C964 Computer Science Capstone

Joshua Novak

Western Governors University

A.1 Letter of Transmittal

June 23, 2022

Barry Allen, Owner

Data Solutions Inc

1215 Meadow St

South Lake Tahoe, Ca 96150

Regards, Mr. Allen

For small organizations that need to safeguard their data transfer, Data Solutions Inc has during the past six months emerged as the go-to platform. Even while safe data transfer is essential for all organizations, small firms are nonetheless subject to security flaws. The discipline of data discovery has exploded in recent years due to the growth of Big Data. While there are numerous beneficial uses for data discovery, it also provides many opportunities for unethical behavior. Consequently, many firms are transferring their data files under the assumption that these files are secure and that their customers' privacy is respected. The release and acquisition of a great deal of duplicate data by other users is unfortunately made possible by the large number of these microdata interactions. These users can then compare a variety of connected datasets to locate records in common and learn new information about a person, breaching that person's privacy. People who work in the big data economy may be seriously threatened by this, which may lead to the leaking of private information to third parties against the will of the original data source. To assist in resolving this issue, our team at DPE has developed a solution that is simple to incorporate into your current program.

A more recent area of machine learning and data analytics called Differential Privacy is expanded upon by our solution, the Differential Privacy Engine. Professor Green at John Hopkins describes the field, sometimes known as DP, in the following way: "One way to look at this is that DP offers a means of determining if your data significantly affects the result of a query. If not, you might as well add information to the database. In other words, the result of using differential privacy demonstrates that including or excluding oneself will have no impact on the statistics and discoveries drawn from a dataset or when comparing it to other datasets. This can only be done by using DP, which serves to emphasize the message. In what sense does "applying" Differential Privacy on a dataset make sense? In the simplest terms possible, it involves utilizing a little amount of noise to generate random fields inside a dataset that may be generated from the entire dataset or from another source. This ensures that the data, from a statistical perspective, doesn't vary too much and that you can't guarantee the reliability of any one record.

To give a straightforward form of DP that can be likened to a coin flip is the goal of the program we are demonstrating. With the first huge client base you have, this strategy will scale the most easily and is the most researched variation of differential privacy. When you enter a dataset into the program, it prompts you to choose which fields (also known as column values) you want to apply DP to. A coin flip follows each iteration of each value. If the coin comes up heads, the value is transmitted through unmodified; if it comes up tails, a new coin flip is conducted. If this coin falls on heads, the value is sent through unaltered; if it lands on tails, the value is adjusted by taking a column from the same column and replacing it with the original value. With a bottom and upper constraint of 3 percent, this ensures that no more than 25% of the data on this value set will change. On the first chosen dataset, the coin flip approach previously mentioned is used. You may also choose another value (column) and apply this technique to it. However, the coin toss is buried one step deeper for the second application. This causes those data values to change by 12.5% with a lower/upper bound of 1.5%. These modifications have a negligible impact on the total amount of data when examining a data collection with several column values. In the total data set, the average change from our original testing was around 6%.

We recognize that you might be hesitant to use our application or trust us. But be assured that we only contacted you because we trusted your moral dedication to protecting client privacy. Our business is certain that merging our services would be to your great advantage since you will be the first service provider to advocate for privacy and security on both the client and consumer markets. Our company has developed a solid reputation for producing dependable software that prioritizes morality and security. Additionally, giving data intrinsic privacy has tremendous ethical significance. The firms that will obtain someone's email and address should not be a concern when they provide this information. Finally, data privacy offers a clear answer to many legal gray areas that have not been fully investigated since these industries are still young.

The demo we have supplied will need to be scaled up and reorganized to be more dynamic with the data it can absorb to receive funding. This will need splitting the project into two phases: first deployment and a significant feature upgrade. To give our developers with the necessary gear, the first phase will cost $150,000. Based on the current design, this phase should provide a usable application in 120 development hours. 160 development hours and an extra $150,000 are needed for the second phase. Due to our conviction that the initial rollout will immediately provide $200,000 in net revenue, the needed funds can be lower in comparison to the required development hours. From the chosen start date, we predict that the combined time needed for the two stages will be seven weeks.

