# C964 Computer Science Capstone

Ryan Hildebrant

Western Governors University

Signature:
Ryan Hildebrant

Date:
[04/15/2020]

## Table of Contents

**A.1 Letter of Transmittal – Secure Enterprise Solutions** 4

**A.2 Project Recommendations**

A.2.1 Problem Summary 6

A.2.2 Application Benefits 6

A.2.3 Outline of the Data Product 6

A.2.4 Data Used in the Data Product 7

A.2.5 Objective and Hypothesis 7

A.2.6 Methodology 7

A.2.7 Funding Requirements 8

A.2.8 Impact of the Solution on Stakeholders 8

A.2.9 Ethical and Legal Considerations 8

1.2.10 Developer's Expertise 9

**B. Project Proposal for IT Professionals** **10**

B.1 Problem Statement 10

B.2 Customer Description and Benefits 10

B.3 Existing Systems Integration 11

B.4 Projected Timeline 11

B.5 Resources and Costs 11

B.6 Outcome 13

B.7 Implementation Plan 13

B.8 Evaluation Plan 13

B.9 Programming Environments and Related Costs 14

B.10 Timeline and Milestones 15

**D. Developed Product Documentation 16**

D.1 Business Requirements and Project Purpose 16

D.2 Raw and Cleaned Data 17

D.3 Code Analysis 19

D.4 Hypothesis Verification 22

D.5 Effective Visualizations and Reporting 22

D.6 Accuracy Analysis 22

D.7 Application Testing 23

D.8 Application Files 24

D.9 User’s Guide 25

**D. Sources 26**

## A.1 Letter of Transmittal

April 13, 2020

Raymond Singer, Owner & Operator

Secure Enterprise Solutions

14412 Johnson St

Salt Lake City, UT 84107

Dear Mr. Singer,

In the past six months, Secure Enterprise Solutions has become the go-to platform for small businesses that need to secure their data transmission. While secure data transmission is critical to all companies, it does not account for all security breaches found within small businesses. In recent years, the expansion of Big Data has spiked the field of data discovery. While data discovery has many useful purposes, it also opens many doors for malicious practices. As a result, many businesses are exchanging their data records, assuming that these records are secure and that their client privacy is protected. Unfortunately, massive amounts of these microdata transactions allow a lot of duplicate data to be released and captured by other users. These users can then compare many related datasets to find common records and discover new things about an individual, thus violating the privacy of that person. This can be a severe threat to people in the big data economy and can result in data being released to other companies that the original data source did not consent to. Thus, our team at DPE has created a solution to help solve this problem that can be easily integrated into your existing software.

Our solution, the Differential Privacy Engine, expands on a newer field of machine learning and data statistics, called Differential Privacy. The field, also known as DP, is described here by Professor Green at John Hopkins: “One way to look at this is that DP provides a way to know if your data has a significant effect on the outcome of a query. If it doesn’t, then you might as well contribute to the database”. In other words, the effect of applying Differential Privacy shows that the statistics and discovery generated from a dataset or comparing other datasets will not change if you include or exclude yourself. To further this point, this is only achievable by applying DP. So, what is “applying” Differential Privacy to a dataset? Put very simply; it is using a minimal amount of noise to random fields in a dataset that can be derived from the overall dataset or another source. This guarantees that the data, in terms of a statistical view, does not change very much and that you can’t guarantee the authenticity of a single record.

The objective of the application that we are presenting is to provide a simple form of DP that can be compared to a coin flip. This approach is the most studied variant of Differential Privacy and will be the easiest to scale with the large client base you have initially. The application works by taking in a dataset then asking which fields (aka column values) you would like to apply DP too. Each value is then iterated through, and a coin flip occurs. If the coin lands on heads, then the value is passed unaltered; if it lands on tails, then another coin flip occurs. If this coin lands on heads, then the value is passed unchanged, if it is tails, then the value is altered by pulling another column from the same column and replacing it with the original value. This guarantees that ~25% of data on this value set is changed with a lower & upper bound of 3%. The coin flip method described above is applied to the first selected dataset. You can also select a second value (column) to use this algorithm too. However, for the second application, the coin flip is nested one level deeper. This results in those data values changing ~12.5% with a lower/upper bound of ~1.5%. When looking at a data set with many column values, these changes result in a minimal change to the overall data. Our initial tests have averaged about a 6% change in the entire data set.

We understand that you may be wary of us and our application. Still, I assure you we have exclusively reached out to you because we believe in your ethical commitment to customer privacy. Our company firmly believes that combining our services will be of great benefit to you because you will become the first provider on the market to promote privacy and security on both the customer and consumer markets. Our firm has created a strong reputation for building clean and efficient software that puts a top priority on ethics and security. Additionally, there is great ethical value in providing inherent privacy to data. When someone submits their email and address, they should not be worried about what companies will get their hands on this data. Lastly, data privacy is a clear solution to many legal grey-areas that have not been thoroughly explored due to the freshness of these fields.

In terms of financing, we will need to scale the demo we have provided and restructure it to be more dynamic with the data it can accept. To accomplish this, we will need to break the project into two phases: initial deployment and a major feature update. The first phase will require $150,000 to provide the hardware needed for our developers. This phase should take 120 development hours and will provide a working application based on the existing architecture. The second phase will require an additional $150,000 and 160 development hours. The required money can be less comparative to development hours because we believe the initial rollout will generate an immediate $200,000 in net income. We estimate that the total time requirements for both phases will be seven weeks from the determined start date.

We look forward to hearing from you soon!

All the best,

Ryan Hildebrant, CEO DPE

# A.2 Project Recommendations Summary

## A.2.1 Problem Summary

Secure Enterprise Solutions offers a suite of software tools that help provide other businesses the ability to send encrypted data over a network or through email. The problem with their current tools is that they do not have a way to control interactions of their client's employees. Furthermore, they do not have a tool to protect client data discovery. Adding the Differential Privacy Engine to its existing software plans will help solve this problem and make Secure Enterprise Solutions a full-service data-security provider.

## A.2.2 Application Benefits

The DP Engine takes an existing dataset and applies a small amount of noise to it. The purpose of adding this noise is remove the absolute validity of the dataset. Without this validity, any data discovery that is done on the dataset cannot be guaranteed to be accurate information. The more amount of data available to manipulate in the dataset, the better this application works. Once DP has been applied, complete privacy is possible for the people or objects that each data entry is related to. This is because there is no significant statistical difference from any one individual including themselves or removing themselves from the data set. The benefit of using this application is that it helps remove liability for data that is comprised as a result of unsecure transmission or third-party interactions. Massive amount of data is moved around every day and the likelihood of data sets becoming compromised is very high. Applying the DP engine before transmitting the data ensures that there is an attempt to protect customer information. This will greatly benefit customers by mitigating data breeches and growing consumer confidence. Lastly, this program can help with changing decisions related to selling data or understanding the validity of purchased data.

## A.2.3 Outline of the Data Product

The DP Engine is written using the Python programming language with the use of a few external libraries. More specifically, we are using Python 3.7 along with Qt Designer, PyQt5, PyQt Graph, and default Python libraries. This instance of the program was written in PyCharm but can be run and edited in any Python supported IDE if the proper libraries are installed and included. The PyQt5 library and associated resources were chosen because of the wide variety of tools it offers. The data graphing UI tools can be seen through the application and can be adjusted in real time. The Python language was chosen for this project due to the large amount of data statistics resources that are built into it. The application is locally hosted on a development machine but can be packaged and zipped out to any client as a locally run program. Included with the application files are 4 CSV files that represent consumer data the application uses to display the power of the DP Engine. There files are discussed in greater detail in the following section.

