Designing a User Interface for Improving the Usability of a Statistical Disclosure Control Tool

Rawat, Anshika; Janssen, Marijn; Bargh, Mortaza S.; Choenni, Sunil

DOI
10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00212

Publication date
2021

Document Version
Final published version

Published in
19th IEEE International Symposium on Parallel and Distributed Processing with Applications, 11th IEEE International Conference on Big Data and Cloud Computing, 14th IEEE International Conference on Social Computing and Networking and 11th IEEE International Conference on Sustainable Computing and Communications, ISPA/BDCloud/SocialCom/SustainCom 2021

Citation (APA)
Rawat, A., Janssen, M., Bargh, M. S., & Choenni, S. (2021). Designing a User Interface for Improving the Usability of a Statistical Disclosure Control Tool. In *19th IEEE International Symposium on Parallel and Distributed Processing with Applications, 11th IEEE International Conference on Big Data and Cloud Computing, 14th IEEE International Conference on Social Computing and Networking and 11th IEEE International Conference on Sustainable Computing and Communications, ISPA/BDCloud/SocialCom/SustainCom 2021* (pp. 1581-1591). (19th IEEE International Symposium on Parallel and Distributed Processing with Applications, 11th IEEE International Conference on Big Data and Cloud Computing, 14th IEEE International Conference on Social Computing and Networking and 11th IEEE International Conference on Sustainable Computing and Communications, ISPA/BDCloud/SocialCom/SustainCom 2021). IEEE. https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00212

2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

Designing a User Interface for Improving the Usability of a Statistical Disclosure Control Tool

Anshika Rawat and Marijn Janssen
Faculty of Technology, Policy and Management
Delft University of Technology
Delft, The Netherlands
e-mails: anshika.rawat2020@gmail.com and
m.f.w.h.a.janssen@tudelft.nl

Mortaza S. Bargh and Sunil Choenni
Research and Documentation Centre
Ministry of Justice and Security
The Hague, The Netherlands
e-mails: m.shoae.bargh@wodc.nl and
r.choenni@wodc.nl

Mortaza S. Bargh and Sunil Choenni
Research Centre Creating 010
Rotterdam University of Applied Sciences
Rotterdam, The Netherlands
e-mails: m.shoae.bargh@hr.nl and r.choenni@hr.nl

***Abstract*— Data sets often contain personal data that are not needed for analysis. Statistical Disclosure Control (SDC) is a personal data minimizing technology to reduce personal data in a data set while maintaining the utility of the data set at an acceptable level. SDC tools usage lags due to the complexity encountered by users. The aim of the research in this paper is to reduce the complexity of SDC tools via improving their ease of use. For this study information system adoption literature is reviewed, some usability challenges are identified, and a User Interface (UI) prototype is developed. From the literature, some design principles are derived related to minimal memory, feature self-descriptiveness, user guidance, navigability, minimal action, and UI familiarity. Based on these principles, a prototype is developed and evaluated. The findings suggest that the designed UI overcomes the identified UI-related issues of an SDC tool called ARX but show no significance in reducing the complexity associated with the SDC-related complexity. The findings suggest that while reducing the complexity of the UI might be a good approach to address the problem of adopting SDC technology within organizations, there is a need for complementary approaches to increase the usability and adoption of an SDC tool.**

Keywords- design principles; personal data minimization; statistical disclosure control; technology adoption; user interface

I. INTRODUCTION *(HEADING 1)*

Collected data sets often have more personal data than needed for data usage purposes. This discrepancy stems from the way that the data are typically collected and used. Sometimes a data set is collected for a purpose (like for provisioning a service), but it is used for another purpose (like for statistical analysis or scientific research). For example, in the medical domain patient data are collected to document medical treatments, while also being reused for medical research purposes. In the justice domain, another example, offender data are collected to try, sentence and treat offenders while they are reused for criminology research. Other times when even statistical analysis and/or scientific research are the primary purpose of data collection, the collected data may contain too much personal data due to, for example, an inappropriate research design.

According to privacy laws and regulations, like the General Data Protection Regulation (GDPR) [1], it is necessary to minimize the amount of personal data in a data set to only the data that are required and allowed for a chosen (legitimate) data usage. Not adjusting the amount of personal data to the data usage is a privacy breach and may have an adverse impact on individuals and society [2, 3]. It may also inflict reputation damages upon organizations responsible for privacy breaches and may bring lawsuits and financial fines against these organizations. Therefore, it is crucial that (governmental) organizations in their data sharing and opening processes exploit the techniques that hide sensitive information of citizens and individuals [4].

Statistical Disclosure Control (SDC) is an important personal data minimizing technology that aims at reducing personal information in a data set as much as possible and/or necessary, while maintaining the utility of the data set for a legitimate purpose in mind [5]. To this end, many sophisticated SDC tools have been developed. Using an SDC tool, an expert applies SDC models, methods and parameters to a data set to protect privacy-sensitive information, while retaining the information needed for the purpose of data analysis. Applying SDC technology, however, has not become a mainstream practice in most data-intensive organizations. This lack of organizational embedding stems from the complexity of applying SDC into practice because it is context-dependent and multi-disciplinary, imposes liability and accountability burdens, and requires having an advanced theoretical SDC background. Likewise, existing SDC tools are complex and there is a need for highly trained SDC experts to employ SDC technology within organizations appropriately.

978-1-6654-3574-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00212

There is a lack of enough SDC expertise in organizations. Therefore, a promising approach for embedding SDC technology within organizations seems to reduce the complexity of SDC technology and tools so that non-SDC experts (e.g., ordinary data stewards) can learn and use SDC tools easily within these organizations. In this contribution, we aim at reducing the complexity of SDC tools via improving their ease of use (i.e., via improving effort expectancy) to facilitate the use and adoption of SDC technology within (public) organizations.

To this end, our contributions are manyfold, namely: Identifying the usability issues of the User Interface (UI) of a typical SDC tool called ARX, eliciting the requirements to overcome these usability issues, translating these requirements into a UI design and prototype, and evaluating the resulting UI design by end-users. Moreover, we explain why personal data minimization is necessary, mention which technologies can be used for personal data minimization, and motivate the importance of reducing the complexity of (the UI of) SDC tools based on technology adaption theories and guidelines.

For conducting this study, as well as for developing our proof-of-concept prototype, we have used various research methods such as literature study, a mix of surveys and interviews to collect data from novice ARX users, proof-of-concept prototyping, and quantitative evaluation of the prototype. Our results can enhance the awareness about and guide devising the roadmap for embedding the SDC technology within data-intensive organizations. To the best of our knowledge, the work presented here is unique within the context of applying innovative personal data minimization technologies within organizations.

The structure of the paper is as follows. In Section II we present some background about data minimization, SDC technology, and the technology adoption theories to motivate the work. Subsequently, in Section III we explain the design principles behind our proposed UI for an SDC tool. In Section IV we present our proposed UI design and implementation. In Section V we discuss the results of our evaluation of the realized proof-of-concept prototype. Finally, we draw some conclusions and present a few directions for future research in Section VI.

