Table of Contents

1. Introduction

1.1 Purpose....................................................................................... 2

1.2 Scope.......................................................................................... 2

1.3 Definitions, acronyms, and abbreviations.................................. 3

1.4 References.................................................................................. 3

1.5 Overview.................................................................................... 3

2. Overall description

2.1 Product perspective................................................................... 3

2.2 Product functions.......................................................................5

2.3 User characteristics................................................................... 7

2.4 Constraints................................................................................ 7

2.5 Assumptions and dependencies............................................... 7

3. Specific Requirements

Team Roles..................................................................................... 9

1) Introduction

**1.1) Purpose**

a) To keep a detailed list of Employee information, such as:

* ID
* First and Last name
* Benefit
* Tax
* Gross Income
* Net Pay

b) Users can also add, edit, and delete information

c) This program is intended for Payroll Employees

**1.2) Scope**

a) The Product is called "Payroll Application."

b) It will have several functions:

* To display employee information
* To add information
* To edit information
* To delete information
* To search for all available data given in the file

c) All information will be displayed in a window on a GUI program  
where the user will be able to open a .txt file to view and update information

d) Users can input data on a person's tax and/or gross income and the net pay will update and display in real time

e) Conditional functions include displaying the location in the machine of the text file being used and being able to filter your search to specific columns

**1.4) References**

a) Lucidchart: the creation of the algorithm chart was developed here.

**Source:**  [www.lucidchart.com](http://www.lucidchart.com)

b) Github: sight used to maintain group assignments, delegate work, and make problems/bugs/area for enhancements aware among the workers.