### A.2.4 Data Used in the Data Product

The data being used for this application is stored in 4 comma-separated-value files (.csv). Each file contains different information pertaining to the dataset it represents. There are two primary datasets that have been used in our demo application: employee data and company data. Both the employees and companies have 2 associated CSV files. One file represents the unaltered or original data set and the other two CSV files represent the altered records after the DP Engine is applied. Each file contains 1,000 records that have been generated based on sample data we have generated for the purpose of the application demo. The employee records csv file contains a primary key value that represents employee id. It also includes first name, last night, email, gender and job title. The DP Engine allows you to select either the first name, last name, or email to apply changes to. The first value will have ~25% or 250 entries changes and the second selection will have ~12.5% or 125 entries changed. The company records include a primary key that represents company id and values for company name, country of origin, company value, and a link to their website. The company name, country of origin, and company value fields can be applied and updated by the DP Engine. The same percentage and bound ranges exist for the company data as well.

### A.2.5 Objective and Hypothesis

The primary objective of this application is to provide an acceptable range of privacy to large datasets. A secondary objective is providing fast and efficient software that can seamlessly integrate with existing systems. The hypothesis is that we can provide about 6% noise on total datasets to reduce or ultimate prevent unwarranted data discovery. This is nearly impossible to achieve because of closely related datasets. For example, if you have two large datasets of similar information, two values can be compared across both datasets that the DP Engine does not provide any noise to. If you know one is accurate then you can compare individual records and determine what values have noise applied to them and correct most of the dataset back to its original state. This scenario is not considered in the demo application because it is highly unlikely that this rigorous data exchange will ever occur in real-world scenarios.

### A.2.6 Methodology

DPE and Secure Enterprise Solutions both actively use and encourage the Agile framework. Continuous testing and feature integration are a key factor to the success of this application. The project is broken down into two phases so that we can get a consumer facing product on the market as soon as possible. Then the second phase will focus on feature rollouts and bug fixes. In addition to this, we will release the program demo right away to key vendors to begin training and we will open channels for feedback. All feedback will be collected and considered then integrated into the second phase deployments. Another primary reason for adopting the

Agile/Scrum framework as our methodology is the use of sprints. We will need to break down key application points into three different sprints in order to achieve the first phase scaling in a timely manner. The first sprint will focus on porting the application into executable software and changing the static structure of the application to accept any properly formatted CSV file. The second sprint will focus on integrating the application into the existing software programs. The final sprint of the first phase will be application testing and rollout. This last phase will be critical as we will want to develop an application that can always support 50 users.

### A.2.7 Funding Requirements

The funding requirements for the application are provided in the Letter of Transmittal but we will provide them here again. All the software and tools required to build the program are of no cost and are open source. This includes Python and its associated libraries. Funding will however be required to account for development hours, server space, hardware, and employee salaries. The program will require two funding installments of $150,000. Each installment will need to be provided before the beginning of each phase. Additionally, we are expecting to generate $200,000 in net income from the deployment of the first phase. This income should be entirely accounted for in the required budget of the second phase, bringing the total required funding to $500,000. If we generate less, then the anticipated amount then that funding will need to be included on top of the second installment.

### A.2.8 Impact of the Solution on Stakeholders

Both parties' stakeholders should be in full agreement for a successful merger. Open communication and frequent progress reports is an absolute must. To be considered a success by the stakeholders, we must generate a project that operates with few bugs and follows the ethical guidelines we have defined our companies around. Additionally, the application must provide the level of noise and privacy outlined in the documentation above. Lastly, the primary success will be based on how much income the application generates for Secure Enterprise Solutions. We estimate that the application can generate $2 million of new income once it has been rolled out to all existing customers. If the application can be ported and sold as an individual tool, we estimated it can generate an additional $4 million.

### A.2.9 Ethical and Legal Considerations

When dealing with any type of secure data, there is serious ethical considerations that must be reviewed and agreed upon. First, we will not store any data that is passed through the DP Engine by our customers. The further this, we will not allow companies to redistribute, repurpose, or offer our services. This will help us ensure that no unlawful data collection is occurring. Legally speaking, we must make sure our customers are not altering data to sell that should be protected under HIPPA or any other federal mandated law. As a publisher, it is important that we ensure our customers are abiding by our ethical stands as well. To ensure this, we will not allow the application or rights to the software to be purchased outright. Instead, we will offer access to the software via a monthly contract. If we feel that a customer is violating our policies and ethical standards, we can terminate their contract and prevent future misuse.

### A.2.10 Developer's Expertise

The developers working on this project will be comprised of a team of most junior software developers. We will have two different software experts on site to facilitate the project progress and act as scrum masters. The primary reason for this we will need to expand our employment base in order to meet the project deadline we have provided above. We think this is a necessary move so that we can train a fresh and prepared team of developers to handle the application as the demand increases. This project will act as an initial learning experience for our junior developers and help provide them with the skill set needed to create bigger projects.

# B. Project Proposal for IT Professionals

## B.1 Problem Statement

One of the biggest threats modern companies face is internal or proprietary data being compromised due to employee neglect. We live in a world where data exchange is persistent throughout a person’s day. These data interactions include email, social media, search engines, and many other resources. While these tools are a great success of modern technology, they also show a staggering negligence that has been developed by the consumers of these tools. The negligence exists in the consumer lack of care about their personal data. Location tracking, microphone privileges, and other tools can gather information about an individual at any time without their knowledge. Unfortunately, much of this neglect is also carried into the workplace where employees interact with sensitive data.

This creates a big problem for security and data firms because it is very hard to track employee usage of data and their daily interactions. You can implement tools such as disallowing cellphones in the workplace, restricting sites you can visit, and logging all user activity but these breaches still occur. However, by using the DP Engine, you can at least mitigate the damage that is incurred when data is misused or stolen. This mitigation is created by filtering the data and adding noise to the dataset. This provides a level of accuracy that this still very high but helps prevent useful data discovery. Adding this tool to the existing suite of security tools will create multi-tier protection plan for data. Additionally, it will provide us with the capital needed to expand this program and create more tools to ensure data privacy.

## B.2 Customer Description and Benefits

The customer base will include all Secure Enterprise Solutions existing customers. This can also expand beyond existing customers and be sold as a stand-alone application. The benefits of a DP application to customers have been outlined above. We will now describe how this will fit customer needs:

1.) Customers are trying to excel in the world of Big Data. Many companies will want to look towards selling data, however this is considered an ethical issue by many. The DP Engine solves the ethical issue for companies because it helps prevent unwarranted data discovery and helps prevent companies from being accused of selling data.
2.) Customers need a tool that is simple, easy to use, and is flexible with their data. The DP Engine provides these features and requires no technical knowledge or formal training. Additionally, the application can be integrated with existing software to automatically apply differential privacy for companies and their employees.
3.) Customers can let their vendors and clients know that they are using a system to help protect data.

## B.3 Existing Systems Integration

All the software tools required for this project have been in use and are publicly available as open-source software. The code is written in Python and can be easily adapted for any environment. In terms of integration, the tool can be ported into an existing program or can be used as a stand-alone tool.

## B.4 Data Needed

The data used in the demo program is stored in four data CSV files and are required for the application to process any meaningful data. These CSV files represent customer datasets and were randomly generated used a tool provided by www.mockaroo.com. There are two primary categories of datasets, employees and companies. Each dataset include a file for unaltered data and another file that altered data is inserted into. The structure of each file can be found in section A.2.4. The reason we choose the CSV file format is because it is the most common way of storing data and is seen as a standard across most database applications. Additionally, CSV files are typically small, easy to parse, and can be plugging into most applications.

We must expand the existing demo for the application to work with any CSV dataset. The file works by using the CSV reader/writer tool that is built into Python. We parse the column values then assign them a hash value. These hash values are group into one of ten hash buckets depending on their value. The existing application will need to be edited to read column values and determine which values are applicable for noise. The column values must have high selectivity and should contain most unique values. We will then grab 10% of the total data in that column dataset and use it to apply the noise. Each value is then iterated through and the data that have been selected to apply noise to will have their current value replaced with a random value from the 10% dataset. This process ensure that no new or unknown data values are included in the dataset.

It should be noted that there are many ways of applying differential privacy and this method does not generate “true” privacy. However, it is a strong starting point and has been studied to show a significant improvement in data protection then not applying any noise to a dataset. What is important here is that we are trying to provide data privacy and that this gives us a platform to build on.

## B.5 Project Methodology

As discussed above, our team will be using the Agile Scrum Methodology to produce the software. We choose the Agile development methodology because we feel that continuous improvements and change process will occur. Additionally, this is the framework both DP Engine and Secure Enterprise Solutions have used in the past. Lastly, we must develop and utilize the sprint tools to ensure that the project meets required deadlines. Below we provide an overview of the workflow process:

B.5.1 – Requirements

- Our application must provide a range of privacy on all datasets that is within our acceptable ranges (~6% noise added to an entire dataset).

- Our application must meet the ethical standards we adhere to and ensure our vendors follow these standards as well.

- We must take into consideration all customer feedback and adjust as needed to produce an application that customers are happy with.

- Stakeholders but be confident in application and their considerations should be discussed and added to the change board.

B.5.2 – Development

- The project will be broken down into two phases that we estimate will take a total of 370 development hours of 7 weeks. We include another week in this estimate for trouble shooting and backlog, bringing the total project development time to 2 months

- The first phase will focus on porting the existing application and getting a prototype out on the market. This phase will have 3 different sprints and should take 3 weeks to complete.

- The second phase will focus on rolling out feature updates and applying patches to fix bugs.

B.5.3 – Testing

- At the start of the project, we will send our key customers the demo version of the DP Engine so that they may work with it and provide feedback. This will be considered our initial consumer facing and black box testing.

- Unit and integration tests will make up a majority of the first phase. We will allocate one week of development time to transition the software and two weeks for white box testing.

- The second phase will collect all the data from initial testing efforts and focus primarily on unit and specific case testing.

B.5.4 – Delivery

- The project will have an initial delivery after the first phase. We will then deploy our second phase which will be deemed complete based on stakeholders sign off.

B.5.5 – Feedback

- Feedback will be collected throughout the first phase and integrated into the second phase

- We will have 3 primary sources of feedback: initial customers who receive the demo deployments, stakeholders, and customers that purchase the application after the first phase deployment.

## B.6 Project Deliverables

The project will have one deliverable, the finished DP Engine application.

## B.7 Implementation Plan

The implementation plan focuses primarily on the two phases and the rollout of the prototype and final application. At the end of each phase, the application and all associated resources will be rolled out to customers. These resources comprise of a brief training video series and two weeks of initial 24/7 support. We will also include a set of training data to ensure that no use trains on a production dataset.

In terms of access to the software, we will restrict the code repository to only members of the assigned development team. This will prevent any hallway testing or edits to the code base by other employees of DP Engine or SES. This will help ensure proper change control management and ensure that no developer can push changes without approval. Lastly, clients will need to store the data locally to plug into the DP Engine as we will not support any data storage through our existing software. If needed, we will provide a tool to convert .DB files or .SQL files to .CSV files so that the data can be properly changed. This conversion tool must be able to also convert data back to the original file format and be easy to use.

## B.8 Evaluation Plan

The evaluation plan will focus on how well the program is implemented into existing daily actives. We must provide surveys and collect data from super users on statistics such as how many times a user opens the application daily. To ensure that our tool is providing useful data protection, we must also track use cases and ensure that proper data is being inserted into the DP Engine. As mentioned before, the program should provide a meaningful service without interfering with sensitive data that could interfere with HIPPA or other data compliance. Additionally, we need to confirm that the noise being added to datasets does not go beyond the existing data or create errors. To accomplish these tasks, we will need to initially track usage, data, and noise statistics for customers that use the product during the first deployment phase. This data must be stored locally with the client and transferred in a secure way. Once we are certain the application is working as intended, this data should be destroyed, and stakeholder sign off should begin.

The initial goal of our testing will be to determine as many errors and data discrepancies as possible. As the amount of errors fall into an acceptable range and deployment is solidified, our testing focus will then switch to ensure a level of data privacy is maintained. To accomplish this, we will ask for periodic customer reports on data statistics that does not reveal any sensitive information to us. Our team will analyze these statistics for future updates and rollouts.

## B.9 Programming Environments and Related Costs

Programming environment:

- Python 3.7 along with Qt Designer, PyQt5, PyQt Graph, and default Python libraries.
- A file converter to convert any common database file types into CSV files
- Statistics tracking software for tracking usage statistics during the first deployment

There are no associated costs with the programming environment as all of these tools are available as no-cost open source software.

Environment Costs:

- We will need workstations and storage hardware to facilitate the deployment phases. Our developers will need workstations that can connect to a central network to begin programming, testing, and collecting data analytics.

We are budgeting $30,000 to the environment costs and these costs will be accounted for in the first installment payment

Human Resource Requirements

- We require a staff of six developers to complete the project. They will each be paid a monthly salary of $5,000 per month. The estimate time of completion is two months so a total of $60,000 of the total budget should go towards employee salaries.
- Phase one (Deployable application): 120 development hours – total cost: $150,000 (this includes the equipment cost and the first month salary payments).
- Phase two (Major Feature Updates and Bug Fixes): 160 development hours + 1 week of allocated backlog – total cost: $350,000. The additional costs for this phase are incurred from the following:
    - More developer hours
    - Deployment costs associated with getting the product to customers
    - Hiring outside testers
    - Increased hardware requirements needed to scale the application
    - A reserve for unseen costs

## B.10 Timeline and Milestones

| Event | Start Date | End Date | Developer hours required | Dependencies | Assigned Resources |
|---|---|---|---|---|---|
| Project Start | May 1st, 2020 | May 1st, 2020 | 0 | None | Project Manager, Stakeholders |
| Phase One | May 1st, 2020 | May 22nd, 2020 | 120 | Project Start | Project Manager, Developers |
| Project integration and scaling for consumer use | May 1st, 2020 | May 8th, 2020 | 40 | Project Start | Project Manager, Developers |
| Deployment of application to select customers and integration testing | May 8th, 2020 | May 22nd, 2020 | 75 | Project integration and scaling for consumer use | Developers |
| Deployment of the first DP Engine to customers | May 22nd, 2020 | May 22nd, 2020 | 5 | Deployment of application to select customers and integration testing | Project Manager, Developers |
| Phase Two | May 22nd, 2020 | June 26th, 2020 | 160 | Phase One | Project Manager, Stakeholders, Developers |
| First major feature update | May 29th, 2020 | June 5th, 2020 | 40 | Phase Two | Project Manager, Developers |
| Second major feature update | June 5th, 2020 | June 19th, 2020 | 80 | First major feature update | Project Manager, Developers |
| Final bug review and stakeholder signoff | June 19th, 2020 | June 26th, 2020 | 40 | Second major feature update | Project Manager, Stakeholders, Developers |

# D. Developed Product Documentation

## D.1 Business Requirements and Project Purpose

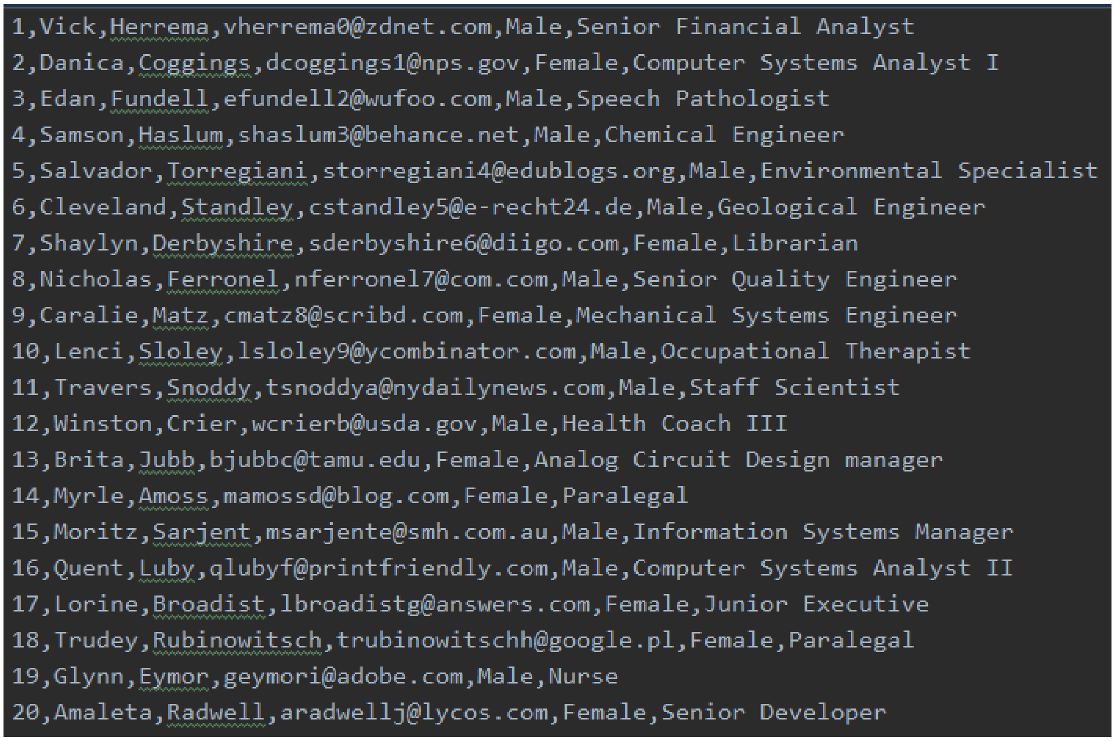
The business requirements have been outlined throughout the document. To recap, we will use the Python programming language along with a set of open-source libraries and tools to complete the project. Additionally, we will need a total of $500,000 for the project to be completed within our eight-week timeline. Lastly, we need stakeholder approval for our contract and deliverable product, the Differential Privacy Engine.