We look forward hearing from you soon!

I wish you well,

Differential Privacy Engine CEO Joshua Novak

Summary of Project Recommendations in A.2

A.2.1 Problem Synopsis

Other organizations can communicate encrypted data via a network or via email with the aid of the software solutions offered by Data Solutions Inc. The issue with their present technologies is that they are unable to regulate staff interactions for their clients. In addition, they lack a device to prevent client data discovery. Data Solutions Inc will become a full-service data-security supplier by including the Differential Privacy Engine into its current software plans, which will assist in resolving this issue.

A.2.2 Applications' Advantages

A little amount of noise is added to an existing dataset using the DP Engine. The aim of this noise addition is to undermine the dataset's absolute validity. Without this validity, it is impossible to ensure the accuracy of any data discovery performed on the dataset. The more data that can be manipulated in the dataset, the more effectively this program operates. For the persons or things that each data entry is associated to, total privacy is feasible after DP has been implemented. This is due to the lack of a statistically significant difference between a person adding or excluding himself from the data set. Utilizing this application has the advantage of assisting in removing culpability for data breaches caused by insecure transmission or interactions with other parties. Daily movement of enormous amounts of data makes it quite likely that data sets may be hacked. Applying the DP engine before transferring the data assures that consumer data is being protected. Customers will gain a lot from this since it will reduce data breaches and boost consumer confidence. Finally, this application may assist in altering choices made in relation to selling data or comprehending the veracity of acquired info.

A.2.3 Data Product Overview

With the aid of a few third-party libraries, the DP Engine was created in the Python programming language. We particularly use Python 3.7, Qt Designer, PyQt5, PyQt Graph, and the standard Python libraries. Despite being created in PyCharm, this version of the program may be run and modified in any Python-compatible IDE provided the necessary libraries are present and included. The extensive selection of tools provided by the PyQt5 library and related resources led to their selection. The program allows for the real-time adjustment of the data graphing UI tools. Due to the extensive data statistics resources that are already integrated into the Python language, it was chosen for this project. The application is hosted locally on a development computer, but it may be packed and sent as a zip file to any client to be executed locally. Four CSV files, which serve as consumer data for the application to use in displaying the DP Engine's power, are included with the application files. The next section goes into further depth about those files.

A.2.4 Information Used to Produce the Data

4 comma-separated value files hold the data needed for this application (.csv). Regarding the dataset each file represents, it contains a variety of information. Employee data and corporate data are the two main datasets that have been used in our example application. There are two CSV files related to both the firms and the personnel. Two CSV files reflect the updated records once the DP Engine is applied, while one file represents the original or unaffected data set. 1,000 records per file are generated based on sample data we created for the program demo and are contained in each file. A primary key value that corresponds to the employee id is contained in the csv file for employee records. In addition, the first name, last night, email, gender, and work position are included. You can choose to adjust the first name, last name, or email in the DP Engine. A change of 25% or 250 entries will be made to the first value, and a change of 12.5% or 125 entries will be made to the second choice. The company records contain a primary key for the company id as well as values for the name of the firm, the nation of origin, the company worth, and a hyperlink to their website. The DP Engine can apply and change the data for business name, country of origin, and company value. The firm data also has the same percentage and bound ranges.

A.2.5 Purpose and Theory

The main goal of this program is to give huge datasets an acceptable level of anonymity. The provision of quick and effective software that can easily interface with current systems is a secondary goal. In order to limit or ultimately prohibit unauthorized data discovery, it is hypothesized that we can add roughly 6 percent noise to all datasets. Due to the highly relatedness of the datasets, this is almost hard to accomplish. Two numbers, for instance, that the DP Engine does not introduce noise into may be compared across both vast databases of comparable data. If you are confident that one is right, you may compare individual records to identify which values had noise applied to them and restore most of the dataset to its initial condition. The sample program does not take this scenario into account because it is extremely improbable that this exacting data exchange would ever take place in actual settings.

