**Customized Financial Tracker and Invoice Application for a Small Business**

**Goals:**

-Create an application centered around specific client needs

-Optimize information presentation given client needs

**Client Needs:**

Summary: Client wants an application to store sales and order information which can then send invoices to the business’ clients. Client also would like to have the ability to visually represent basic statistical information about past sales data.

1. Storage/Retrieval of sensitive data, such as:

-Business Client Information

-Product Information

-Order Information

2. Automatic Invoice Generation and Sending using Client’s business email address

3. Ability to calculate basic statistical analyses on past data with visual representation

-Trend analysis

-Sales Over Time

-per Business Client

-per Product

-per Client “Purchasing Class” (defined by order size and frequency)

-Seasonal Sales

-User can define periods of time to indicate a specific “season” or event and can then compare against past sales numbers

**Deliverables:**

1. Application interface to a secure local database to store and retrieve data
2. Password Encryption for database to protect against local users reading sensitive data
3. GUI for inputting new client information and editing old information
4. Feature for automatic invoice delivery by email
5. GUI for statistical analysis on past data

**Stretch Goals:**

1. Implementing further statistical analyses on past data
2. Basic phone application to record client data and upload to DB (currently, business uses pen/paper)
3. OR parser that can read-in notes from a phone application to the DB
4. Further develop security protection measures

**Software/Libraries Used:**

-Python 3

-PyQt5 (Qt5 under LGPL)

-Sqlite3 for DB

-SQLCipher (pysqlcipher3) (256-bit AES)

-numpy

-“email” python pkg with smtplib for invoice sending

- or “Yagmail” for easy gmail integration

**Timeline:**

October:

-Begin/Finalize GUI Design (Main Page GUI, Invoicing GUI, Statistical Analysis GUI)

-Finalize libraries used for deliverables (libs may change due to usability, e.g. email interface may be easier depending on email client used)

-Finalize Database Design

-Complete UML Diagram for application

Late-October/Early-November:

-Begin/Complete GUI and Code for Main Page

-Begin/Complete Database functionality

-Begin/Complete Auto-Invoice Email integration

-Begin Statistical Analysis Code & GUI implementation

Late-November:

-Complete Statistical Analysis Code and begin full application testing

Early-December:

-Complete application testing/debugging and enhance GUI and/or add-in stretch goals