In terms of the application, the primary requirement is to create an easy to use program that can integrate into existing software systems and provide data privacy. The problem this requirement addresses is that our client's employees access sensitive data daily and cause frequent data breaches. Additionally, there is a problem of data discovery on customer datasets that violates consumer confidence. Our solution to these issues of data theft and privacy is to help Secure Enterprise Solutions create a multi-facing software solution. This solution will capitalize on their existing security software infrastructure and introduce our privacy engine on the data level. The purpose of providing this service to customers is to help reduce the ethical and financial responsibility they take on when dealing with personal data.

## D.2 Raw and Cleaned Data

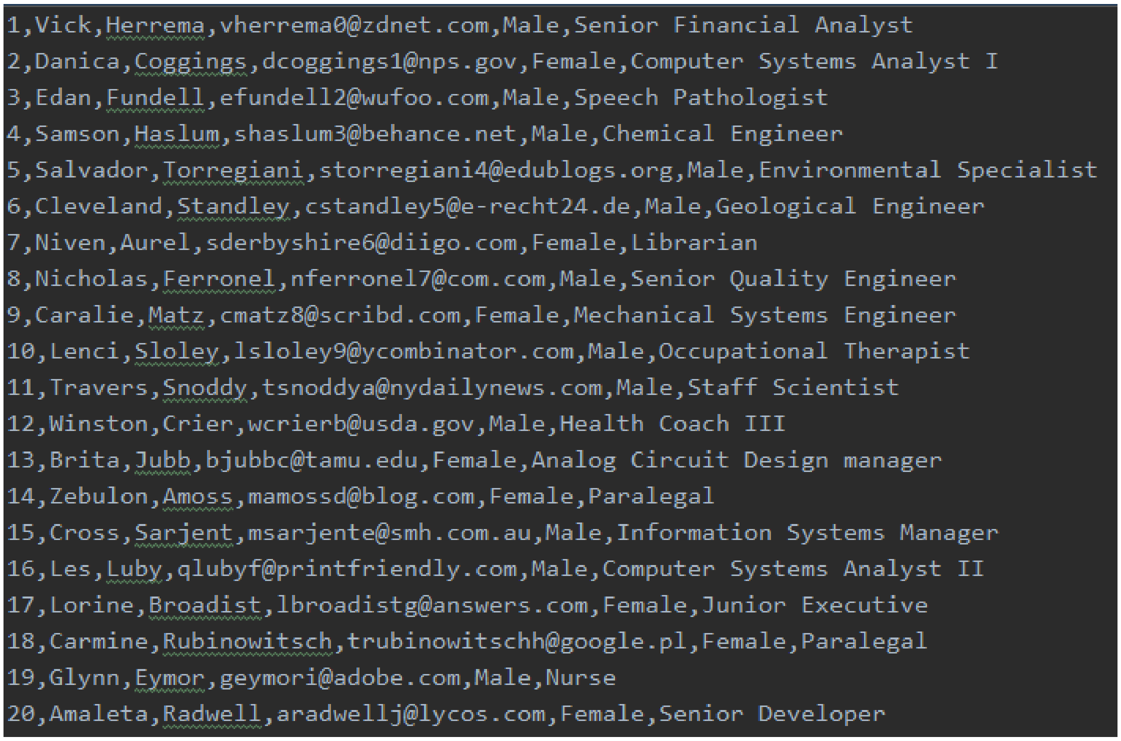
The data required for the application is properly formatted .CSV files that contain data to be plugged into the DP Engine. The sample CSV data that is included in the demo was generated using www.mockaroo.com. These CSV files can be viewed by opening the files within the application files.

The data in the two files, Read_Company_Data.csv and Read_Employee_Data.csv are considered the raw datasets. This is the unaltered data before it is plugged into the DP Engine. The data is cleaned by running the application, selecting a dataset to operate on, then selecting two fields to apply noise to. Once this has been done, you can look at the Write_Company_Data.csv and the Write_Employee_Data.csv files to view the dataset with DP applied. These two CSV files are empty files by default and will only contain data if the DP Engine has been run on their respective dataset. **Figure D.2.1** shows an example of a raw dataset and **Figure D.2.2** shows an example of a cleaned dataset after differential privacy has been applied. Notice that the following rows 7, 14, 15, 16 and 18 have either had their first name or last name values changed.



```
1,Vick,Herrema,vherrema0@zdnet.com,Male,Senior Financial Analyst
2,Danica,Coggings,dcoggings1@nps.gov,Female,Computer Systems Analyst I
3,Edan,Fundell,efundell2@wufoo.com,Male,Speech Pathologist
4,Samson,Haslum,shaslum3@behance.net,Male,Chemical Engineer
5,Salvador,Torregiani,storregiani4@edublogs.org,Male,Environmental Specialist
6,Cleveland,Standley,cstandley5@e-recht24.de,Male,Geological Engineer
7,Shaylyn,Derbyshire,sderbyshire6@diigo.com,Female,Librarian
8,Nicholas,Ferronel,nferronel7@com.com,Male,Senior Quality Engineer
9,Caralie,Matz,cmatz8@scribd.com,Female,Mechanical Systems Engineer
10,Lenci,Sloley,lsloley9@ycombinator.com,Male,Occupational Therapist
11,Travers,Snoddy,tsnoddya@nydailynews.com,Male,Staff Scientist
12,Winston,Crier,wcrierb@usda.gov,Male,Health Coach III
13,Brita,Jubb,bjubbc@tamu.edu,Female,Analog Circuit Design manager
14,Myrle,Amoss,mamossd@blog.com,Female,Paralegal
15,Moritz,Sarjent,msarjente@smh.com.au,Male,Information Systems Manager
16,Quent,Luby,qlubyf@printfriendly.com,Male,Computer Systems Analyst II
17,Lorine,Broadist,lbroadistg@answers.com,Female,Junior Executive
18,Trudey,Rubinowitsch,trubinowitschh@google.pl,Female,Paralegal
19,Glynn,Eymor,geymori@adobe.com,Male,Nurse
20,Amaleta,Radwell,aradwellj@lycos.com,Female,Senior Developer
```

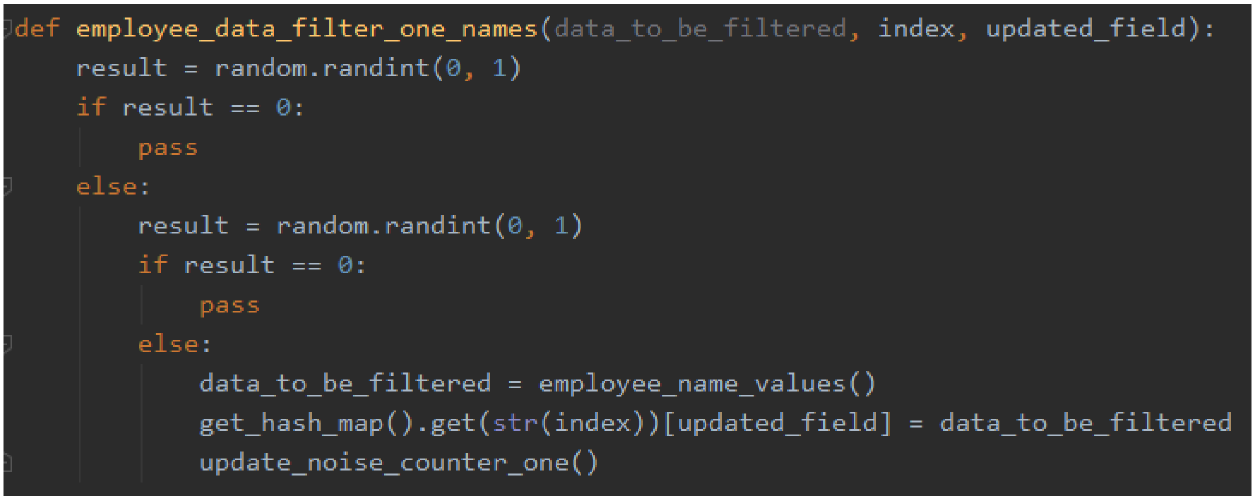
**Figure D.2.1** – Raw Read_Employee_Data.csv file



```
1,Vick,Herrema,vherrema0@zdnet.com,Male,Senior Financial Analyst
2,Danica,Coggings,dcoggings1@nps.gov,Female,Computer Systems Analyst I
3,Edan,Fundell,efundell2@wufoo.com,Male,Speech Pathologist
4,Samson,Haslum,shaslum3@behance.net,Male,Chemical Engineer
5,Salvador,Torregiani,storregiani4@edublogs.org,Male,Environmental Specialist
6,Cleveland,Standley,cstandley5@e-recht24.de,Male,Geological Engineer
7,Niven,Aurel,sderbyshire6@diigo.com,Female,Librarian
8,Nicholas,Ferronel,nferronel7@com.com,Male,Senior Quality Engineer
9,Caralie,Matz,cmatz8@scribd.com,Female,Mechanical Systems Engineer
10,Lenci,Sloley,lsloley9@ycombinator.com,Male,Occupational Therapist
11,Travers,Snoddy,tsnoddya@nydailynews.com,Male,Staff Scientist
12,Winston,Crier,wcrierb@usda.gov,Male,Health Coach III
13,Brita,Jubb,bjubbc@tamu.edu,Female,Analog Circuit Design manager
14,Zebulon,Amoss,mamossd@blog.com,Female,Paralegal
15,Cross,Sarjent,msarjente@smh.com.au,Male,Information Systems Manager
16,Les,Luby,qlubyf@printfriendly.com,Male,Computer Systems Analyst II
17,Lorine,Broadist,lbroadistg@answers.com,Female,Junior Executive
18,Carmine,Rubinowitsch,trubinowitschh@google.pl,Female,Paralegal
19,Glynn,Eymor,geymori@adobe.com,Male,Nurse
20,Amaleta,Radwell,aradwellj@lycos.com,Female,Senior Developer
```

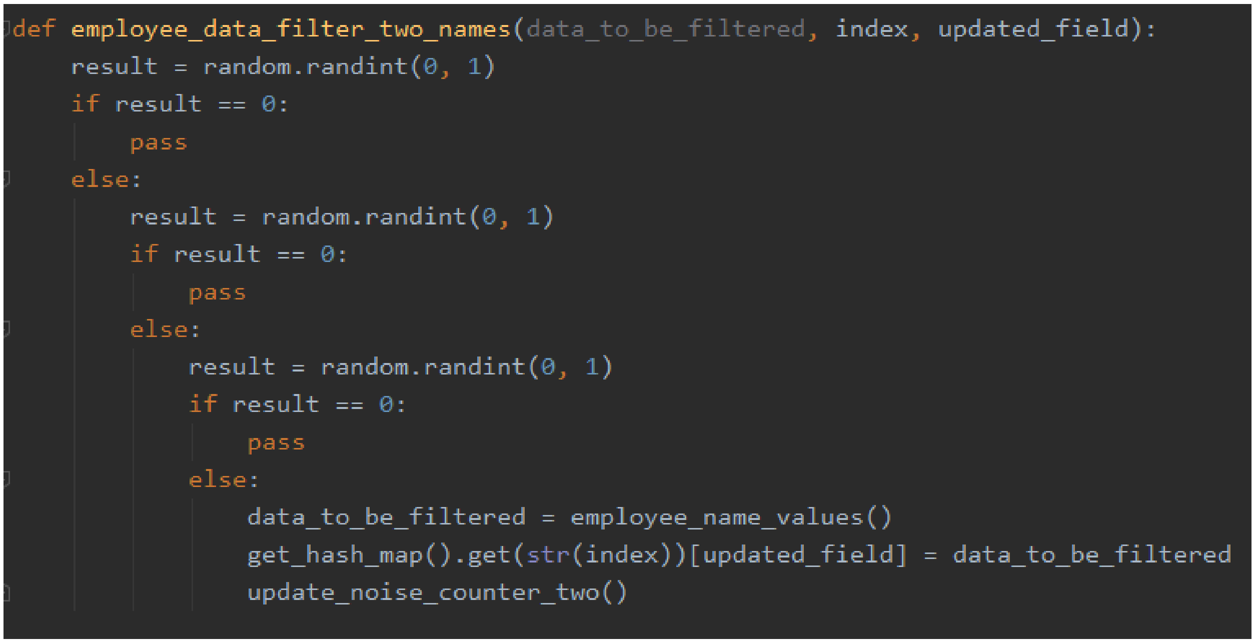
**Figure D.2.2** – Cleaned Write_Employee_Data.csv file

The DP Engine works by using a coin flip algorithm (see **figure D.2.3** and **figure D.2.4**) to filter and apply noise to the algorithm. The first data column value you choose will have a minimum of one-coin flip and a maximum of two-coin flips as shown in **figure D.2.3**. The second data column value you choose will have a minimum of one-coin flip and a maximum of three-coin flips as shown in **figure D.2.4**. The example code shown below can be located in the employee_records_window.py file. Similar methods existing for the company data can be found in the company_records_window.py file.



```
def employee_data_filter_one_names(data_to_be_filtered, index, updated_field):
    result = random.randint(0, 1)
    if result == 0:
        pass
    else:
        result = random.randint(0, 1)
        if result == 0:
            pass
        else:
            data_to_be_filtered = employee_name_values()
            get_hash_map().get(str(index))[updated_field] = data_to_be_filtered
            update_noise_counter_one()
```

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



```
def employee_data_filter_two_names(data_to_be_filtered, index, updated_field):
    result = random.randint(0, 1)
    if result == 0:
        pass
    else:
        result = random.randint(0, 1)
        if result == 0:
            pass
        else:
            result = random.randint(0, 1)
            if result == 0:
                pass
            else:
                data_to_be_filtered = employee_name_values()
                get_hash_map().get(str(index))[updated_field] = data_to_be_filtered
                update_noise_counter_two()
```