II. BACKGROUND

A. Personal Data Minimization

Personal data minimization is concerned with limiting the amount of personal data processing to the data usage purpose. Realizing personal data minimization is necessary to satisfy legal requirements and establish trust among involved stakeholders, particularly data subjects when collecting and sharing privacy-sensitive data sets. Limiting personal data processing is emphasized in, for example, the GDPR [1] and – in relation to the operational affairs in the Justice domain – the EU Law Enforcement Directive (LED) [6]. The purpose limitation, data minimization, and data accuracy principles are the relevant principles of the GDPR and the LED, see Article 5(1-b, c, d) of GDPR and Article 4(1-b, c, d) of LED. Applying the data minimization principle appropriately is a way to nurture the trust of data subjects and citizens in the (public) organizations that hold and process their personal data. Establishing this trust, in turn, encourages data subjects to share more information for research and statistical purposes, and thereby improves the quality of the data-driven applications that often rely on high-quality data.

The necessity of applying personal data minimization is also mentioned by the experts we interviewed. We conducted four interviews with two data minimization experts (who actively use data minimization in their daily practice), one legal expert (who is knowledgeable about data minimization principles), and an expert on managing data-driven research, see [7] for more information about these interviews. All the interviewees acknowledged the necessity of minimizing personal data to the amount needed for data usage purposes in mind. Furthermore, we can trace the necessity of data minimization to similar principles in other common practices and fields. For example, it resembles the need-to-know principle in the military and information security domains [8].

B. SDC Technology

The important categories of minimizing personal data technologies are de-identification, privacy-sensitive information leakage prevention, and SDC. De-identification techniques remove, suppress, or pseudonymize directly identifying information items (like names and social security numbers) in a data set. As such, they deal with the intrinsic disclosure risks in the data set and are used in data processing applications where the identity of data subjects is not needed (like for scientific research and statistical analyses). Note that in some countries (like those in North America) the term de-identification is used differently than the way is used here.

The techniques used for preventing privacy-sensitive information leakage aim at limiting the amount of the probabilistic belief change (or so-called information gain) about data subjects after accessing a released data set. Inferring such privacy sensitive information may occur, for example, (a) when some attributes in the released/protected data set leak information about a privacy-sensitive attribute that is removed intentionally from the original data set (for example, the attribute residence postal code may reveal information about the attribute income, in case that the former is maintained and the latter is removed) or (b) when some attributes in the transformed data set leak information about a privacy-sensitive attribute that was not in the original data set (for example, some symptom attributes that are present in the released data set may indicate the type of the disease that a patient has).

SDC techniques extend de-identification techniques via removing or minimizing indirectly identifying information in a data set. Combinations of indirectly identifying information (like the combination of the values of birthdate, postal code and gender attributes) can be used to uniquely identify a large number of individuals in a population based on auxiliary data sources available to intruders [9, 10, 11]. These auxiliary data sources that are, in addition to the minimized data set, available to intruders are called background knowledge in literature. Depending on the data usage purpose, SDC techniques can be used to adjust the amount of indirectly identifying information about individuals in the data set to a desired or allowed level. To this end, SDC tools can provide insights into and mechanisms for transforming raw data sets, assessing the utility of the original and the transformed data sets, estimating the data disclosure risks of the original and the transformed data sets, and making trade-offs between data utility aspects and data disclosure risks.

In this paper, we focus on SDC techniques and their embedding in organizations because they are supported and provisioned by more mature toolsets relative to privacy-sensitive information leakage prevention techniques, and because they are more complex and challenging to be deployed within organizations relative to de-identification techniques.

C. Complexity of SDC Adoption

Applying SDC technology into practice can be a complex and demanding task for non-SDC-experts (i.e., those without prior experience and affinity with this technology). Three main reasons behind this complexity are: (a) lack of adequate coverage of theoretical and practical foundations of SDC technology within the current education curricula of data science experts, (b) the context-dependency of applying SDC technology, which demands the involvement of competent users to estimate and deal with the peculiarities of the data environment into which the data are released (e.g., estimate the background knowledge available for intruders and the motivations of intruders), and (c) the need for and necessity of adopting a multi-disciplinary approach where SDC experts confer with non-SDC experts for collaborative decision-making about all aspects of data minimization process (like how to estimate the disclosure risks and deal with the residual disclosure risks after applying SDC technology). This conferring with multi-disciplinary experts is needed to adequately represent and cover the utility, legal, ethical, cybersecurity and policy related aspects of personal data minimization.

D. On Facilitating SDC Adoption

For most organizations, including government agencies and ministries, SDC technology is a new and innovative technology. Few organizations have deployed SDC technology systematically and have integrated it into the fabric of their organizations. Examples of these organizations are national statistics bureaus and medical research centers. Therefore, many organizations should still plan and take appropriate measures to embed the SDC technology successfully.

Technology adoption theories, which have been developed for identifying and modeling those factors that contribute to the successful adoption and use of innovative technologies, can be used to partially inform us about how to effectively embed SDC technology within organizations. We use the extended UTAUT model of [12], see Fig. 1, to list several theory-informed guidelines for the successful adoption of the SDC technology within (public) organizations. We choose this model because it includes the technology, context and individual-related factors, as it

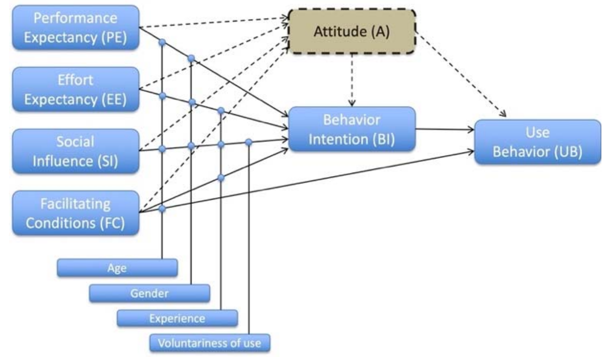

Figure 1. The UTAUT model and its extension, indicated with the dashed connections and box by [12].

integrates previous studies, showing “excellent fit statistics” for 162 prior studies on IS/ IT acceptance and use [12]. The constructs of the extended UTAUT model are shown in Fig. 1. In the following, we mention several theory-informed guidelines from literature along with the constructs of the extended UTAUT model that are related to the scope of our contribution. Note that we are not going to elaborate on the UTAUT moderators (i.e., gender, age, experience, and voluntariness) as they are not included in the study and model of [12], and as their contributions to our topic are included in the constructs of the extended UTAUT model.

Performance Expectancy (PE) refers to the degree to which an individual believes that using the system will help him or her to attain gains in job performance [13]. The construct has a significant positive influence on individuals’ intention to use information technology. It captures perceived usefulness (i.e., a person’s expectation that using the technology will result in improved job performance), extrinsic motivation, job fit, relative advantage, and outcome expectations [14]. We believe that the performance expectancy for the SDC technology is (going to be) trivial for its users, considering the current proliferation of personal data (volume-wise, type-wise, velocity-wise, etc.) for which the traditional ways of data minimization (like simply removing sensitive attributes from a data set) are not effective anymore. This triviality can be attributed to, among others, the presence of an urgent problem and minimizing risks for organizations [15]. In our context, the urgent problem is the issue of personal data protection and the risks for organizations are those associated with personal data disclosures.

