Code Miners

*“Never dig straight down!”*

**-SCRUM LOG-**

26 OCT 2019:

In attendance:

Moe Soliman

Po Jen Su

Adam Hurd (Scrum Master)

George Good

Not in attendance:

Sami (Product Owner)

Topics: Team rules, coding standards, initial Agile stories

Team rules:

1) Be on-time for meetings

2) Finish assignments before sprint checkpoints

3) Give advance notice if you’re going to miss a meeting

4) Give team updates on assignment progress. Avoid overlapping assignments.

5) Members are responsible for writing and submitting their agile stories to the product owner.

Coding standards:

1) Frameworks: SQLite for database, Qt (C++) for UI, Github (Version control; individual branches for each member), Trello (Agile management), Doxygen (documentation)

2) Provide sufficient commenting that the code function is evident.

3) Use descriptive identifier names (“i” for index is fine)

Agile Stories (Writing): DUE MONDAY & ONE UML ITEM PER MEMBER. Planning poker will be played during team meeting after class.

Mo – 4, 5, 6

Po Jen – 13, 14

Adam – 8, 9 ,10

George – 1, 2, 3

Sami – 7, 11, 12

**Agile story template:**

#) Description: As a \_\_\_\_\_\_,

a. Assumptions:

b. Assignee:

c. Story Point Estimation:

d. Priority:

e. List of Tasks and Tests:

i. Tasks:

1.

ii. Tests:

1.

f. Definition of Done

**EXAMPLE:**

1. Description: As a customer, I want to be able to click on a help option that will explain how to operate the program.

a. Assumptions: Main window is working and space on the window is allocated for the help button.

b. Assignee:

c. Story Point Estimation: 1

d. Priority: 1

e. List of Tasks and Tests:

i. Tasks:

1. Create a button named “HELP”.

2. Make the help button go to another window.

3. Come up with some text that explains how to operate the program that will be displayed on the help window.

ii. Tests:

1. Clicking on the button will take you to a new window.

2. The new window displays the correct text.

f. Definition of Done: This story is done when there exists a help button on the main window that takes the customer to a separate window with information regarding how to operate the program.

General tasks:

~~- Establish Github~~

~~- Establish database~~

~~- Plan out UI~~

~~- Plan out program structure (UML)~~

~~- Team rules~~

~~- Coding standards~~

~~- Agile stories- Trello board?~~

- Continuous Integration? (Extra credit) -ASK PROF

- Planning poker (need agile stories)

- UML (Class, State, Use case, Activity)

Adam: in-progress

8) Description: As an administrator, I want to manage the store’s inventory list by adding and deleting items from the list.

a. Assumptions: The inventory list (database) is created and the admin account is created.

b. Assignee: TBD

c. Story Point Estimation: TBD

d. Priority: TBD

e. List of Tasks and Tests:

i. Tasks:

1. Create buttons for “add item” and “Delete item”.

2. Connect add button to “insert” new item with price (float/double)

3. Display record to be added and create “Confirm addition” button

4. Add record

5. Connect delete button to “delete” existing item (price not included)

6. Display record to be deleted and create “Confirm deletion” button. Create “Confirm deletion

7. Delete record (item).

8. Error-check input: check for missing fields (“add”) or invalid item names (“delete”).

ii. Tests:

1. Click “add item”. A window should pop up with text prompts for item name and price [ADDITIONAL INFO?]

2. Enter information. Click “accept”.

3. Confirmation window should appear. Click “confirm”

f. Definition of Done:

Sami: in-progress

7) Description: As a team member, I want the ability to create *“dummy”* purchases for new customers that do not affect the stores, to validate the accuracy of the software.

1. Assumptions: The database is established as well as the GUI to interact with the checkout method.
   1. Assignee: TBD
   2. Story Point Estimation: TBD
   3. Priority: TBD
   4. List of Tasks and Tests:
      1. Tasks:
         1. Create the Database to hold our products information
         2. Create the check out/ purchase method
         3. Create another method that calls upon the checkout method, however it does not affect the numbers of the database and is only used to validate the software.
         4. This should not be accessed by the customer, however **may** be useful to administrators of the program. But should be allowed to team members working on the project:
      2. Tests:
         1. Click “Trial Purchase Method” which should call upon our check-out method.
         2. Team member selects the products they desire and once they click “Confirm Purchase” they are prompted with a receipt.
         3. Team member should check the database to make sure the query has executed properly with the database.
2. Definition of Done:
   1. The team member should have a method which should only show when logged in as a team member, that allows them to test the validity of the checkout method. This method allows the team member to make a “*dummy”* purchase which does not affect the database values but rather tests functionality.

11) As a team member, I want to the application to determine if any Regular Customer should be prompted to promote to an Executive Status when it is beneficial to them based on their spending.

1. Assumptions: A hierarchy is set up that denotes which customers are regular customers and which customers are executive customers. In addition to the customer check-out method is also established.
   1. Assignee: TBD
   2. Story Point Estimation: TBD
   3. Priority: TBD
   4. List of Tasks and Tests:
      1. Tasks:
         1. Establish the threshold when it is considered beneficial to upgrade to executive status.
         2. Establish the algorithm used to determine when it the promotion should be prompted.
         3. Set up a pop up to ask the user if they wish to upgrade, if they accept their rank in the database should reflect their decision.
      2. Tests:
         1. Attempt checking out below our set threshold and proceed without being prompted.
         2. Attempt checking out above our set threshold, and be prompted to upgrade status.
         3. Accept the status upgrade and check if it reflects the database.
2. Definition of Done: When checking out as a customer if a certain threshold is met you will be prompted asking if you wish to be upgraded to executive status. Otherwise if the threshold is not met you will not be prompted to upgrade your status.