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

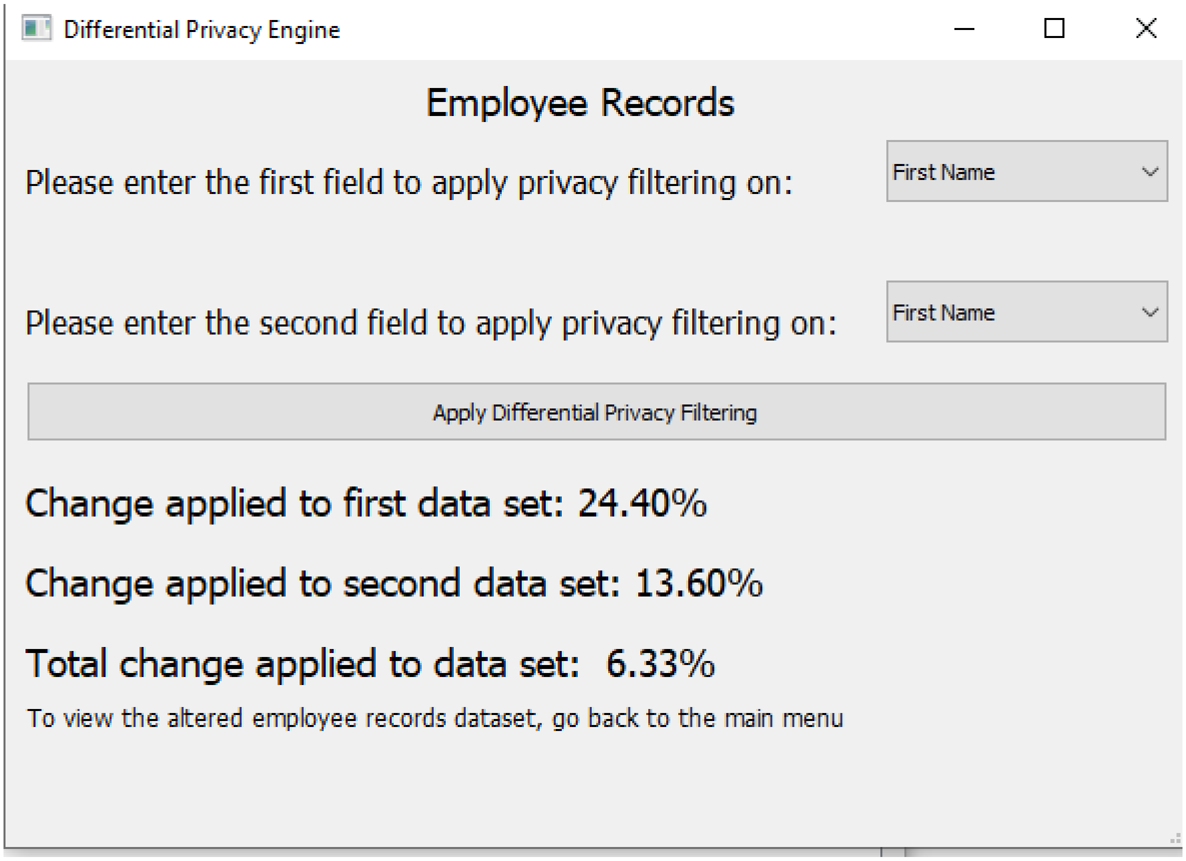
## D.3 Code Analysis

To better understand the data and how the privacy filtering changes the dataset, we have created a GUI environment using PyQt5. The GUI provides three primary tools to produce predictive and descriptive data analysis. These tools are described below:

Predictive Tools:

1.) An overview of the amount of noise applied to dataset

To apply privacy filtering in the application, you must select the “Apply DP to Employee Records” or “Apply DP to Company Records” buttons. Once you are in the correct menu, you can apply privacy filtering and the results of total changes to the dataset will be displayed. This example can be found in the employee_records_window.py and the same instance exists for companies in the company_records_window.py See **figure D.3.1** for an example…


Differential Privacy Engine
Employee Records
Apply Differential Privacy Filtering

Please enter the first field to apply privacy filtering on: First Name

Please enter the second field to apply privacy filtering on: First Name

Change applied to first data set: 24.40%

Change applied to second data set: 13.60%

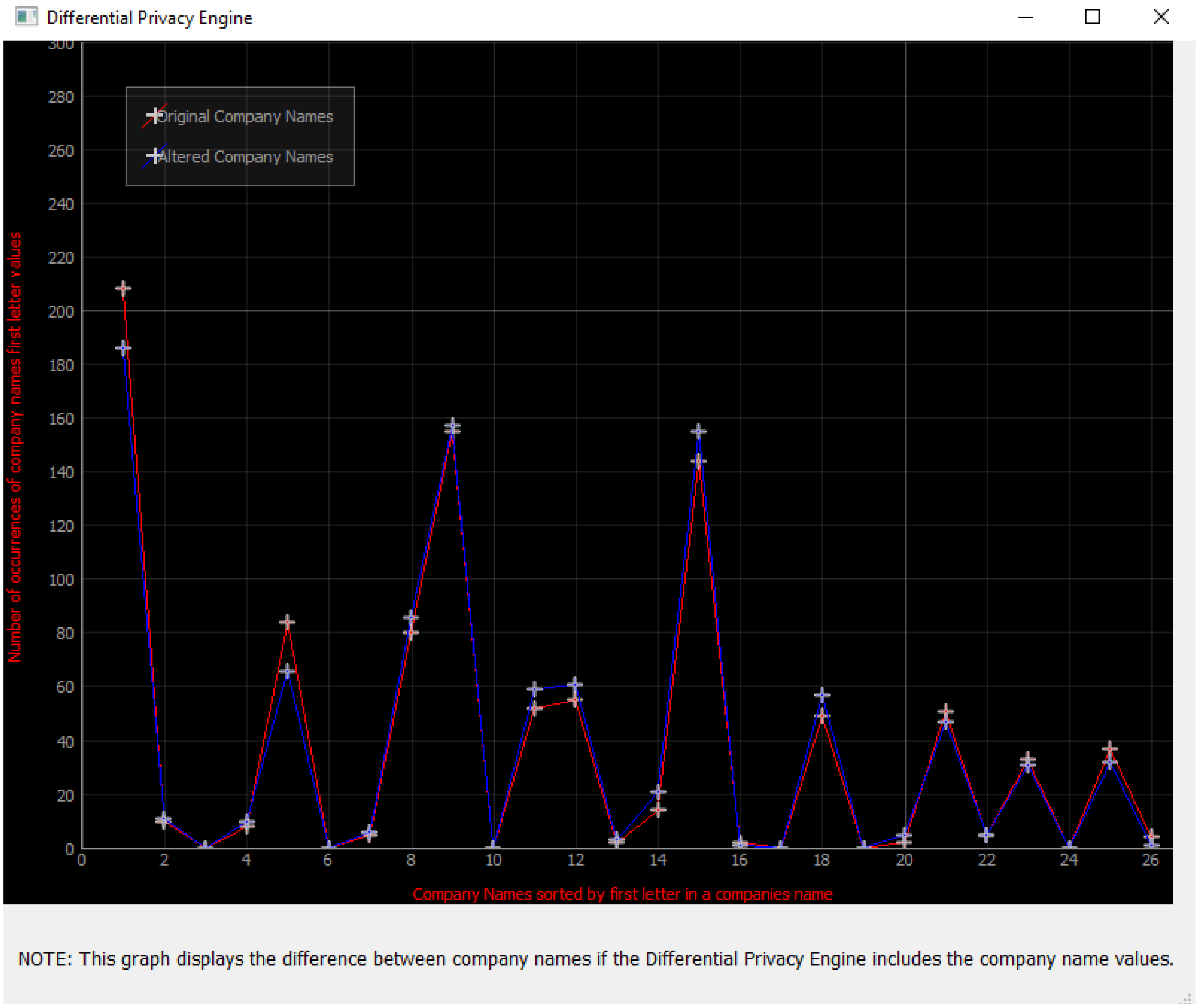
Total change applied to data set:  6.33%

To view the altered employee records dataset, go back to the main menu

**Figure D.3.1** – Percentage of privacy filtering applied to a dataset

2.) Scatter plot diagrams that show the difference on a column value field across the whole dataset

The scatter plots are designed to show the difference between the employee first names or company names. The red line represents the original values while the blue line represents the changed values. The numbers on the x-axis (0 through 26) represent the letters in the alphabet (A is equivalent to 1 and B is equivalent to 2). It is important to note that the blue line will only be present if DP was applied to a name field in the respective dataset. This example can be found in the company_scatter_plot.py and the same instance exists for companies in the employee_scatter_plot.py See **figure D.3.2** for an example…