Effort Expectancy (EE) refers to the degree of ease associated with the use of the system [13]. The construct has a significant positive influence on individuals’ intention to use information technology. It captures perceived ease of use (i.e., “the degree to which an individual believes that using a particular system would be free of physical and mental effort”) and complexity (i.e., “the degree to which a technology is perceived as relatively difficult to understand and use”) [14]. For example, healthcare professionals appreciate and use healthcare when the deployed Information System (IS) is perceived as not too complex [16].

Social Influence (SI) refers to the degree to which an individual perceives the importance of what others believe about using the new system by him/her [13]. The construct has a significant positive influence on an individual's intention to use information technology. This factor is not relevant to our contribution.

Facilitating Conditions (FC) refer to the degree to which an individual believes that an organizational and technical infrastructure exists to support the use of the system [13]. The construct has a significant positive influence on individuals' intention to use information technology. It captures perceived behavioral control, compatibility, and the objective factors that "make an act easy to do" [14]. According to [14, 17], training users and assisting them when they encounter difficulties are examples of facilitating conditions that influence technology utilization. Study [18] considers the adoption of a computerized clinical support system by paramedics and identifies the factors affecting its implementation and use. They conclude that the implementation of such a new technology can be effective when there is support at the organizational level to deal with the technical limitations, lack of integration with other systems, and practical usability problems. Study [14] investigates how construction firms can utilize IT to better manage geographically dispersed projects and conclude that a successful technology adoption requires technical support. Study [16] considers technical infrastructure available to system users and other internal support such as organizing training sessions and programs as a necessary part of the facilitating conditions to enhance end-users' knowledge and skills (i.e., a mandatory means to enable end-users to adjust to the new system, to build positive attitudes towards the system and to appreciate the benefits offered by the system). Emphasizing the ability of the staff in an organization on the success or failure of an IS adoption, the authors recommend training before and during the introduction of the IS.

Individuals' Attitude (AT) encompasses individuals' positive or negative feelings about performing the target behavior [12]. The construct influences individuals' intentions and behaviors (i.e., Behavioral Intention (BI) which is a measure of the strength of one's intention to perform a specific behavior, see [12]) to use a new system. To this end, as illustrated in Fig. 1, the mechanisms for enhancing the FC construct (like the technical and organizational infrastructure relating to an IS/IT, e.g., helpdesks and training programs), the PE construct (like those for usefulness), The EE construct (like those for ease of use), and for SI construct (like having the advice of champions) can be instrumental.

Training users and supporting them to use a new system are concerned with knowledge transfer from experts to practitioners. Study [19] aims at understanding organizational technology adoption by looking at different types of innovation knowledge used in the technology adoption process. They identify three types of innovation knowledge as important for mobilizing potential adopters towards technology adoption, namely: Awareness-knowledge about the existence and key properties of an innovation, how-to-knowledge about how to use an innovation properly at individual and organizational levels, and principle-knowledge about the functioning principles underlying the innovation. The authors conclude that balancing the principles knowledge and the how-to knowledge early in the innovation process enhances successful technology adoption.

For successful adoption of the SDC technology within (public) organizations, in this section we mentioned several theory-informed guidelines that are based on technology adoption theories and that, considering the scope of this study, are concerned with designing the User Interface (UI) of SDC software tools. Reducing the complexity of SDC tools, in the following section, we aim at designing a UI for SDC tools that satisfy a subset of the mentioned technology adoption guidelines, namely: (a) easing the (perceived complexity of) SDC tool/system usage, (b) facilitating technology training via providing a simplified version of the SDC tool/system, and (c) transferring the how-to-knowledge and the principle-knowledge about SDC technology to users within (public) organizations.

III. DESIGN PRINCIPLES

Designing a UI for a software application often involves a considerable investment of time and effort. This investment can be reduced by adhering to previously established design guidelines [20]. These guidelines can serve as a starting point for eliciting software requirements for developing a software prototype. However, note that these guidelines can only indicate the right direction and defining requirements and assessing how a system will be used by the user are equally important.

To design an easy-to-use UI for SDC tools we need to understand the relevant usability issues of existing tools that users face. To this end, we start with studying the literature on software usability and how it is measured, i.e., specifying software usability criteria in Section III-A. We use these criteria to assess the usability of the UI of an existing SDC tool in Section III-B. Using the result of this UI assessment, we formulate the design principles for our proposed UI of the SDC tool in Section III-C.

A. *Software Usability Criteria*

Software usability has been defined in different ways in the literature. According to [21], software usability relates to how a system interacts with the user and cannot be defined as a specific aspect of a system. Regardless of how it is defined, measuring usability usually helps in determining the quality of the system.

Many models have been proposed to measure software usability. One of such models is called Quality in Use Integrated Measurement (QUIM). This model unifies existing models into a single consolidated, hierarchical model of usability measurement [22]. The QUIM model proposes 10 factors on which the usability of the software can be assessed. Each of these 10 factors corresponds to a specific facet of usability that has previously been identified in existing models (e.g., Metrics for Usability Standards in Computing, the MUSiC) or standards (e.g., ISO 9241, ISO/IEC 9126 and IEEE Std.610.12), see p. 168-169 [22]. The QUIM factors are:

1. *Efficiency:* Capability of the software to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.

2. *Effectiveness:* Capability of the software to enable users to achieve specified tasks with accuracy and completeness.
3. *Productivity:* The amount of useful output that is obtained from user interaction with the software.
4. *Satisfaction:* Subjective responses from users about their feelings when using the software.
5. *Learnability:* Capability of the software to enable users to feel that they can productively use it right away and then quickly learn any subsequently new functionalities.
6. *Safety:* Capability of the software to meet the user requirements during normal operation without harm to other resources and the environment.
7. *Trustfulness:* The faithfulness a software offers to its users.
8. *Accessibility:* Capability of a software to be used by persons with some type of disability.
9. *Universality:* Capability of a software to accommodate a diversity of users with different cultural backgrounds.
10. *Usefulness:* Capability of a software to enable users to solve real problems in an acceptable way.