A.2.6 Approach

The Agile methodology is actively used and promoted by DPE and Data Solutions Inc. The success of this application depends largely on continuous testing and feature integration. The project is divided into two stages so that we may launch a customer-facing product as soon as feasible. The second phase's emphasis will be on adding new features and fixing bugs. In addition, we will immediately give the software demo to important suppliers so they may start receiving training, and we will establish channels for feedback. The second phase deployments will consider each and every piece of feedback that has been received. The utilization of sprints is yet another key factor in our methodology's adoption of the Agile/Scrum framework. To complete the first phase scaling as soon as possible, we will need to divide the important application points into three distinct sprints. The first sprint will be primarily concerned with turning the program into executable software and adapting its static structure to accept any correctly structured CSV file. The second sprint's main goal will be to incorporate the application into the current software systems. Application testing and rollout will be the last sprint of the first phase. This final stage will be crucial since we want to create an application that can constantly sustain 50 users.

A.2.7 Required Funding

The Letter of Transmittal outlines the financing conditions for the proposal, but we'll repeat them here. Each piece of software and equipment needed to create the application is free and open source. Python and its companion libraries fall under this category. However, funding will be needed to cover the costs of staff wages, server space, and development time. There will be two $150,000 funding installments needed for the program. Each installment must be delivered before the start of each phase. In addition, we anticipate that the deployment of the first phase will result in a net profit of $200,000. The second phase's necessary budget should fully account for this income, increasing the total amount of funds needed to $500,000. If we produce less revenue than expected, further financing will have to be added on top of the second installment.

Impact of the Solution on Stakeholders (A.2.8)

A successful merger requires complete consent from the stakeholders of both businesses. There must be constant progress updates and open communication. For the project to be judged successful by the stakeholders, it must run smoothly and adhere to the moral standards that we have built our businesses on. The program must furthermore provide the amount of privacy and noise described in the guidelines. Finally, the application's main success will depend on how much money it brings Data Solutions Inc. Once it is made available to all current users, we predict that the application will be able to bring in $2 million in additional revenue. We calculated that if the program could be transferred and offered as a standalone utility, it could bring in an extra $4 million.

A.2.9 Moral and legal issues to think about

There are significant ethical issues that need to be evaluated and approved when working with any form of sensitive data. First off, we won't keep any of the information that clients pass via the DP Engine. We will not let businesses to redistribute, reuse, or provide our services in the future. This will enable us to make sure that no unauthorized data collecting is taking place. From a legal standpoint, we must make sure that none of our clients are selling data that should be protected by HIPPA or any other federally imposed legislation by modifying it. As a publisher, it's crucial to make sure that our clients share our commitment to moral principles. We won't permit the application or software rights to be bought altogether to guarantee this. Instead, we will provide a subscription-based plan for access to the program. We have the right to revoke a customer's contract if we believe they are acting unethically or in violation of our policies to stop any further abuse.

A.2.10 Developer Proficiency

A group of software engineers that are quite new to the field will be working on this project. Two distinct software specialists will be present to assist project development and serve as scrum masters. The main reason for this is that in order to fulfill the project date we have listed above; we will need to increase the number of people working for us. This, in our opinion, is an essential step so that we can educate a new, well-prepared team of engineers to manage the program as demand grows. Our junior developers will use this project as their first opportunity to learn and to gain the necessary skill set to work on more complex projects.

B. A proposal for a project for IT experts

Issue Statement in B.1

Internal or private data breach brought on by employee carelessness is one of the major hazards facing modern businesses. The interchange of data is constant throughout an individual's day in the world we live in. Email, social media, search engines, and a lot more are just a few of the tools that interact with data. Although these technologies represent a major advance in modern technology, their use reveals a shocking level of carelessness on the part of its users. Consumers' carelessness regarding their personal data is negligence. Any time without their awareness, tools like location tracking, microphone access, and others can collect information about a person. Sadly, a lot of this carelessness is transferred into the job when workers encounter private information.

