[Pick the date]

Team CyberDragon

iCyberSecurity Internet Pamphlet Project

Chris Bassar

Team CyberDragon

iCyberSecurity Internet Pamphlet Project

Introduction

This project was developed in response to the need for an internet pamphlet, selling our skills as secure programmers. Serving only top clients, we took a Spartan approach to the advertising element while allowing users to efficiently communicate with us through dialog windows. We believe this will function on all operating systems with little or no compatibility issues. Administrative users have maximum flexibility in adding and deleting customers, or otherwise managing the database.

By providing all possible information to the customer up front, and allowing them to access their information in their own login to the upper right, both guests and customers can readily see what we, the administrators wish them to see. Our Admin window is only available through a file prompt. As long as we don't distribute the admin password to the public, the data is safe from harm, and our administrative team will be able to access what we need to see and control. It is a simple text file comparison, in a wrapper class, the data separated from the front end.

This document represents the final submittal and includes the final deliverables as requested by the sponsor (Professor). The best example of our Agile development having been submitted with the Project Scope Report (signed version), we are requesting that those be used for final evaluation (grading). Should you select our product for distribution it is possible to implement it immediately.

The URL to the GitHub Repository has been previously provided to you as follows:

https://github.com/JaredRauch/Cyber-Dragon-Repository

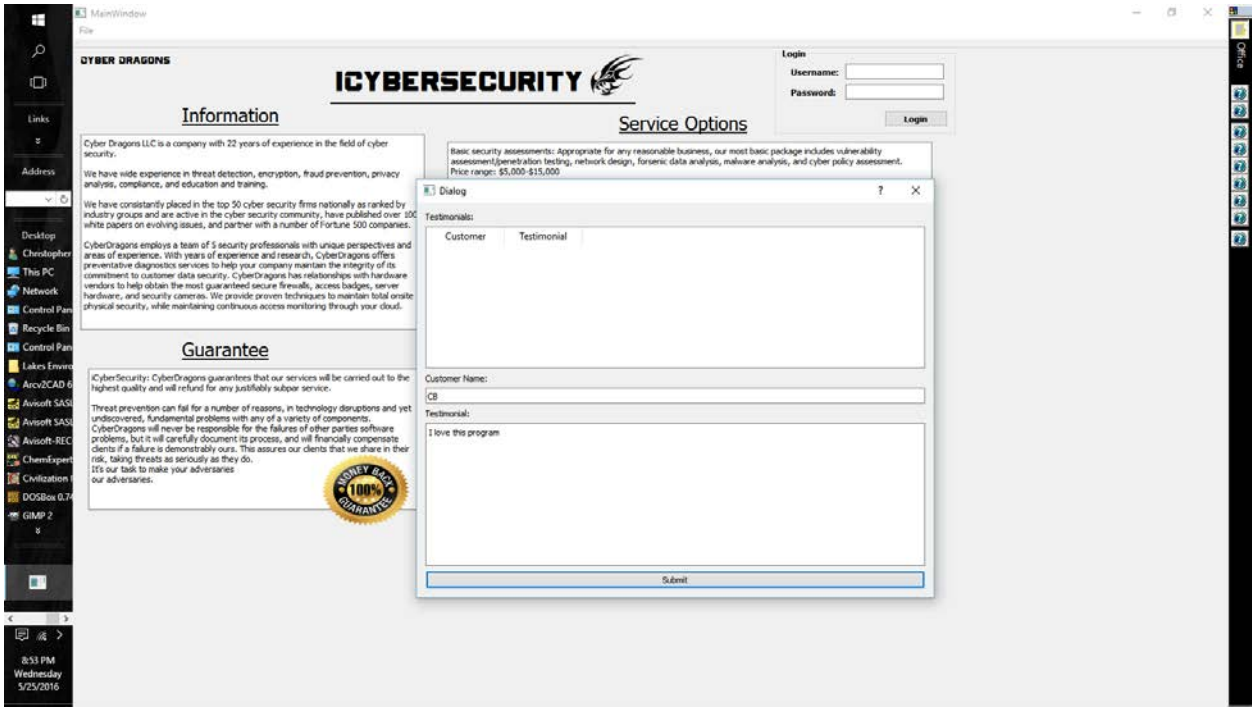
Zip Files to be attached in Email.

QT Project- Finished Product (Bitmaps of Project)

Opening Screenshot

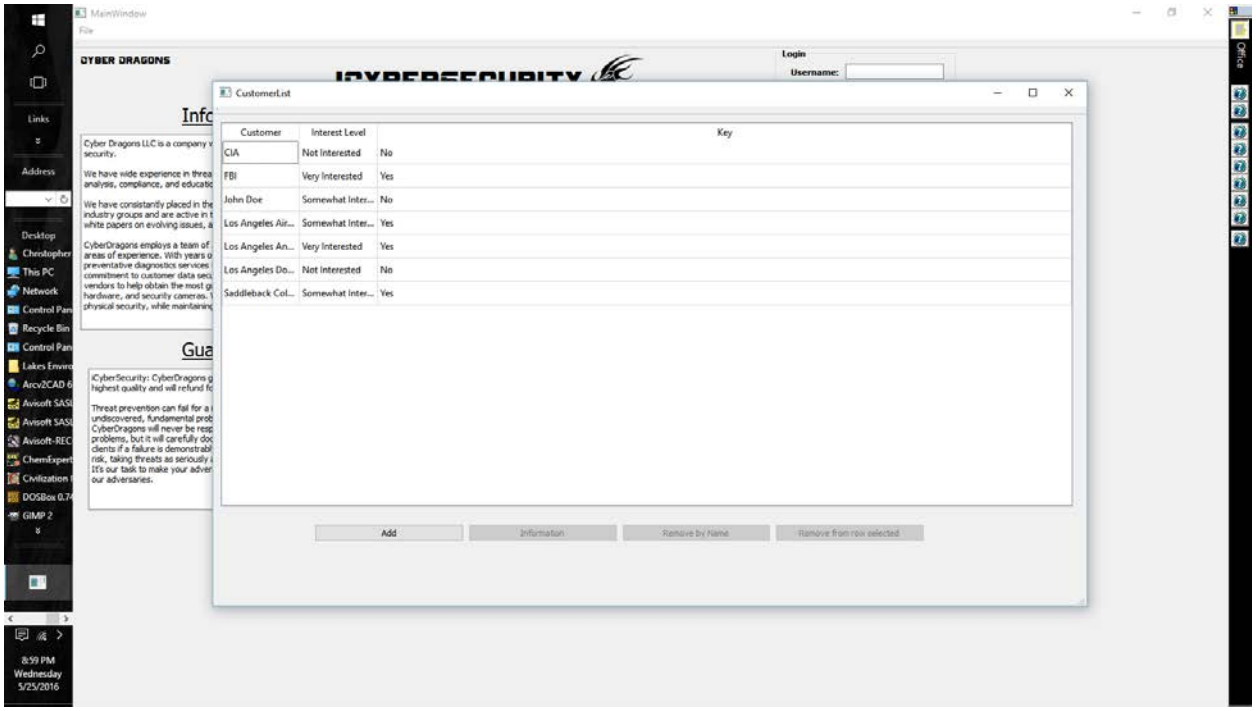
Request Dialog

Testimonial Diaolog

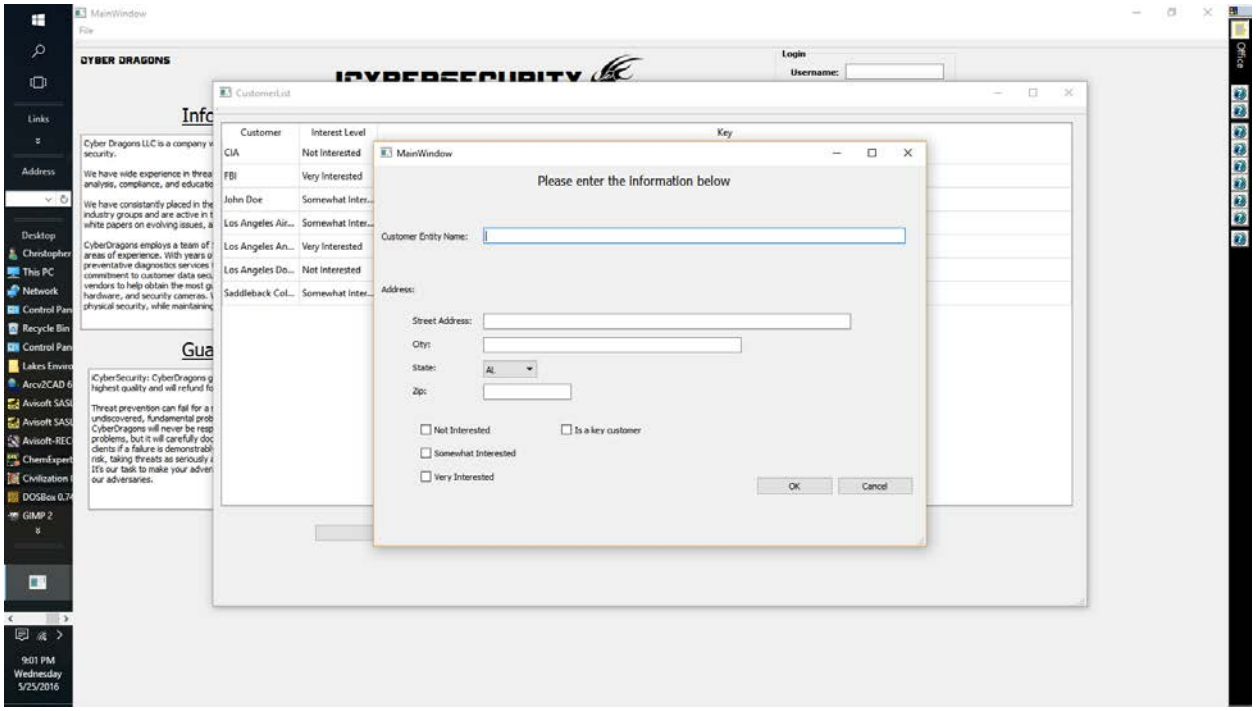


Order Dialog

Admin Window



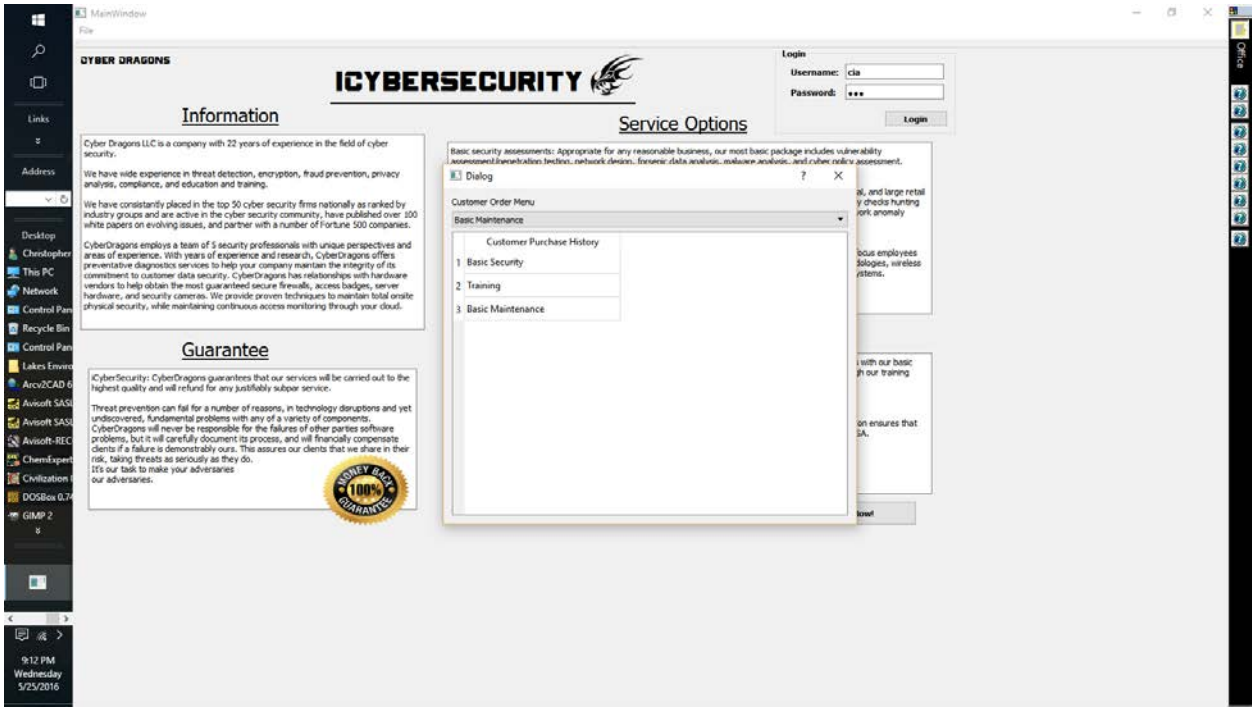
Add Customer Window



Instruction Dialog

iCyberSecurity Contact Information Window

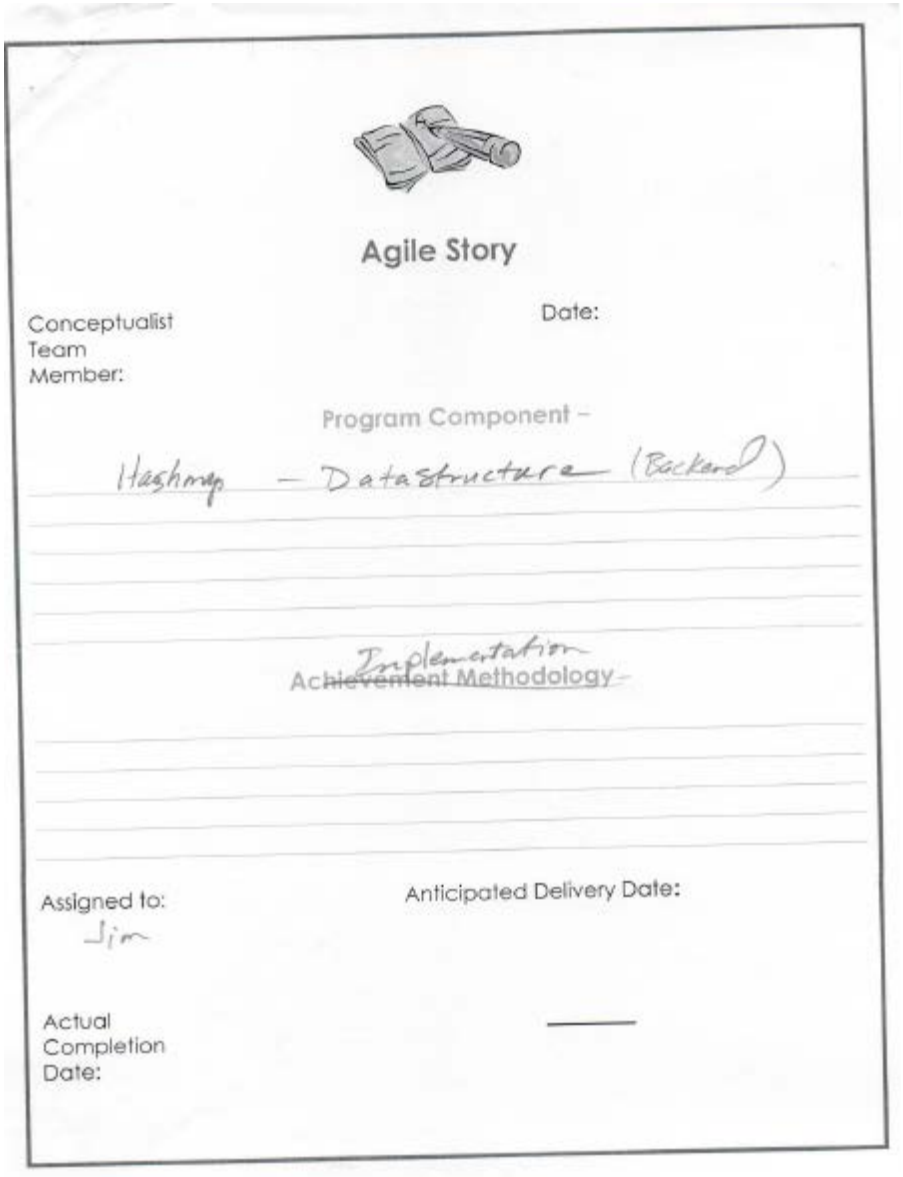
CIA Customer Login and Order Details



Initial Agile Stories (subjected to change)

Herein are the initial Agile stories requested in the first week of the project, beginning on April 20,2016. The project ideas changed between the time the initials were drafted and when the final Agile was submitted at Checkpoint 1, on 5/2/2016. Hence the best Agile Stories are in the PSR submittal for this project.

The following are "Note" form. Please use the PSR submittal. All team members were asked to develop 10 stories on the first round.



Agile Story

Conceptualist Team Member:

Date:

Program Component –

Hashmap – Datastructure (Backend)

Implementation

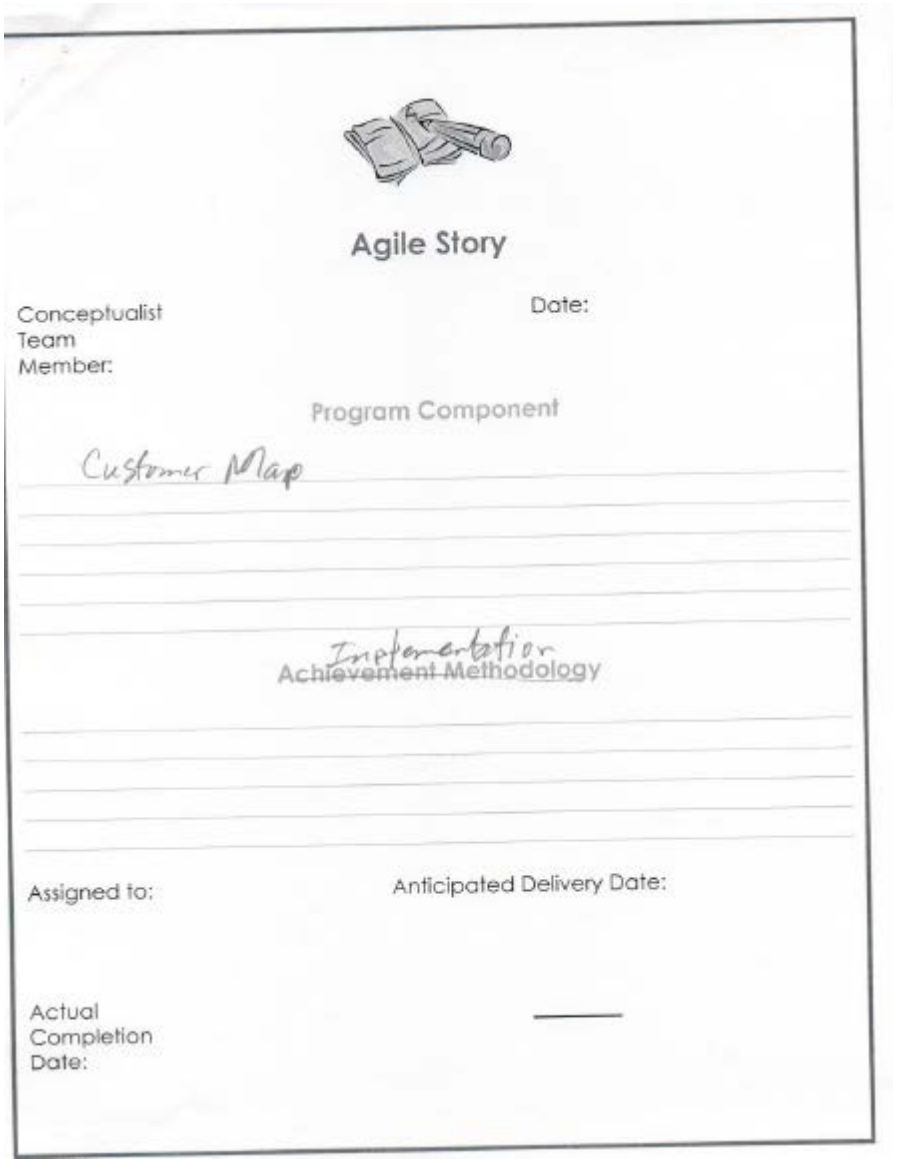
Achievement Methodology –

Assigned to:

Jim

Anticipated Delivery Date:

Actual Completion Date:



Agile Story

Conceptualist Team Member:

Date:

Program Component

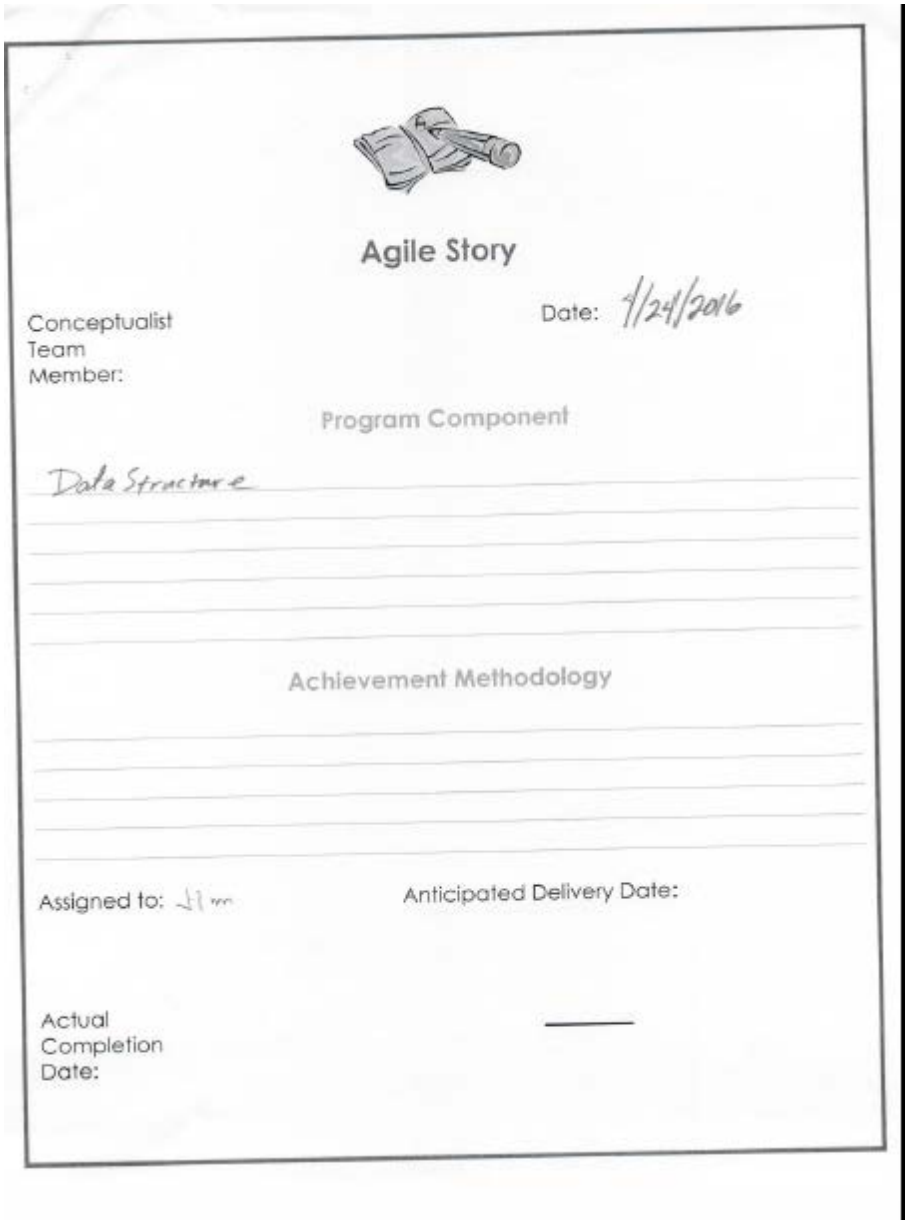
Customer Map

Implementation
~~Achievement~~ Methodology

Assigned to:

Anticipated Delivery Date:

Actual Completion Date:



Agile Story

Conceptualist Team Member:

Date: 4/24/2016

Program Component

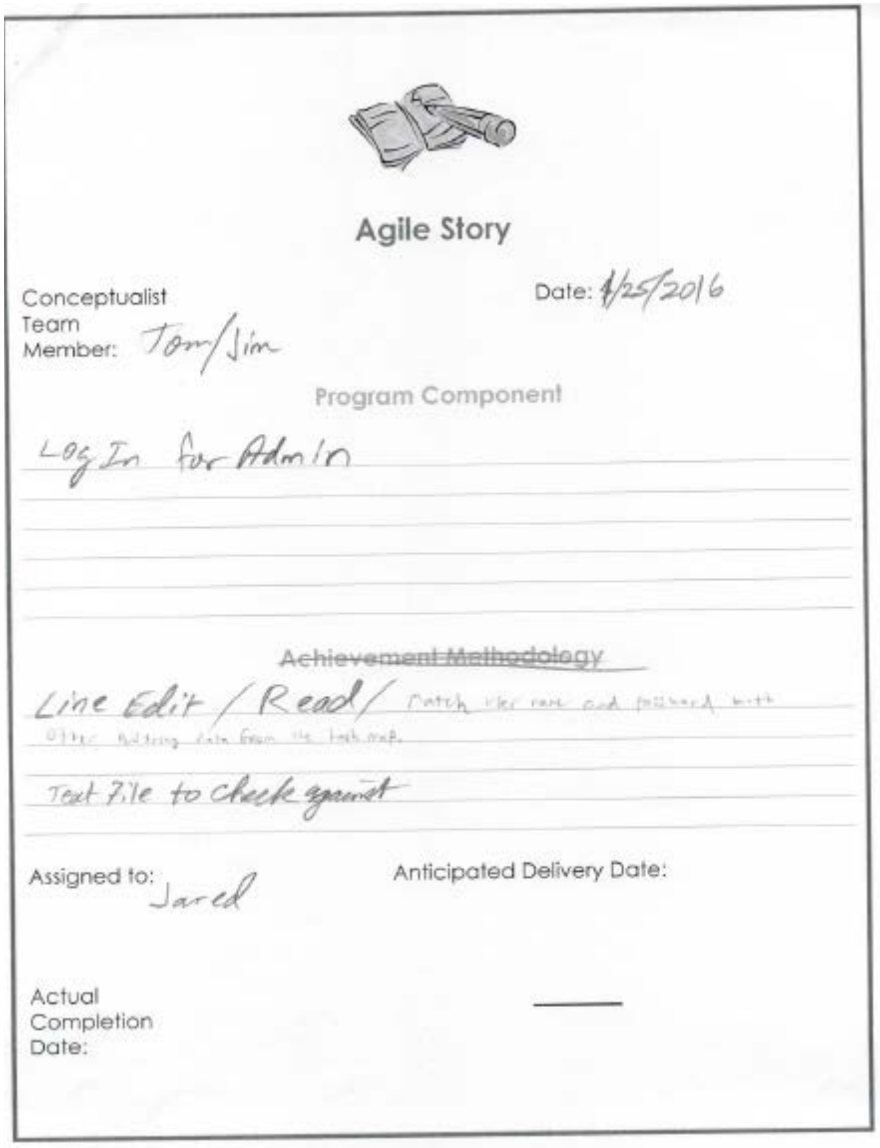
Data Structure

Achievement Methodology

Assigned to: Jim

Anticipated Delivery Date:

Actual Completion Date:



Agile Story

Conceptualist Team Member: Tom/Jim

Date: 4/25/2016

Program Component

Log In for Admin

~~Achievement Methodology~~

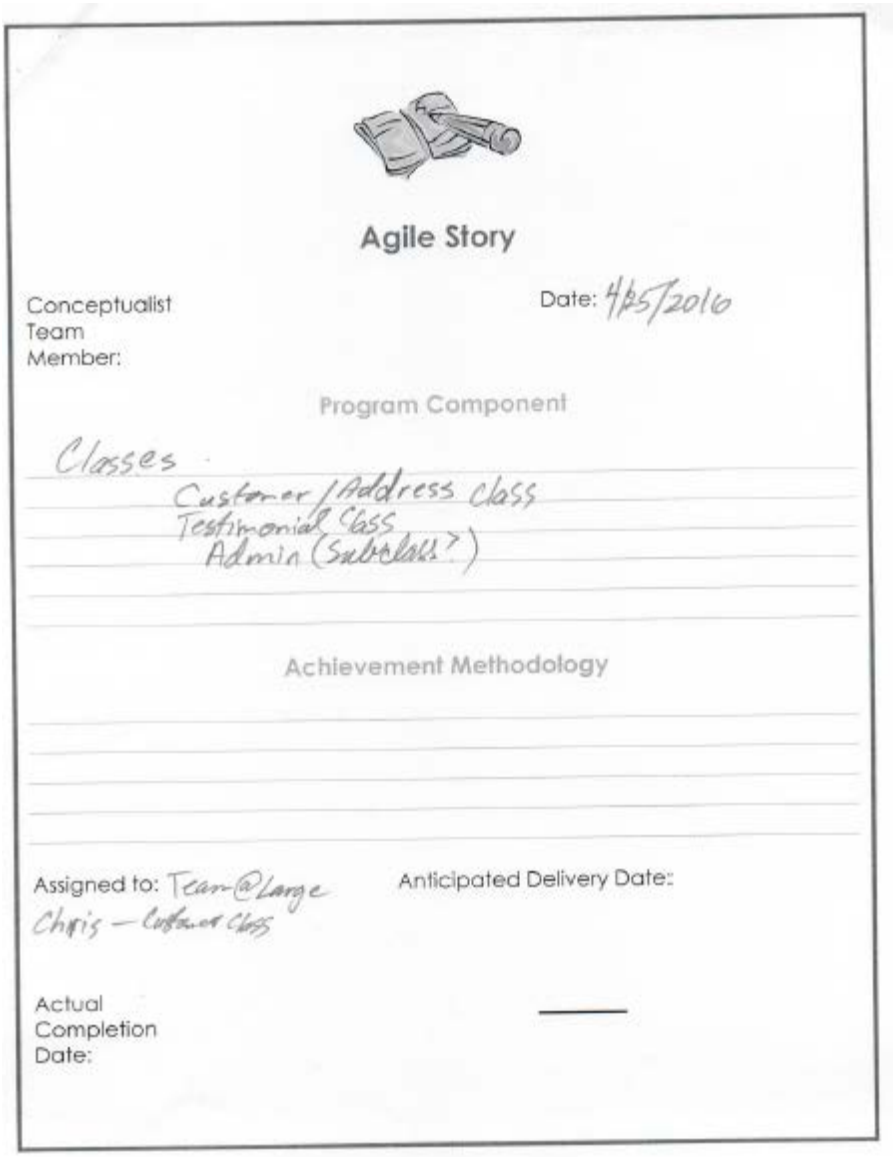
Line Edit / Read / Match user name and password with other matching data from the text file.

Text File to check against

Assigned to: Jared

Anticipated Delivery Date:

Actual Completion Date:



Agile Story

Conceptualist Team Member:

Date: 4/25/2016

Program Component

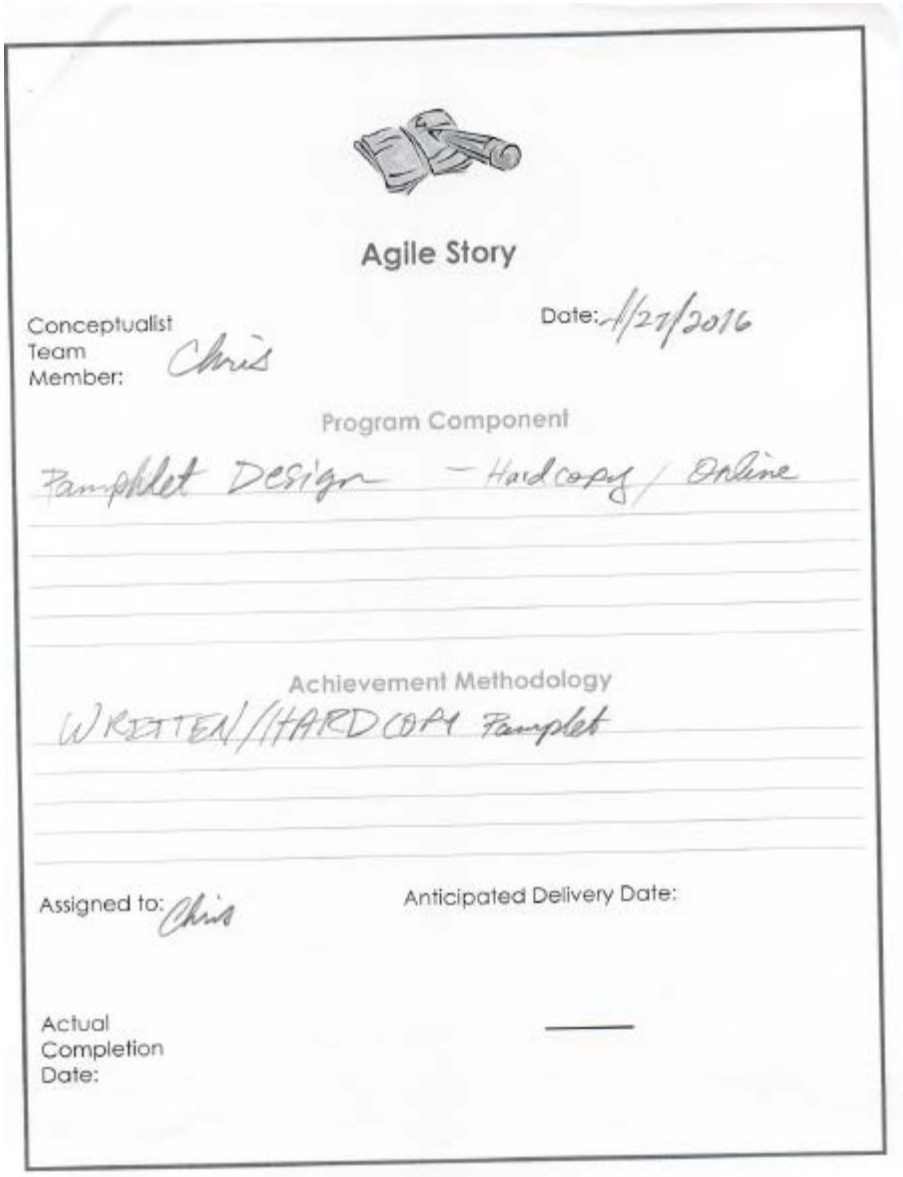
Classes
Customer / Address class
Testimonial Class
Admin (Subclass?)

Achievement Methodology

Assigned to: Team@Large
Chris – Customer Class

Anticipated Delivery Date:

Actual Completion Date: _____



Agile Story

Conceptualist Team Member: Chris

Date: 4/27/2016

Program Component

Pamphlet Design – Hard copy / Online

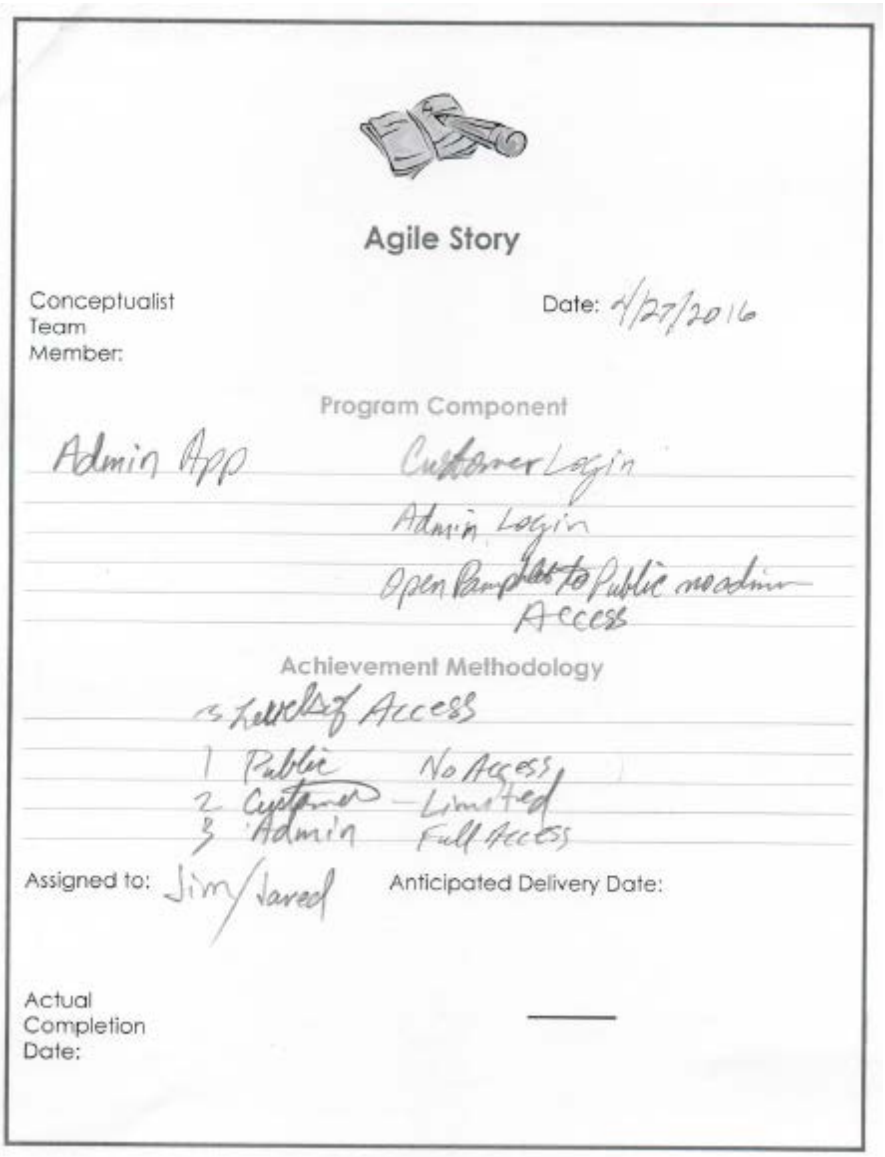
Achievement Methodology

WRITTEN/HARD COPY Pamphlet

Assigned to: Chris

Anticipated Delivery Date:

Actual Completion Date: ———



Agile Story

Conceptualist Team Member:

Date: 4/27/2016

Program Component

Admin App

Customer Login

Admin Login

Open Page ~~Not~~ to Public no admin Access

Achievement Methodology

3 Levels of Access

1 Public No Access

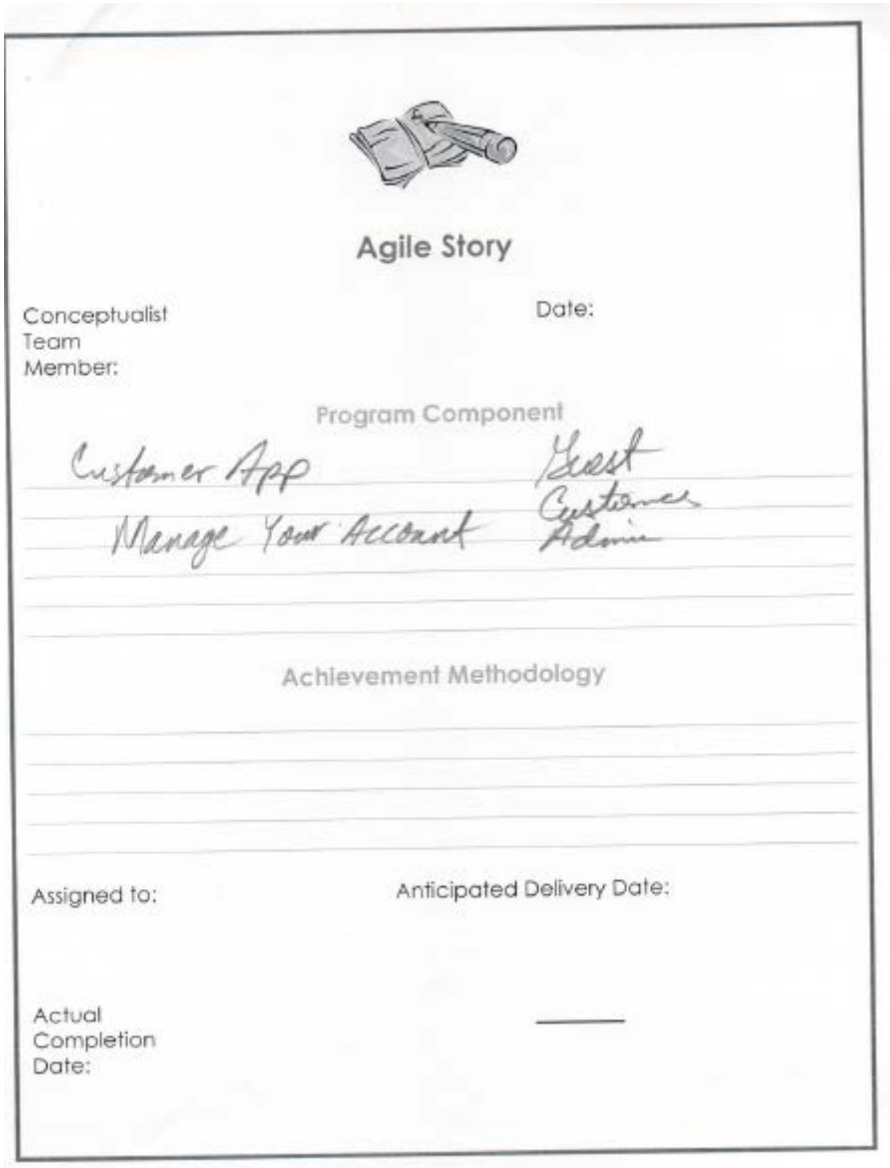
2 Customer Limited

3 Admin Full Access

Assigned to: Jim/Jared

Anticipated Delivery Date:

Actual Completion Date:



Agile Story

Conceptualist Team Member:

Date:

Program Component

Customer App — Guest, Customer, Admin

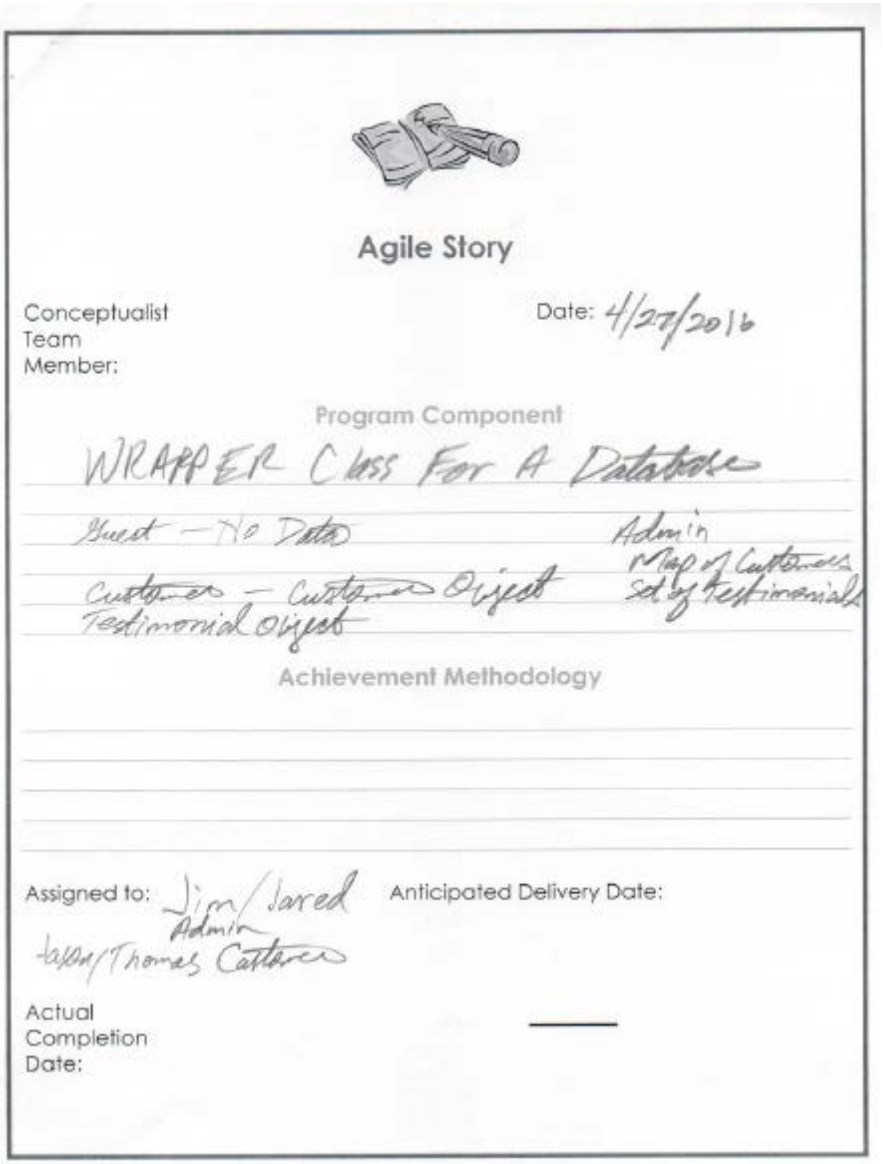
Manage Your Account

Achievement Methodology

Assigned to:

Anticipated Delivery Date:

Actual Completion Date: _____



Agile Story

Conceptualist Team Member:

Date: 4/27/2016

Program Component

WRAPPER Class For A Database

Guest – No Data

Admin

Map of Customers

Set of Testimonials

Customer – Customer Object

Testimonial Object

Achievement Methodology

Assigned to: Jim/Jared

Admin

Jason/Thomas Customer

Anticipated Delivery Date:

Actual Completion Date:

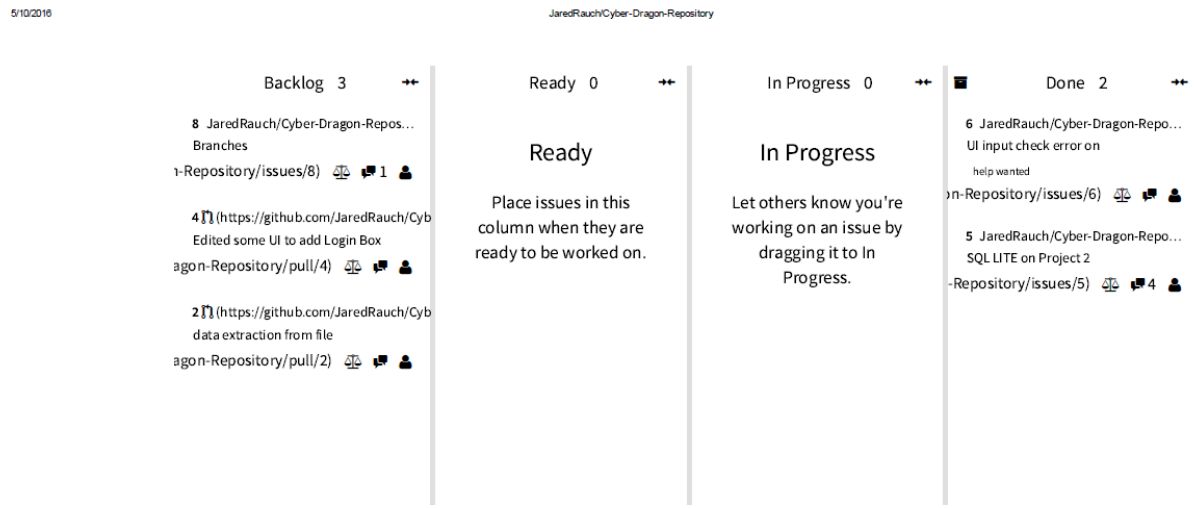
Thomas Gorney Agile

PRIORITY	AS A/AN	THEME	I WANT TO	SO THAT	NOTES	STATUS
1	Developer	User Interface	Set up the skeleton of the UI	Developers can being work on features		ToDo
2	Developer	Back end	Create two classes for administrator	Admins and users can log in and see different interfaces	Different UI elements based on class	ToDo
3	Developer	Back end	Set up linked list/dynamic array for user list	We can access and modify the file	Recommend linked list	ToDo
4	Developer	Back end	Begin work on UML diagram	We can organize our data		ToDo
5	User	User Interface	Be able to log in as a user or admin	Our UI views change based on who logs in	Can be done through a login window	ToDo
6	Developer	Back end/UI	Work on admin panel to allow addition and deletion of users and sending	We can change the user list based on admin command and send pamphlets		ToDo
7	Developer	Back end/UI	Work on user panel to allow for ordering products	We can update the order information based on user selection		ToDo
8	Developer	Back end	Create class for different product options	Determine which product to add to their cart	There are 3 service options	ToDo
9	Developer	Commenting	Finish Doxygen commenting for final product	It has a nice readable HMTL file		ToDo
10	Developer	Testing	Clean up and test the project	The product is finished and working for presentation		ToDo

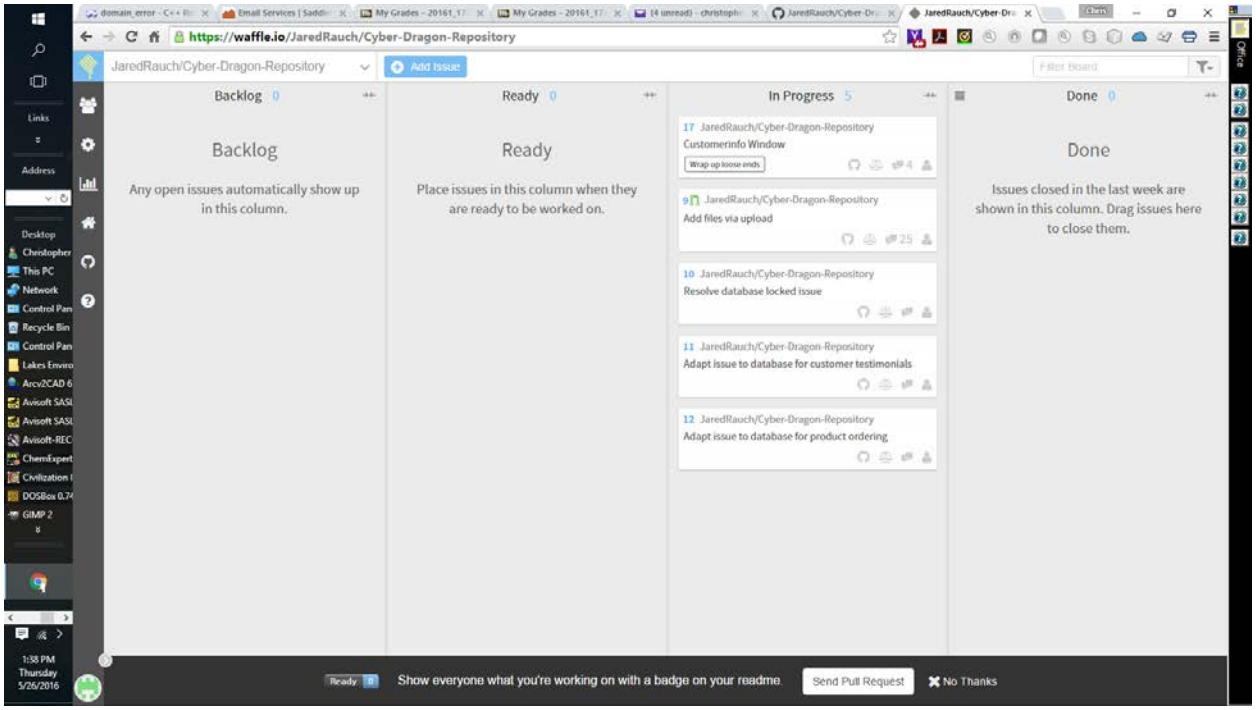
Scrum Procedures and Product BackLog (From the Wiki and Waffle)

Our scrum and backlog were produced formally throughout. While Chris began the meeting minutes and backlog, by the second checkpoint, Jason was actively participating in writing meeting minutes. Jared's github ID was used as the main repository, and as the GitHub Manager, he had the Maintenance tasks , and some of the organizational aspects. James came up with most of the structural Ideas for the database, and the team had to let him develop the hashmap and database, thus, most of the Scrum and Agile was not up to Jim. Tom was identified as a second sprint Scrum Master/Product Owner, however, it was later determined that Chris was the best "point man" for this task throughout the project by the team.

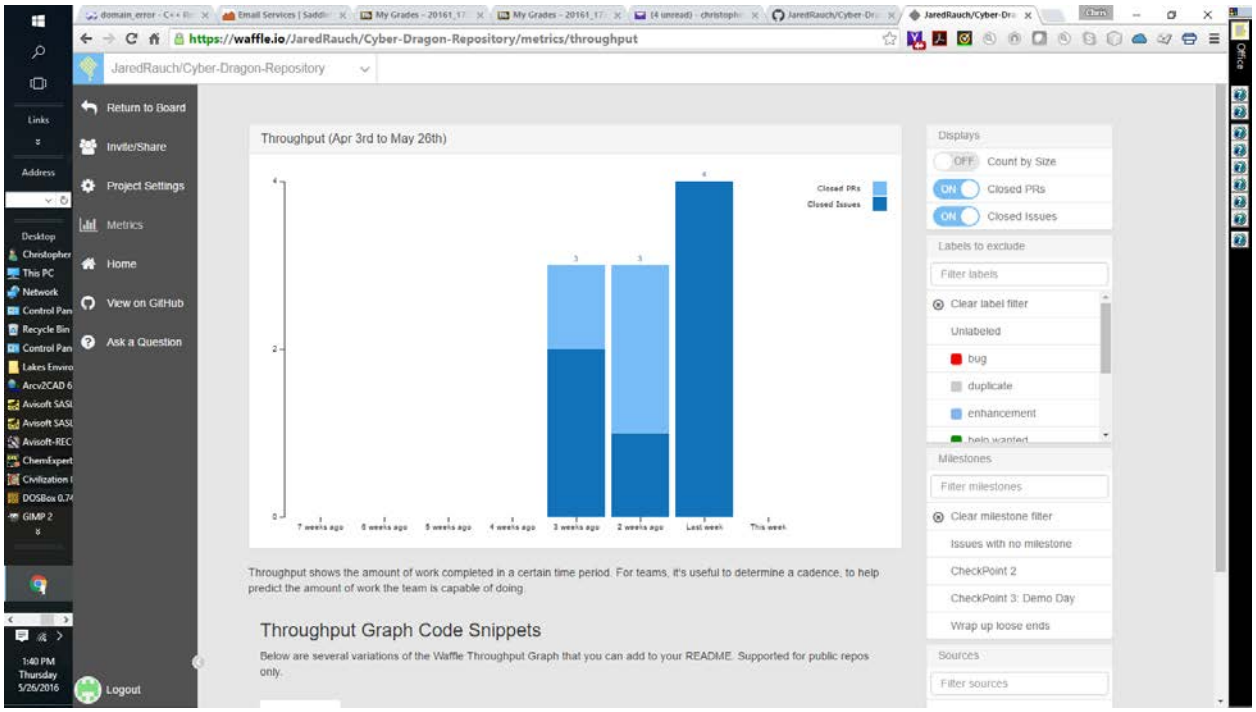
Waffle at CheckPoint 2



Waffle at CloseOut



Final Graphs



Meeting Minutes from Wiki Page from 4/20/2016-5/9/2016

Kick Off Meeting Minutes

Cyber Security Project

Date: April 20, 2016

Agenda:

1. Meet Team Members
2. Establish Roles for the Project
   a. Sprint 1
      i. Chris Bassar
         1. Project Owner/Manager/Scrum Master
         2. Developer
      ii. Jared Rauch
         1. Developer
         2. GitHub Manager for entire project
      iii. Thomas Gorney
         1. Doxygen/ Documenter
         2. Developer

iv. Jason Weisner

1. Developer

2. Miscellaneous

3. Exchange Information

a. Jared Rausch

b. Chris Bassar

c. Thomas Gorney

d. Jason Weiser

4. Read over Assignment

a. Project BreakDown

b. Component Parts

5. Preliminary Discussion

6. Agile Stories/UML

a. Develop PseudoCode-Each member will draft their own ideas

b. Discussion by team on components

7. Data Storage/ Manipulation

a. Considered SQL

```
i.    Voted against due to skill level of team

ii.   Dynamic Array

iii.  Database/file browser

iv.   Administrative Functions

v.    Pamphlet Requirements

8.    Started QT Layout

a.    Based on Preliminary Requirement Discussion
```

Second Meeting- 4/27/2016

1. Action Items for next meeting
    i. Prepare Project Study Report (Budget) by Checkpoint 1
    ii. Prepare Project Design Report (UML/PseudoCode) by Sprint 2.
    iii. Establish preliminary Testing Plan
    iv. Final Testing Plan by Start of Sprint 2.

v. Each team member will develop 10 Agile stories

vi. Vote on Best at the next meeting

vii. Code Uniformity will be decided by 2 meetings from next.

1. Discuss New Software Architecture a. SQL/SQL Lite may be required to wrap a database i. Professor clarified assignment instructions to define a tri-layered architecture
2. external no password required
3. customer password access required for customer interface
4. Administrative access interface
5. PHP/WebServer floated
6. Himachi (all jim)
7. Bilevel access considered where 2 clients suddenly became 1.

Action Items. 1. Thomas and Jason to work on customer interface, while Jim and Jared work on Administrator interface.

Completed SubTasks

1. Customer List Completed.

Meeting 3, 5/4/2016

With most of the different aspects of our project coming together and the deadline approaching in a week and a few days, we reviewed how our as of now completed work will sync together, overcame problems with the database syncing with the login, and split up the remaining work.

Thomas - Improve ui appearance James - continue testing backend Jason - Further integrate window ui system Jared - After James's notfication, work on data manipulation in Admin UI files. Chris - waffle.io

Meeting # 4, 5/9/2016

Decided to do final linkup on both uis, do the design report finished, finalize agile for the homework assignment, pamphlet finished, just need to link to it.

James, Jason, Jared will send UI diagrams to be compiled into report.

James will send black box testing of backend, for test plan.

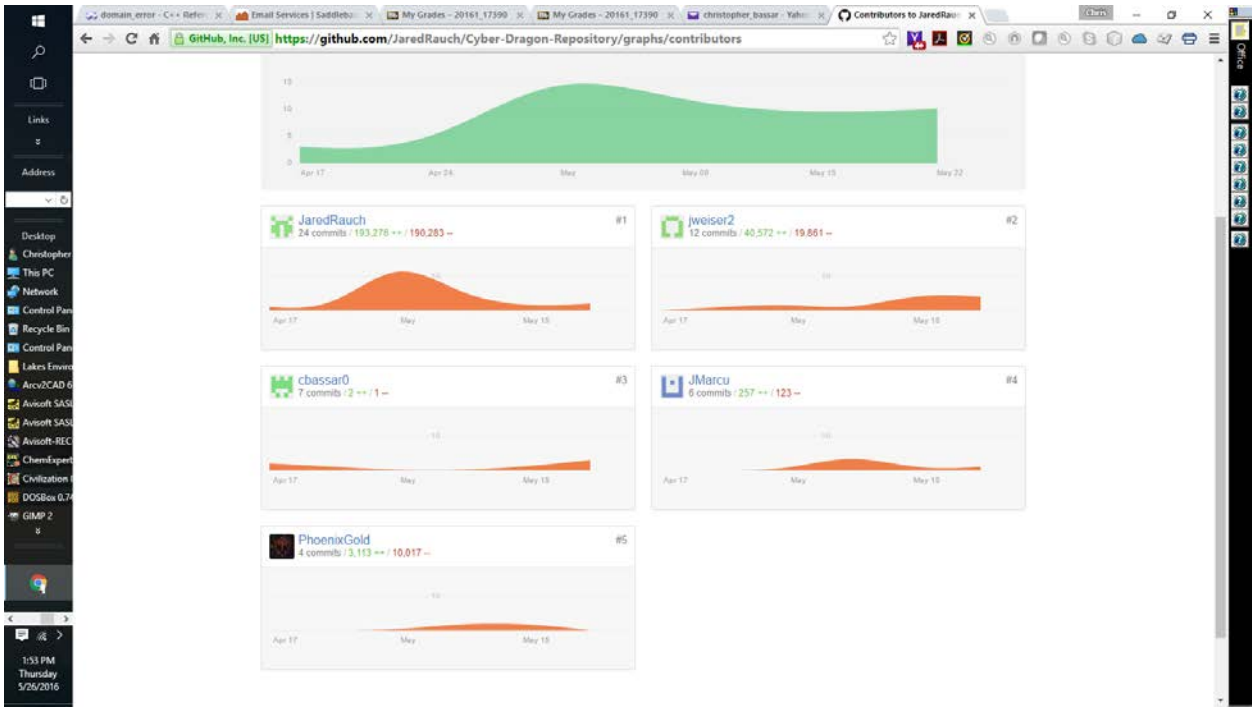
Thomas revised overall design of login window to be more simplified.

Waffle.io has been accessed and should be running.

Project backlog should exist.