Differential Privacy Engine
Original Company Names
Altered Company Names
Number of occurrences of company names first letter values
Company Names sorted by first letter in a companies name
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26

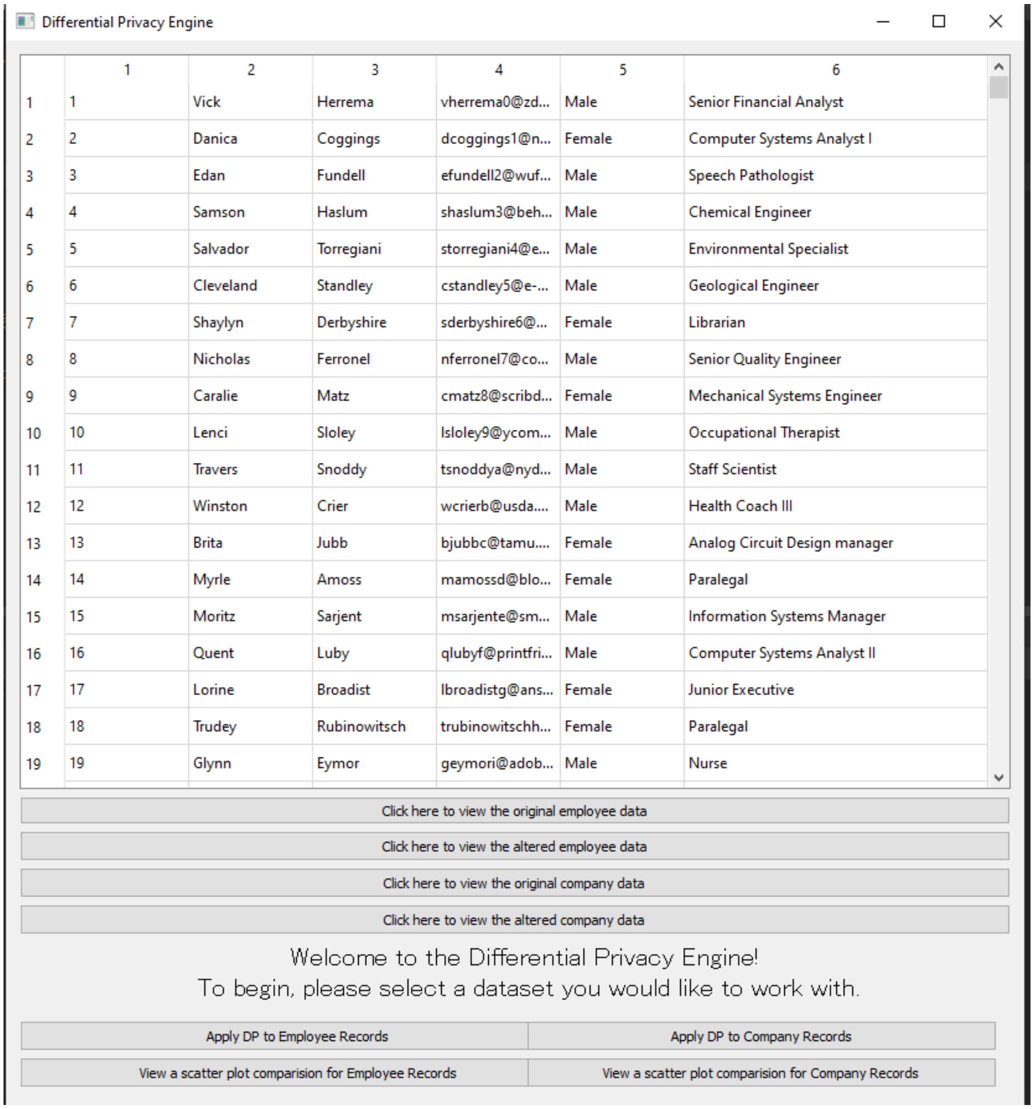
NOTE: This graph displays the difference between company names if the Differential Privacy Engine includes the company name values.

**Figure D.3.2** – Scatter Plot displaying company name changes between both datasets

Descriptive Tools:

1.) A Database view of a specified Dataset

We provide a database view where you can analyze the unaltered and altered datasets. This allows you to see if the changes the DP Engine applied are valuable and that there is enough selectivity on a certain column to make use of DP. It also allows you to manually change data values to better optimize the dataset. For example, in the rare instance that the DP Engine changes multiple sequential rows to the same value then you can manually update a few rows to create more selectivity. Lastly, the main screen that this tool is available in acts as a dashboard for the DP Engine. You can view or alter any data set from this screen. This example can be found in employee_records_original_window.py file. See **figure D.3.3** for an example…



|  | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | Vick | Herrema | vherrema0@zd... | Male | Senior Financial Analyst |
| 2 | 2 | Danica | Coggings | dcoggings1@n... | Female | Computer Systems Analyst I |
| 3 | 3 | Edan | Fundell | efundell2@wuf... | Male | Speech Pathologist |
| 4 | 4 | Samson | Haslum | shaslum3@beh... | Male | Chemical Engineer |
| 5 | 5 | Salvador | Torregiani | storregiani4@e... | Male | Environmental Specialist |
| 6 | 6 | Cleveland | Standley | cstandley5@e-... | Male | Geological Engineer |
| 7 | 7 | Shaylyn | Derbyshire | sderbyshire6@... | Female | Librarian |
| 8 | 8 | Nicholas | Ferronel | nferronel7@co... | Male | Senior Quality Engineer |
| 9 | 9 | Caralie | Matz | cmatz8@scribd... | Female | Mechanical Systems Engineer |
| 10 | 10 | Lenci | Sloley | lsloley9@ycom... | Male | Occupational Therapist |
| 11 | 11 | Travers | Snoddy | tsnoddya@nyd... | Male | Staff Scientist |
| 12 | 12 | Winston | Crier | wcrierb@usda.... | Male | Health Coach III |
| 13 | 13 | Brita | Jubb | bjubbc@tamu.... | Female | Analog Circuit Design manager |
| 14 | 14 | Myrle | Amoss | mamossd@blo... | Female | Paralegal |
| 15 | 15 | Moritz | Sarjent | msarjente@sm... | Male | Information Systems Manager |
| 16 | 16 | Quent | Luby | qlubyf@printfri... | Male | Computer Systems Analyst II |
| 17 | 17 | Lorine | Broadist | lbroadistg@ans... | Female | Junior Executive |
| 18 | 18 | Trudey | Rubinowitsch | trubinowitschh... | Female | Paralegal |
| 19 | 19 | Glynn | Eymor | geymori@adob... | Male | Nurse |

Click here to view the original employee data
Click here to view the altered employee data
Click here to view the original company data
Click here to view the altered company data

Welcome to the Differential Privacy Engine!
To begin, please select a dataset you would like to work with.

Apply DP to Employee Records
Apply DP to Company Records
View a scatter plot comparision for Employee Records
View a scatter plot comparision for Company Records

**Figure D.3.3** – Main menu currently displaying the unaltered employee dataset

## D.4 Hypothesis Verification

Our initial hypothesis was that we could create an easy to use data product that could easily integrate with existing software and provide a base-level for privacy filtering. Given the visual data we have provided above and our results staying within predicted bounds after hundreds of unit tests, we believe our hypothesis was correct. Our data shows that every dataset will display ~6% total noise and applies a noticeable amount of privacy filtering. However, total and complete privacy is not possible with our current application. While this is not the stated goal of the project, it should be our final goal. To reach this goal, more research and testing will need to be done. For now, the objective and purpose of this program are inline with what we predicted and planned for.

## D.5 Effective Visualizations and Reporting

The visual elements of our program are outlined and described in further detail in section D.3. These visual aids show meaningful changes on the dataset as a result of the data filtering in the DP Engine. You can view the data visually or in a graph distribution. Additionally, you can also view the percentage of noise applied to each field and the dataset. These visualizations help bringing meaning to the data and show that the privacy filtering is working.