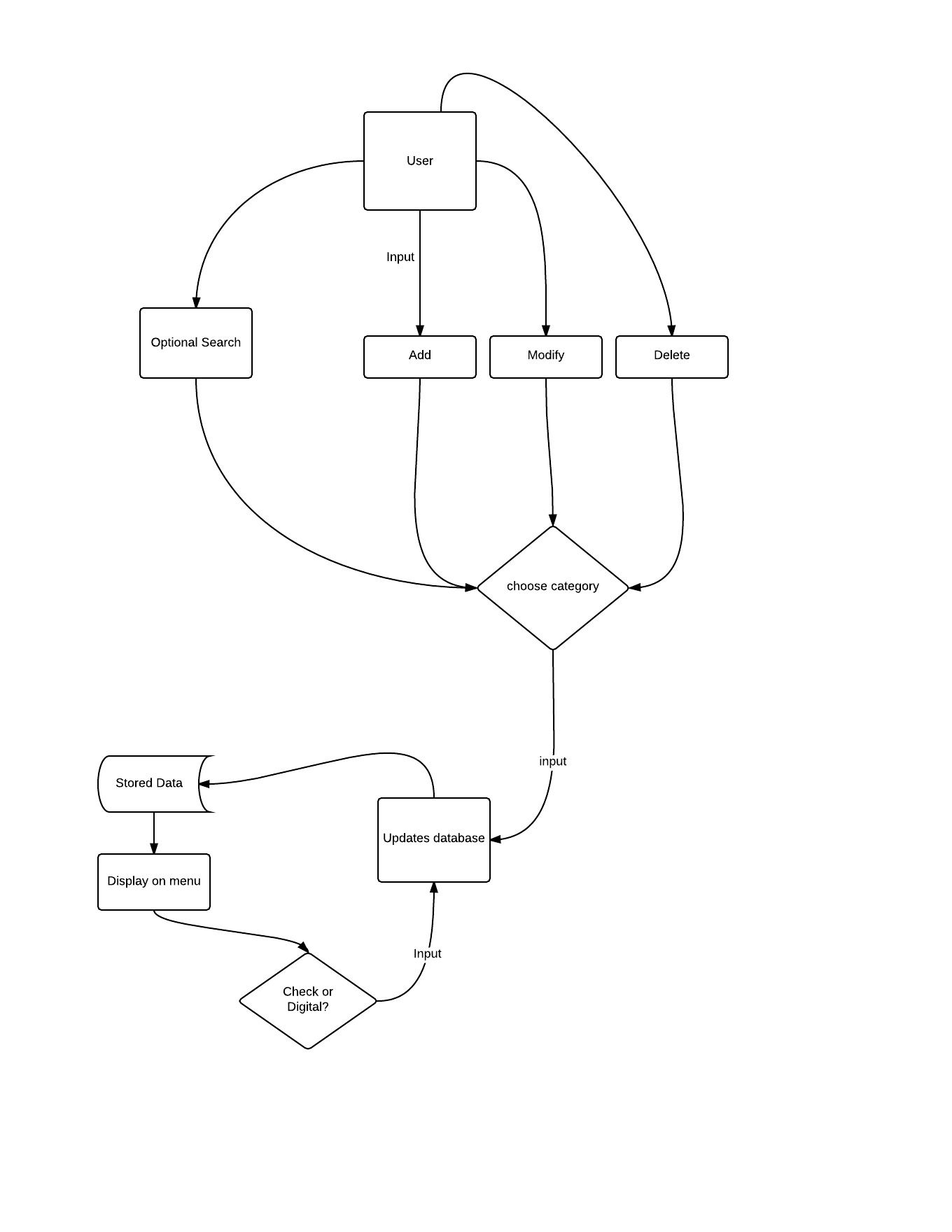
**Source:** <https://github.com/>

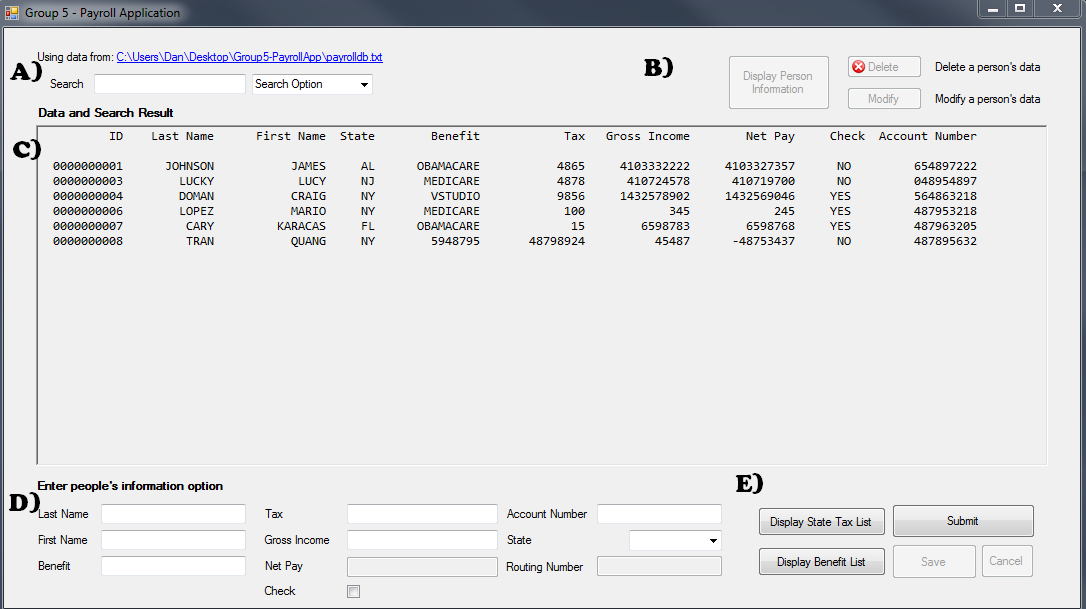
**1.5) Overview**

The rest of the SRS document will go into specifics of the functionality of Payroll Application, as well as go into detail on the process of building it, how it was organized, and any complications along the way.

2) Overall description

**2.1) Product perspective**

This product is self-contained, and not made up of additional, outside software. This is a diagram of the basic functionality of the software:** 2.2) Product functions**

****

**A):**

* This will display the location of the text file being used. In addition, if you click the blue text, the file currently in use will open for easy access.
* Below that, you will be able to use the search function by selecting a tab in the drop down window to limit the search to a specific column, type in your desired search, and the results will be displayed in the window below the search.

**\*\*\* Constraint #1: see Constrains section \*\*\***

**B):**

* **Delete Button:** Allows user to delete desired entry
* **Modify Button:** Allows user to modify desired entry. NOTE: modifying entries will be displayed in the "D" section of the GUI, towards the bottom.

**C):** Window that displays all available employee data. When a search is made, this window will immediately update and display all desired search information.

**D):**

* User may enter information to be submitted to the database here.
* Information includes: Last name, first name, benefits, taxes, and gross income.
* In addition, the Net Pay for the user will be calculated and displayed in real time.
* The check box is for the employee's paycheck options. If the box is checked off, the user will be marked for requesting payment with a check. If left unchecked, the user will get direct bank deposit.

**E):**

* **Submit Button:**  Will take the user information submitted in section "D" and update the database, as well as display the new input data in section "C" of the GUI.
* **Save Button:**  If the user chooses to modify information (see "B" section), they need to press this button to confirm the changes made.
* **Cancel Button:**  Will cancel any potential changes made to a particular data entry entered in "D" section during a modify.

**2.3) User Characteristics**

The intended audience for this product are people in payroll. They can manage their employee's payment options with ease and in real time with an accessible display window. Users should at least be able to comprehend basic data entry.

**2.4) Constraints**

**#1:** Typing inside of the "Search Option" will not allow you to filter your search. Users should stick with the template list of tabs from the drop down.

**2.5) Assumptions and dependencies**

* In order for any data to be displayed, there must be a text file being used.
* If there is no data inside the .txt file prior to being used, it will not display any information, but will be able to take in user data.
* The "Search Option" drop down box must not be tampered with by the user, or the search function will not work properly.

3. Specific Requirements

* program uses Circular Linked List to read and write data
* the code is written in C# on Visual Studio

When tested during an idle startup state on an Intel Core i5 processor rubbing Windows 7, the application used 27 Megabytes of RAM.

**Complications:**

* 10/7: GUI needed an enhancement to display employee info and state tax info
* 10/9: Need a newer version of Payroll Application to take out the State drop down list. Miscommunications were made between Chris and group with specific delegated assignments.
* Chris wants the rest of the group to master and use Github more, as we didn’t originally take full advantage of it.

Team Roles:

**Chris: (Team Leader)**

* created class prototype (functions and variables)
* main function (generating a text file to display that the worker has had a direct deposit or check given to them)
* delegated roles to the group
* gave presentation
* Proof read source code
* Fixed the GUI to be more user-friendly and presentable
* Fixed Errors

**Tuan:**

* created read in functions
* created text files and ability to read in lines from the text file into a new class, as well as writing new classes to a text file
* Created circular linked list based algorithm to store data
* Fixed Errors

**Daniel:**

* Wrote all documentations, including the SRS contract and team roles and lessons learned.
* Proof read source code
* Fixed Errors

**Stan:**

* Assisted with the creation of the Gross Income function
* Proof read source code
* Fixed errors
* Assisted in writing up documentation

**Michael:**

* Helped make the benefits function more robust
* Proof read source code
* Proof read SRS documents
* Fixed errors

**ETA: 12 days**

**ATA: 23 days**