These factors are further decomposed into 26 measurable usability criteria [22], providing a metric through which the factors can be measured. These usability criteria, whose relationships with QUIM usability factors are given in Fig. 2, are: (a) Time behavior, i.e., the capability to consume appropriate task time when performing its function, (b) resource utilization, i.e., the capability to consume appropriate amounts and types of resources when the software performs its function, (c) attractiveness, i.e., the capability of the software product to be attractive to the user through, for example, use of color or graphic design, (d) likeability, i.e., users' perceptions, feelings, and opinions of the product, (e) flexibility, i.e., whether the UI of the software product can be tailored to suit users' personal preferences, (f) minimal action, i.e., the capability of the software product to help users achieve their tasks in a minimum number of steps, (g) minimal memory load, i.e., whether a user is required to keep minimal amount of information in mind in order to achieve a specified task, (h) operability, i.e., the amount of effort necessary to operate and control a software product, (i) user guidance, i.e., whether the UI provides context-sensitive help and meaningful feedback when errors occur, (j) consistency, i.e., the degree of uniformity among elements of the UI and whether they offer meaningful metaphors to users, (k) self-descriptiveness, i.e., the capability of the software product to convey its purpose and give clear user assistance in its operation, (l) feedback, i.e., the responsiveness of the software product to user inputs or events in a meaningful way, (m) accuracy, i.e., the capability to provide correct results or effects, (n) completeness, i.e., whether a user can complete a specified task, (o) fault tolerance, i.e., the capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface, (p) resource safety, i.e., whether resources (including people) are handled properly without any hazard, (q) readability, i.e., the ease with which visual content (e.g.,

	Factors									
Criteria	Efficiency	Effectiveness	Satisfaction	Productivity	Learnability	Safety	Trustfulness	Accessibility	Universality	Usefulness
Time behavior	+			+						
Resource utilization	+			+						+
Attractiveness			+						+	
Likeability			+							
Flexibility		+	+					+	+	+
Minimal action	+		+		+			+		
Minimal memory load	+		+		+			+	+	+
Operability	+		+				+	+		+
User guidance			+		+			+	+	
Consistency		+			+	+		+	+	
Self-descriptiveness					+		+	+	+	
Feedback	+	+							+	+
Accuracy		+				+				+
Completeness		+				+				
Fault-tolerance						+	+			+
Resource safety						+				
Readability								+	+	
Controllability							+	+	+	+
Navigability	+	+					+	+	+	
Simplicity					+			+	+	
Privacy							+		+	+
Security						+	+			+
Insurance						+	+			
Familiarity					+		+			
Loading time	+			+					+	+

Figure 2. Relationship between factors and criteria in QUIM [22].

text dialogs) can be understood, (r) controllability, i.e., whether users feel that they are in control of the software product, (s) navigability, i.e., whether users can move around in the application in an efficient way, (t) simplicity, i.e., whether extraneous elements are eliminated from the UI without significant information loss, (u) privacy, i.e., whether users' personal information is appropriately protected, (v) security, i.e., the capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access, (w) insurance, i.e., the liability of the software product vendors in case of fraudulent use of users' personal information, (x) familiarity, i.e., whether the UI offers recognizable elements and interactions that can be understood by the user, (y) load time, i.e., the time required for a Web page to load (i.e., how fast it responds to the user), and (z) appropriateness, i.e., whether visual metaphors in the UI are meaningful.

In applying QUIM into practice, its developers suggest measuring the 26 criteria on a Likert scale and calculating the response by mapping the results according to the effect of the criteria on enhancing the usability factors. The QUIM developers also propose to use it not only for measuring the usability quality of a system but also for guiding how to incorporate usability into a software design [22]. Thus, we use the model to measure the usability of ARX in Section III-B, and to identify the usability criteria that are measured low, i.e., the improvement aspects and/or the design principles of the proposed prototype UI in Section III-C.

B. ARX Usability Evaluation

Using a mixed-method approach, we investigated the challenges users face while using a typical and open-source SDC tool called ARX. ARX is a freely available SDC tool for protecting microdata sets, which undergoes continuous development, testing and documentation updates from its contributors [23]. It supports most personal data protection models, disclosure risk models and data quality models. These features are made accessible through a UI and a java software library containing Application Programming Interfaces (APIs) to access all ARX features. Owing to its active development, ARX has been transformed into a flexible and versatile tool, supporting almost all arbitrary combinations of a wide range of techniques in a salable manner [24]. As its methods are complex from a theoretical perspective, ARX can usually only be operated by experts [24].

The data collection for measuring the usability of ARX is based on one-to-one sessions through a video conferencing application (Skype). Every data collection session lasted for about 30 to 45 minutes. The data collection process was divided into two parts: Semi-structured interviews for qualitative data collection (N=5) and surveys for quantitative data collection (N=7). Five participants carried out both parts and two additional persons participated in only the quantitative data collection process. The participants were selected based on their levels of proficiency with ARX, technical skill level and age. All users can be categorized as entry-level users who did not have in-depth knowledge or ample hands-on experience with using SDC tools (thus being non-experts in this field). However, the participants were cognizant of the SDC theories at a beginner or medium level. They could, therefore, use ARX to protect a microdata set and analyze the impact it has on the data utility and risks.

During the first part of the data collection process (i.e., the qualitative part), the participants were asked a series of open-ended questions to facilitate discussion about the ease of use. The questions ranged from basic to more specific topics, loosely based on the QUIM model, and took about 15 to 20 minutes of the interview time. Additionally, they were asked to do a heuristic evaluation of the ARX application.

The second part of the data collection process (i.e., the quantitative part) aimed at confirming the data collected in the previous step. This included an online survey based on the criteria of the QUIM model described in Section III-A. The QUIM model criteria are designed to measure the quality of all kinds of software products (e.g., Operating Systems and Web Applications). As such, some QUIM criteria do not apply to desktop applications such as ARX. Therefore, we used only 18 out of the 26 QUIM usability criteria to measure the usability of ARX. For example, criteria like ‘insurance’ and ‘resource safety’ could not be considered. Other criteria like ‘time behavior’ and ‘resource utilization’ that were related to optimizing software performance were not considered as they were not within the scope of this study. The relevant 18 usability criteria were presented on a 5 points Likert scale to measure the level of agreement or disagreement the respondents of the survey had for the statements.

Qualitative data analysis is done by following three steps: data reduction, data display and drawing conclusions [25]. First, data reduction is done using a streamlined code-to-theory model [26]. The model follows a step-by-step process from the collected data, i.e., the interview transcripts (a total of approximately 6500 words) being coded and subsequently combined into categories with similar attributes through axial coding [27]. For categorizing the data, the codes and categories are developed deductively first and then inductively. Miles & Huberman support this by saying that when there exists a preliminary theory, one can construct an initial list of codes and categories from it and change or refine these during the research process as new codes and categories emerge inductively [28]. Therefore, the codes and categories are generated from the literature reviewed in Section III-A. The benefit of the adoption of existing codes and categories is that one can build on and/or expand prevailing knowledge [25]. The result of this data analysis is displayed in Table 1. For an overview of the used method see Appendix C of [29].

Table 1. Results of qualitative data analysis for ARX

Code	Category
Prior knowledge needed Guidance required	Minimal memory load
Limited information Not instructive Minimal explanation Not self-explanatory	Self-descriptiveness
Inconsistent	Consistent elements
Confusing Non-uniform structure Switching between tabs	Navigability
Cluttered Outdated Unattractive	GUI design
Effortful	Operability
Advanced language Cannot interpret data Language not helpful	Data display (textual)
Easy visualisation	Data display (visual)
Discouragement Difficult to use Robustness	Feelings invoked

The abovementioned model of qualitative data analysis is found to yield richer results when it is used in mixed-methods research [30]. The numerical results of the quantitative data analysis are displayed in Table 2, where the mean, variance and Standard Deviation (SD) are calculated for each question of the conducted survey among the 7 participants, where each question corresponds to one of the 18 chosen QUID criteria. Refer to Appendix D of [29] for individual responses to the survey.