Due to how challenging it is to monitor employee data usage and everyday interactions, this poses a significant challenge for security and data companies. Although you may utilize methods like forbidding smartphone usage at work, limiting the websites you can access, and recording all user behavior, these breaches still happen. But by utilizing the DP Engine, you can at least lessen the harm that results from data being stolen or utilized improperly. Filtering the data and introducing noise to the dataset produces this mitigation. The precision provided by this is still quite good while also assisting in preventing the discovery of relevant data. Data will be protected on several levels by adding this tool to the current set of security tools. It will also provide us the money we require to develop this program further and produce other technologies to guarantee data protection.

Description of the client and advantages

The clientele will be made up of all current clients of Data Solutions Inc. The ability to sell this as a standalone application allows for expansion beyond current clients. Above was a list of the advantages a DP application offers to users. Here's how this will satisfy the wants of the clientele:

1. Clients strive for success in the Big Data industry. Even though many people view the sale of data by firms as unethical, many will want to explore this option. Due to its role in preventing unauthorized data discovery and accusations that they are selling customer information; the DP Engine helps businesses resolve their ethical dilemma.

2. Customers need a solution that is straightforward, simple to use, and adaptable to their data. These functionalities are offered by the DP Engine, which doesn't need any special training or technical know-how. The program may also relate to other programs to automatically apply varying levels of privacy to businesses and their staff.

3.) Clients and vendors can be informed that a system is being used to assist secure data.

Integration of Current Systems (B.3)

The whole set of software tools needed for this project has already been in use and is available to the public as open-source software. Python-based, the code is flexible enough to be used in any setting. The tool has two options for integration: it may be used independently or as a port of an existing software.

B.4 Data Are Needed

For the application to handle any meaningful data, the four data CSV files used in the example program must be present. These CSV files were created at random using a tool made available by www.mockaroo.com and represent customer datasets. Company and personnel datasets make up the two main dataset categories. For unmodified data, each dataset contains a file; for changed data, a different file is used. We use the CSV file format since it is the most popular method of storing data and is regarded as a standard across most database programs (section A.2.4 contains the structure of each file). Furthermore, CSV files can often be plugged into most applications and are short and simple to interpret.

For the program to function with any CSV dataset, the current demo must be expanded. Python's built-in CSV reader/writer is used to open and edit the file. Our process involves parsing the column values before hashing them. The 10 hash buckets are used to categorize these hash values according to their value. To read column values and identify which values apply to noise, the current application will need to be modified. The values in the column need to be highly selective and contain the majority of unique values. Then, we will choose a random sample from that column dataset made up of 10% of the entire data and apply noise to that sample. The data that have been chosen to add noise will subsequently have their current value replaced with a random value drawn from the 10% dataset when the process is repeated for each value. By doing this, the dataset is guaranteed not to include any new or unidentified data values.

It should be emphasized that there are other applications for differential privacy, and that this approach does not produce "real" privacy. It has been researched and found to significantly increase data protection compared to not applying any noise to a dataset, although it is a strong beginning point. The fact that we're attempting to provide data privacy and that this provides us with a foundation upon which to expand is what matters in this situation.

Project Management, Section B.5

The Agile Scrum Methodology will be used by our team to create the product, as was already mentioned. Since we believe that constant change and improvement will occur, we choose the Agile development technique. DP Engine and Data Solutions Inc have both previously used this framework as well. Finally, to guarantee that the project is completed by the deadline, we must create and use the sprint tools. The workflow procedure is summarized below:

B.5.1: Requirements

- Our application must offer a level of privacy for each dataset that is within our allowed tolerances (generally, a dataset with 6% more noise).

- We have ethical standards we uphold for our application, and we make sure our vendors do the same.

- To create an application that users are satisfied with, we must take into account all client input and make adjustments as necessary.

- Stakeholders should be discussed and added on the change board, but they should have confidence in their application.

B.5.2 - Development - We will divide the project into two stages, each of which should take a total of 370 development hours over the course of seven weeks. This estimate includes one additional week for backlog reduction and troubleshooting, making the entire project development duration 2 months.

- The existing application will be ported during the first stage, and a prototype will be released to the public. It should take three weeks to finish this phase, which will consist of three distinct sprints.

The release of feature updates and the application of bug-fixing patches will be the main goals of the second phase.

B.5.3 Testing: At the beginning of the project, we will distribute the demo version of the DP Engine to our important clients so they may use it and give comments. This initial consumer-facing and black box testing will be regarded as such.

The first phase will consist mostly of unit and integration tests. Two weeks will be allotted for white box testing after one week of development time is spent transitioning the program.

- The second phase's primary focus will be on unit and special case testing while collecting all the data from the first testing efforts.

Delivery is a feature of B.5.4. After the first phase, the project will be delivered initially. As soon as stakeholders give their approval, we will launch our second phase, which will then be considered finished.

B.5.5 - Input - We will gather feedback from stakeholders, first-time customers who receive demo deployments, and customers who buy the application after the first phase deployment. This feedback will be included into the second phase.

B.6 Deliverables for the Project

One product, the final DP Engine application, will be the project's delivery.

Plan of Execution for B.7

The two phases, the release of the prototype, and the completion of the application are the main topics of the implementation plan. The application will be made available to consumers at the conclusion of each phase, together with any related materials. A short training video series and two weeks of introductory 24/7 assistance make up these materials. To preventing anyone from training on a production dataset, we will additionally supply a set of training data.

We shall limit the code repository's accessibility to only those on the designated development team in terms of access to the program. As a result, no additional DP Engine or SES staff members will be able to test software in public spaces or make code base modifications. As a result, no developer will be able to push changes without permission, helping to maintain effective change control management. Finally, clients will have to save the data locally to plug it into the DP Engine because we will not support any data storage through our current software. In order to correctly alter the data, we will, if necessary, give a program to convert.DB files or.SQL files to.CSV files. It must be simple to use and capable of converting data back to the original file format.

Evaluation Strategy B.8

The program's integration into routine everyday activities will be the focus of the assessment plan. Surveys must be made available, and super users must offer information on statistics such how frequently a user accesses the program each day. We must also keep track of use cases and make sure that the right data is being entered into the DP Engine if we are to be certain that our solution is offering practical data security. As was previously stated, the software should offer a useful function without affecting sensitive data that could conflict with HIPPA or other data compliance. We also need to make sure that the noise that is introduced to datasets doesn't contaminate or erroneously extend the data. We must first monitor use, data, and noise statistics for users of the product in the early stages of deployment if we are to complete these responsibilities. It is necessary to transport and store this data securely locally on the client side. This data ought to be deleted once we're sure the program is operating as intended, and stakeholder sign-off ought to start then.

Finding as many flaws and data inconsistencies as we can be the initial aim of our testing. Our testing efforts will then shift to ensuring a degree of data privacy is maintained after the number of mistakes decreases to an acceptable range and deployment is finalized. To do this, we will periodically request client reports on data statistics that don't give out any sensitive information to us. For next releases and upgrades, our team will examine these statistics.

B.9 Programming Environments and Associated Fees

Python 3.7 as well as the standard Python libraries, PyQt5, Qt Designer, and PyQt Graph make up the programming environment.

- A tool for creating CSV files from any form of common database file.

- Tracking program for statistics to monitor use data during the initial deployment

The programming environment is free to use because all these tools are open source and are thus all available for download.

cost to the environment

For the deployment phases, we will require workstations and storage equipment. To start developing, testing, and gathering data analytics, our developers will require workstations that can connect to a central network.

The environment is a line item in our budget for $30,000, and the first installment payment will cover these expenses.

We need a team of six developers to finish the project, thus human resources are needed. A monthly compensation of $5,000 will be given to each of them. Given that the project's estimated completion time is two months, the wages of the employees should account for a total of $60,000 of the entire budget.

- Phase one (Deployable application): 120 development hours - total expense: $150,000 (this includes the equipment cost and the first month salary payments).

The second phase, which consists of "Major Feature Updates and Bug Fixes," will require 160 development hours and $350,000 in total. The following expenses contribute to this phase's higher costs:

More developer hours, Deployment expenses to bring the product to clients, hiring external testers, increasing hardware needs to grow the program, A reserve for unforeseen expenditures

B.10 Timeline and Milestones

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event | Start Date | End Date | Developer hours required | Dependencies | Assigned Resources |
| Project Start | June 1st, 2022 | June 1st, 2022 | 0 | None | Project Manager, Stakeholders |
| Phase One | June 1st, 2022 | June 22nd, 2022 | 100 | Project Start | Project Manager, Developers |
| Scaling and integrating projects for consumer usage | June 1st, 2022 | June 8th, 2022 | 20 | Project Start | Project Manager, Developers |
| Application deployment to a small number of clients and integration testing | June 8th, 2022 | June 22nd, 2022 | 70 | Scaling and integrating projects for consumer usage | Developers |
| Deployment of the first DP Engine to customers | June 22nd, 2022 | June 22nd, 2022 | 10 | Application deployment to a small number of clients and integration testing | Project Manager, Developers |
| Phase Two | June 22nd, 2022 | June 26th, 2022 | 160 | Phase One | Project Manager, Stakeholders, Developers |
| First major feature update | June 29th, 2022 | June 5th, 2022 | 20 | Phase Two | Project Manager, Developers |
| Second major feature update | June 5th, 2022 | June 19th, 2022 | 40 | First major feature update | Project Manager, Developers |
| Final bug review and stakeholder signoff | June 19th, 2022 | June 26th, 2022 | 20 | Second major feature update | Project Manager, Stakeholders, Developers |

Documentation for the Developed Product

D.1 Business Needs and Project Objectives

The paper includes a list of the business needs. To summarize, to accomplish the project, we will utilize a variety of open-source libraries and tools in addition to the Python programming language. For the project to be finished within our eight-week deadline, we will also require a total of $500,000 in funding. The Differential Privacy Engine, our contract's deliverable, needs stakeholder clearance before moving further.

For the application, the main requirement is to develop a user-friendly program that can fit into current software systems and offer data privacy. The issue that this need attempts to solve is the everyday access to sensitive data by our client's workers, who also frequently cause data breaches. Furthermore, there is an issue with data discovery on customer datasets that betrays customer trust. By assisting Secure Enterprise Solutions in the development of a multi-facing software solution, we can address these challenges of data theft and privacy. This solution will introduce our privacy engine at the data level while leveraging their current security software architecture. Customers receive this service to let them shoulder less moral and financial responsibility while handling personal data.

D.2 Clean and Raw Data

The information needed for the application is correctly structured.

Data CSV files that must be fed into the DP Engine. Utilizing www.mockaroo.com, sample CSV data for the demo was produced. You may see these CSV files by accessing the relevant files inside the program files.

The two files Read Company Data.csv and Read Employee Data.csv contain data that is regarded as the raw datasets. This is the raw data that will be inserted into the DP Engine. The program is launched, a dataset is chosen for operation, and two fields are chosen to receive noise. You may examine the dataset with DP applied by looking at the Write Company Data.csv and Write Employee Data.csv files when this has been completed. The DP Engine must have been run on each of these two CSV files' respective datasets for data to be present; otherwise, the files are empty by default. A sample of a raw dataset is shown in Figure D.2.1, and a sample of a cleaned dataset after differential privacy has been used is shown in Figure D.2.2. You'll see that the values for rows 3, 7, 8, 11, 15, 16, 17, 18 have been updated to reflect the first name and the same thing can be done for last name and email.

Text

Description automatically generated

Figure D.2.1 – Raw Read\_Employee\_Data.csv file

Text

Description automatically generated

Figure D.2.2 – Cleaned Write\_Employee\_Data.csv file

In order to filter and apply noise to the algorithm, the DP Engine uses a coin flip method (see figures D.2.3 and D.2.4). Figure D.2.3 illustrates the lowest and maximum coin flip values for the first data column value you select. Figure D.2.4 illustrates the lowest and maximum coin flip values for the second data column value you select. You may find the employee records window.py file where the example code for this sentence is stored. The company records window.py file has similar techniques that are used for business data.

