UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Rain Oksvort

A Prototype for Learning Privacy-Preserving Data Publishing

Master's Thesis (30 ECTS)

Supervisor: Raimundas Matulevičius, PhD

Tartu 2017

A Prototype for Learning Privacy-Preserving Data Publishing

Abstract: Our data gets collected every day by governments, different organizations and individuals for data mining. It is often not known who the receiving part of data is and whether data receiver can be trusted. Therefore it is necessary to anonymize data in a way what it would be not possible to identify persons from released data sets. This master thesis will discuss different threats to privacy, discuss and compare different privacy-preserving methods to mitigate these threats. The thesis will give an overview of different possible implementations for these privacy-preserving methods. The other output of this thesis is educational purpose software that allows students to learn and practice privacy-preserving methods. The final part of this thesis is a validation of designed software.

Keywords: Privacy, data, publishing

CERCS: T120 Systems engineering, computer technology

Prototüüp privaatsust säilitava andmete avalikustamise õppimiseks

Lühikokkuvõte:
Erinevad organisatsioonid, valitsusasutused, firmad ja üksikisikud koguvad andmeid, mida on võimalik hiljem uute teadmiste saamiseks andmekaeve meetoditega töödelda. Töötlejaks ei tarvitse olla andmte koguja. Sageli ei ole teada andmetöötleja usaldusväärsus, mistõttu on oluline tagada, et avalikustatud andmetest poleks enam võimalik tagantjärgi privaatseid isikuandmeid identifitseerida. Selleks, et isikuid ei oleks enam võimalik identifitseerida, tuleb enne andmete töötlejatele väljastamist rakendada privaatsust säilitavaid meetodeid. Käesolevas lõputöös kirjeldatakse erinevaid ohte privaatsusele, meetodeid nende ohtude ennetamiseks, võrreldakse neid meetodeid omavahel ja kirjeldatakse erinevaid viise, kuidas andmeid anonümiseerida. Lõputöö teiseks väljundiks on õpitarkvara, mis võimaldab tudengitel antud valdkonnaga tutvuda. Lõputöö viimases osas valideeritakse loodud tarkvara.

Võtmesõnad: Privaatsus, andmed, avalikustamine

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia

Contents

Definitions and Acronyms

PPDP - Privacy-Preserving Data Publishing.
QID - Quasi-Identifier
CSV - Comma Separated File.
Import table - read table from CSV file into program memory.
Export table - write table from program memory to CSV file.
EMD - Earth Mover Distance
Open table - select table that is in program's memory to for further processing.
Identifier column - column that is either primary key for table row (such as 'Id') or that is part of table key (such as 'first name' or 'last name').
Attribute type - type of attribute (column) that is represented by that column. One of the following: Explicit_identifier, Quasi_identifier, Sensitive_attributes or Non-sensitive_attributes.
Attribute action - action to take to this attribute (column). One of the following: Keep as is, Remove or generalize.
TCP - Transmission Control Protocol.
Attribute - characteristic of data table. In other words column name. Examples: Id, Name, Age, Job, etc.
JSON - JavaScript Object Notation.
HTML - HyperText Markup Language.
CSS - Cascading Style Sheets.

1 Introduction

Data mining is often used to gain knowledge about our everyday life. Our data gets collected daily and then either sold or given to data miners. Digital information is collected by governments, organizations and individuals. For example in California hospitals are even required to publish data by the law to publish demographic data. Different parties collect and publish different data. Original data from data publisher's database often contains personal information about people whose data was published. This will cause threats to our privacy because not all data miners are trustworthy. It may happen that data publisher does not even know if they can trust data recipients. This means that data recipient might be attacker. An attacker could find a victim from one record by some identifying attributes and then link the row in the first table with a row from the second table using attributes that are common for both tables. This can lead to loss of privacy due to record linkage as an attacker could misuse the data. This is a problem because person's private data could help an attacker to take advantage of victim. For example, an attacker could impersonate victim in order to get some prescription drug or even commit a crime so that guilt would fall on victim instead. Therefore it necessary to publish it in a way that preserves privacy. Privacy-preserving data publishing offers means to publish data in a way that is still useful for data miners but preserves privacy of individual whose data was published [FWCY10].
It is necessary to protect the people from potential consequences of privacy linkage. Privacy-preserving data publishing methods will preserve person's privacy by making person not identifiable from published data. This can mean removing of identifying attributes such as first name and last name but in addition to that there different methods to generalize data in the way that makes it even harder to identify some specific person from a dataset. Such operations will improve protection of privacy while still keeping data useful for data miners.
Since there are different methods for anonymizing data, it is hard to know which method is good for which situation. The aim of this master's thesis is to compare the different methods to each other to find out which the strengths and weaknesses of each method. In the first part of this master's thesis author gives an overview of different threats to privacy, and for each threat one or more solution and comparison of these solutions by highlighting their strong and weak points. The second part of this master's thesis will be an educational program that lets users to try different privacy-preserving methods on different datasets to see and learn how they work. The third part of this thesis will be validation of this software.
The information is obtained by reading different articles on privacy-preserving data publishing [FWCY10].
The first chapter gives an overview of privacy-preserving data publishing and related work. The second chapter describes different threats to privacy that can

occur when data is published in its original form and discusses and compares some privacy-preserving methods. The third chapter discusses and compares different anonymization operations, that can be used to make data to meet the requirements set by privacy-preserving methods. The fourth chapter specifies the educational purpose prototype application which is intended to help students to learn more about privacy-preserving data publishing. The fifth chapter validates the usefulness of prototype using feedback that was received from students who used this program. In addition to that, this chapter also discusses possible threats to validity.
This thesis will address the following research questions:

MRQ How to publish personal data in a way that preserves privacy?

RQ1 What are the risks of publishing personal data and what methods can be used to mitigate these risks?

RQ2 What anonymization operations implement privacy-preserving methods?

RQ3 What is the prototype for learning these privacy-preserving methods?

RQ4 What is usability of the prototype?

Research questions 1 and 2 are answered by doing systematic literature review and summarizing reviewed literature in the chapters 1 and 2. Research question 3 is answered by designing and implementing a prototype application for learning a privacy-preserving data publishing.

2 Privacy Assets, Threats and Mitigations

This chapter will answer to research question "What are the risks of publishing personal data and what methods can be used to mitigate these risks". Since the data is collected by different parties every day and it is not possible to know whether or not data miner is trustworthy it is necessary to apply different anonymization operations on data before it is published. If data is not anonymized it can be used to perform different attacks. For example, AOL(America Online) query logs were published but removed after reidentification of a searcher [FWCY10]. Anonymization is a set of operations on data that are designed to hide the identity of the data owner. In this chapter we are going to analyze different threats to privacy and anonymization methods [FWCY10].

2.1 Related Work

Different research works have been done in the field of privacy-preserving data publishing for years. In [Swe02] where are discussed the threat of publishing data, re-identification of the victim and a privacy-preserving method called k-anonymity. In [WF06] the authors show the weakness of k-Anonymity and propose (X, Y)-Anonymity as a solution. In [NCN07] authors extend the definition of k-Anonymity to work with relational data and define a new method called MultiRelational k-Anonymity. The authors also explain why k-Anonymity cannot be used on relational data. In [MKGV07] the authors discuss different attacks on k-Anonymity and propose l-Diversity to mitigate these attacks. In [LLV07] the authors discuss record linkage and attribute linkage, show the limitation of l-Diversity and proposes t-Closeness to overcome this limitation. Cynthia Dwork has published several papers about ϵ-Differential privacy. From these papers [Dwo06] was used for writing about ϵ-Differential privacy in this master's thesis. In [Dwo06] is stated that risk to person's privacy must not increase as a result of participating in data release which is the essence of ϵ-Differential privacy. In addition to that, there are several papers that discuss how these privacy-preserving methods could be implemented. Article [FWCY10] summarizes these papers and also other papers that I have not mentioned. Privacy-preserving Data Publishing: A Survey of Recent Developments was used as a starting point for writing this masters thesis and other papers were used to gain additional information. Papers that have been published in this field are mainly informative. They explain different attacks to privacy such as record linkage, attribute linkage or table linkage. Published work also proposes mitigation methods for such attacks such as k-Anonymity, (X, Y)-Anonymity, MultiRelational k-Anonymity, l-Diversity, t-Closeness, ϵ-Differential privacy and distributional privacy. Studies also propose algorithms for anonymizing data released [FWCY10].

However by thesis author's observation research papers do not compare different privacy-preserving methods by highlighting their strengths and weaknesses. Research papers mainly highlight weaknesses of previously known methods and after that focus on proposing an other method. In this master's thesis author will compare privacy-preserving methods to each other to find out what are the weaknesses and strengths of different privacy-preserving methods. This research work will also compare the purposes of different privacy-preserving methods and the author of this thesis will write a prototype application that allows users to try different methods.

2.2 Privacy-Preserving Data Publishing Methods

Generally in PPDP (Privacy-Preserving Data Publishing), the table that is to be published is in the following form

$$T(Explicit\_identifier, Quasi\_identifier, Sensitive\_attributes, Non-sensitive\_attributes)$$

Where Explicit_identifier is a list of attributes that directly identify the person such as name, person code, address, social security number and so on. Quasi_identifier (QID) is a list of attributes whose combination could possibly identify the person such as sex, age, job, zip-code and so on. The combination of quasi-identifiers can either specify the unique record from a second table or small set of records. Sensitive_attributes are sensitive attributes that are specific to the person such as diseases or salary. Although there can be several attributes that can belong to sensitive attribute set, in this thesis we only cover privacy-preserving methods that are designed for tables that have only one sensitive attribute. Non-sensitive attributes are all other attributes that do not fit into previous categories [FWCY10]. In this master's thesis we discuss three different types of threats.

1. Record linkage attack which in other words is identity disclosure. After successful record linkage attack, attacker knows whether or not his victim is present in the data release.

2. Attribute linkage which means that value of some sensitive attribute is revealed about the victim.

3. Table linkage occurs when attacker will get to know whether or not victim is present in given table. The difference from record linkage is that in table linkage attacker does not need to find exact victim's record.

Record linkage can often lead to attribute linkage [LLV07].

The following table will give an overview of threats to privacy and methods for each threat that can be used to mitigate the threats.

Table 1: Privacy models (adapted from [FWCY10])

Privacy Model	Attack Model		
	Record Linkage	Attribute Linkage	Table Linkage
k-Anonymity	x		
(X, Y)-Anonymity	x		
MultiR k-Anonymity	x		
l-Diversity	x	x	
t-Closeness		x	
ϵ-Differential Privacy			x
Distributional Privacy			x

From table 1 it can be seen that most of the privacy-preserving methods protect against specific attack type. The only exception of methods discussed in this thesis is l-Diversity which protects against both record linkage and attribute linkage.

2.3 Record Linkage

Record linkage is an attack where attacker tries to map one or more records released dataset to victim. To do this, attacker will try to match victims' QID from released data. This could lead to exposure of data owner's private data such as political or religious views for example [FWCY10].

The following tables are to demonstrate record linkage attack.

Table 2: Work (adapted from [FWCY10])

First name	**Last name**	**Age**	**Gender**	**City**	**Work**
Juhan	Olev	39	Male	Elva	Programmer
John	Doe	35	Male	Elva	Analyst
Gebede	Ahmed	37	Male	Tartu	Developer
Beth	Smith	41	Female	Tartu	Designer
Laura	Saar	44	Female	Elva	Painter

Table 3: Disability (adapted from [FWCY10])

Age	**Gender**	**City**	**Disability**
39	Male	Elva	No
35	Male	Elva	Yes
37	Male	Tartu	No
41	Female	Tartu	Yes
44	Female	Elva	No

In the following two examples we have {First name, Last name} in role of explicit_identifier, {Age, Gender, City} in role of Quasi_Identifier, and {Work}

from table 2 and {Disability} from table 3 in role of Sensitive_Attribute. Both tables 2 and 3 contain common attributes **Age**, **Gender**, **City**. These attributes individually do not specify any individuals, but when together they can either specify person uniquely or leave attacker with only few options to choose from.

Example 2.1. (Adapted from [FWCY10]). Suppose that attacker has access to table 2 and table 3. If both tables contain same people then attacker could join these tables together on **Age**, **Gender** and **City**. This way attacker can learn whether victim has disability or not.

It is also possible to use record linkage attack where only one table is available.

Example 2.2. (Adapted from [FWCY10]). Suppose that attacker knows victims Quasi_Identifier (QID). In that case, attacker can find out if the victim has a disability because knowing the combination of Age, Gender and City will uniquely define the row in tables 2 and 3.

The following chapters will discuss different methods to mitigate record linkage attacks what were shown in examples 2.1 and 2.2.

2.3.1 k-Anonymity

Suppose that data publisher wants to publish data to data miners. This will raise a question how to publish this data in a way that attacker would not be able to uniquely identify the victim from published data. One way of doing this is to take one or more quasi-identifiers and generalize them into groups so that data owner would not be identifiable inside that group. For example, if attribute job is part of QID then programmer, analyst and tester could be generalized to a software developer. Different ages of data owner could be generalized into 5-year intervals. For example, ages 21, 23 or 24 years could be generalized to 20-25 years. After such generalization exact values will be replaced with generalized valued. This will prevent record linkage trough QID. The idea described above is called k-anonymity. Next the author will give more formal definition [Swe02].

Definition 2.1. Let T be published table and k number of records. Table T is k-anonymous when record owner is not distinguishable from least k-1 other record owners. [Swe02]

In other words, k-anonymity means that groups size on QID is k or more. Therefore probability of linking victim trough QID is $\frac{1}{k}$ or less. [FWCY10]

The number of attributes in QID can affect the value of k for which table satisfies k-Anonymity. For example, if we take QID = Age, Gender we can make a 2-anonymous table of table 3 by generalizing Age to intervals with a size of five.

Table 4: 2-anonymous disability (adapted from [FWCY10])

Age	Gender	City	Disability
[35-40)	Male	Elva	No
[35-40)	Male	Elva	Yes
[35-40)	Male	Tartu	No
[40-45)	Female	Tartu	Yes
[40-45)	Female	Tartu	No

According to definition (2.1) table 4 is 2-anonymous because the smallest group on QID of indistinguishable records has a size of 2. If we would take QID to be {Age, Gender, City} then, however, our table would only be 1-anonymous because there is no other male whose age is in range 35 to 40 years and who lives in Tartu. From table 4 we cannot tell whether the male with a disability is Juhan or John. However, since the table is only 2-anonymous we can assume the name of record owner with 50% of probability. Hence the greater the k the less probability attacker has to successfully link records from different tables. Also the more elements we include in QID the better protection k-anonymity would provide. The downside of choosing more attributes in QID is that the data would become more distorted compared to choosing fewer attributes in QID [FWCY10].
The disadvantage of k-anonymity is that it causes data distortion. For example our 2-anonymous table above will not give us exact information about age, instead it gives us the age range. Such distortion might make information less valuable for data miners [Swe02].