Lastly, we interpreted the results of both qualitative and quantitative data analyses and draw conclusions. On merging these results, we found that some results were convergent. The mixed-method enabled us to identify several problem areas of ARX usability as summarized below:

1. *Minimal memory load:* ARX requires its users to recall the data anonymization concepts from memory to complete a task. Thereby, it increases the burden on memory rather than promoting recognition.
2. *Self-descriptiveness:* ARX lacks self-explanatory features. This is compounded by a lack of adequate external supporting documentation.
3. *User guidance:* Low self-deceptiveness of ARX results in poor user guidance.

Table 2: Results of quantitative data analysis

Strongly Disagree = 1, Disagree = 2, Neutral =3, Agree = 4, Strongly Agree = 5 *to 3 decimal places, sample size = 7				
S. No	Usability Criterion	Mean*	Variance*	SD*
1	Attractiveness	2.571	1.387	1.178
2	Likeability	2.428	0.816	0.903
3	Flexibility	1.571	0.530	0.728
4	Minimal action	2.14	0.408	0.638
5	Operability	1.857	0.979	0.989
6	Minimal memory load	1.857	0.122	0.349
7	User guidance	1.571	0.244	0.494
8	Consistency	3	1.714	1.309
9	Self-descriptiveness	1.857	0.408	0.638
10	Feedback	2.714	1.632	1.277
11	Accuracy	4	1.142	1.069
12	Completeness	3.857	0.693	0.832
13	Readability	2.142	0.979	0.989
14	Controllability	2.571	1.102	1.049
15	Navigability	2.428	1.387	1.178
16	Simplicity	2.571	1.102	1.049
17	Familiarity	2.285	0.204	0.451
18	Appropriateness	2.714	1.061	1.030

4. *Navigability:* The design elements of ARX impede a smooth navigation experience for the user and adds to more confusion.
5. *Minimal action:* Lack of information and guidance leads to users finishing a task in a larger number of steps than needed.
6. *Familiarity:* Given the extensiveness of ARX's features, its UI (such as a content display) does not invoke feelings of familiarity in the user, forcing users to constantly look up things.

We note that some of the ARX usability problem areas are interdependent. One problem might be perceived as an enabler for another problem. ARX tries to provide a range of functionalities to anonymize microdata. This distinguishes it from its peers. However, having a wide range of functionalities is also its drawback, reducing the range of its user base. The lack of a large user base impedes having more supporting documentation and weighing down its usability. Therefore, there is a need to simplify ARX functionalities.

C. *Principles of a New UI Design*

Design principles specify some software requirements. Software requirements describe the functions and features that a software system must provide and the constraints under which it must operate [31]. These requirements can be categorized as: user and functional requirements.

The ARX problems identified in Section III-B indicate plenty of room for its improvement. These problems can be addressed by applying certain software design principles (i.e., user requirements) that we derived from our literature review. These design principles, i.e., user requirements, are summarized in Table 3, followed by a detailed explanation.

Additionally, the tool should be capable to anonymize microdata while providing a means to balance information loss and disclosure risks. Such requirements are categorized

Table 3. User requirements for addressing six usability problems of ARX

Problem Area	User requirement
Minimal Memory Load	Minimalistic design to avoid visual clutter
	Consistent interface elements based on existing mental models
	Offloading tasks by using default values or visual clues for decision making
Self-descriptiveness	Intrinsic methods to relay information
	Use of simple, unassuming language
	Providing contextual functions and information
	Instinctive placing of visual metaphors
User Guidance	Principle of tunnelling and selective attention through multi-step pathway forms with inline validation for task completion
Navigability	Defining a clear primary navigation area
	Minimal hierarchical structures that embrace predictability such as a left-hand side navigation menu
Minimal Action	Streamlining and grouping similar task actions on one page/tab of the screen
Familiarity	Incorporating predictable design elements in pace with current trends

as functional requirements in this contribution. These functional requirements specify the required properties of the tool from the perspective of the data anonymization domain (i.e., the SDC models, methods, and parameters relevant to the protection of the microdata sets in mind).

During the investigation of the usability of ARX, it was surmised that the functionalities of ARX have to be simplified as an overload of options does not necessarily lead to better usability results. This can be achieved by reducing the domain-related functionalities. Providing users with fewer options can result in making better-informed decisions by users without facing decision fatigue, as observed in the literature review. We carried out this functionality reduction based on some domain knowledge, as studying (the impact of) these requirements was out of the scope of this study. The reduced set of functionalities for the prototype are streamlined into the following categories:

- *Anonymization models*: We opted for k-anonymity [32] with l-diversity [33].
- *Data utility measures*: We opted for three general-purpose metrics of Average Equivalence Class Size, Non-Uniform Entropy, and Granularity.
- *Disclosure risk measures*: We opted for three metrics which are based on the prosecutor, journalist, and marketer attacker models [34].

Concerning the functionalities chosen above, ARX has around 5 configuration parameters (like the suppression limit for k-anonymity) that we set to the default recommended values.

There are also some global requirements that are common for such software tools like users being able to open (a new) project, save a project, import/export microdata sets in various formats, close, minimize or maximize the application, filter raw data to remove direct identifiers from the microdata set through an edit function, receive hints on SDC related terminologies via pop-ups while hovering on a particular term, disable or enable the hint option from the settings, and search within the documentation of the application about the related theories.

As mentioned above, for our UI design we focused on resolving the user requirements as summarized in Table 3, while reducing the number of functional requirements based on some domain knowledge.

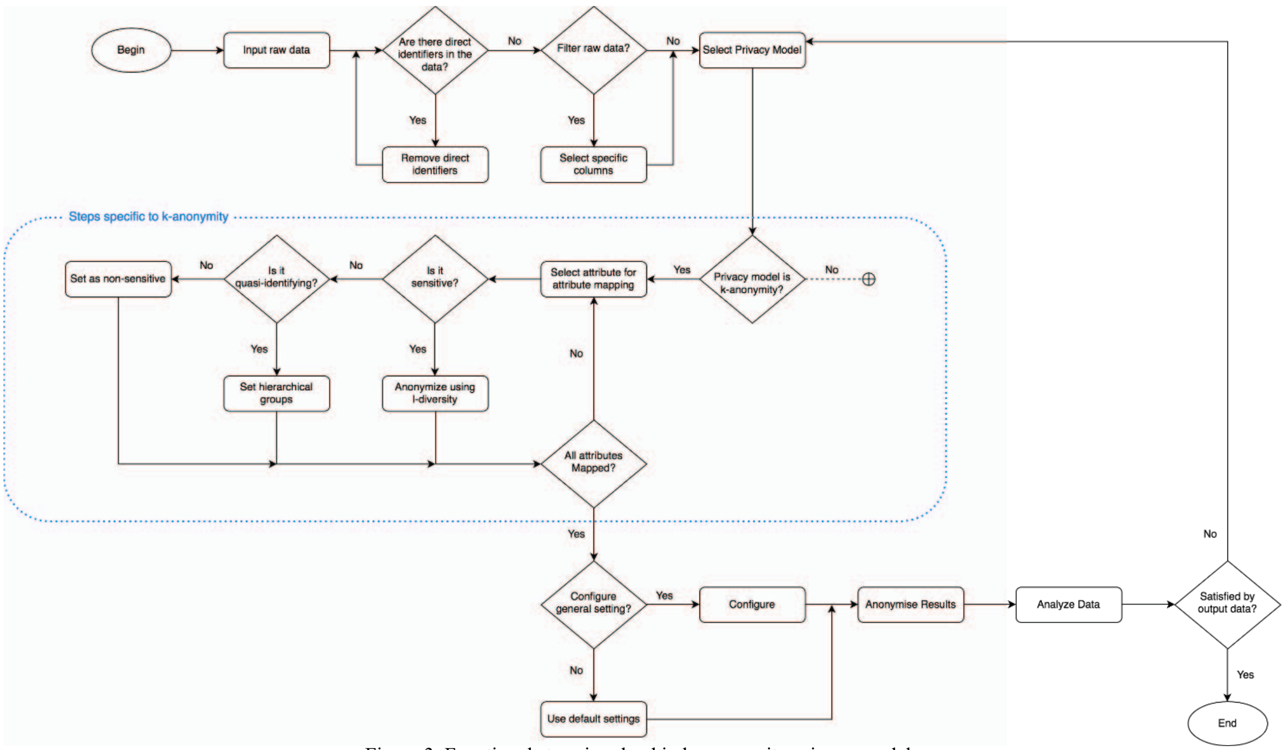

Figure 3. Functional steps involved in k-anonymity privacy model.

IV. PROPOSED DESIGN

Our UI prototype is given the name Danaamta which is a combination of two words – anaam (a Hindi word for 'one without a name') and data. In the following, we describe the workflow of the chosen functional requirements and the features of Danaamta prototype.

A. Functional Workflow

In Fig. 3 the workflow of anonymizing a data set using k-anonymity is shown. The dotted box divides the process by differentiating between steps that will be common for all other privacy models and the steps that are specific only for k-anonymity. This is done to make the prototype more flexible in design. Flexible arrangement of simplified operations can help in extensibility of the software and integration with other operations with the software [35].

B. Design Features

The wireframes of the Danaamta UI prototype are shown in appendix E of [29]. The prototype is given a modern outlook, see Fig. 4, by taking examples of commonly used software tools to promote familiarity. A minimalist look and a dark color scheme are some of the attributes of recent UI designs. Gradients and subtle drop shadows are used to draw user attention to interactive elements.

The Danaamta UI is divided into a static and dynamic section, as can be seen in Fig 4. A left-hand side menu is the primary navigation area that remains static. Placed adjacent to this navigation area is the dynamic main body which takes up most of the screen to display context-sensitive content based on the menu option selected. Fig. 5 presents a sitemap that indicates the structure of the UI prototype and all the pages contained, along with major functionalities. The pages represent context-sensitive content which dynamically load on the main body of the prototype depending on the menu option selected. The prototype can be accessed here [36].

A general use case can be defined to describe the working of the prototype. First, the user creates a project in the prototype application. Next, the user uploads the microdata file that is required to be anonymized. On uploading the microdata file, the user can edit this file to remove certain columns such as direct identifiers from the data set. Then, the user can configure settings required to anonymize the file. The

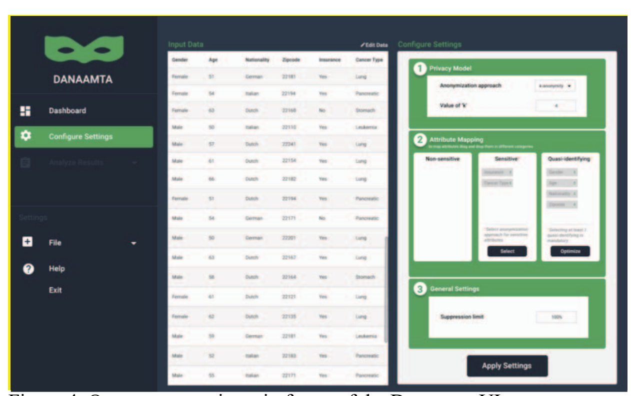

Figure 4. One representative wireframe of the Denaamta UI prototype.

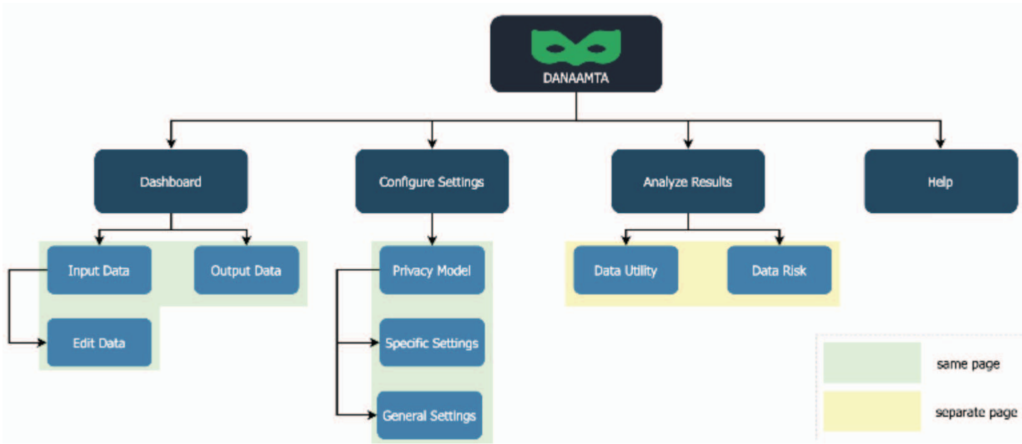

Figure 4. Fig. 5 Sitemap of the Danaamta prototype.

user can select the desired privacy model. Additionally, the user can perform attribute mappings and apply general privacy settings. On configuring the settings, the user can now apply it to reveal the transformed data set. The transformed data set can then be analyzed based on the measures provided for assessing data utility and data disclosure risks. Lastly, the user can download the anonymized microdata file for sharing. This is a general use case, however, in a real-life scenario the user will probably go back and forth between changing the settings till the desired level of anonymity is achieved.

V. EVALUATION

The Danaamta prototype was evaluated by allowing participants to interact with the prototype and conducting a quantitative survey to measure its usability contrast with ARX. The evaluation was carried out with five participants (N=5) who had earlier been interviewed on ARX (so they all were familiar with ARX). The survey examined the 15 statements which analyzed the identified problems of the ARX application during the previous data collection stage (See Section III-B) and then addressed in the prototype. For example, the measurable criteria identified in the QUIM model [22] formed some of the statements that directly measured the improvement in the identified problem areas of ARX. The survey required the participants to rate their agreement with the statements on a 5 points Likert scale. The participants were first asked to fill the survey evaluating ARX and, then they were given the link to the Danaamta prototype and asked to explore it. On completing this, they were asked to fill the same survey, but now evaluating the prototype. Besides, they were asked to share their textual feedback and experience with the prototype. The evaluation survey can be found in Appendix F of [29].

