# CS 3330 Final Project Documentation

**UI Development:** This project was created using JavaFX. There are two .fxml documents in the project including StartWindow.fxml and TrackingWindow.fxml

| | | |
|---|---|---|
| **Architecture:** | Models | {Switcher.java, Requests.java} |
| | Views | {StartWindow.fxml, TrackingWindow.fxml} |
| | Controllers | {StartWindowController.java, TrackingWindowController.java} |
| | Main | {HickmanjvMaintenanceTracker.java} |

**Required Elements**

1. Object Oriented Elements that you write the code for:
    a. Classes
    **Requests.java, Switcher.java, etc;**
    b. Subclasses
    **StartWindowController.java, and TrackingWindowController.java**
    **Both of these classes are subclasses of Switcher.java**
    c. At least one Abstract Class
    **Switcher.java is abstract**
    d. At least one Interface
    **Initializable and Serializable are interfaces for StartWindowController.java and TrackingWindowController.java**

2. Code elements that you utilize:
    a. One or more collection classes
    **I utilize ArrayList to store the Request Objects that are created. Line 40 and 136 of TrackingWindowController.java**
    b. Exception Handling
    **Exception Handling with FileIO because the program is serialized. Lines 90-101, 112-123, 138-149, 171-189 in the TrackingWindowController.java**
3. The application must have a clearly defined model
    **The Models are Switcher.java which holds the data for transitioning scenes, and Requests.java which holds the data of the Request Objects that are to be created**
4. The UI must have multiple scenes and one of the scenes will have the contents changed based on application state.

**There are 2 scenes, a Start page and a Tracking page. The Tracking page is changed based on the requests entered and the ones saved prior to re-opening the program.**

5. There must be a way to access “About” Information

   **On the Start page there is an About button that will provide application and author information to the user**

6. The application must save and load data

   **The application saves each time a new request is entered into the tracking table, a request removed from the tracking table, and upon opening the program after leaving data, the program will open and pull the data from prior uses. These solutions are found in the initialize, addNewRequest, deleteRow, and deleteAllRequests methods of TrackingWindowController.java**