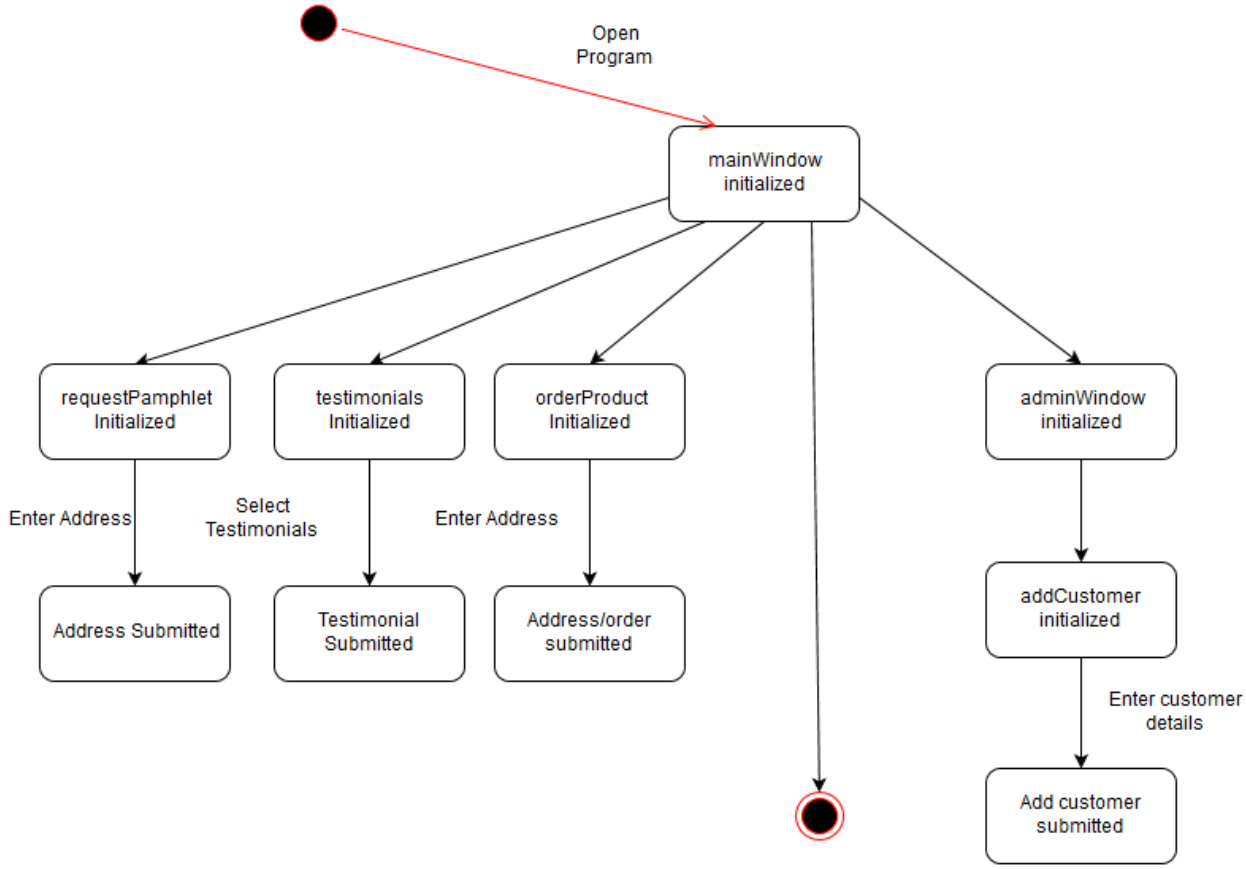
5/16/ 2016 –Demo Day Occurred

Contributor/ Contribution to GitHub Graph

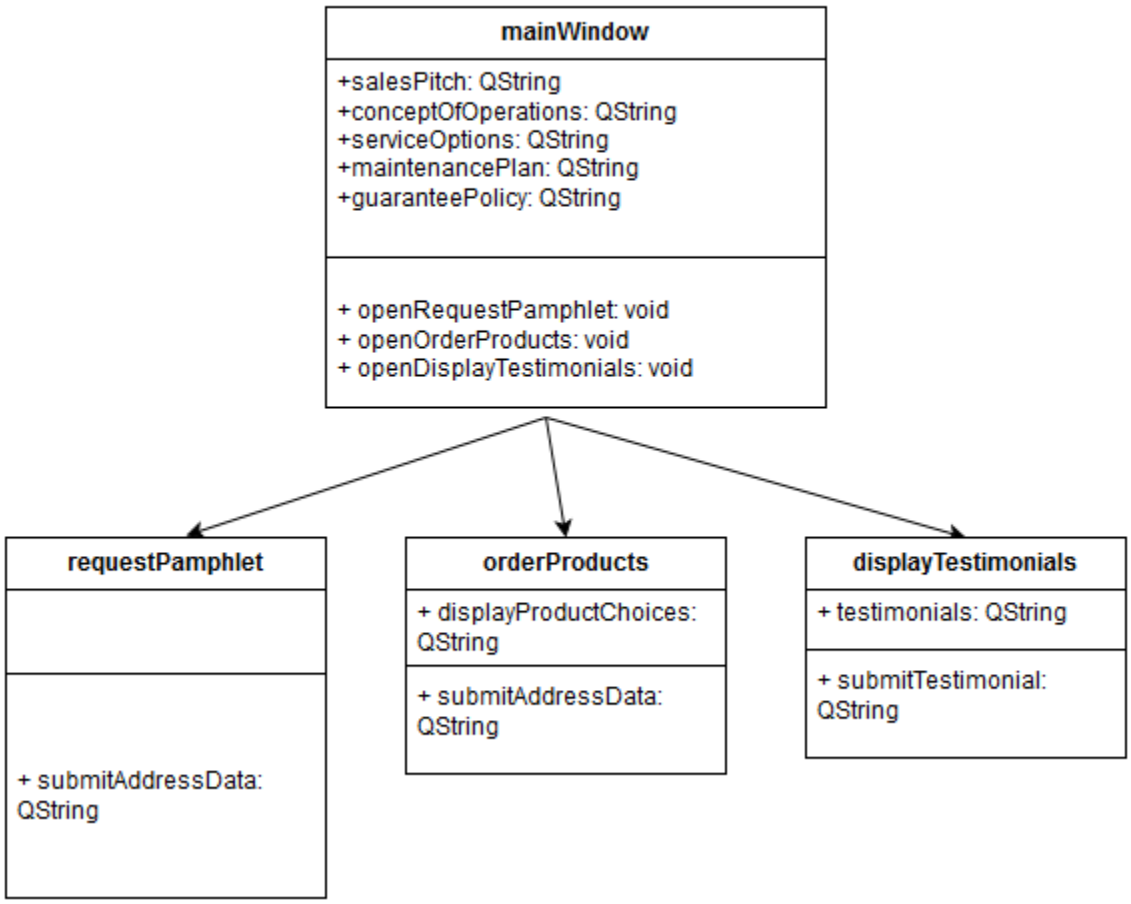


UML Class Diagrams and Use Cases (from Software Design Report)

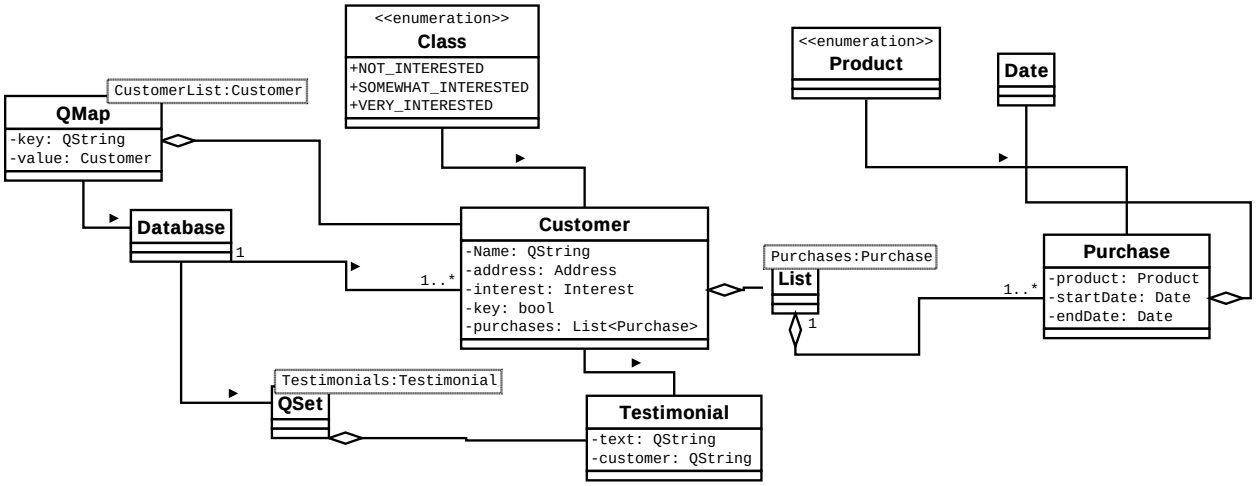
Main Introductory Window and General Program Execution



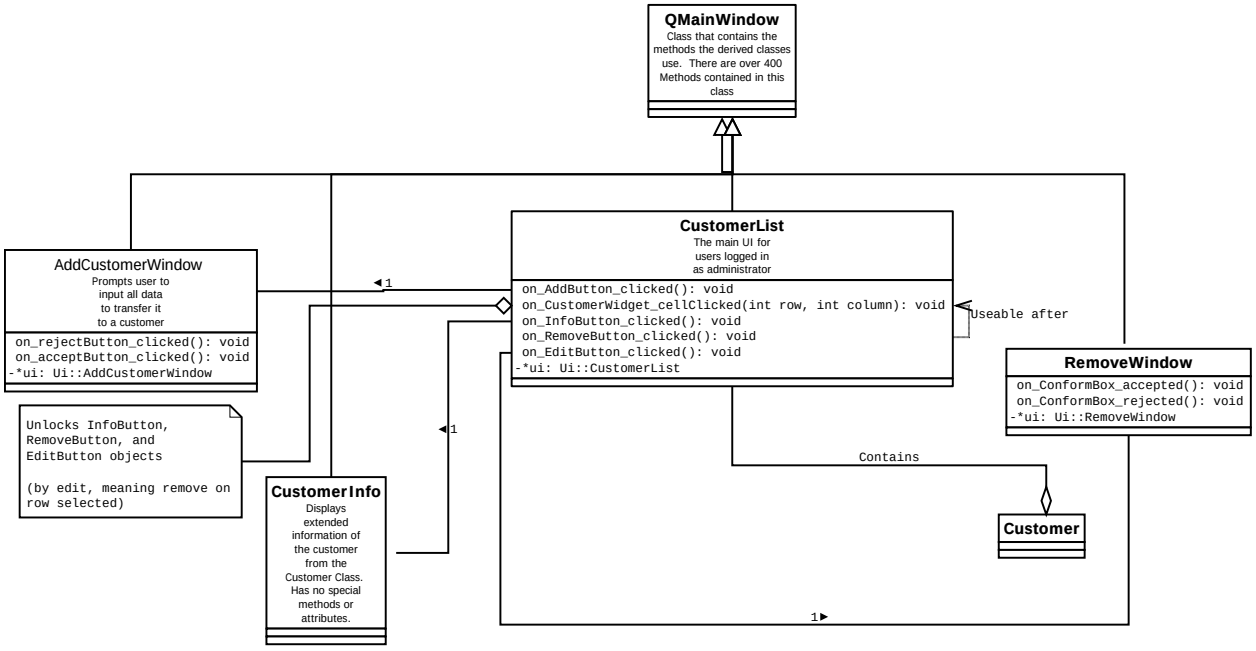
Guest UML class diagram

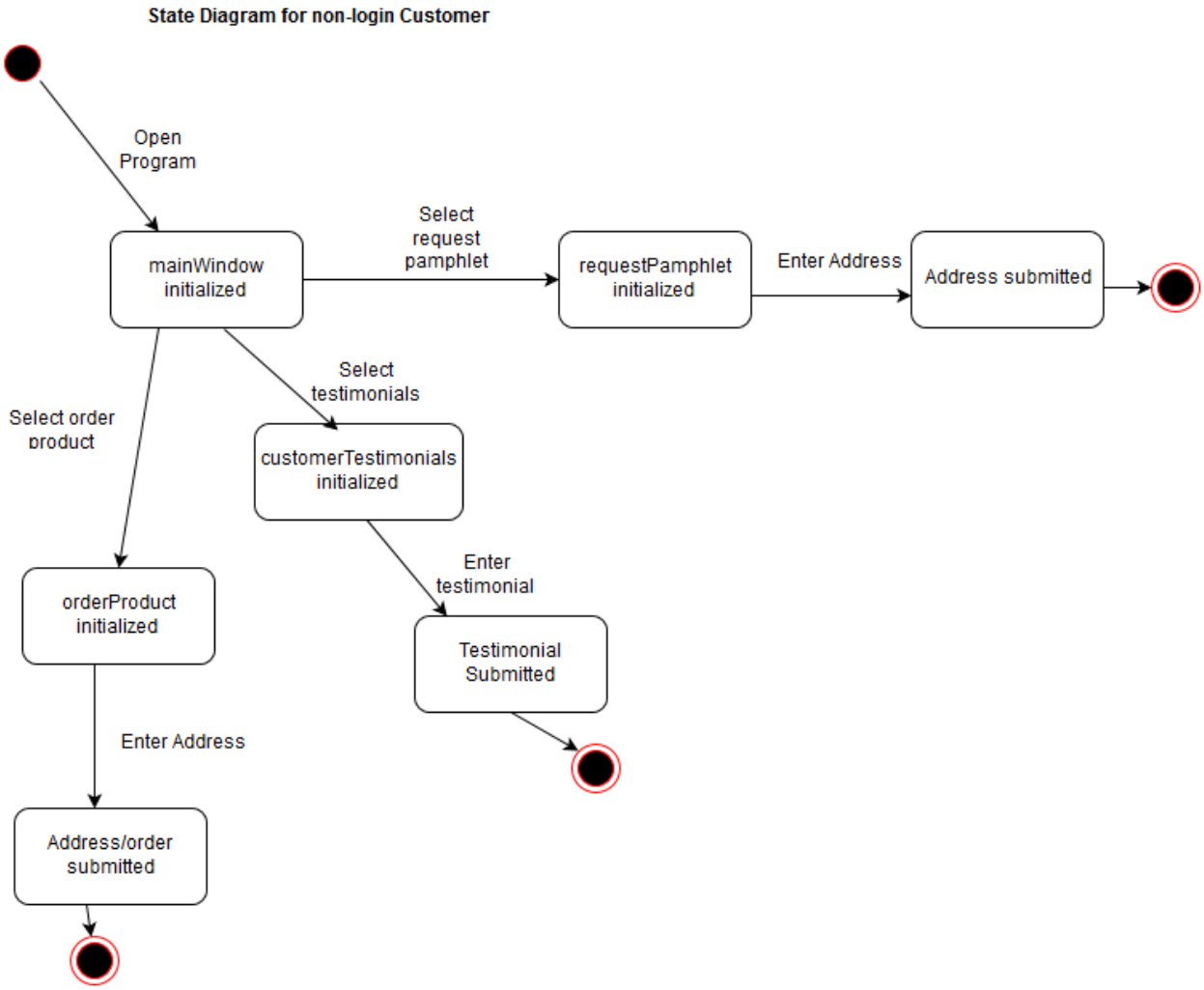


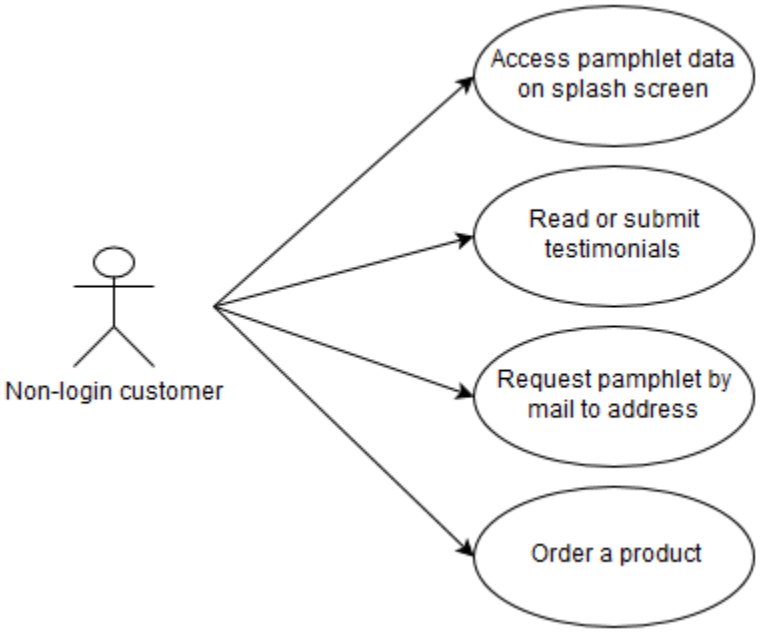
Class UML



Admin UI diagram




State Diagram for non-login Customer
Open Program
mainWindow initialized
Select request pamphlet
requestPamphlet initialized
Enter Address
Address submitted
Select order product
Select testimonials
customerTestimonials initialized
orderProduct initialized
Enter testimonial
Testimonial Submitted
Enter Address
Address/order submitted



Class Diagram Content

Name: mainwindow

Attributes: salesPitch: Qstring

conceptOfOperations: QString

serviceOptions: Qstring

maintenancePlan: Qstrig

guaranteePolicy: QString

Methods: login:

openRequestPamphlet:

openOrderProducts:

openDisplayTestimonials:

Name: requestPamphlet

Methods: submitAddressData: QString

Name: orderProducts

Attributes: displayProductChoices: QString

Methods: submitAddressData: QString

Name: displayTestimonials

Attributes: testimonials: QString

Methods: submitTestimonial: QString

Use Case Content

Access Pamphlet Data on Splash Screen

Use Case Number: 1

Application: iCyberSecurity non-login access

Use Case Name: Pamphlet Data

Use Case Description: Access to the various data entries regarding the sales pitch, the concept of operations, the service options, the maintenance plan, and the guarantee. Anyone can access this information.

Primary Actor: Anyone who opens the program should have access to this information

Precondition: None

Trigger: Opening the program

Basic flow: User opens program, reads through data

Alternate flows: Unless the program fails, or the data is corrupted, this use case has no alternate flows or exceptions

Read or submit testimonials

Use Case Number: 2

Application: iCyberSecurity non-login access

Use Case Name: Testimonials

Use Case Description: User can open the testimonials section and immediately read other users submitted experiences. In a window below the testimonials, they may submit their own and immediately see the testimonial added.

Primary Actor: Anyone who opens the program should have access to this functionality

Precondition: The program should check that this person is a verified customer

Trigger: Opening the testimonial section on the splash menu.

Basic flow: The program operates normally, specifically, the database displays information in the testimonials window without problems, and the window accepts data from the customer, displaying it on submission.

Alternate flows: The database becomes corrupted, or the program does not start

Request pamphlet by mail to address

Use Case Number: 3

Application: iCyberSecurity non-login access

Use Case Name: Mail pamphlet

Use Case Description: User can enter their address and be entered into data base, for processing to send to their address.

Primary Actor: Anyone who opens the program should have access to this functionality

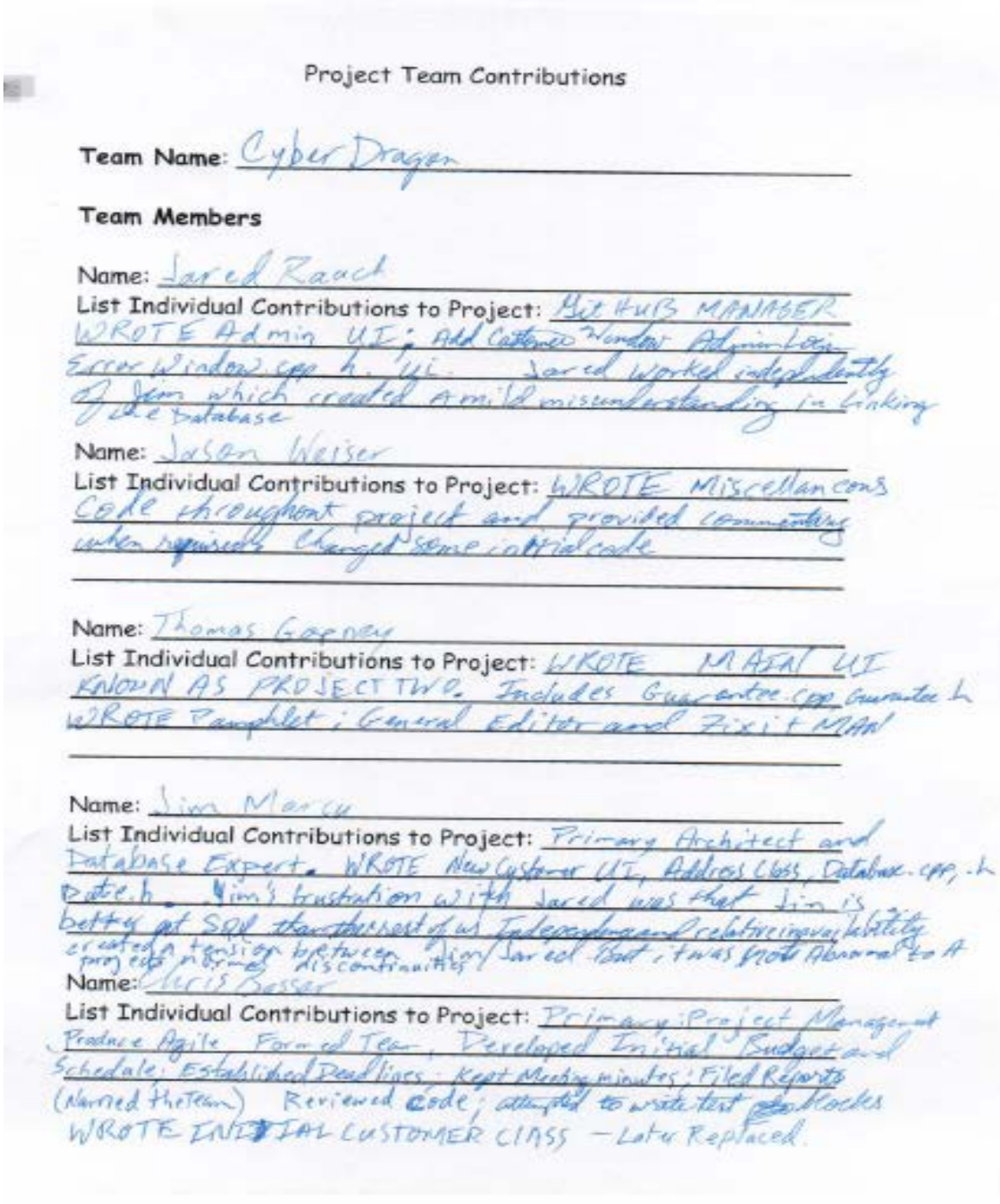
Precondition: None

Trigger: Access order window and enter address

Basic flow: The user opens the program, opens the order window, and enters address. The database accepts the address

Alternate flows: The program fails to open, the window fails to open, or the database fails to accept the data entry

Team Participation/Contribution Form



Project Team Contributions

Team Name: Cyber Dragon

Team Members

Name: Jared Raudt
List Individual Contributions to Project: Git HUB MANAGER WROTE Admin UI; Add Customer Window Admin Login Error Window.cpp h. ui. Jared worked independently of Jim which created a mild misunderstanding in linking the database

Name: Jason Weiser
List Individual Contributions to Project: WROTE Miscellaneous Code throughout project and provided commenting when required. Changed some initial code

Name: Thomas Gagney
List Individual Contributions to Project: WROTE MAIN UI KNOWN AS PROJECT TWO. Includes Guarantee.cpp Guarantee.h WROTE Pamphlet; General Editor and Fix it Man

Name: Jim Mercu
List Individual Contributions to Project: Primary Architect and Database Expert. WROTE New Customer UI, Address Class, Database.cpp, .h Date.h. Jim's frustration with Jared was that Jim is better at SQL than the rest of us. Independent relative inexperience created a tension between Jim/Jared but it was more Abnormal to A project normal discontinuities

Name: Chris Bassar
List Individual Contributions to Project: Primary Project Management Produce Agile Formed Team, Developed Initial Budget and Schedule; Established Deadlines; Kept Meeting minutes; Filed Reports (Named the Team) Reviewed code; attempted to write test stubs WROTE INITIAL CUSTOMER CLASS – Later Replaced.

It is fair to say that all team members assisted with development of the software design report, as their components had to be integrated.

Major Contributions of Team Members

Chris Bassar-Product Owner/Scrum Master, Designer

Product Owner/Scrum Master

Organized Team, came up with team name, and team logo.

Agile Development

Scope/Schedule, and Project Progression Documentation

Wrote initial customer class, later scrapped

Wrote Initial Doxygen

Wrote initial Pamphlet as guide (in Agile Folder)

Ran Independent Black Box Testing.

Tended to be consumed by process and procedure than writing code-lost track of where the team was at a point with regard to the code.

Most likely perceived as a Boss figure by some.

Jared Rauch- Designer

Managed GitHub for Project

Wrote Admin UI

Provided diagram of his code.

Occasionally acted to try to organize the team (Scrum Master Role).

Demonstrated interesting features in his component (Slider)

Found linking to James' components challenging.

Engaged in the success of the project.

Thomas Gorney- Designer

Wrote Main UI.

Came up with miscellaneous items like the guarantee symbol, and flying dragon symbol, (noted by Professor in Demonstration).

Participated in Agile Development by providing 10 individual Agile Stories as requested by Product Owner/Scrum Master.

Wrote Pamphlet elements as part of the Main UI.

Provided diagram of his code.

Wound Up writing the master *.pro file, project two.pro, out of necessity.

Jim Marcu – Senior Designer

Came up with the Big Picture in Agile development.

SQL/Hashmap was James' Idea

Designed Customer Class associated to a Guest and customer Login.

Provided diagram of his code.

Wrote a "Linking to the Database" Instruction Manual on Wiki of GitHub

Stated to Project Owner- "We lost a week because of Jared", to which the Project Owner replied "I know .....Off in his own world. But that's normal on large team interactions. ";

James was frequently unavailable by phone, but would send in critical components and assist with Jared's confusion when needed.

Jason Weiser- Designer Level 2

Tried to learn SQL when James was unavailable, not always successful.

Occasionally entered meeting minutes and product Backlog.

Wrote Order Dialog box, and other dialogs as needed in the Main UI.

Felt more comfortable with Linked Lists as a data storage structure.

Displeased by having to use SQL.

Provided Guest Diagram of Code.

Wrote Use Cases for the Software Design Report

Cooperative and willing to do anything Attitude.

Team Interaction /

A limitation to success is that James was the only knowledgeable SQL programmer.

Despite our squabbles, this was a well functioning team willing to work hard to get things done on schedule. Blaming was not significant, and it allowed us to work toward our objective , utilizing each other's best skills.