A. Results

The results revealed that the user group preferred Danaamta over ARX. Findings suggested that Danaamta overcame the problems of ARX, which were identified in Section III-B. However, there was no significance in the results that measured whether Danaamta actually reduced the complexity that is associated with the theories of data anonymization process (i.e., the complexity associated with the functional requirements). It was perceived that the user could complete the specific task using either of the applications. This observation could have been better justified if the evaluations could measure the accuracy and cost of completing the task using the two applications given the diversity in the skill level of the participants with a larger sample size.

B. Discussion

ARX packs a lot of functionalities in its design to provide users with an arsenal of anonymization techniques in hopes of resulting in acceptable anonymized data. There is this implicit assumption among the ARX developers that ARX will be used by SDC experts. Therefore, ARX developers are not aiming at mass adoption but rather at a stable software solution that incorporates the best research results on SDC – an assumption that stems from the fact that handling data for the process of anonymization is a task best left to experts. Therefore, the ARX design does not take into consideration the needs of users who might not be SDC experts. This assumption that ARX users will be well versed in the theories of SDC is its limitation of being adopted at a large scale. A reason for this can be the infancy and instability of the SDC field.

During the research, it was found that the complexity of a software tool is deeply rooted in the functionality it provides. A messaging application is a mere communication platform, nothing more, nothing less. On the other hand, a data anonymization tool holds rather a much more complex functionality. The fact that there is no one-shot operation of eliminating the risk of data disclosure but only measures to reduce it is quite tricky to explain in practice.

Thus, ARX provides multiple functionalities to give its users the capabilities to anonymize microdata sets to the best of their abilities. But this overwhelming choice of options mostly leads to a higher cognitive effort by novice users. This leads the user to make rather mediocre decisions which is posited by the literature on the paradox of choice [37].

Moreover, given the low user base of ARX, there are not many supporting materials to explain its functionalities in detail or tutorials to see how it can be used at its maximum potential. For cases like these, software needs to be self-contained, i.e., without depending on third-party materials to explain its functionalities but being self-descriptive through its design. This is partially explained by technology adoption models where facilitating conditions like technical infrastructure can influence user behavior.

However, there has been an oversight in the assumption of data anonymization should be left to SDC experts. As Helen Hayes rightly put “every expert was once a beginner” [38]. Everyone must start somewhere to become an expert and hence beginning from a complex, but stable solution like ARX might not be the right approach. There is a need to start small, to master the basics before moving onto more sophisticated concepts to build that expertise. And this is the gap that our UI prototype might be able to bridge.

C. Limitations

This study was restricted to a small sample size. This small size might not be significant for the exploratory study conducted on the problems that users face with ARX, because a large sample size is not needed for an explorative research for problem finding. However, the small sample size study can be a drawback during the evaluation of the prototype where the reliability of the results is affected. Conducting a non-parametric test on Likert data with a sample size < 10 and with 15 Likert items could result in low reliability of the survey results for evaluating the prototype. Moreover, a small sample size also restricts diversity in the participants. In this case, all participants were from an IT background and the same set of participants took part in both interview/survey processes. This could have impacted the evaluation results as the participants were aware of ARX problems and, therefore, would incline towards what had been improved in the prototype.

The QUIM model provided some Likert data items that could measure the usability of the applications. However, the model cannot be applied to measuring the time taken to complete a task and judge the goodness of the results. This was also restricted by the prototype design as it is only a simulation and not a fully functional tool.

The prototype was made with the intention of guiding entry-level users through the anonymization process. Thus, with the use of concepts like tunneling and selective attention, the prototype could be interpreted to be influential in determining the path of the privacy analysis. This could make the prototype persuasive in design. Persuasive design uses principles of psychology to design effective and engaging interfaces [39]. While this works for gaming applications or social networking websites, it might not be acceptable for an SDC tool. Here, the process of privacy analysis of sensitive data could become biased for entry-level users who rely on such tools with a persuasive design. The existence and extent of this bias can be revisited in future research.

Lastly, while the Denaamta prototype might be a good approach to address the problem of adopting SDC technology within organizations, there could be other complementary approaches needed to increase the usability and adoption of ARX. For example, by creating a learning framework to understand SDC technology, organizing training workshops about ARX and establishing SDC/ARX helpdesks as suggested by technology adoption theories (see Section II-D).

VI. CONCLUSION

Minimizing the amount of personal data in data sets, in general, and applying SDC techniques, in particular, are necessary when collecting or sharing privacy-sensitive data. The personal data should be minimized only to the data that are required and allowed for a chosen (legitimate) data usage. Applying SDC technology into practice can be a complex and demanding task for non-SDC-experts due to lack of adequate education of theoretical and practical foundations of SDC technology, the context-dependency of applying SDC technology, and the necessity of adopting a multi-disciplinary approach when applying SDC technology. This complexity hinders the adoption of SDC technology within organizations.

Inspired by technology adoption theories, our study aimed at reducing the complexity of the UI of SDC tools for successful adoption of the SDC technology within (public) organizations. To this end, we conducted user studies to identify the relevant UI-related usability issues of an existing tool called ARX. The identified UI-related issues of ARX (i.e., minimal memory, feature self-descriptiveness, user guidance, navigability, minimal action, and UI familiarity) were addressed in a new UI design for ARX by applying the software design principles derived from literature. Further, we used some domain knowledge to limit the SDC-related functionalities of ARX to reduce its complexity.

The findings suggested that the designed UI overcomes the identified UI-related issues of ARX but show no significance in reducing the complexity associated with the SDC-related complexity (i.e., how to apply the appropriate SDC models and methods). The latter outcome seems reasonable because investigating how to reduce SDC-related complexity was out of the scope of this study. This study suggests that while reducing the complexity of the UI might be a good approach to address the problem of adopting SDC technology within organizations, other complementary approaches might be needed to increase the usability/adoption of an SDC tools like ARX within organizations. Examples of these complementary approaches are creating a learning framework for employees to understand SDC technology, organizing training workshops for employees about ARX, and establishing SDC/ARX helpdesks within organizations.

During the evaluation of the designed UI prototype, it was found that the UI prototype indeed was preferred over that of ARX by entry-level users, however, it did not (attempt to) shield the dependency on the complex concepts of SDC theories. This could be explained by how the prototype was developed and evaluated. In conclusion, the prototype was able to bridge the gap for non-experts to get familiar with SDC technology and thus can be helpful for the first steps of SDC adoption and deployment within organizations.

REFERENCES

[1] GDPR 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC.

[2] M.S. Bargh and S. Choenni, "On preserving privacy whilst in- tegrating data in connected information systems," In Proceedings of the 1st International Conference on Cloud Security Management (ICCSM), October 17-18, Seattle, USA, 2013.

[3] S. Kalidien, S. Choenni, and R. Meijer, "Crime statistics online: potentials and challenges," in Proceedings of the 11th Annual International Conference on Digital Government Research, Public Administration Online: Challenges and Opportunities, DG.O 2010, Puebla, Mexico, May 17-20, 2010, pp. 131-137.

