Jared Moore, Max Mueller, Ben Nowacki

CSE 2102

MIS Project

**Assignment 2a**

Project Breakdown:

The project breakdown consisted of a separation of classes where each member initially took a different class to write. Jared Moore worked on Patient.java, Max Mueller worked on PatientProfileInterface.java, and Ben Nowacki worked on PatientDatabase.java. Once most of the methods were written for each class, they were tested individually. Each class took between 4-12 hours for the implementation and testing. With the classes tested, we all worked together (in person) to integrate them. This process took roughly 8 hours to get a solid, functional GUI using all three classes. The final stage of work was primarily testing and minor changes, which took another 5 hours or so.

Most of the difficulties encountered stemmed for the interface class. This is likely due to GUI being a relatively new topic for most of us. No one had experience with Swing, which made it more time consuming to fix errors. We were able to figure out workarounds for most of the problems (one being the GUI dropdowns not showing enumerated types) by looking up similar problems or watching tutorials.

If we were to do another assignment of this scope, it would be beneficial to layout a better schedule in completing certain tasks. There were certain tasks that didn’t seem too difficult, and we didn’t leave enough time to properly complete it. This then caused some delays in completing other portions of the project. Thus, it would help to create a deadline for each individual part so that the group can properly continue advancing the project without falling behind.

Test Plan:

Rationale in creation of test cases, actual test cases, any additional code written to test

Our testing consisted of first testing each class individually and then testing the system as a whole. That is, class functionality was tested first. The patient class was tested for the getter and setter methods. The print summary method was also tested to make sure the proper information was returned. The patient database needed to be tested for its ability to insert, delete, update, display, and search through patients. For this a test case was written to test one method at a time. For example, a patient was inserted and then the entire database was printed out to make sure the patient was correctly added. The file was then closed and reopened to make sure data was being saved. The GUI portion of the system needed to be tested a little differently. We couldn’t hard code test cases, so we had to manually use the GUI to try to create errors. Any time the system crashed, we addressed the cause and proceeded to fix it, after which it was tested again. This manual process was repeated for many scenarios, including when text fields were empty or improperly used. Eventually, we reached a point where we couldn’t cause anymore problems.