Text

Description automatically generated

Figure D.2.3 – First Data Filter Algorithm (two-coin flips)

Text

Description automatically generated

Figure D.2.4 – Second Data Filter Algorithm (three-coin flips)

Analyzing the D.3 Code

Using PyQt5, we have built a GUI environment to help users comprehend the data and how the privacy filtering affects the dataset. To provide predictive and descriptive data analysis, the GUI offers three key capabilities. Here is an explanation of these tools:

Statistical Analysis:

1. A summary of the dataset's noise application

You must choose the "Apply DP to Employee Records" or "Apply DP to Company Records" option in the application to apply privacy filtering. When you are in the right menu, you may apply privacy filters and see the consequences of all dataset modifications. This example may be found in the employee records window.py, and the company records window.py contains an identical version for businesses. Figure D.3.1 provides an illustration.

Graphical user interface, text, application

Description automatically generated

Figure D.3.1 - A dataset's privacy filtering percentage

2.) Diagrams of a scatter plot that display the variation on a column value field over the whole dataset.

The scatter plots are intended to demonstrate the distinction between employee first names or business names. The red line shows the initial values, while the blue line shows the modified values. Alphabetical letters are represented by the digits 0 through 26 on the x-axis (A is equivalent to 1 and B is equivalent to 2). Remember that the blue line will only appear if DP was applied on a name field in the relevant dataset. The company scatter plot.py has this example, and the employee scatter plot.py contains a similar instance for businesses. Figure D.3.2 provides an illustration.

A screenshot of a computer

Description automatically generated with low confidence

Changes in firm names between the two datasets are shown in Figure D.3.2.

A database view of the provided Dataset is one of the Descriptive Tools.

You can examine the unmodified and edited datasets using the database view that we provide. This enables you to determine whether the DP Engine's modifications were worthwhile and whether a particular column has sufficient selectivity to support DP. Additionally, it lets you manually alter data values to improve dataset optimization. For instance, you can manually update a few rows to increase selectivity in the unusual case when the DP Engine updates numerous sequential rows to the same value. Last but not least, the primary screen where this function is accessible serves as the DP Engine's dashboard. From this page, you may see or modify any set of data. The employee records original window.py file contains this example. For an illustration, see figure D.3.3.

Graphical user interface, table

Description automatically generated

Figure D.3.3 shows the main menu's unchanged employee dataset at the moment.

D.4 Verifying Hypotheses

Our original assumption was that we could build a user-friendly data product that would be simple to connect with current applications and offer a fundamental degree of privacy screening. We feel our hypothesis was true because the visual data we have supplied above and the fact that our findings remained within projected boundaries after countless unit tests. Our findings indicate that each dataset performs a measurable amount of privacy filtering and will display about 6% of the overall noise. However, our existing application cannot provide absolute and total privacy. Although not the project's explicit objective, this ought to be the end goal. More investigation and testing will be required in order to achieve this aim. As of right now, the goal and intent of this initiative are consistent with our expectations and plans.

Effective Visualizations and Reporting, Section D.

Section D.3 provides a more thorough summary and description of the program's visual components. These visual tools demonstrate significant dataset modifications brought about by the DP Engine's data filtering. The data can be shown graphically or as a distribution in a graph. The amount of noise that has been introduced to each field and the dataset can also be seen. These data visualizations aid in giving the data meaning and demonstrate that the privacy filtering is effective.

With regard to reporting, we needed to make sure that our application was portraying a random distribution. The order in which the rows should have been chosen for filtering and the value with which the noise should be applied should have been random. Two layers of randomization should be used to further this idea:

1.) Random selection of the choose values is required while filtering a data value.

2.) The noise value that substitutes for the original value should also be random for the random values that are chosen.

For the concept of real data privacy, randomization is crucial, but we also need to impose restrictions to make sure the randomness is directed and connected to the dataset. To do this, we used a restricted range for random values that might be chosen based on the order in which they are chosen. We also verify that the noise replacement values are included in the original dataset and fall within a range of the first 100 values.