[4] M.S. Bargh, R. Meijer, M. Vink, Walter Schirm, S. van den Braak and S. Choenni, "Opening privacy sensitive microdata sets in light of GDPR," in the 20th Annual International Conference on Digital Government Research, DG.O 2019, Dubai, United Arab Emirates, June 18-20, 2019, pp. 314-323.

[5] M. Elliot, F. Mackey and C. Tudor, "The anonymisation decision-making framework," A technical report published by UK anonymization network (UKAN), 2016. Manchester, UK. [Online] Available: https://ukanon.net/wp-content/uploads/2015/05/The-Anonymisation-Decision-making-Framework.pdf.

[6] LED 2016. "Law Enforcement Directive or Directive EU 2016/680 (2016). Directive (EU) 2016/680 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data by competent authorities.

[7] M.S. Bargh, A. Latenko, S. van den Braak, M. Vink and R. Meijer "Personal data protection in the justice domain: Guidelines for statistical disclosure control", Technical Report, reeks Cahier 2021-10. WODC, The Hague, The Netherlands, 26 May 2021. [Online] Available: http://hdl.handle.net/20.500.12832/3057.

[8] R. Choenni and C. Leijnse. "A framework for the automation of air defence systems," NLR, 1999, Amsterdam.

[9] M.S. Bargh and S. Choenni, "On preserving privacy whilst integrating data in connected information systems," In Proceedings of the International Conference on Cloud Security Management (ICCSM'13), Seattle, US, 17-18 October 2013.

[10] L. Rocher, J.M. Hendrickx and Y.-A. de Montjoye, "Estimating the success of re-identifications in incomplete datasets using generative models," In Nature Communications, 10(1), 3069. [Online] Available: from https://doi.org/10.1038/s41467-019-10933-3

[11] L. Sweeney. Maintaining patient confidentiality when sharing medical data requires a symbiotic relationship between technology and policy, 21 May 1997, pp. 1-21. [Online] Available: http://414.kelehers.me/papers/sweeney.pdf.

[12] Y.K. Dwivedi, N.P. Rana, A. Jeyaraj, M. Clement and M.D. Williams, "Re-examining the unified theory of acceptance and use of technology (UTAUT): Towards a revised theoretical model," In Information Systems Frontiers, 21(3), 2019, pp. 719-734.

[13] V. Venkatesh, J.Y. Thong and X. Xu, "Unified theory of acceptance and use of technology: A synthesis and the road ahead," Journal of the association for Information Systems, 17(5), 2016, pp. 328-376.

[14] K. Sargent, P. Hyland and S. Sawang, "Factors influencing the adoption of information technology in a construction business," In Australasian Journal of Construction Economics and Building, 12(2), 2012, p.72-86.

[15] R. Urquhart, et al. "The role of scientific evidence in decisions to adopt complex innovations in cancer care settings: a multiple case study in Nova Scotia, Canada," In Implementation Science, 14(1), 14, 2019.

[16] P. Ifinedo. "Technology acceptance by health professionals in Canada: An analysis with a modified UTAUT model," In the 45th Hawaii international conf. on system sciences. January, 2012, pp. 2937-2946.

[17] R.L. Thompson and C.A. Higgins. "Personal computing: toward a conceptual model of utilization," In MIS Quarterly, 15(1), 1991, pp. 125-143.

[18] A. Porter, J. Dale, T. Foster, P. Logan, B. Wells and H. Snooks, "Implementation and use of computerized clinical decision support (CCDS) in emergency pre-hospital care: a qualitative study of paramedic views and experience using Strong Structuration Theory," In Implementation Science, 13(1), 91, 2018.

[19] Y. Kyratsis R. Ahmad and A. Holmes, "Technology adoption and implementation in organisations: comparative case studies of 12 English NHS Trusts," BMJ Open 2012.

[20] S.L. Smith and J.N. Mosier, "Guidelines for designing user interface software," 1986.

[21] X. FerreÃÅ, N. Juristo, H. Windl, and L. Constantine, "Usability basics for software developers," In IEEE software, 18(1), 2001, pp.22-29.

[22] A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda. Usability measurement and metrics: A consolidated model. Software quality journal, 14(2), 2006, pp.159-178.

[23] "ARX data anonymization tool," [Online] Available: https://arx.deidentifier.org/.

[24] F. Prasser, J. Eicher, H. Spengler, R. Bild, and K. A. Kuhn, "Flexible data anonymization using ARX - Current status and challenges ahead," Software: Practice and Experience, 2020.

[25] U. Sekaran and R. Bougie, "Research methods for business: A skill building approach," John Wiley & Sons, 2016.

[26] J. SaldanÃÉa, "The coding manual for qualitative researchers," Sage, 2015.

[27] A. Strauss and J. Corbin, "Basics of qualitative research techniques," Sage publications Thousand Oaks, CA, 1998.

[28] M.B. Miles and A. M. Huberman, "Qualitative data analysis: An expanded sourcebook," sage, 1994.

[29] A. Rawat, "Designing for usability: A statistical disclosure control tool for microdata sets," Master thesis, Delft University of Technology. 2020. [Online] Available: http://resolver.tudelft.nl/uuid:f999c3fc-d9c2-4e53-9209- d863136b0dc1.

[30] T. Carmichael and N. Cunningham, "Theoretical data collection and data analysis with gerunds in a constructivist grounded theory study," in Electronic Journal of Business Research Methods, 15(2), 2017.

[31] "Software engineering chapter 2: software requirements," [Online] Available: http://www.inf.ed.ac.uk/teaching/courses/cs2/LectureNotes/CS2Ah /SoftEng/se02.pdf.

[32] L. Sweeney, "k-anonymity: A model for protecting privacy," In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05), 2002, pp. 557-570.

[33] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," In ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1):3-es, 2007.

[34] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," In Proc. VLDB Endow., 11(10):1071-1083, June 2018. ISSN 2150-8097.

[35] G. Reitmayr and D. Schmalstieg, "OpenTracker: A flexible software design for three-dimensional interaction," Virtual reality, 9(1), 2005, pp. 79-92.

[36] Denaamta web-based prototype, [Online] Available: https://www.figma.com/proto/NFga12tiWWW0XzSQ3S3uKv/DAN AAMTA?node-id=2%3A418&viewport=430%2C255%2C0.13771803677082062&s caling=min-zoom

[37] N. Patel. The Paradox of Choice: Why Less is More in UX Design, 2017. [Online] Available: https://usabilla.com/blog/paradox-choice-less-ux -design/.

[38] "Every expert was once a beginner. Are there other famous quotes that mean the same thing?", 2015. [Online] Available: https://www.enotes.com/homework-help/every-expert-was-once-beginner-other-quotes-mean-559169.

[39] M. Oduor, T. AlahaÃàivaÃàlaÃà, and H. Oinas-Kukkonen, "Persuasive software design patterns for social influence," In Personal and Ubiquitous Computing, 58, 2014, pp. 1689-1704.