2.3.2 (X, Y)-Anonymity

K-anonymity assumes that each individual appears only once in a data table. However when the same person appears more than one time in a data table, k-anonymity may not provide required anonymity.

Example 2.3. One person may have many different diseases. This could mean that hospital's data set may contain more than one row for given person. If a person has 4 diseases then that person would have 4 rows in table and thus k-anonymity could give him anonymity of $\frac{k}{4}$ instead of k.

To solve this problem (X, Y)-Anonymity was proposed by Wang and Fung in 2006 [WF06] where X and Y are disjoint sets of attributes.

Definition 2.2. Let X and Y be disjoint sets of attributes in table T. When each value from set X has to be linked to k or more values from Y then table T is (X, Y)-Anonymous.

This means that if X represents some QID and Y represents identifier (person) then (X, Y)-Anonymity would require that each QID group would represent at least k different persons. This would guarantee that QID groups would offer anonymity of k [WF06].

2.3.3 MultiRelational k-Anonymity

So far we have looked different methods to anonymize data that consists of a single table. In this section we are going to discuss method called multirelational k-Anonymity [NCN07].
In often cases data is kept in multiple relational tables instead of a single table. In relational tables, not all sensitive or identifying information is kept in each table separately. Instead there is one table that contains identifying information and other tables would refer to it. In this case k-anonymity may not be the best option. It may fail to ensure anonymity or cause too much distortion making data useless for data miners [NCN07]. The example below will illustrate multirelational database.

Table 5: Person (adapted from [NCN07])

Id	**First name**	**Last name**	**Age**	**Gender**	**City**
1	Juhan	Olev	39	Male	Elva
2	John	Doe	35	Male	Elva
3	Gebede	Ahmed	37	Male	Tartu
4	Beth	Smith	41	Female	Tartu
5	Laura	Saar	44	Female	Elva

Table 6: Work (adapted from [NCN07])

Id	**Work**
1	Programmer
2	Analyst
3	Developer
4	Designer
5	Painter

Table 7: Disability (adapted from [NCN07])

Id	**Disability**
1	No
2	Yes
3	No
4	Yes
5	No

Table 5 Person is used to store the details of a specific person such as name, age,

etc. Table 6 Work will not store any other identifying details other than *Id* that refers to Id column of Person table 5. Likewise the Disability table 7 only stores *Disability* attribute and *Id* that refers to Person table 5. When data comes from multiple tables like these, it is necessary to guarantee that it would be difficult to link records between those tables. For such case multi-relational anonymity [NCN07] is proposed.

Definition 2.3. (Multirelational k-anonymity) Let PT be Person-specific table, $T_1, ..., T_n$ be a set of tables that linked to PT using identifier from PT, $T = PT \bowtie T_1 \bowtie ... \bowtie T_n$ be join of all tables and o the owner of record in table T. We say that table T is k-anonymous if for each record owner o there is at least $k - 1$ record owners with same QID [FWCY10].

In example tables above PT is Person table, T_1 is Work table and T_2 is Disability table. In this case table T would have columns {Id, First name, Last name, Age, Gender, City, Work, Disability }. Choosing QID would depend on to whom data is published [NCN07].

2.3.4 Comparison of Solutions

In this chapter, author will compare highlight strong and weak points of each privacy-preserving method discussed so far.

K-anonymity is designed for single data table where each row represents a different person. In situations where the same person is present in multiple rows k-anonymity may not offer sufficient protection. In the case of relational database, k-anonymity might distort data too much or leak privacy. Another weakness of k-anonymity is that attacker can still successfully infer the value of sensitive attribute when the diversity of sensitive attribute in QID group is too small [FWCY10] [MKGV07].

(X, Y)-Anonymity is designed for situations where the same person is present in multiple rows. (X, Y)-Anonymity ensures that each QID group has at least k different values meaning that it would be more difficult to guess some specific attribute value for given QID group [FWCY10].

Multirelational k-anonymity is designed for relational databases. Compared to k-anonymity multirelational k-anonymity protects privacy in multirelational database sufficiently while still keeping it useful for data miners [FWCY10].

Table 8: Comparison table

Criteria	**k-Anonymity**	**(X, Y)-Anonymity**	**Multirelational k-Anonymity**
Purpose	Single table where each row represents a distinct person.	Single table where one person can appear in multiple rows.	Multiple relational tables.
Weakness	Does not guarantee diversity of sensitive attribute meaning that attribute linkage is still possible. Does not guarantee sufficient privacy when person is present in multiple rows. Does not offer sufficient privacy or renders table useless for dataminers when used on multi relational database. In practice, privacy does not depend much on QID size but rather on number of distinct values in QID group [XT06].	Does not offer sufficient privacy or renders table useless for dataminers when used on multi relational database. Does not guarantee diversity of sensitive attribute.	Does not guarantee diversity of sensitive attribute. Does not guarantee sufficient privacy when person is present in multiple rows.
Strength	Easier to understand and check than (X, Y)-Anonymity	Works wit tables where the same owner is present in multiple rows.	Does not over or under anonymize relational data.

Although in table 8 k-anonymity has more weaknesses than other privacy-preserving methods. (X, Y)-Anonymity and Multirelational k-Anonymity overcome these issues by adding additional requirements to k-Anonymity.

2.4 Attribute Linkage

Privacy-preserving methods that this master's thesis covered in previous chapters work by hiding data owner in a number of other data owners with same QID. However this may not offer sufficient protection to privacy [FWCY10].

Example 2.4. For example when hospital releases a dataset of patient's diseases. In some QID group there could be four people with HIV and one person with AIDS. In this case, even if attacker does not manage to exactly identify target's record, he can guess with 80% of confidence that his target has HIV [FWCY10].

Such release is still k-anonymous but it still does not offer sufficient protection to privacy.

Attribute linkage is attack where attacker who knows QID of his target can infer sensitive attribute of target based on sensitive attributes in target's QID group. The main idea of methods in the following chapters is to reduce the correlation between QID and sensitive attribute [FWCY10].

2.4.1 l-Diversity

L-diversity is proposed by Machanavajjhala [MKGV07] to prevent attribute linkage attack. L-diversity requires that each QID group has at least l different sensitive attribute values. This requirement also satisfies the requirement of k-anonymity where $k = l$. L-diversity differs from k-anonymity for that while k-anonymity requires a group to contain at least k individuals with same QID, l-diversity requires a group to contain at least l different sensitive attributes [AY08] [FWCY10].
Next, the author of this will explain two instances of l-diversity called entropy l-diversity and recursive (c, l)-diversity.

L-diversity does not offer sufficient protection against probabilistic attack because some attributes appear more often than others. In probabilistic, the sensitive attribute is inferred because it appears more frequently than other sensitive attributes and therefore attacker can infer that his victim must also have that value for the sensitive attribute. To protect the record owner against such attacks there are stronger notations of l-diversity called entropy l-diversity [FWCY10].

Definition 2.4. Table T is entropy l-diverse when for each QID group

$$-\sum_{s \in S} P(qid, s) \cdot log(P(qid, s)) \geq log(l) \tag{1}$$

Where s is sensitive attribute $P(qid, s)$ is a probability that random person from that QID group has sensitive attribute s [FWCY10].

As a result of this condition, every QID block has to have at least l different sensitive attribute values.

Example 2.5. Let $QID = \{Age, Gender\}$ in table 4 and $P(qid, s)$ be probability that sensitive attribute 'disability' has value of yes. Using formula 1 find entropy in first QID group $-\frac{1}{3}log(\frac{1}{3}) - \frac{2}{3}log(\frac{2}{3}) = log(1.9)$ and for second QID group $-\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = log(2)$. Therefore when $QID = \{Age, Gender\}$ table 4 will satisfy entropy l-diversity when $l < 1.9$. (adapted from [FWCY10])

Recursive (c,l)-diversity assures that most common attribute does not appear too frequently in data release and lease common attributes does not appear too rarely [FWCY10].

For the following two definitions author will first define some variables. Let S sensitive attribute and $s_1, ...s_n$ possible values of S in given QID group. Lets sort counts of the occurrences of $s_1, ...s_n$ in descending order and name the new sequence $r_1, ..., r_n$ where r_1 is the most frequent element of S and r_n be the least frequent element of S. Let c be some constant chosen by data publisher [MKGV07] [FWCY10].

Definition 2.5. We say that QID block is recursive (c, l)-diverse when $r_1 < c \cdot (r_l + r_{l+1} + ... + r_n)$ [MKGV07].

Definition 2.6. We say that table T is recursive (c, l)-diverse when every QID group is (c, l)-recursive [MKGV07].

Limitation of l-diversity is that it assumes that each sensitive attribute comes from uniform distribution. When data is skewed, l-diversity may cause data loss. [FWCY10]

Example 2.6. Suppose we have data of 1000 patients and sensitive attribute disease. Lets also suppose that only 5 patients have HIV and others have all flu for disease. To satisfy k-anonymity where $k = l$ we need at lease one patient who has HIV in each QID group. This means that data loss is high [FWCY10].

2.4.2 t-Closeness

Similarly to previous example t-Closeness was proposed to prevent attribute linkage.

Example 2.7. suppose that in the dataset the value 90% of people have sensitive value a and 10% have sensitive value b. In such table there could be QID that still satisfies l-diversity requirement but where 50% of sensitive values are b. In that case person from that QID group could be inferred to have attribute b with 50% of confidence.

T-closeness was proposed to overcome skewness problem that was described in example 2.7. T-closeness requires that distribution of sensitive attribute in QID is similar to the distribution of that same attribute in the overall table. A table satisfies t-closeness when all QID groups satisfy t-closeness. T-closeness uses the Earth Mover Distance (EMD) to measure the closeness between two distributions of sensitive values. T-closeness requires that this distance is less than or equal to t. [LLV07] [FWCY10].
T-closeness has the following limitations:

1. Specification of different protection levels is not flexible.

2. EMD function does not prevent attribute linkage on numerical attributes.

3. Enforcing t-closeness would reduce the data usefulness because it requires the distribution to be same in all QID groups.

2.4.3 Comparison of Solutions

In this chapter author will compare methods for preventing attribute linkage.

L-diversity was designed to overcome weaknesses of k-anonymity, however it will not offer sufficient protection to privacy when a distribution of values of a sensitive attribute in some QID group is different from the distribution of sensitive attributes in the table. [LLV07].

To overcome this weakness of l-diversity **t-closeness** was proposed. There are several problems with t-closeness as it can reduce the usefulness of data or threaten privacy when the sensitive attribute is numerical [LLV07].

Table 9: Comparison table

Criteria	**l-diversity**	**t-closeness**
Purpose	To prevent attribute linkage for which k-anonymity is not enough.	To prevent attribute linkage in cases where l-diversity does not offer enough protection.
Weakness	Does not work when distribution of sensitive attribute is skewed. Knowing about global distribution of sensitive attribute can help attacker to assume sensitive value for victim Diversity requirement does not take into account diversity requirement [LLV07].	No good way defining protection level. EMD function does not offer protection when sensitive attribute is numerical. Strict requirement for distributions to be similar reduces usefulness of data. EMD function is difficult to compute.
Strength	Does not reduce the data usefulness.	Offers protection in cases where sensitive attribute does not come from uniform distribution.

Table 9 shows that l-Diversity and t-Closeness have both their strengths and weaknesses. While t-Closeness offers stronger protection to privacy than l-Diversity, t-Closeness is more computationally more expensive than l-Diversity.

2.5 Table Linkage

Record linkage and attribute linkage attacks are based on assumption that victim's record is already known to attacker. However, sometimes it is already enough when attacker know whether or not victim is present in released table. This can happen when the sensitive attribute in released data table is too specific. For example, suppose that a hospital released data table of patients who are mentally ill. In this case exact record does not matter to attacker anymore because no matter what exact disease hs victim has, attacker will know that his victim is mentally ill. In this case just knowing that fact that the victim is present in the data table can already be damaging. The following chapters will discuss some methods to mitigate record linkage attack [FWCY10].

Example 2.8. The following is adapted from [Man13] and illustrates where table linkage could be used. Suppose that employer has two candidates to choose from. Lets also suppose that there is a hospital that releases information about a disease that what will cost extra money for an employee when he hires a person with that disease. In that case employer might be interested in using that released data table to pick a candidate that would cost him less.

2.5.1 ϵ-Differential Privacy

In 2006 Dwork proposed another privacy-preserving method called ϵ-differential privacy, which says that threat to record owner's privacy should not significantly increase as a result of being present in a data table. ϵ-Differential privacy model compares risk to record owner's privacy before and after adding it to data table [FWCY10].

Definition 2.7. (ϵ-Differential privacy) Let F br randomized function, T_1 and T_2 be two datasets that differ exactly one record and $S \subseteq Range(F)$ a set of all possible outputs of function F. A function F gives ϵ-differential privacy for all sets T_1 and T_2 and all $S \subseteq Range(F)$ holds:

$$\left| \ln \frac{P(F(T_1)) = S}{P(F(T_1)) = S} \right| \leq \epsilon. \tag{2}$$

ϵ-Differential privacy guarantees to record owners that when they submit their sensitive information to the database, there is either very little or nothing new that attacker could learn from that data release about them. However, the downside of ϵ-differential privacy is that does not protect against record linkage and attribute linkage attacks [FWCY10].

Achieving Differential Privacy Differential privacy is achieved by adding random noise to query answer. In other words answer $a = f(X) + b$ where f is query function, X is database and b are some random noise from distribution that is close to distribution on that sensitive attribute in database [Dwo06].

2.5.2 Distributional Privacy

Distributional privacy was proposed by Blum in 2008. It suggests that when records in data table origin from some distribution then that table should only reveal information about that specific distribution. Distributional privacy offers stronger protection to privacy than ϵ-differential privacy however it has high computational cost [FWCY10].

2.5.3 Comparison of Solutions

This chapter will compare ϵ-Differential privacy to Distributional privacy.

ϵ-Differential privacy is developed to estimate whether or not it is safe to add given person to data release. It does not protect against record or attribute linkage.

Distributional privacy from the other hand does not try to answer the question whether or not it is safe to add given person to data table but rather sets limitations on what table in general can and what it cannot publish.

Table 10: Comparison table

Criteria	**ϵ-Differential privacy**	**Distributional privacy**
Purpose	To limit the maximum risk to data owner's privacy when data his record is added data release	To limit the information that gets revealed by data release within specific distribution.
Weakness	Does not offer protection against record linkage and attribute linkage.	Computation is time consuming.
Strength		Stronger privacy than ϵ-differential privacy.

Table 10 offers side by side comparison of ϵ-Differential privacy and Distributional privacy.

3 Anonymization Operations

So far we have discussed what needs to be done in order to achieve required anonymity. In this chapter we are going to answer the research question "What anonymization operations implement privacy-preserving methods".
It often happens that original table does not satisfy required anonymity criteria. In this case data needs to be modified in a such way that privacy requirement would be satisfied and at the same time data would still be useful for data miners. Such operations that lead to privacy criteria fulfillment are called anonymization operations. There are many different anonymization operations but in this master's thesis we are going to look at generalization, suppression, anatomization, permutation, and perturbation [FWCY10].

3.1 Generalization and Suppression

Generalization and suppression operations are used to hide details in QID. For example, we could generalize jobs such as tester, programmer and administrator under information technology and jobs such as a surgeon, family doctor and nurse could be generalized to medicine. If further generalization is needed information technology and medicine could be generalized to *any job*. Such generalizations can be represented using taxonomy trees. Likewise exact values such as weight or age, for example, could be generalized to intervals. However, since in generalization specific values will get replaced with more general values, data usefulness will decrease. Another weakness of Generalization is that the more columns there are in QID, the more accuracy gets lost [FWCY10] [XT06].
Suppression works by replacing some values of QID with special value to indicate that this record was not meant to be published or record could be left out of release entirely [Pal14].

Figure 1 is the sample taxonomy tree that illustrates jobs generalization described in the previous paragraph. Below we are going to give an overview of some generalization schemes.

1. Full-domain generalization. This scheme requires that all values generalized to same level of taxonomy tree. For example, when programmer is generalized to information technology then also surgeon needs to be generalized to medicine. Because of this requirement, this method has highest distortion [FWCY10].

2. Subtree generalization. For given non-leaf node either all or none of its child nodes are generalized. For example, if tester is generalized to *information*

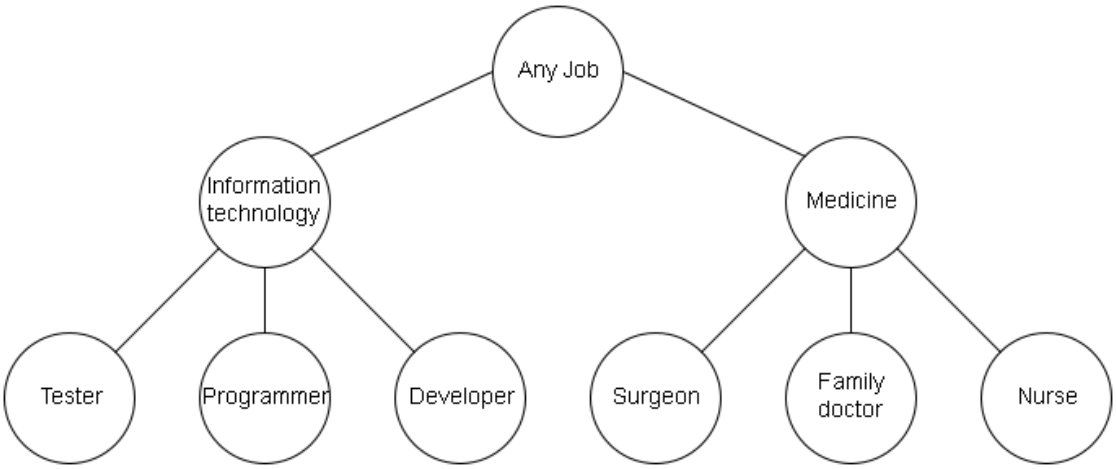

Figure 1: Sample taxonomy tree of jobs (adapted from [FWCY10])

technology then programmer and developer should be also generalized. However children of *medicine* can be left specific [Pal14].

3. Sibling generalization is similar to subtree generalization with the difference that it allows some nodes to be left un-generalized. For example if *tester* is generalized then *programmer* and *developer* can be left specific. When some value is generalized then all the other instances of this value must also be generalized [Pal14].

4. Cell generalization is a little bit more flexible than other generalization schemes. While the full-domain generalization, subtree generalization and sibling generalization require all the instances of given value to be generalized, cell generalization allows some values to stay specific. For example, if in table T there are two records that have same values for sensitive attribute we can choose to generalize one record from programmer to information technology, but keep an other record as programmer [Pal14].

First three schemes are called global recoding while the fourth one is called local recoding [FWCY10].

3.2 Anatomization and Permutation

Anatomization is a method for anonymizing data release which overcomes the weaknesses that generalization has. It works releasing data in two separate tables. One table contains QID and an other table contains sensitive attribute. Both tables have group id column which can be used to join these two tables together.

The table that contains QID groups is called QIT and table that contains sensitive data is called ST. After applying an atomization technique to a data table, an attacker would randomly have to guess sensitive attribute for a specific row in QIT table. Since anatomization will keep sensitive attribute values unchanged, data would be more useful for data miners. Since it keeps data unchanged, the correlation between QID and the Sensitive attribute is also more accurate [XT06].

Permutation is similar to an atomization but it does not associate QID with the sensitive attribute in case it has a numerical value. This disassociation is achieved by shuffling values of the sensitive attribute within QID group [FWCY10].

3.3 Perturbation

Perturbation is known for its long history, simplicity and effectiveness. It works by replacing original data with synthetic data which has similar statistical properties. Attacker cannot gain sensitive information from perturbed data because it does not correspond to original data. The downside of perturbation is that the data is meaningless for humans and it is only useful for computing statistical properties such as minimum, maximum, average, mean and so on. As a result of this, data publisher could already just publish data mining results as well. Some perturbation methods will be discussed below [FWCY10].
Additive noise is perturbation method that works by adding some random value to original value so that statistical properties of the original table would not differ too much from original ones. The downside of additive noise is that it does always offer sufficient protection to sensitive attribute. For example, when there is high correlation between QID and sensitive attribute and noise is low, the sensitive attribute's original value can be covered from perturbed data [FWCY10].
Data swapping is perturbation method that swaps sensitive attribute values with each other.
Rank swapping is similar to data swapping but first the sensitive attribute values will be divided into ranks and then the swapping is limited within one given rank. In other words value from one rank cannot be swapped with value from an other rank [FWCY10].
Synthetic data generation is an approach where a statistical model is built from original data and then data table is generated based on that model by using sampling. After that the synthetic data will be published instead of actual data.
Condensation works like synthetic data generation, but it first divides data into groups and then generates synthetic data for each group such that statistical properties of generated data would match the statistical properties of original data in that group [FWCY10].

3.4 How Operations Help To Implement method

3.4.1 k-Anonymity

Generalization of QID attributes to more general value enables to reach k-anonymity on QID size.
Suppression is not meant to be used alone, instead it should be used when after other methods do not offer sufficient protection (*).
Anatomization does not modify original data but rather releases QID and ST in different tables. Later those two tables are joined using group_id. So if we take group size = k then we have k-anonymity.
Permutation contributes to k-Anonymity the same way that anatomization does but it enhances privacy by swapping sensitive attribute values within each QID group.
Perturbation is not directly related to k-Anonymity as it generates new synthetic data instead of protecting data (*).

(*) The same applies for other privacy-preserving methods.

3.4.2 (X, Y)-Anonymity

Generalization is used to generalize specific data into QID groups.
Anatomization will lead to (X, Y) anonymity when we choose X to be QID and Y to be a sensitive attribute. In that case we would require that each element from X have at least k different matches from Y.
Permutation contributes to (X, Y)-Anonymity the same way that anatomization does but it enhances privacy by swapping sensitive attribute values within each QID group.

3.4.3 Multirelational k-Anonymity

In Multirelational k-Anonymity tables from relational database are first joined together and after that Generalization, Suppression, Anatomization, Permutation and Perturbation work exactly the same way as the work with k-anonymity.

3.4.4 l-Diversity

Generalization of attributes in QID will lead to large enough QID group that can have at least l distinct sensitive attribute values.
Anatomization can be used to implement l-Diversity when we require that in each group in ST there would be at least l distinct values.

Permutation does not directly contribute to l-Diversity as l-Diversity sets requirement on diversity of sensitive attribute but shuffling of sensitive attribute values within same QID group would not change privacy in case of l-Diversity.

3.4.5 t-Closeness

In t-Closeness Generalization, Suppression, Anatomization, Permutation and Perturbation work exactly the same way as the work with l-Diversity.

3.4.6 ϵ-Differential Privacy

Generalization of attributes increases protection to privacy making it more difficult for attacker to learn whether or not victim is in data table.
Anatomization makes it more difficult to link victim with specific record.
Permutation has the same effect as anatomization but enhances privacy protection even more by swapping sensitive attribute values.

3.4.7 Distributional Privacy

Generalization methods that we have discussed so far do not offer any guarantee that table would not leak info about other distributions.

3.5 Comparison of Solutions

In this chapter author of the thesis will compare different anonymization operation. In this master's thesis author has discussed five different anonymization operations. They all have purpose of protecting record owner against attacker but they all work a little bit different.

Generalization changes some specific value in QID group to more general value thus hiding individual in large QID group. The problem, however, is that modifying values in QID will make data less useful for data miners.

Suppression is like generalization except that it completely hides some values. It may also use special values to indicate that something is left unpublished. For example, it could have minus one for weight value to indicate that it is not to be published. This can make data mining more difficult since such will affect statistical properties if they are not left out.

Anatomization from the other hand does not generalize or hide any value behind some special value. Instead, it publishes all the data as it is but divides the QID and sensitive attribute into two separate tables that both have group id in

common. This, however, does not offer sufficient protection to privacy.

Permutation is a version of anatomization which shuffles sensitive attribute values within QID group, thus offering stronger protection for privacy than anatomization.

Perturbation is data anonymization operation that publishes synthetic data instead of actual data. This will protect record owners because their actual data will not be published at all. Instead of that, a table that is entirely generated will be published. The negative side is that such data is uninteresting for humans.

Table 11: Comparison table

Criteria	**Generalization**	**Suppression**	**Anatomization**	**Permutation**	**Perturbation**
Purpose	Hide information in QID.	To indicate that some values are not released.	Preserve data as it is.	To overcome weaknesses of anatomization.	To keep original data from being published.
Weakness	Will reduce the usefulness of data. The more attributes in QID, the greater information loss.	Special value is useless for datamining and dataminers. Datamining software must be able to handle special values.	Unmodified data in QIT and ST is danger to privacy.	Shuffling will remove correlation between QID and sensitive attribute.	Generating synthetic data might be difficult if data mining tasks are not known. Additive noise may not offer sufficient protection to privacy.
Strength	Is flexible due to different generalization schemes.	Offers strong protection for privacy	Preserves original values of sensitive attribute will cause datamining results to be more accurate. More accurate correlation between QID and sensitive attribute.	Stronger protection for privacy since correlation due to shuffling.	Has long history. Simple. Effective. Preserves statistical information.

Table 11 offers side by side comparison of anonymization methods described in this chapter. The advantage of generalization, suppression and perturbation is that they keep data in a single table rather than splitting it into multiple tables. The advantage of anatomization and permutation is that they offer a stronger protection to privacy than generalization and suppression can offer. The advantage of perturbation is that it protects the record owner's privacy but the disadvantage is that instead of actual data, synthetic data is published.

4 Prototype For Learning Privacy-Preserving Data Publishing

This chapter will answer the research question "What is the prototype for learning these privacy-preserving methods". A prototype is an educational software with the purpose of helping students to better understand Privacy-Preserving Data Publishing. This program allows users to select data table(s) and then guides it's users trough data anonymization process step-by-step. At each step, a user will be explained why this step is necessary and what needs to be done in order to complete this step. After going trough anonymization steps program will show the resulting table and statistics about it. Users can learn about PPDP by going trough steps of anonymization and comparing the results of anonymization operations. Prototype is a Spring boot [Gut02] web application with Thymeleaf [Fer11] front-end. In addition to Thymeleaf, also JavaScript [Eic95] frameworks such as jQuery [Res06], jQuery UI [Bak07], jQuery table sorter [Bac07], ChartJS [Dow13] and jQuery File Upload Plugin [Mor13] are used. JavaScript that is used by frontend is generated by TypeScript [Mic12] and icons used in this program are from icones.pro [Coz09]. The program has sample data sets to demonstrate different attacks and to allow students to experiment with different anonymization methods and parameters. A user can access content via web browser at localhost:8080.

4.1 Scope

The PPDP learning tool is designed to allow users to use different anonymization methods to anonymize data. This program allows users to practice data anonymization with different parameters and learn how data usefulness and privacy change when anonymization parameters are changed. The primary purpose of this software is help users to understand how different anonymization methods work. This program is intended to let users to fill in required fields in each step while displaying them instructions and hints. This program is not intended to automatically go trough all steps for two reasons: firstly, students would think and learn more about anonymization methods if they are asked to enter parameters themselves because then they would be actively engaged in process instead of passively observing it and secondly because automatically filling in required form fields often requires artificial intelligence which is not in scope of this master thesis. Distributional privacy is out of the scope of this program as it defines the required distribution of released table [FWCY10] but distribution theory is not part of this thesis. The program will support the following anonymization methods: k-Anonymity, (X, Y)-Anonymity, MultiRelational k-Anonymity, l-Diversity, t-Closeness and ϵ-differential privacy.

4.2 Overall Description

This chapter will give a brief overview of the purpose of created prototype application, describes the main characteristics of users of this prototype and specifies assumptions and dependencies of this prototype application. The source code of prototype can be found in Appendix 8.

4.2.1 Product Perspective

The product is a program that is created with primary intention to help users to learn and understand how privacy-preserving methods work. PPDP study assistant uses back-end to read tables from the disk and data anonymization related operations are performed in front-end. PPDP study assistant goes trough different steps of anonymization process giving explanations and hints on each step. After anonymizing table it shows user anonymization result and statistics about anonymized table.

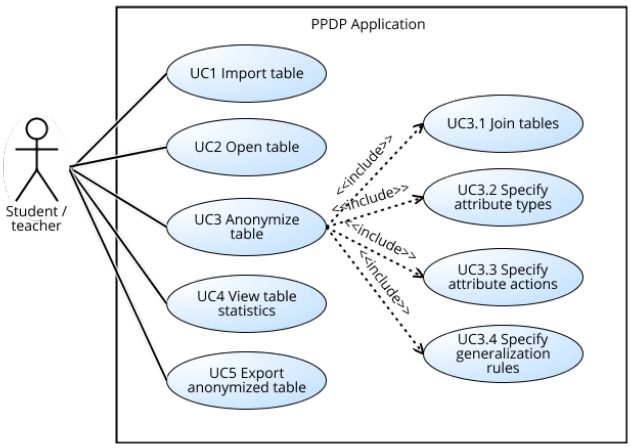

Figure 2: PPDP study assistant Use Cases

Figure 2 shows that there are 5 major use cases. The use case 3 can be divided into four subcategories. To see detailed use cases with Use Case Number, Use case Name, Use Case Description, Primary Actor, Precondition, Trigger and Normal Flow, see Appendix 6 Use cases.

