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: