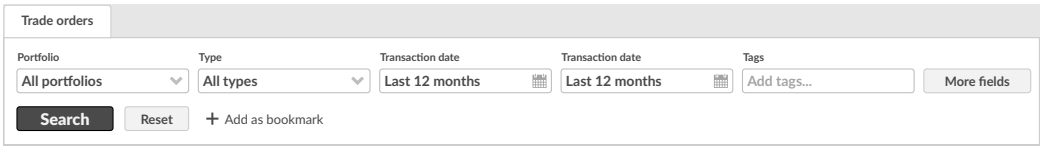



Figure C.7: Proposed SUI for Trade orders View


Figure C.8: Proposed SUI for Transactions View

# Appendix D

# User-Satisfaction Questionnaire

The following survey ins intended to understand and record your satisfaction after performing the usability tasks.

Date: _____
Name: _____
Position: _____
Company: FA Solutions Oy
Place: Kamppi, Helsinki

Kindly answer the following questions with response on the scale of 1-5 regarding the tasks you just performed.

5 being very easy and 1 being very hard.

| **Question** | Existing Application (1-5) | New Design (1-5) |
|---|---|---|
| How easy was it to find portfolios with type equal to 'Asset Management'? | | |
| How easy was it to find portfolios with tax-country equal to 'Finland'? | | |
| How easy was it to find transactions with period of 'Current month'? | | |
| How easy was it to find transactions with currency equal to 'EUR'? | | |

Kindly answer the following questions as openly as possible:

- Were the tasks assigned relatable to the daily-usage?

- Was it generally simple to perform the tasks in the new search view?

- How was your experience when using the new search design?

- Any other suggestions that you would like to add :

For further questions, recommendations and issues, contact the researcher:

Siddhant Gupta
UX Designer, FA Solutions Oy
siddhant@fasolutions.com