In terms of reporting, we wanted to ensure that our application was telling a story of random distribution. This randomness should have been applied first in the selection of rows to apply filtering to and then again in the value it chooses to apply the noise with. To reinforce this concept the randomization should occur at two levels:

1.) When filtering a data value, the select values must be randomly selected.
2.) For the random values that are selected, the noise value that replaces the original value should be random as well.

Randomization is very important to the idea of true data privacy, but we also have to apply constraints to ensure the randomness is directed and correlated to the dataset as a whole. To do this we applied a bounded range for random values that can be selected based on the order in which they are selected. Additionally, we make sure that the noise replacement values exist within the original dataset and that they are within a range of the first 100 values.

## D.6 Accuracy Analysis

As discussed throughout the documentation, the project is designed to apply meaningful randomization to a dataset in order to create a baseline of privacy filtering. In terms of accuracy, we have created and rand over 300 use cases with the provided demo application and have maintained the reported bounds. An overview of these bounds is provided below:

First data filter: ~25% of data in this group should be changed. This is bounded by 3% in either direction, creating an effective range of 22-28%.

Second data filter: ~12.5% of data in this group should be changed. This is bounded by 1.5% in either direction, creating an effective range of 11-13.5%.

Total data filter across all records: ~6 of data in the total data set should be changed after combining the first and second data filters. This range is also bounded by 1% in either direction, creating an effective range of 5-7%.

## D.7 Application Testing

Application testing is a major part of the product. Testing phases can be broken down in to the two lifecycle phases we have discussed already.

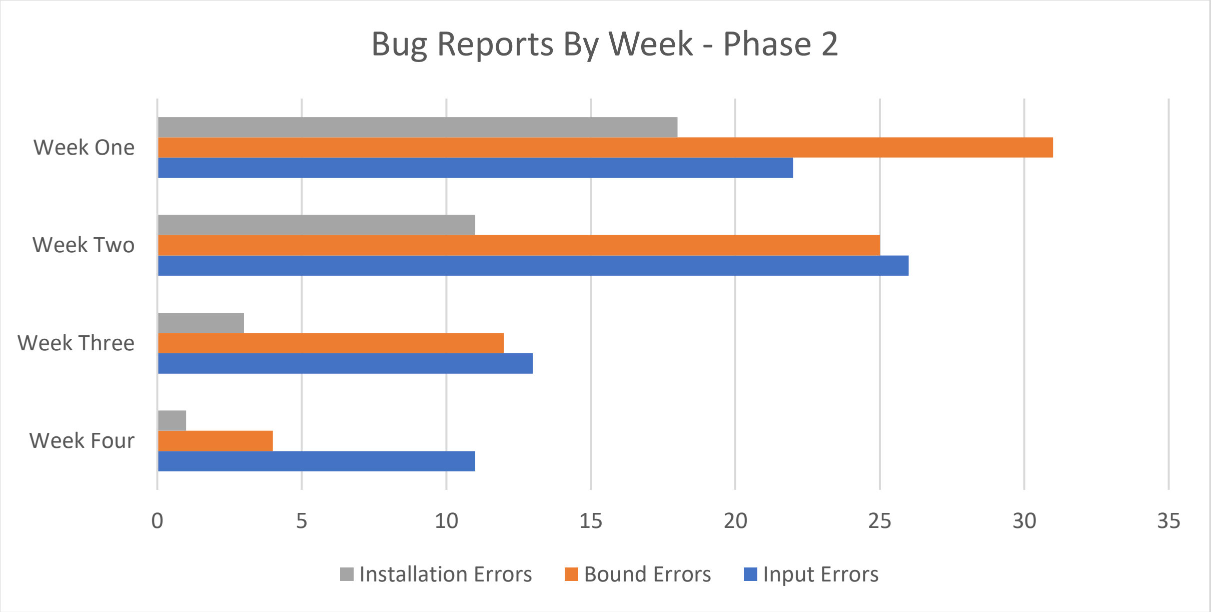
Phase 1: Initial Deployment Testing

During this phase, we released the program demo and a prototype to super users and selected customers. The program demo is used initially for training and the prototype allows users to begin inserting their own datasets. During this two-week period, companies and testers are asked to submit bug reports for our time to review. Very little white box testing is done during this period as most testing is being executed by end users

Phase 2: Feature Update Review

During the second phase, we will collect user feedback and begin white box testing. We will create 3 primary categories to place bug reports in and add them into the changelog as we progress. The 3 categories are as follows: errors caused by input into the dataset, errors with reports outside of set bounds, errors related to product installation. This is where most of the revisions will occur.

In addition to the testing described above, we will perform demos to stakeholders for final sign off. A tracking report for bugs reported per week in phase 2 is provided below…



## D.8 Application Files

All of the required files to view and run the initial application demo can be found here: https://github.com/Ryanhilde/WGU-C964-Capstone. These files are also submitted with the documentation. The project is designed to be ran in an IDE environment that supports Python 3.7. To successfully run the application, you must install the PyQt5 and PyQtGraph libraries. Additionally, you need to also include the following Python libraries: numpy, csv, random, string, and counter from the collection's library.

A description of each file in the repository is provided below:

employee_records_original_window.py – This is the main application window and should be the file that is run to start the application.

employee_hash_map.py – This file creates buckets and a hash map for employee records to be inserted into for data manipulation

employee_records_window.py – This file is where the DP filtering algorithm exists and is applied to the Read_Employee_Data.csv file.

employee_scatter_plot.py – This file contains the scatter plot for viewing the unaltered and altered employee records

company_hash_map.py – This file creates buckets and a hash map for company records to be inserted into for data manipulation

company_records_window.py – This file is where the DP filtering algorithm exists and is applied to the Read_Company_Data.csv file.

company_scatter_plot.py – This file contains the scatter plot for viewing the unaltered and altered company records

Read_Employee_Data.csv – This file contains the unaltered employee records in a csv file format

Write_Employee_Data.csv – This file contains the altered employee records in a csv file format

Read_Company_Data.csv – This file contains the unaltered company records in a csv file format

Write_Company_Data.csv – This file contains the altered company records in a csv file format

For further assistant, please refer to the User's Guide below

## D.9 User's Guide

For evaluators and users trying to startup the application, please follow the instructions below:

1.) Unzip the associated files and place them into a project folder.
2.) Make sure that you have python 3.7 installed on the desktop you are accessing the program from. This can be done by navigating to your local command prompt and entering "python -version". If you do not have Python installed, please see the Advanced Trouble Shooting section below.
3.) Load that folder into an IDE that can support Python 3.7 (We recommend using JetBrains Pycharm Community Edition 2019.1.3).
4.) Download and install the PyQt5 and PyQtGraph libraries.

    a. This can be done in PyCharm by navigating to file/settings/project/project interpreter and locate the "+" on the right-hand side of the menu. From here, type in the package names and install them into your environment. More detailed instructions can be found here: https://www.jetbrains.com/help/pycharm/installing-uninstalling-and-upgrading-packages.html

5.) Navigate to the python file called employee_records_original_window.py and start the application by right clicking the file name and selecting "Run". Tip: if you cannot run the file this way, hold down your shift key and press F10.
6.) Once the application starts you can use any of the top four buttons to view datasets. Tip: the altered files will appear as empty at first because no data has been filtered and written to these files yet.
7.) Use the button 4 buttons to apply privacy filtering on a dataset or view a scatter plot comparison of the datasets. Tip: the scatter plot data will show only the unaltered records if the privacy filter has not been applied yet

Note: The bottom four buttons in the employee_records_original_window.py will open new windows. To close these windows, simple clicked the red X at the top righthand side of the screen.

Advanced Trouble Shooting:

1.) If you cannot locate Python on your machine or you have the wrong version installed, please follow this link to properly install Python and set the correct path: https://phoenixnap.com/kb/how-to-install-python-3-windows
2.) If you cannot locate PyQt5 or PyQtGraph, you can manually install the packages by using this link: https://www.riverbankcomputing.com/software/pyqt/download5

## E. Sources

Schneider, Michael, et al. "What Is Differential Privacy?" *A Few Thoughts on Cryptographic Engineering*, 18 July 2016, blog.cryptographyengineering.com/2016/06/15/what-is-differential-privacy/.