**See Own Personal Purchase History**

List all the orders the customer him/herself made to date after logging into their account.

Priority: High Cost: 2 days

**See Own Number of Refills Left for ALL Prescription Forms**

List all the prescription forms customer/patient made with the number of refills remaining after logging into their account.

If time: see how many medications have less than 3 refills left

Priority: High Cost: 2 days

**See Own Total Money Spent**

Output total money spent by customer/patient at pharmacy by logging into their account.

Priority: High Cost: 2 days

**See Own Personal Information**

See all personal information (name, phone number, address) after logging into account.

Priority: High Cost: 2 days

**Search Patient by Health Card ID to See Entire Purchase History**

List all the orders (OTC and Prescription based) a patient made by searching their health card ID in the system. *Note: dependent on an above detailed story’s implementation (so should take less time)*

If time: give option of searching patient by name

Priority: Medium Cost: 1 days

**Search Patient by Health Card ID to See # of Refills Left for ALL Prescription Forms**

List all the prescription forms under customer/patient with the number of refills remaining by searching their health card ID in the system. *Note: dependent on an above detailed story’s implementation (so should take less time)*

If time: give option of searching patient by name

Priority: Medium Cost: 1 days

**Search Patient by Health Card ID to See Total Money Spent**

\*\***ONLY OWNER**\*\* Output total money spent by each customer at pharmacy by searching their health card ID in the system. *Note: dependent on an above detailed story’s implementation (so should take less time)*

If time: give option of searching patient by name

Priority: Low Cost: 1 days

Z

**CODE SMELL: Duplicated Code**

As a developer, remove the code smell of duplicated code found in the Owner.java and Pharmacist.java classes.

Priority: High Cost: 2 days

**CODE SMELL: Long Methods**

As a developer, shorten the method “displayDescription()” in the DisplayDescription class to remove the code smell of long methods. Also try to shorten any other long methods found in the presentation layer.

Priority: High Cost: 4 days

**CODE SMELL: Large Classes**

As a developer, shorten the Inventory class by creating more classes to remove the code smell of large classes. Also try to shorten any other long classes found in the presentation layer.

Priority: High Cost: 4 days

**Testing #1**

As a developer, modify existing tests to enhance their capabilities by adding more tests for detailed stories implemented in previous iterations. Also, use SQL queries inside tests.

Priority: Medium Cost: 3 days

**Testing #2**

As a developer, create integration and unit tests for new detailed stories implemented this iteration.

Priority: Medium Cost: 3 days

**Testing #3**

As a developer, create customer/end-to-end tests for all big stories implemented to date.

Priority: Medium Cost: 3 days

**Fix Bugs/Issues from Previous Iterations**

As a developer, try to fix all bugs/issues left from the previous iterations that were identified previously or have been identified now.

Priority: High Cost: 3 days

**Fix Issues Identified By QA Team**

As a developer, try to fix as many issues that the QA team identified to improve our application.

Priority: High Cost: 3 days

**Hardening Phase**

As a developer, clean up and polish the project before the final release.

Priority: Low Cost: 2 days