4.2.2 User Characteristics

The main user of this program is a student who is learning PPDP. Student's goal is to learn how different anonymization operations anonymize data and how combining them either increases or decreases anonymity. The author of this software

assumes that users of this program have already basic understanding about PPDP. Program author also assumes that users of this program are familiar how joining tables in relational database work.

4.2.3 Assumptions and Dependencies

Since this application is written in Java, it is required that Java version 8 or newer is installed. To run this web application it is required that TCP port 8080 is free at the time if starting the program. Since the user interface is designed in HTML, JavaScript and CSS it is required that user has browsers that support JavaScript.

4.3 Specific Requirements

This chapter will show the screenshots of the views of the program and explain what user is able to do in each view, define the functional and non-functional requirements.

4.3.1 Mock-ups

This chapter will show screenshots of different views in created prototype application and under each screenshot explain what user can do in that view.

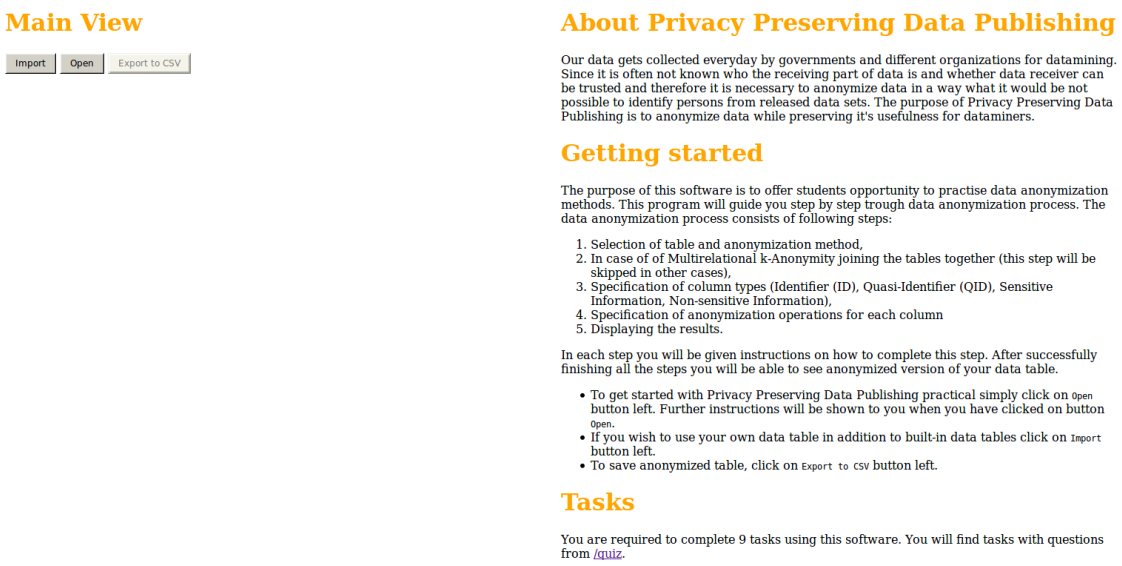

Figure 3: Main view

The main view shown on figure 3 is shown to a user when he starts the program, with Import and Open buttons active. After a user has performed anonymization

operations he will also see the resulting table and statistical information about the anonymized table. Export to CSV button becomes active after opening a table. It can be used to export table to CSV file. On the right side user will see information about PPDP and instructions on how to use this program.

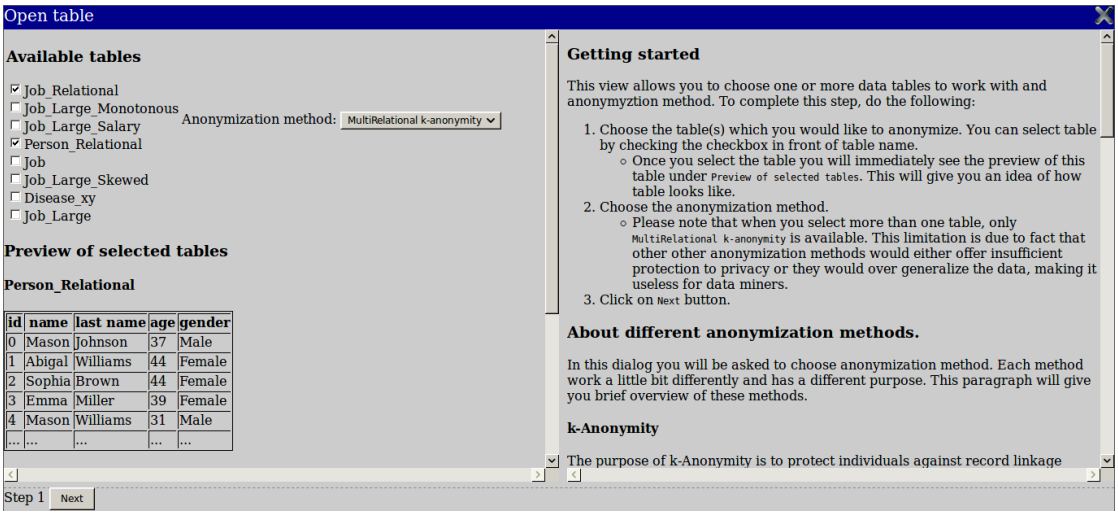

Figure 4: Open table view

Open table view shown on figure 4 will be shown after a user clicks on 'Open' button in the main view. On the left side user will see the list of tables under 'Available tables' header. In this list, both built-in and imported tables will be shown. A user is able to select one or more tables by checking checkboxes in front of table names. Under 'Preview of selected table' a short preview of each selected table is shown to help a user to know better what table(s) is he going to work with. From table preview user can already see what are the column names in selected table(s) and what kind of data is in selected table(s). This preview area will be updated every time when a user checks or unchecks tables. In addition to that user is also asked to specify anonymization method before he can continue. In case the user selects ϵ-Differential Privacy, he will be asked to enter a value for ϵ into input that will become visible below anonymization method selection. On the right side user will see a step-by-step tutorial on how to complete this step and explanations on how different anonymization methods work and what they are for.

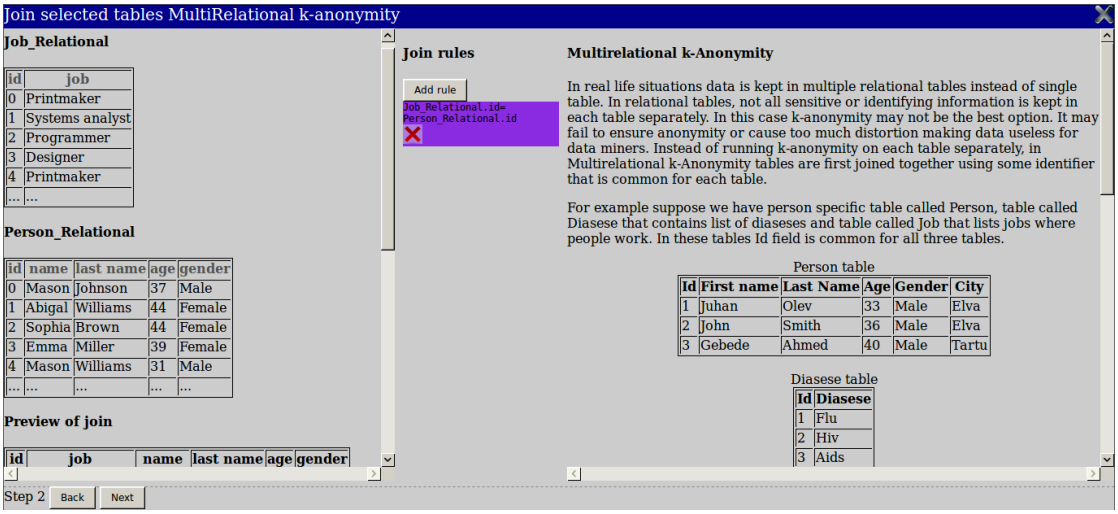

Figure 5: Join table view

Join table view shown on figure 5 will be shown when a user has checked more than one table in open table view. When a user selects only one table, this view will be automatically skipped. This view allows a user to specify join rules in order to work with relational tables. On the left user will see the tables that he had previously checked in open table view. Below the header 'Preview of join' user will see the preview of the join operation. When the user has not defined any rules yet a text 'No rules defined' is shown. When a user clicks on 'Add rule' button he will be asked click on some column in the first table and then corresponding column in the second table. In addition to that 'Add rule' gets replaced with 'Ok' button. When a user clicks 'Ok' program creates new joining rule based on selected column names and updates preview of the join operation. In addition to that when a user clicks 'Ok' the 'Ok' button changes back into 'Add rule' and created rule will also appear in the purple box under 'Add rule' button. When it is possible to join tables using the rule that is defined by a user then a user will see the result of join operation below the header 'Preview of join'. If it is not possible to use that rule to join the tables together, a user will see notification which says 'Rules are defined but seem to be invalid. Please check your rules and remove any rules that incorrect. Hint: can you expect Job to be equal to disease instead of join result. If a user has defined a wrong rule, he can delete it by clicking delete button (which is red X icon) inside rule box. On the right side user will see explanation on MultiRelational k-Anonymity and joining columns and also a step-by-step tutorial on how to complete this step.

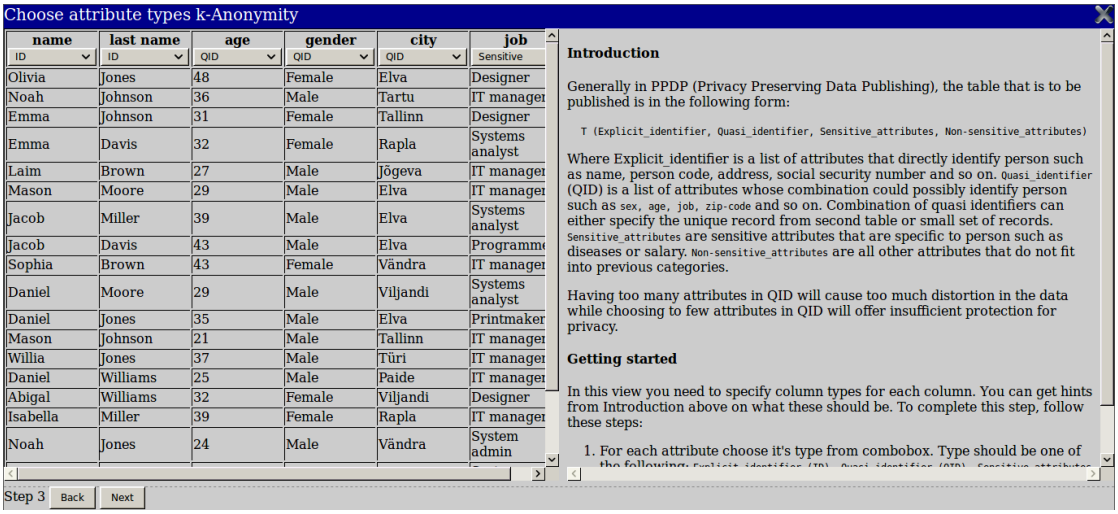

Figure 6: Attribute type specification view

Attribute type view shown on figure 6 will be shown after user has opened a table. This view will allow user to specify attribute types for each attribute. Attribute types are required to compute statistics and also as input for attribute actions view. This view shows data table with attribute type selection below each attribute on the left side. On the right side information about attribute types and step-by-step instructions on how to complete this step is shown.

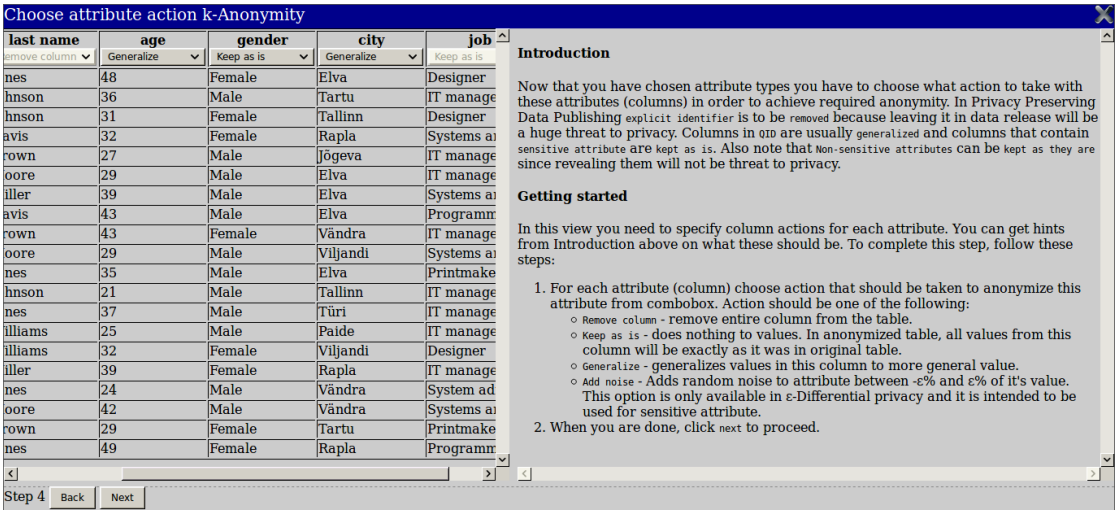

Figure 7: Attribute action specification view

Attribute action specification view shown on figure 7 allows user specify anonymization operations for each attribute that is part of QID or that is non-sensitive. On

the left side user will see the table that he is currently working on and specify what action should be taken for each attribute. On the right side user will see an explanation of possible actions and a step-by-step guide on how to complete this step.

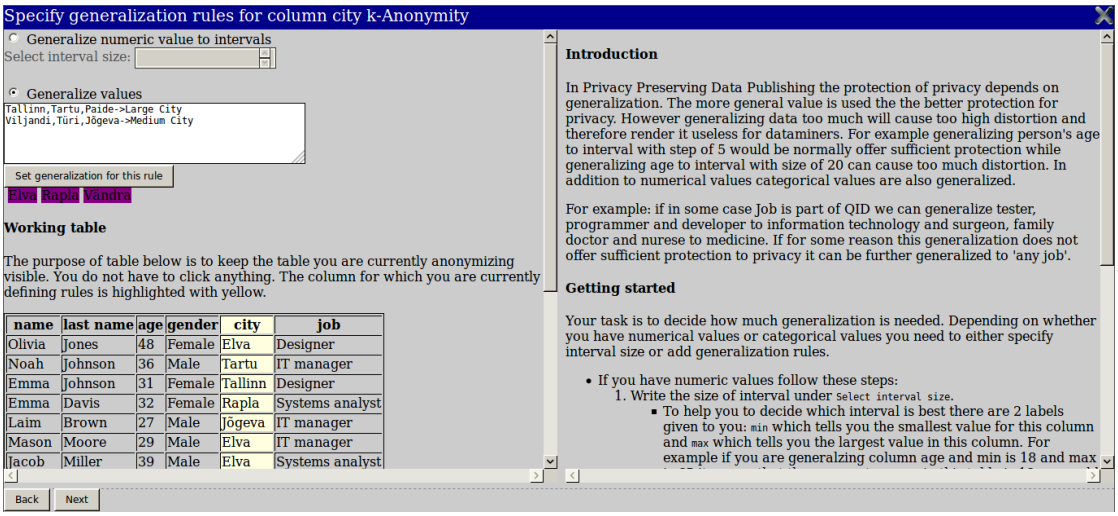

Figure 8: Generalization rules specification view

Generalization rules specification view shown on figure 8 allows user specify rules for generalization for given attribute. On the left side user will see two options: the first option is 'Generalize numeric value to intervals' and second option is 'Generalize values'. The program will automatically choose the right option for user. When the first option is selected, user will be asked to enter the size of the interval. This means that all the values in this column are numbers and therefore it is best to generalize them into intervals. When the second option is selected user will be asked to generalize specific values to some more general value. This option is selected when the column contains values that are not numbers. User will be able to select values by clicking on purple boxes under 'Set generalization for this rule'. After that user will be asked to specify a more general value for selected values. For example on this screenshot 'Tallinn' and 'Tartu' are generalized to 'Large City' This means that all occurrences of 'Tallinn' and 'Tartu' in city column will become 'Large City' and therefore become indistinguishable from each other. Under the 'Working table' header user will see the table on which he is currently working on and highlighting shows the column for which he is currently specifying generalization rules. The purpose of this table is to give user visual overview of the table and to allow him to see values in context. On the right side user will see an explanation about generalization and a step-by-step tutorial on how to complete this step.

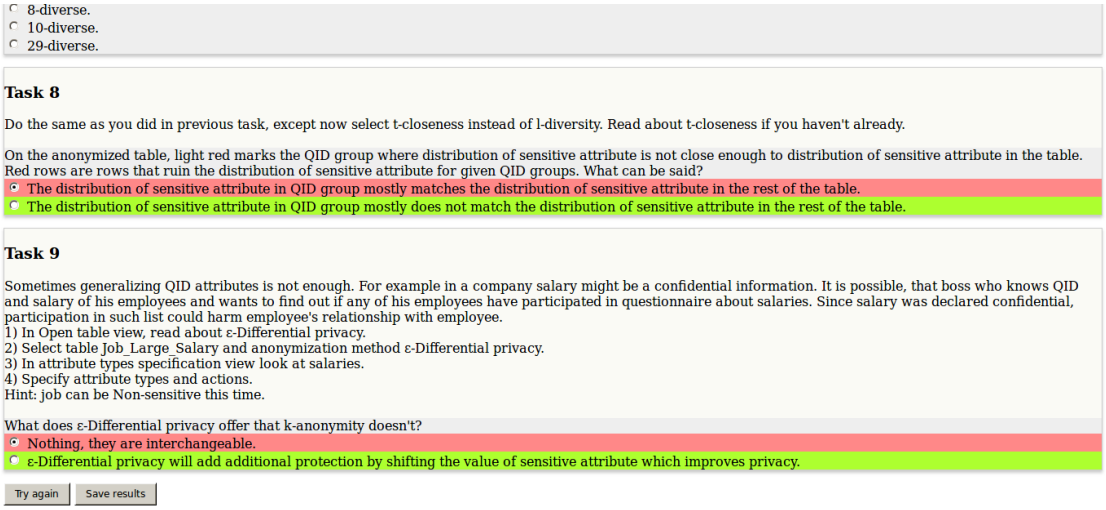

Figure 9: Quiz

Quiz shown on figure 9 can be accessed by visiting clicking on "quiz" link on main page 3. On quiz page users can see 9 tasks with multiple choice questions below these tasks. Tasks are always in the same order but questions below tasks are randomized each time user visits quiz page. After answering all questions a user can click on submit button. If all questions are answered then a user will receive feedback for the quiz where user answer is highlighted with green if it was the correct answer and with red when it was a wrong answer. In case user selected a wrong answer, the right answer for that question is highlighted with green. After clicking Submit button user can see his score on top of the page. If he wishes to retake the test he can click on 'Try again' or if he wants to download feedback to his computer he can click on 'Save results'.

4.3.2 Functional Requirements

This chapter specifies functional requirements of prototype application. Functional requirements specify what user must be able to do with the data tables in this program, how the program should interact with a user and what information should be displayed to a user. All functional requirements have the following fields: Requirement#, Requirement Type, Use case#, Description, Rationale, Created By, Fit Criterion, Dependencies, Supporting Materials, Supporting Materials, Priority and View id. Functional requirements in tabular form can be seen in appendix 4.

4.3.3 Nonfunctional Requirements

In this chapter author will specify nonfunctional requirements for given software. Nonfunctional requirements specify understandability of instructions and user interface and information it displays. These requirements also specify maximum waiting time. All nonfunctional requirements have the following fields: Requirement#, Requirement Type, Use case#, Description, Rationale, Created By, Fit Criterion, Supporting Materials, Priority and View id. Nonfunctional requirements in the form of tables can be seen in appendix 5.

4.4 Back-end Description

In this chapter author will specify major concepts and components using a class diagram. This program consists of two parts: front-end and back-end. The back-end is used to allow front-end to have access to file system and to generate and grade quizzes. Everything else that is not related to file system is done in front-end part.

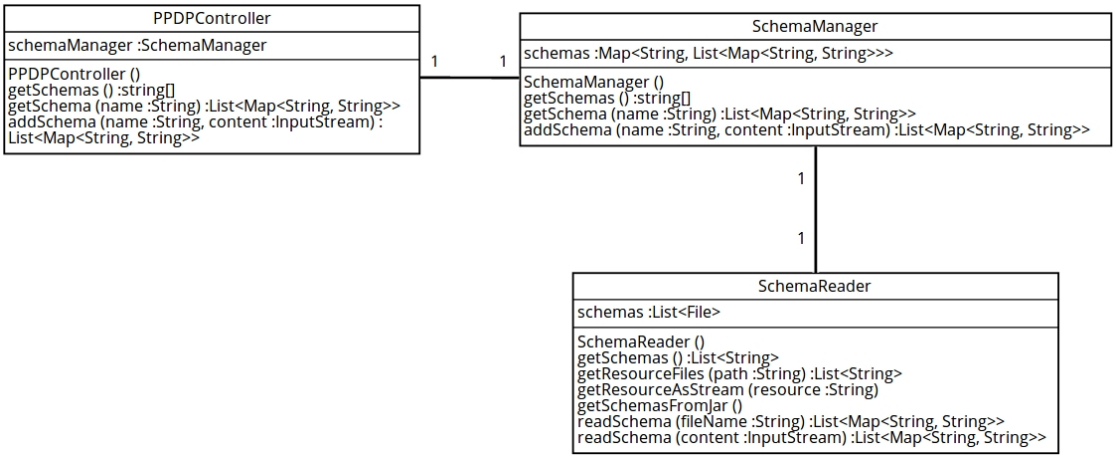

Figure 10: PPDP study assistant back-end class diagram

Figure 10 shows classes from back-end side that are used to manage .csv files. PPDPController is a class that exposes SchemaManager functionality to web and rest controllers. SchemaManager is used for listing and reading .csv files. It also handles uploaded .csv files by using SchemaReader to read uploaded a .csv file into memory and then adding it to schema list SchemaReader class is used to read .csv files from resources folder into memory. Uploaded .csv files are turned into InputStream and then read to memory using SchemaReader's readSchema method.

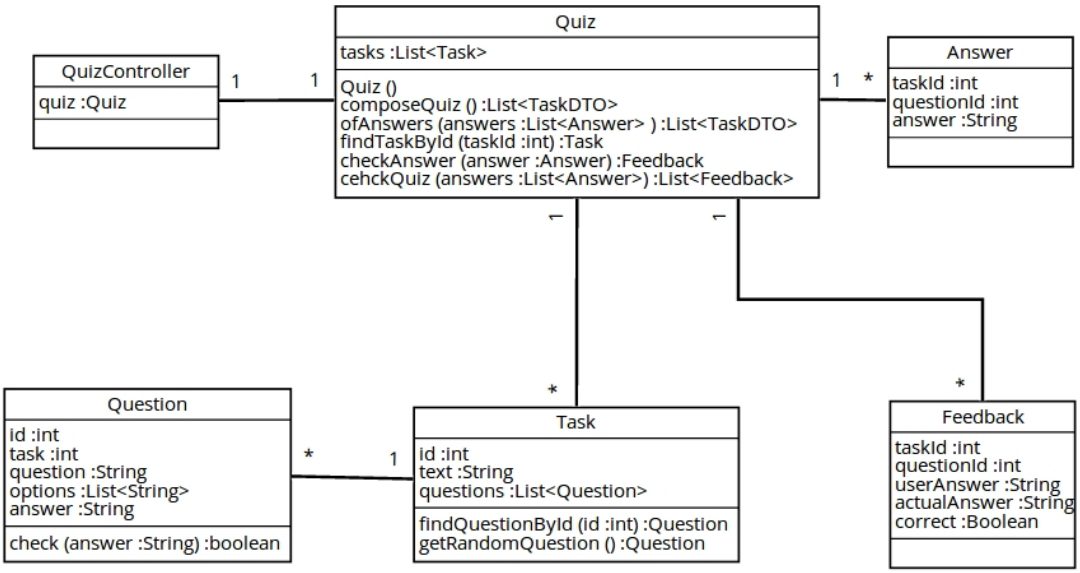

Figure 11: PPDP study assistant back-end class diagram for quiz

Figure 11 shows classes from back-end side that handles quiz part. A Quiz class is used to compose and check quiz. A quiz is made of tasks. Each task has at least one multiple choice question. Every time quiz is composed a random question is chosen for each task from the list of questions that belong to that specific task. When a student submits a quiz, answers are sent as a list of Answer objects. Quiz class then gives Feedback for each Answer, sending client back the list of Feedback.

4.5 Front-end Description

Front-end part requests the list of available tables from back-end. Later, when user wants to use that specific table, it will send an other request to get the contents of selected table. Not downloading all the tables at once helps to reduce network traffic by not downloading tables that are not needed. Front-end guides users trough different steps and the performs anonymization operations based on user input. When user gets past the open table view, internet connection to back-end is no longer required because the anonymization of data and computing the statistics is done in front-end side.

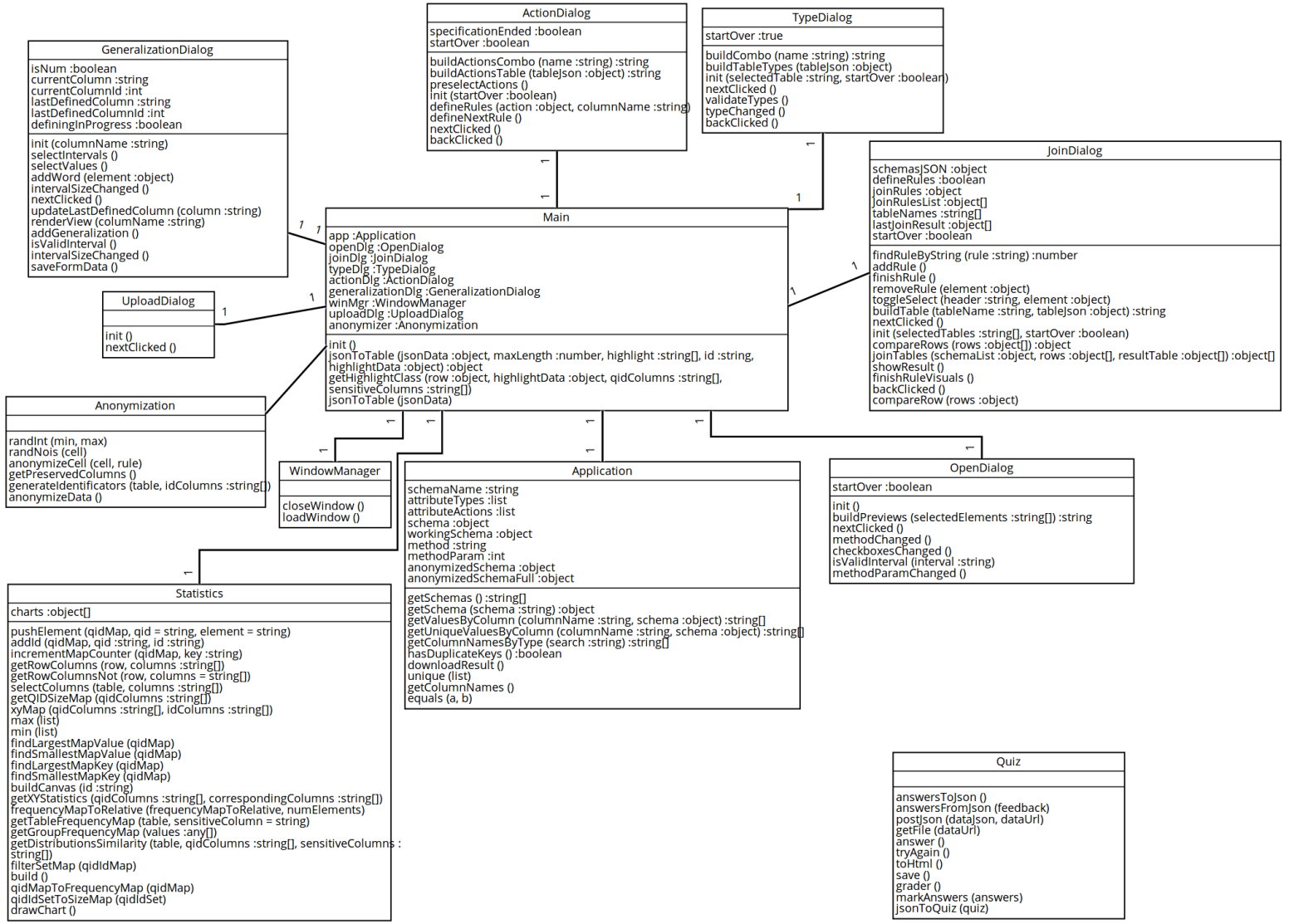

Figure 12: PPDP study assistant front-end class diagram

Figure 12 shows classes in front-end side. Front-end starts from the Main class which initializes Application and dialogs. Application class is used to interact with back-end to get the list of available tables and to get the contents of a specific table by its name. It has also functionality processing received tables and extracting information from these. Frontend also has one class for each dialog that handles rendering of that specific dialog and actions that can be performed in that dialog. Anonymization class is used to anonymize data after the user has specified rules for anonymization. Statistics class is used to generate descriptive statistics for anonymized table.

4.6 Process Diagram

This chapter gives overview of anonymization process in PPDP study assistant.

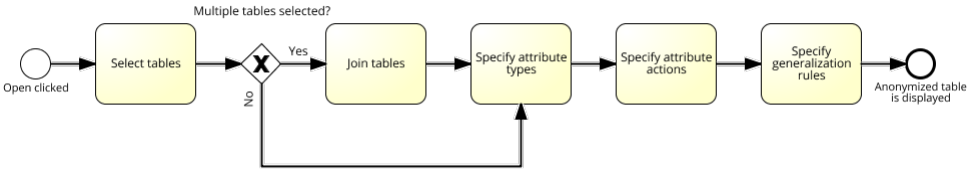

Figure 13: PPDP study assistant process diagram

Figure 13 First user selects one or more table in open table view shown on figure 4. In case user selects two or more tables user will have to specify how to join tables in join table view shown on figure 5. In case user only selects one table in open table view or when he clicks next on join table view He will be shown he will be asked to specify attribute types for each attribute in attribute type specification view that is shown on figure 6. After that user is asked to specify attribute actions using attribute actions specification view shown on figure 7. For each column for which generalization was specified, user is asked to specify generalization rules using generalization rules specification view. After that program performs anonymization operations using input received from user and displays the result.

4.7 Summary

In this chapter, we introduced the prototype for learning privacy-preserving data publishing. We defined the scope and the goal of our prototype application. We showed the screenshots of different views and explained what user can do in each view. We specified functional and nonfunctional requirements for that prototype. We created class diagrams for both back-end and front-end and explained how they work. We did process diagram and described the workflow of prototype.

5 Validation of the Prototype

In this chapter the author of the thesis will answer the question "What is usability of the prototype". Since the purpose of this software is to help students to learn and understand about PPDP the validation of this prototype will consist of two parts:

1) 9 different tasks that students have to solve,

2) Feedback questionnaire.

The aim of the tasks is to test if students can learn PPDP using this prototype tool and to test if they can use this tool independently. In addition to tasks, feedback survey was used to estimate usability.

5.1 Tasks

Before the students were given tasks to solve independently they were given a 30 minutes long introduction to PPDP with demonstration of this software which included solving of the first task. After completing each task, students had to answer questions based statistics that will be shown after anonymizing te table. These questions will test whether students understood instructions given by the prototype application. Since this tool teaches PPDP by guiding students trough different anonymization steps it is necessary to know whether or not students who use this will understand these instructions and if they are able to solve tasks independently with help of these instructions. In addition there are also questions that will test students understanding of PPDP in general, to see if this tool helped them to learn the main concepts and principles of PPDP.

5.1.1 Datasets

This prototype has built-in data sets for each privacy-preserving method. For some data sets there are smaller data set and larger data set of the same data. The purpose of different size data sets is to demonstrate that the same amount of generalization can offer sufficient protection to privacy on larger datasets while offering insufficient protection to smaller datasets. Detailed information about datasets can be seen in appendix 3.

5.1.2 Tasks

Tasks were made with purpose to help students to better understand PPDP. To complete the tasks students follow steps in each task. It was also required that students read explanations that given to them by this program on the right side.

After each task students had to answer the multiple choice questions at the end of the task. After finishing all tasks students had to submit quiz answers via the course web page. Tasks assumed that students used their knowledge from previous tasks and therefore each next task may have given less information than the previous task. Students who could not figure out task were expected to memorize what they did in the previous task or read the instructions given by this program. If that did not help the were able to ask the practical class assistant. Tasks can be seen in appendix 1.

5.2 Feedback Survey

In addition to tasks the author of this thesis also used survey questions to get more feedback from students. Since the answers to tasks may not answer all the questions about the usability of this application, additional feedback was used to find out the following aspects:

1) Usefulness of this program in learning PPDP,

2) Usefulness of tasks and questions in learning PPDP,

3) Understandability of instructions, tasks, questions and statistics,

4) Understandability and learnability of program itself.

Feedback questions were presented as statements which students were able to agree or disagree. each question had five possible answers 1-4 where 1 means strongly disagree and 4 means strongly agree. The fifth option was "Don't know" to indicate that student either did not know how to answer this question or did not want to answer it. Feedback questions can be seen in appendix 2.

5.3 Background of the Students

Students who used this software had all technical background. They were masters students of cybersecurity and software engineering. The majority of these students had finished their bachelor's in computer science but some finished their bachelor's in some non-technical field and then started to study either cyber security or software engineering masters.

5.4 Treatment of the Students

In University1 students received 30 minutes long lecture about privacy-preserving data publishing. After the lecture, the first task was solved together by guiding

students through each step. After solving the first task together we discussed the answer to the first question together with students. After that students were asked to solve the rest of the tasks independently.
In University2 students were not given any lecture on privacy-preserving data publishing. Instead of that they had to read about privacy-preserving data publishing from [FWCY10]. In class they were not given any instructions on how to use this program by lecturer and they had to learn to use this program on their own by reading the instructions that were given to them by the program itself.
After finishing each task students had to answer the question for that task. After answering all questions students were given feedback to their answers. After seeing feedback students could either retake the quiz or download the feedback and submit it to the course web page. After finishing the quiz students were also asked to answer the questions in the feedback survey that can be seen at appendix 2. In feedback survey, students were able to rate questions in four point scale where 1 was worst and 4 was worst rating. If students did not finish all tasks during the class they had one week to finish the tasks at home.

5.5 Results from University1

The average score of all questions was 3.22. The lowest average was 2.64 for the statement "You would like to use this program to prepare for the exam". The second lowest average was for the statement "Statistics in this program were easy to understand". The third lowest average was 3.07 for the statement "Questions in quiz were easy to understand". The most likely reason for this was that second task caused confusion. In second task students were asked to generalize gender attribute to demonstrate that generalizing more attributes will increase the protection of privacy. In the third task students were asked to generalize the way they did in the second task but the actual right solution required that gender would be left ungeneralized. It made many students to think that they had to generalize gender in tasks three to nine also, but in fact, they had to keep gender attribute as it was instead of generalizing. The issue about wording the tasks was verbally corrected after it became evident and since there was planned one more practical with this prototype in the other university, I also corrected tasks in handouts.
In addition to that, in class I also noticed that user interface was not correctly rendered with all web browsers. The instructions area was wider than it should have been and the working area was narrower than it should have been. That bug rendered user interface inconvenient for those who had it. I tried to fix that bug but I could not reproduce that bug after trying various different browsers on different operating systems. In addition to that the observation also showed that the majority of students managed to complete the tasks with out asking for help. Since students were able to re-take quiz as many times as they wanted and edit

the feedback to their attempt before submitting it, the quiz results should not be taken very seriously. Quiz was graded in nine point scale. Quiz results were submitted together with feedback survey via the course website. The majority of students had 9 points out of 9. The lowest score that was submitted was 7 out of 9. The average score was 8.81. Since many students manually added their name to quiz feedback, which required editing the HTML code, there is no guarantee that the scores submitted were actual scores. However out of those few who had scored less than 9 out of 9 had mainly mistakes in the seventh question which was about l-diversity.

5.6 Results from University2

In University2 the average score of all feedback questions was 3.16 which is higher than the average score of feedback questions in University1. The lowest average was 2.4 for the statement "Questions in quiz were easy to understand". The second lowest average was 2.6 for the statement "You would like to use this program to prepare for the exam". For quiz part all students received 9 points of 9. Average scores for feedback questions can be seen from appendix 7.Figure 14 shows that in University2 the median of scores was higher than scores in the University1. On figure 14, the box of the University1 is shorter than the box of University2 which means that suggests that in the University1 students opinions were more similar to each other than in University2.

5.7 Comparison of Results

The average score of all questions was quite similar in both universities. In both universities 3 was most common answer for feedback questions. In University1 the average score of feedback questions was 3.22 and in University2 it was 3.16. In both results statements "You would like to use this program to prepare for the exam" and "Questions in quiz were easy to understand" had lowest scores. However in University2 the score for the statement "Questions in quiz were easy to understand" was 2.4 which was significantly lower than the score for that same question in University1 which was 3.07. Average scores for feedback questions can be seen from appendix 7.
Below is the box plot [MSvCS] that was generated in R language [IG93] to compare the results from University1 and University2.

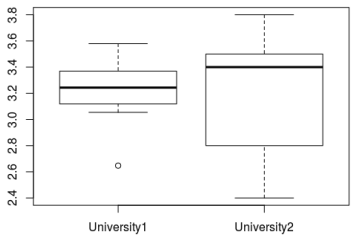

Figure 14: Box plot of results from University1 and University2.

Figure 14 shows that in University2 the median of scores was higher than scores in the University1. On figure 14, the box of the University1 is shorter than the box of University2 which means that suggests that in the University1 students opinions were more similar to each other than in University2.

5.8 Threats to Validity

There are different threats to the validity of results. The first threat is that questions in feedback survey may not reveal all the benefits or problems of this program. The second threat to validity is that a prototype application allowed students to retake tests as many times as they wanted which means that feedback to tasks may not truthfully reveal students knowledge. Students who wanted to get good points could have retaken the quiz until they have tried all different answers and then use the feedback from previous attempts to correctly fill in the last attempt. The third threat to validity is that since this quiz results are not saved in database but rather exported from the prototype application in HTML format, they could have manually edited the .html that contained the results and then submit corrected feedback instead of original. The fourth threat to validity is that multiple choice questions limit the opportunities of testing the knowledge because the options were specified which simplified answering the questions compared to letting students fill in the input with text. The fifth threat to validity is that students may fill in feedback survey randomly. The sixth threat is the shortage of students who tested this program in University2. In University2 this program was used by five students which means that the average scores may not truthfully reflect the actual average.

6 Conclusion

In this thesis we discussed threats to privacy such as record linkage, attribute linkage and table linkage that could occur when personal data is collected and published without any protection. We discussed and compared different privacy-preserving methods such as k-Anonymity, (X, Y)-Anonymity, Multirelational k-Anonymity, l-Diversity, t-closeness, ϵ-Differential privacy and Distributional privacy and highlighted their strengths and weaknesses. We discussed how these privacy-preserving methods could be implemented and then we wrote educational purpose software that allowed students to learn about privacy-preserving data publishing. Finally we created tasks that students had to solve using this software and feedback survey to measure how useful created software was for learning privacy-preserving data publishing.
In addition to writing educational software and creating tasks and feedback survey we also gave an introductory lecture to privacy-preserving data publishing and my software in the lecture of Principles of Secure Software Design course and in practical class we gave students tasks to solve independently.
The results of using this prototype application snow that in general this application is suitable for teaching students privacy-preserving data publishing. However, this program needs some enhancements before using it in class. First, the quiz results should be stored in database to avoid the manipulating with feedback. Secondly, bugs from the user interface should be removed.

6.1 Limitations

In this thesis we focused on data sets that have single sensitive attribute however in real life data sets could contain multiple sensitive attributes. This thesis did not offer a solution to such cases. Another limitation of this thesis is the selection of privacy-preserving methods. In this thesis we only discussed 6 different privacy preserving methods but [FWCY10] alone already mentioned 15 different privacy-preserving methods.
The prototype application supports k-Anonymity, (X, Y)-Anonymity, Multirelational k-Anonymity, l-Diversity, t-closeness and ϵ-Differential privacy. From anonymization operations that were described in this thesis the prototype only supports generalization. Another limitation of the prototype is that it is possible for users to cheat the in quiz by manipulating with the quiz feedback since results are not stored in database but directly displayed to user after he submits the quiz.

6.2 Answers to Research Questions

In this thesis we had four research questions for which we provide short answers:

RQ1: What are the risks of publishing personal data and what methods can be used to mitigate these risks? When personal data is used with out any protection, the following attacks could occur: record linkage, attribute linkage or table linkage. There are different privacy-preserving methods which can mitigate these risks of which in this thesis we discussed k-Anonymity, (X, Y)-Anonymity, Multirelational k-Anonymity, l-Diversity, t-closeness, ϵ-Differential privacy and Distributional privacy.
RQ2: What anonymization operations implement privacy-preserving methods? In this thesis we discussed five different ways to implement these methods: generalization, suppression, anatomization, permutation, and perturbation. Each of these methods have their own advantages and disadvantages.
RQ3: What is the prototype for learning these privacy-preserving methods? The prototype is Spring Boot web application which allows users to learn about privacy-preserving data publishing. Users can learn about privacy-preserving data publishing by reading the instructions that are given to them by this prototype and by solving the tasks that are provided by this prototype.
RQ4: What is usability of the prototype? This prototype was used in two universities where students had to solve the tasks and also answer the feedback questions. Although it is possible to cheat the quiz, the observation in class showed that majority of students managed complete the tasks without asking for help. After finishing tasks students also had to fill in the feedback survey where they had to rate different aspects of this prototype from 1 to 4 where 1 was worst and 4 was worst. The most frequently used answer was 3.

6.3 Conclusion

The experiments with the prototype in two universities showed that the prototype is usable for learning about privacy-preserving data publishing. Feedback survey results showed that majority of students found that this prototype helped them to learn about privacy-preserving data publishing. However feedback survey showed that questions in quiz should be improved in order to increase the usefulness of this prototype.

6.4 Future Work

In addition to that other privacy-preserving methods with their implementations could be discussed and software could be updated to include these methods. For example in this thesis we discussed the following data anonymization methods: generalization, suppression, anatomization, permutation and perturbation but in implemented software students and can currently only use generalization. It would be interesting to see all of these implemented. An other topic that was left out of

this thesis was measuring the data usefulness. This topic is good for future work because it measuring the usefulness of anonymized data would help to estimate if anonymized data is generalized too much or not.

References

[AY08] Charu C. Aggarwal and Philip S. Yu. *A General Survey of Privacy-Preserving Data Mining Models and Algorithms*, pages 11–52. Springer US, Boston, MA, 2008.

[Bac07] Christian Bach. jquery table sorter. MIT, 2007. `http://tablesorter.com` (Visited: May 18, 2017).