D.6 Accuracy Evaluation

As said in the description, the project's goal is to build a baseline for privacy filtering by applying significant randomization to a dataset. With the example application that has been given, we have generated and rand over 300 use cases while maintaining the specified boundaries. Below is a description of these bounds:

First data filter: This group should have 25% of the data modified. This has a 3 percent boundary on either side, giving it an effective range of 22 to 28 percent.

A second data filter should modify 12.5% of the data in this group. This has a 1.5 percent limit on either side, giving it an actual range of 11 to 13.5 percent.

Total records filtered by the data: After merging the first and second data filters, six out of the data in the entire data set should be modified. This range has an effective range of 5-7 percent because it is also constrained by 1 percent in each way.

D.7 Testing of Applications

The product's main component is application testing. The two lifecycle stages that we have just covered serve as a basis for the testing phases.

Initial deployment testing is the first phase.

We gave super users and certain clients access to a prototype and program demo at this time. Users can start entering their own datasets into the prototype after first using the software demo for training. Companies and testers are requested to submit bug reports within this two-week period for our assessment. During this time, end users are performing the majority of the testing, hence very little white box testing is done.

Second stage: feature update review

The second stage will see us gathering user input and starting white box testing. As we move forward, we'll establish three main categories for issue reports, which we'll then add to the changelog. The three categories are as follows: errors with dataset input, errors with reports outside of specified boundaries, and issues with product setup. Most of the changes will be made here.

We will also provide demos to stakeholders for final approval in addition to the testing mentioned above. Below is a tracking report for the problems that were reported each week in phase 2.

Applications D.8

You can find the files to run the program in the submission. An IDE environment that supports Python 3.7 is required to run the project. Installing PyQt5 and PyQtGraph libraries is necessary for the application to work properly. Numpy, csv, random, string, and counter from the collection's library are other Python libraries that you need additionally include.

The following is a description of every file in the repository:

The primary application window is represented by the file employee records original window.py, which is what should be executed to launch the program.

For data manipulation, the employee hash map.py file constructs buckets and a hash map for employee records.

The Read Employee Data.csv file is processed using the DP filtering method included in the employee records window.py file, and the employee scatter plot.py file provides the scatter plot used to compare employee records that have been edited and those that have not.

Company records window.py is where the DP filtering method is present and is applied to the Read Company Data.csv file. Company scatter plot.py is where the scatter plot for examining the unmodified and edited company records is contained.

Using the csv file format, Read Employee Data.csv contains the unmodified employee records.

The updated employee records are contained in Write Employee Data.csv, a file with the csv file extension.

Read Company Data.csv is a file in the csv file format that contains the original company records.

The changed company entries are stored in Write Company Data.csv, which uses the csv file format.

Please see the user's manual below for more assistance.

Manual for D.9

1.) Uncompress the related files and put them in a project folder.

2. Confirm that Python 3.7 is installed on the desktop from which you are accessing the software. Access your local command prompt and type "python -version" to do this. Please go to the advanced troubleshooting section below if you don't have Python installed.

3.) Open up that folder with an IDE that can handle Python 3.7. (We recommend using JetBrains Pycharm Community Edition 2019.1.3).

4. Download the PyQt5 and PyQtGraph libraries, then install them.

5.) Open the program by right-clicking the employee records original window.py file and choosing "Run."

6.) Navigate to the employee records original window.py file. To execute the file if you can't do it that way, hit F10 while holding down the shift key.

6.) To view datasets, choose one of the top four buttons after the program has launched. One helpful hint: because no data has yet been filtered and written to the updated files, they will first seem to be empty.

7.) Apply privacy filtering to a dataset using the button 4 buttons, or compare the datasets' scatter plots, by using the button 5. One helpful hint: if the privacy filter has not yet been applied, the scatter plot data will only display the unmodified records.

The employee records original window.py's bottom four buttons will launch new windows, so take note of that. The red X in the upper right corner of the screen must be clicked to close these windows.