[Bak07] Paul Bakaus. jquery ui. MIT, 2007. `http://jqueryui.com` (Visited: May 18, 2017).

[Coz09] Frédéric Cozic. icones.pro, 2009. `http://http://icones.pro` (Visited: May 18, 2017).

[Dow13] Nick Downie. Chart.js. MIT, 2013. `http://chartjs.org/` (Visited: May 18, 2017).

[Dwo06] Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052, pages 1–12, Venice, Italy, July 2006. Springer Verlag.

[Eic95] Brendan Eich. Javascript, 1995.

[Fer11] Daniel Fernández. Thymeleaf. Apache License 2.0, 2011. `http://www.thymeleaf.org` (Visited: May 18, 2017).

[FWCY10] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42(4):14:1–14:53, June 2010.

[Gut02] Felipe Gutierrez. Spring boot. Apache License 2.0, 2002. `https://projects.spring.io/spring-boot/` (Visited: May 18, 2017).

[IG93] Ross Ihaka and Robert Gentleman. R language. GNU GPL v2, 1993. `www.r-project.org` (Visited: May 18, 2017).

[LLV07] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, April 2007.

[Man13] Andrei Manta. Literature survey on privacy preserving mechanisms for data publishing. Literature survey, Delft University of Technology, Delft, June/2013 2013.

[Mic12] Microsoft. Typescript. Apache License 2.0, 2012. `https://www.typescriptlang.org/` (Visited: May 18, 2017).

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[Mor13] Daniel Morales. jquery file upload plugin. MIT, 2013. `http://www.daniel.com.uy/projects/jquery-file-uploader/` (Visited: May 18, 2017).

[MSvCS] D. L. Massart, A. J. Smeyers-verbeke, Capron, and Karin Schlesier. PRACTICAL DATA HANDLING Visual Presentation of Data by Means of Box Plots.

[NCN07] M. E. Nergiz, C. Clifton, and A. E. Nergiz. Multirelational k-anonymity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 1417–1421, April 2007.

[Pal14] Amit Kumar Pal. Achieving k-anonymityusingfull domain generalization. Master's thesis, National Institute of Technology, Rourkela, Odisha, 769 008, India, May 2014.

[Res06] John Resig. jquery. MIT, 2006. `http://jquery.com` (Visited: May 18, 2017).

[spe98] Ieee recommended practice for software requirements specifications. *IEEE Std 830-1998*, pages 1–40, Oct 1998.

[Swe02] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[WF06] Ke Wang and Benjamin C. M. Fung. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 414–423, New York, NY, USA, 2006. ACM.

[XT06] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB '06, pages 139–150. VLDB Endowment, 2006.

Appendices

Appendix 1 Tasks

Task 1: (sample)
1) Open table Job_Large and select k-Anonymity as anonymization method.
2) Click Next.
3) Define attribute types. Read the instructions (Introduction and Getting Started) on right hand side. Think what type should each attribute (column) be and select right attributes.
In case you are confused, here is the list of attribute types: name - ID, last name - ID, age - QID, gender - QID, city - QID and job - Sensitive.
4) Click next
5) You are now asked to specify attribute actions for each attribute. Note that some attributes are already preselected based on your choices in previous step. However you still have to specify attribute actions for age gender and city. Again, first thing that you need to do is read the explanations on right hand side. Think what action should be done to each attribute.
In case you are stuck, here are actions: age - Generalize, gender - Keep as is (can you guess why?) and city - generalize.
6) Click next
7) You are now asked to specify interval size for values in column age. Again first thing to do is to read instructions of right hand side to get a better overview of what you need to do.Enter 10 and click Next.
8) Now you are asked to generalize city names. In real life situation, a more general value should be something that is common for all specific values. However in these tasks we care more about getting few groups that are about equal in size than about good general value.
8.1) Click on city names that could belong to one group and then click on Set generalization rule and enter some general value.
8.2) Repeat step 8.1 until you have no city names left.
In this sample task we are going to define the following rules:
Tallinn,Tartu,Paide->Large city
Viljandi,Türi,Jõgeva->Medium city
Vändra,Rapla,Põltsamaa,Elva->Small city

8.3) Click Next.
9) You should now see the resulting table. Since this is the sample task you do not have to submit result of this task.
If you did everything correctly, you should see the following information about QIDs

Unique QIDs: 29
Smallest QID group: 2
Largest QID group: 14
Since the smallest QID group contains only 2 different people this table is only 2-anonymous.

question 1:
Does 2-anonymos table offer sufficient protection?
1) Yes, because generalizing QID values make it difficult to find specific person from given table.
2) No, because with 2-anonymous table sensitive attribute value can be easily guessed (with 50% confidence).

Task 2:
After solving first task, you ended up with having 2-anonymous table. However 2-anonymous table may not offer sufficient protection to privacy. Your task is to come up with a better solution. For example 5-anonymous table would be already better.
1) Follow steps 1-8 from first task but this time, let's generalize gender values male and female to any gender.
We only generalize gender in this task to any gender in this task. In the following tasks we do not generalize or remove gender.

Question 1:
Did protection to privacy increase?
1) Yes, because there are larger QID groups.
2) No, because we used the same table as we did in previous task.

Question 2: This time table is:
1) Still 2-anonymous because we used the same table as previous time.
2) 3-anonymous because we generalized 3 columns instead of 2.
3) 5-anonymous.
4) Something else.

Question 3:
How many different persons are in smallest QID group?
1) We cannot know this because QID hides person specific details.
2) We cannot know because two persons with same explicit identifier could in same group
3) 5 because this table is 5-anonymous.

Task 3:
By now you should have learned what to do when you need to increase the protection to privacy. Now lets see an other factor that plays part in protection for table. In task 2, you anonymized table that had 200 records. Now let's look at table that has 20 records.
1) Open table Job (not Job_Large)
2) Do everything else like you did in task 1 to increase protection to privacy (select same types, same actions and specify same generalization rules as you did in task 1.

Question 1:
Let a and b be two tables. How does the size of table affect protection to anonymity given that the only difference is that table a has more rows than table b?
1) Table a is less anonymous because is has more rows than table b.
2) Table a is more anonymous because it has more rows than table b and therefore bigger QID groups will form.
3) table size doesn't matter.

Question 2:
This table is:
1) 2-anonymous.
2) 5-anonymous.
3) Same as previous time because we used same parameters as previous time
4) More anonymous than previous time because we used smaller table this time

Task 4:
We have so far anonymized tables where each row represents one person. However in some cases it could be that one person is present in multiple rows.
1) Go to Open table view.
2) On right side, scroll down to (X, Y)-Anonymity and read about (X, Y)-Anonymity.
3) Choose table Disease_xy.
4) This table is about different diseases that people have. One person can have one or more diseases. Using this hint choose proper anonymization method and click Next.
5) By now we expect that you already know what column types to choose. Choose right column type for each column and click next. If you have forgotten them, you can read about attribute types in right panel.
6) Choose proper actions for each attribute.
7) Generalize the values. Interval size for age: 10, all cities to estonia.

Question 1:
How many unique people are in smallest QID group?
1) 6
2) 8
3) 26

Question 2:
Suppose that X = {age, gender, city} and Y = {personId}. Suppose that this table satisfies (X, Y)-Anonymity for integer k=4. Suppose that each person has exactly 2 diseases. How many rows are there in smallest QID?
1) 1
2) 2
3) 3
4) 4
5) 6
6) 8

Question 3:
X = {age, gender, city} and Y = {_id}, anonymized table satisfies (X,Y)-Anonymity for k=
1) 1
2) 6
3) 8
4) 26

Task 5:
So far we have looked data sets that consist of single table, however in real life data is often stored in relational database.
1) Go to Open table view and on the right panel read about MultiRelational k-anonymity.
2) Select Job_Relational and Person_Relational.
3) MultiRelational k-anonymity is automatically selected for you. Click Next.
4) You are now asked to Join selected tables. Read the instructions on right hand side. Getting started will tell you what to do.
5) Add required rule(s). Click Next.
6) Specify attribute types and attribute actions.
Hint: id is unique identifier, you dont have to worry about name and last name (Not Defined), you can choose to remove them (Remove column).

7) Generalize age to interval size of 10.

Question 1:
What is the correct attribute type for column id?
1) ID
2) QID
3) Sensitive
4) Non-Sensitive

Question 2:
What could go wrong if relational data is anonymized before joining?
1) Nothing, the order of performing these activities does not matter.
2) Data could be distorted too much.
3) Insufficient protection could be offered.
4) Both can happen: data could be too distorted or privacy might be insufficiently protected.

Task 6:
Introducing additional requirement for better privacy protection.
1) In Open table view, read about l-Diversity.
2) Select table Job_Large_Monotonous and anonymization method l-Diversity.
3) Specify attribute types and actions.
4) Use these generalization rules:
Interval for age: 10
City names:
Tallinn,Tartu,Paide->Large city
Viljandi,Türi,Jõgeva->Medium city
Vändra,Rapla,Põltsamaa,Elva->Small city
Hint: if you don't wish to click on city names that much you can copy-paste these three lines above.

Question 1:
ow many different sensitive attribute values are in the QID group that contains the least amount of unique sensitive attribute values contains?
1) 1
2) 2
3) 6
4) 30

Question 2:
On the top of the anonymized table you can see a red QID group that contains 6 different persons. Does this group offer sufficient protection to the privacy if table is required to be 5-anonymous?
1) Yes because QID with size of 6 also satisfies requirement of 5-anonymity.
2) No because even tough QID group contains enough records, sensitive attribute can be successfully guessed.

Question 3:
On the top of the anonymized table you can see a red QID group that contains 6 different persons. Is there anything wrong with that group?
1) No, it's fine because the size of this QID group is good enough.
2) Yes, the sensitive attribute for this QID group can be guessed with 100% confidence and therefore the size of QID does not help.

Task 7:
Introducing additional requirement for better privacy protection 2.
1) Do steps 1-4 from Task 6 but this time choose Job_Large instead of Job_Large_Monotonous.

Question 1:
Suppose that person x is 60-70 years old female from small city who works as a Therapist(scroll down if you need) and that job is the sensitive attribute that needs to be kept private. What's the probability of guessing her's sensitive attribute when her's QID is known?
1) 1/1
2) 1/2
3) 1/3
4) 1/8
5) 1/10
6) 1/29

Question 2:
Why is diversity of sensitive attribute required?
1) Because the greater the diversity the harder to guess the value of sensitive attribute.
2) It actually doesn't matter, only thing that matters is the size of QID group.

Question 3:
This table is:

1) 1-diverse.
2) 2-diverse.
3) 4-diverse.
4) 8-diverse.
5) 10-diverse.
6) 29-diverse.

Task 8:
Do the same as you did in previous task, except now select t-closeness instead of l-diversity. Read about t-closeness if you haven't already.

Question 1:
On the anonymized table, light red marks the QID group where distribution of sensitive attribute is not close enough to distribution of sensitive attribute in the table. Red rows are rows that ruin the distribution of sensitive attribute for given QID groups. What can be said?
1) The distribution of sensitive attribute in QID group mostly matches the distribution of sensitive attribute in the rest of the table.
2) The distribution of sensitive attribute in QID group mostly does not match the distribution of sensitive attribute in the rest of the table.

Task 9
Sometimes generalizing QID attributes is not enough. For example in a company salary might be a confidential information. It is possible, that boss who knows QID and salary of his employees and wants to find out if any of his employees have participated in questionnaire about salaries. Since salary was declared confidential, participation in such list could harm employee's relationship with employee.
1) In Open table view, read about ϵ-Differential privacy.
2) Select table Job_Large_Salary and anonymization method ϵ-Differential privacy.
3) In attribute types specification view look at salaries.
4) Specify attribute types and actions.
5) generalize age and city as in previous tasks.
Hint: Job can be Non-sensitive this time.

Question 1:
What did happen to salary values?
1) Nothing because sensitive attribute is not generalized.
2) They were changed by +/- ϵ; % of their value.

Question 2:
What does ϵ-Differential privacy offer that k-anonymity doesn't?
1) Nothing, they are interchangeable.
2) ϵ-Differential privacy will add additional protection by shifting the value of sensitive attribute which improves privacy.

Appendix 2 Feedback Questions

Thank you for using this software to learn privacy-preserving data publishing. I would like to collect your feedback for this tool to evaluate it's usefulness and usability. Please circle the option you would like to choose.

Table 12: Feedback questions.

Please indicate the extent to which you agree with following statement.	**Strongly disagree**			**Strongly agree**	**Don't know**
1. Using program helped to learn about privacy-preserving data publishing.	1	2	3	4	-
2. Program helped to understand the difference between anonymization methods.	1	2	3	4	-
3. Program was easy to learn.	1	2	3	4	-
4. Program was easy to understand.	1	2	3	4	-
5. Program was easy to remember.	1	2	3	4	-
6. You would like to use this program to prepare for the exam.	1	2	3	4	-
7. Instructions given by this program helped to solve the tasks.	1	2	3	4	-
8. Instructions given by this program were easy to follow.	1	2	3	4	-
9. Instructions given by this program were easy to remember.	1	2	3	4	-
10. Tasks were easy to follow.	1	2	3	4	-
11. Solving tasks helped to understand privacy-preserving data publishing better.	1	2	3	4	-
12. Questions in quiz were easy to understand.	1	2	3	4	-
13. Questions in quiz helped to learn privacy-preserving data publishing.	1	2	3	4	-
14. Statistics at the right of anonymized table were helpful for answering the questions in quiz.	1	2	3	4	-
15. Statistics in this program were easy to understand.	1	2	3	4	-

Appendix 3 Dataset Descriptions

Built-in Datasets
Dataset 1:
Name: Job
Description: A small table about different jobs.
Purpose: To demonstrate that anonymization methods do not offer good protection with small tables.

Dataset 2:
Name: Job_Large
Description: A large table about different jobs.
Purpose: To demonstrate that anonymization methods offer good better protection with large dataset than with small tables.

Dataset 3:
Name: Person_rel, Job_rel
Description: A relational dataset about job. Person_rel is a person specific table and Job_rel is a job specific table. These tables are not meant to be used alone but to be joined together using id attribute.
Purpose: To demonstrate how anonymizing relational datasets (Multi Relational k-Anonymity) works.

Dataset 4:
Name: Disease_xy
Description: A table about different diseases. In this table same person can be present in multiple rows.
Purpose: To demonstrate (X, Y)-Anonymity.

Dataset 5:
Name: Job_Large_Salary
Description: A table about different jobs and their salaries.
Purpose: To demonstrate ϵ-Differential privacy.

Dataset 6:
Name: Job_Large_Monotonous
Description: Monotonous table about jobs.
Purpose: To demonstrate that the lack of diversity of sensitive attribute could lead to threat to privacy.
Dataset 7:
Name: Job_Large_Skewed

Description: Skewed table of jobs.
Purpose: To demonstrate that l-Diversity does not offer very good protection to privacy in case of skewed data in sensitive attribute.

Appendix 4 Functional Requirements

Table 13: Functional requirement 1

Requirement#	1
Requirement Type	Functional
Use case#	2
Description	User must be able open tables.
Rationale	Program needs to know which table user wants to work on.
Created By	Author
Fit Criterion	This requirement will be fulfilled when user can see and select tables that are imported into program.
Dependencies	All other requirements.
Supporting Materials	
Priority	High
View id	Figure 4

Table 14: Functional requirement 2

Requirement#	2
Requirement Type	Functional
Use case#	1
Description	User must be able to import his own table.
Rationale	User might be interested in running anonymization algorithms on his own table instead of built in tables.
Created By	Author
Fit Criterion	For this requirement to be fulfilled: 1) User must be able to import his data table to program. 2) Imported table must be listed in open table view.
Dependencies	
Supporting Materials	
Priority	Medium
View id	Figure 3

Table 15: Functional requirement 3

Requirement#	3
Requirement Type	Functional
Use case#	3.1
Description	Program must support relational tables.
Rationale	In real life situations, data is often kept in relational databases and therefore it is necessary to support relational data.
Created By	Author
Fit Criterion	This requirement is fulfilled when: 1) Open table view allows user to select multiple, tables 2) Program asks user how it should join selected tables together.
Dependencies	
Supporting Materials	[FWCY10]
Priority	Medium
View id	Figure 5

Table 16: Functional requirement 4

Requirement#	4
Requirement Type	Functional
Use case#	3.3
Description	Program must enable user to specify anonymization operations for each attribute.
Rationale	In order for program to know how to perform anonymization operations user has to tell it explicitly what to do with each column.
Created By	Author
Fit Criterion	This requirement is fulfilled when after opening (and joining if necessary) a table a program asks user to specify anonymization operations for each attribute.
Dependencies	5 and 6
Supporting Materials	
Priority	High
View id	Figure 7

Table 17: Functional requirement 5

Requirement#	5
Requirement Type	Functional
Use case#	3.2
Description	Program must enable user to specify attribute types for each attribute.
Rationale	Program needs to know attribute types in order to calculate statistics later. Also some of the data in next steps can be automatically filled in when attribute types are known.
Created By	Author
Fit Criterion	This requirement is fulfilled when after opening (and joining if necessary) a table a program asks user to specify anonymization operations for each attribute.
Dependencies	5 and 6
Supporting Materials	
Priority	High
View id	Figure 6

Table 18: Functional requirement 6

Requirement#	6
Requirement Type	Functional
Use case#	3.3
Description	Program must support the following anonymization operations: generalization, suppression, anatomization, permutation and perturbation.
Rationale	Anonymization operations are essential to protect the privacy.
Created By	Author
Fit Criterion	This requirement is fulfilled when all these operations are selectable and program properly handles them.
Dependencies	
Supporting Materials	[FWCY10]
Priority	High
View id	Figure 7

Table 19: Functional requirement 7

Requirement#	7
Requirement Type	Functional
Use case#	4
Description	Program must display statistics about table.
Rationale	Ability to see some statistics about table will add educational value to this software as users can compare protection to privacy before and after running anonymization operations.
Created By	Author
Fit Criterion	This requirement is fulfilled when software will display statistics after applying anonymization operations.
Dependencies	
Supporting Materials	
Priority	High
View id	Figure 3

Table 20: Functional requirement 8

Requirement#	8
Requirement Type	Functional
Use case#	5
Description	Program must enable user to export anonymized table into CSV file.
Rationale	Allowing users to export resulting table into CSV file would allow them to compare the outputs and further use these in their own data mining applications
Created By	Author
Fit Criterion	This requirement is fulfilled when after applying anonymization operations program displays download button.
Dependencies	
Supporting Materials	
Priority	Medium
View id	Figure 3

Table 21: Functional requirement 9

Requirement#	9
Requirement Type	Functional
Use case#	3.3
Description	Program must enable user to exclude columns from table
Rationale	Some information such as first and last name or social security number is to be removed.
Created By	Author
Fit Criterion	This requirement is fulfilled when user is able to choose "Remove" as anonymization operation for given column.
Dependencies	
Supporting Materials	[FWCY10]
Priority	High
View id	Figure 7

Table 22: Functional requirement 10

Requirement#	10
Requirement Type	Functional
Use case#	3
Description	User must be able to choose replacement values for each suppression rule.
Rationale	In suppression method, value gets replaced by special value to indicate that this value is not to be released. However in different situations we need different special values. For example sometimes it may be fine to replace string value with empty string but some other time there might be need to replace it with NULL.
Created By	Author
Fit Criterion	This requirement is fulfilled when after applying anonymization operations program will ask replacement value for each record for which suppression is needed.
Dependencies	
Supporting Materials	[FWCY10]
Priority	High
View id	

Table 23: Functional requirement 11

Requirement#	11
Requirement Type	Functional
Use case#	3.4
Description	User must be able to specify generalization rules.
Rationale	In order for program to know generalize data, user must tell it how to do it.
Created By	Author
Fit Criterion	This requirement is fulfilled when before applying anonymization operations program will ask replacement rules for each attribute for which generalization has been chosen.
Dependencies	
Supporting Materials	[FWCY10]
Priority	High
View id	Figure 8

Appendix 5 Nonfunctional Requirements

Table 24: Nonfunctional requirement 1

Requirement#	12
Requirement Type	Nonfunctional
Use case#	
Description	Users should not require manual for using this program.
Rationale	Users may loose their interest in program that is difficult to use.
Created By	Author
Fit Criterion	This requirement is fulfilled when 90% or more of users are able to use this program by reading instructions given to them by the program itself with out asking for help.
Priority	Medium
View id	Figure 3

Table 25: Nonfunctional requirement 2

Requirement#	13
Requirement Type	Nonfunctional
Use case#	
Description	Error messages should be understandable.
Rationale	Error messages such as "Column type not specified" generally result in better user experience than just saying NullPointerException
Created By	Author
Fit Criterion	This requirement is fulfilled when 90% or more of users are able to resolve problems without asking help.
Priority	Medium
View id	Figure 4

Table 26: Nonfunctional requirement 3

Requirement#	14
Requirement Type	Nonfunctional
Use case#	
Description	Anonymizing operations on built-in datasets should not take more than one second.
Rationale	Program that works fast will result in better user satisfaction.
Created By	Author
Fit Criterion	This requirement is fulfilled when applying anonymization operations and generating resulting table to any of built-in data table takes less than a second.
Priority	Medium
View id	Figure 4

Table 27: Nonfunctional requirement 4

Requirement#	15
Requirement Type	Nonfunctional
Use case#	
Description	Information that is displayed should be meaningful.
Rationale	
Created By	Author
Fit Criterion	
Priority	Medium
View id	

Appendix 6 Use Cases

Table 28: Use case 1

Use Case Element	**Description**
Use Case Number	1
Use case Name	Table management - import table.
Use Case Description	User imports CSV file into program.
Primary Actor	User
Precondition	Program must be running
Postcondition	CSV file is imported into program and user can select it from open table view.
Trigger	User's need for anonymizing custom table.
Normal Flow	1) User goes to main view. 2) User clicks on 'Import'. 3) User chooses file and clicks open. 4) Contents of CSV file is loaded into program.

Table 29: Use case 2

Use Case Element	**Description**
Use Case Number	2
Use case Name	Table management - open non-relational table.
Use Case Description	User opens table that is imported to program.
Primary Actor	User
Precondition	Program must be running and table must be opened.
Postcondition	User is in attribute type specification view.
Trigger	Need to anonymize table.
Normal Flow	1) User goes to main view. 2) User clicks 'open'. 3) User selects a table. 4) User selects anonymization method. 5) User clicks 'Next'. 6) User will see a attribute types view.

Table 30: Use case 3

Use Case Element	**Description**
Use Case Number	2
Use case Name	Table management - open relational table
Use Case Description	User opens relational table that is composed of other tables.
Primary Actor	User
Precondition	Program must be running.
Postcondition	User is in join table view.
Trigger	Need to anonymize relational data.
Normal Flow	1) User goes to main view. 2) User clicks 'open'. 3) User selects two or more tables. 4) User selects anonymization method. 5) User clicks 'Next'. 6) User will see a join table view.

Table 31: Use case 3.1

Use Case Element	**Description**
Use Case Number	3.1
Use case Name	Table anonymization - joining relational tables.
Use Case Description	User joins relational table into one.
Primary Actor	User
Precondition	Program must be running.
Postcondition	Tables are joined into one table and user proceeds from attribute types specification view with the table he got by joining tables together.
Trigger	Need to anonymize relational data.
Normal Flow	1) User is in join table view. 2) User clicks 'Add rule'. 3) User clicks on identifier column in one table. 4) User clicks on corresponding identifier column in second table. 5) User clicks 'Ok'. 6) If necessary user repeats steps 2 to 5. 7) User clicks 'Next'. 8) Program has joined tables together and user is in specify attribute types specification view.

Table 32: Use case 3.2

Use Case Element	**Description**
Use Case Number	3.2
Use case Name	Table anonymization - specify attribute types.
Use Case Description	User specifies attribute type for each attribute.
Primary Actor	User
Precondition	Program must be running table must be opened. In case of relational data UC 3.1 must be performed.
Postcondition	Attribute types are specified and user is in attribute actions specification view.
Trigger	Program needs to know attribute types in order to work properly.
Normal Flow	1) User is in select column type view. 2) User selects column type for each column. 3) User user clicks 'Next'. 4) User is in attribute actions specification view.

Table 33: Use case 3.3

Use Case Element	**Description**
Use Case Number	3.3
Use case Name	Table anonymization - specify attribute actions
Use Case Description	User specifies actions to be taken to each column.
Primary Actor	User
Precondition	Program must be running and attribute types must be specified.
Postcondition	Attribute actions are specified and user will see generalization rules specification view for each attribute for which extra input is needed from user.
Trigger	Program needs to know for each attribute, which action it needs to apply to it.
Normal Flow	1) User is in specify column actions view. 2) User chooses anonymization methods for each column. 3) User clicks 'Next'. 4) User will see generalization view for each attribute for which generalization was chosen as attribute action.

Table 34: Use case 3.4

Use Case Element	**Description**
Use Case Number	3.4
Use case Name	Table anonymization - generalization rules for numerical values
Use Case Description	User specifies generalization rules for attribute that has numerical values.
Primary Actor	User
Precondition	Program must be running and attribute actions must be specified.
Postcondition	Program has performed the anonymization operations and user will see anonymized table with statistics.
Trigger	Program needs to know how to exactly generalize data.
Normal Flow	1) User is in generalization rules specification view. 2) Option 'Generalize numeric value to intervals' is selected. 3) User enters interval size. 4) User clicks 'Next'. 5) If there are any attributes left for which generalization rules are needed, user will see generalization view for next attributes for which generalization rules are needed. Otherwise program will anonymize the table and user will see the anonymized table and related statistics.

Table 35: Use case 3.4

Use Case Element	**Description**
Use Case Number	3.4
Use case Name	Table anonymization - generalization rules for categorical values
Use Case Description	User specifies generalization rules for column that has numerical values.
Primary Actor	User
Precondition	Program must be running and attribute actions must be specified.
Postcondition	Program has performed the anonymization operations and user will see anonymized table with statistics.
Trigger	Program needs to know how to exactly generalize data.
Normal Flow	1) User is in generalization rules specification view. 2) Option 'Generalize values' is selected. 3) User clicks on values that are in same category. 4) User clicks on 'Set generalization for this rule'. 5) User enters more general value for values that were selected in step 3. 6) User repeats steps 3 to 5 until there are no unused values left. 7) User clicks 'Next'. 8) If there are any columns left for which generalization rules are needed, user will see generalization view for next column for which generalization rules are needed. Otherwise program will anonymize the table and user will see the anonymized table and related statistics.

Table 36: Use case 4

Use Case Element	**Description**
Use Case Number	4
Use case Name	Statistics - view statistics.
Use Case Description	User views statistics about anonymized table.
Primary Actor	User.
Precondition	Program must be running and attribute actions must be specified.
Postcondition	Program has performed the anonymization operations and user will see anonymized table with statistics.
Trigger	Program needs to know how to exactly generalize data.
Normal Flow	1) User has performed anonymization operations on table. 2) Statistics will be visible next to anonymized table in main view.

Table 37: Use case 5

Use Case Element	**Description**
Use Case Number	5
Use case Name	Table management - export table.
Use Case Description	User exports anonymized table into CSV file.
Primary Actor	User
Precondition	Program must be running and anonymized table must be displayed in main view.
Postcondition	CSV file is saved on disk.
Trigger	User's need to save anonymized table.
Normal Flow	1) User has performed anonymization operations on table. 2) User clicks 'Export to CSV'. 3) User chooses file destination. 4) table is saved on specified destination.

Appendix 7 Summary of Results

The following table summarizes the results.

Table 38: Feedback questions scores.

Please indicate the extent to which you agree with following statement.	**Average (University1)**	**Average (University2)**
1. Using program helped to learn about privacy-preserving data publishing.	3.42	3.60
2. Program helped to understand the difference between anonymization methods.	3.24	3.40
3. Program was easy to learn.	3.28	3.40
4. Program was easy to understand.	3.23	3.60
5. Program was easy to remember.	3.33	3.40
6. You would like to use this program to prepare for the exam.	2.64	2.60
7. Instructions given by this program helped to solve the tasks.	3.57	3.80
8. Instructions given by this program were easy to follow.	3.47	2.80
9. Instructions given by this program were easy to remember.	3.13	2.80
10. Tasks were easy to follow.	3.40	3.40
11. Solving tasks helped to understand privacy-preserving data publishing better.	3.24	3.60
12. Questions in quiz were easy to understand.	3.07	2.40
13. Questions in quiz helped to learn privacy-preserving data publishing.	3.10	2.80
14. Statistics at the right of anonymized table were helpful for answering the questions in quiz.	3.18	3.00
15. Statistics in this program were easy to understand.	3.05	2.80

Appendix 8 Source Code

The source code of prototype application can be found from:
`https://github.com/rain1/Privacy-Preserving-Data-Publishing`

Non-exclusive Licence to Reproduce Thesis and Make Thesis Public

I, Rain Oksvort (date of birth: 13th of January 1992),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

A Prototype For Learning Privacy-Preserving Data Publishing

supervised by Raimundas Matulevičius

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, May 18, 2017