**DRS-Enhanced – Group Report**

**Unit:** COIT20258 – Software Engineering Project  
**Assessment 3 – Group Report Submission**  
**Project Title:** Disaster Response System (DRS-Enhanced)

**Team Name: DRS-DevForce**

**Team Members:**

**1. [Your Full Name] – [Your Student ID]**

**2. [Teammate Name] – [Teammate Student ID]**

**Submission Date: [Insert Date]**

Table of Contents

[1. Introduction 4](#_Toc200752146)

[2. Requirement Specification 5](#_Toc200752147)

[3. Design Documentation 6](#_Toc200752148)

[3.1 Use Case Diagram 6](#_Toc200752149)

[3.2 Class Diagram 6](#_Toc200752150)

[3.3 Sequence Diagrams 7](#_Toc200752151)

[3.4 ER Diagram 8](#_Toc200752152)

[4. Test Plan 9](#_Toc200752153)

[5. Evidence of Testing 10](#_Toc200752154)

[1. Application launch screen in NetBeans 10](#_Toc200752155)

[2. Filled report form with "Submit Report" clicked 10](#_Toc200752156)

[3. Dashboard view showing all reports 11](#_Toc200752157)

[4. ComboBox filter applied (with different result shown) 11](#_Toc200752158)

[5. Auto-priority suggestion after typing e.g., “fire” 13](#_Toc200752159)

[6. Folder Structure 14](#_Toc200752160)

[7. How to Run the Application 15](#_Toc200752161)

# **1. Introduction**

DRS-Enhanced is a desktop application (written in JavaFX) designed to facilitate the process of disaster reporting and management at the local level. This system enables its users to communicate the most serious events (floods, fires, building collapses, and so on) through an intuitive GUI form.

A dashboard equipped with real-time information based on MySQL database allows administrators to view and filter submitted reports. The system was created and developed as a part of the COIT20258 - Software Engineering Project unit, and accordingly, in accordance with the assignment specifications, to expand and improve an existing DRS-Initial system.

Two functional improvements were incorporated in this improved version:

1. Priority and status based report filtering.

2. Automatic priorities proposal by keywords in report description.

The application follows the Model-View-Controller (MVC) system architecture, and the entire development was done with Java 17, NetBeans, Scene Builder, and MySQL. In this report, the requirements, the design of the system, the testing procedure, and the implantation of the system are recorded in a formal way, as an academic exercise and also as an evaluation exercise.

# **2. Requirement Specification**

**Functional Requirements**

* Report disasters using JavaFX form.
* Store reports in a MySQL databank.
* See reports in a table on dashboard.
* Report filtering by priority and status ComboBoxes.
* Provide the possibility of switching scenes between the form and the dashboard.

**Non-Functional Requirements**

* Adheres to MVC structure to keep things clean.
* JavaFX and Scene Builder based GUI.
* The application is written on Java 21 and NetBeans 20+.
* MySQL is used to store data even after closing and restarting the app.

**System Requirements**

* Java JDK 17+
* NetBeans IDE (20 or above)
* JavaFX SDK ( module-info.java)
* MySQL Server 8.0 or later

**User Requirements**

* The user should be capable of entering reports without any difficulty.
* Admin must look at data and filter data in an intuitive way.
* Applications should be sensitive and show definite alerts or acknowledgments.

**Other Features Added**

1. **comboboxes-filtering-on-dashboard.**

* Admin filtering by Priority (CRITICAL, HIGH, etc.), or Status (NEW, RESOLVED, etc.).

1. **Auto-Priority Suggestion:**

* The keywords such as fire, explosion are identified by the app and the priority field in the report form is auto-filled.

# **3. Design Documentation**

## **3.1 Use Case Diagram**

* Actors: User, Admin
* Use cases: Submit Report, View Dashboard, Apply Filter, Auto Suggest Priority

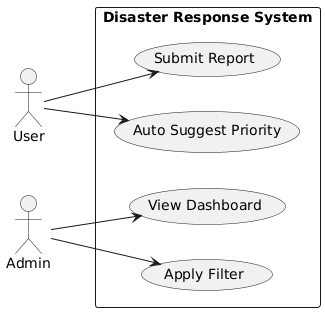
****

Figure 1:Use case Diagram

## **3.2 Class Diagram**

* App, ReportController, DashboardController, DBConnection, DisasterReport, SceneSwitcher

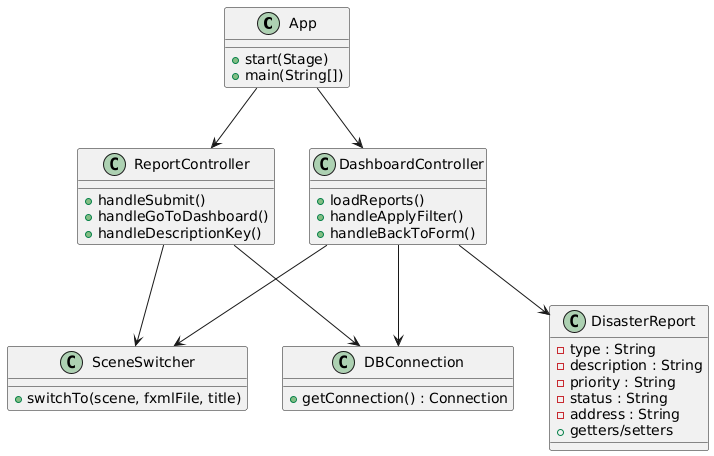


Figure 2:Class Diagram

## **3.3 Sequence Diagrams**

* Form Submission Sequence:  
  User → ReportController → DBConnection → Database

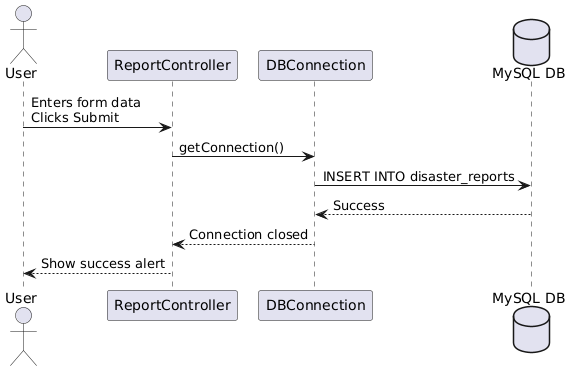
****

Figure 3:Form submission sequence diagram

* Filter Application Sequence:  
  Admin → DashboardController → DBConnection → Database → TableView

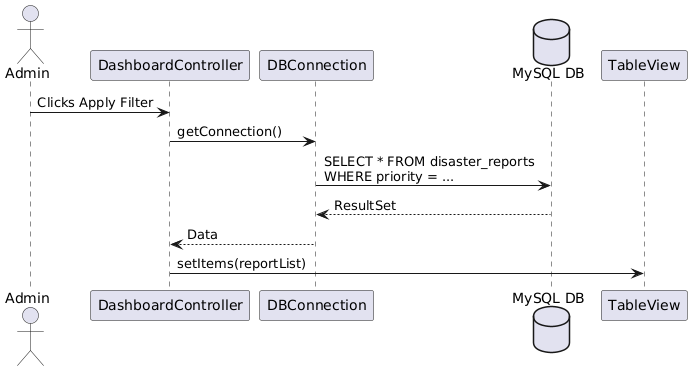
****

Figure 4:Filter application sequence diagram

## **3.4 ER Diagram** Table: disaster\_reports Fields: id, type, description, priority, status, address

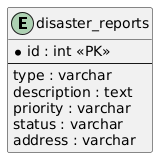
****

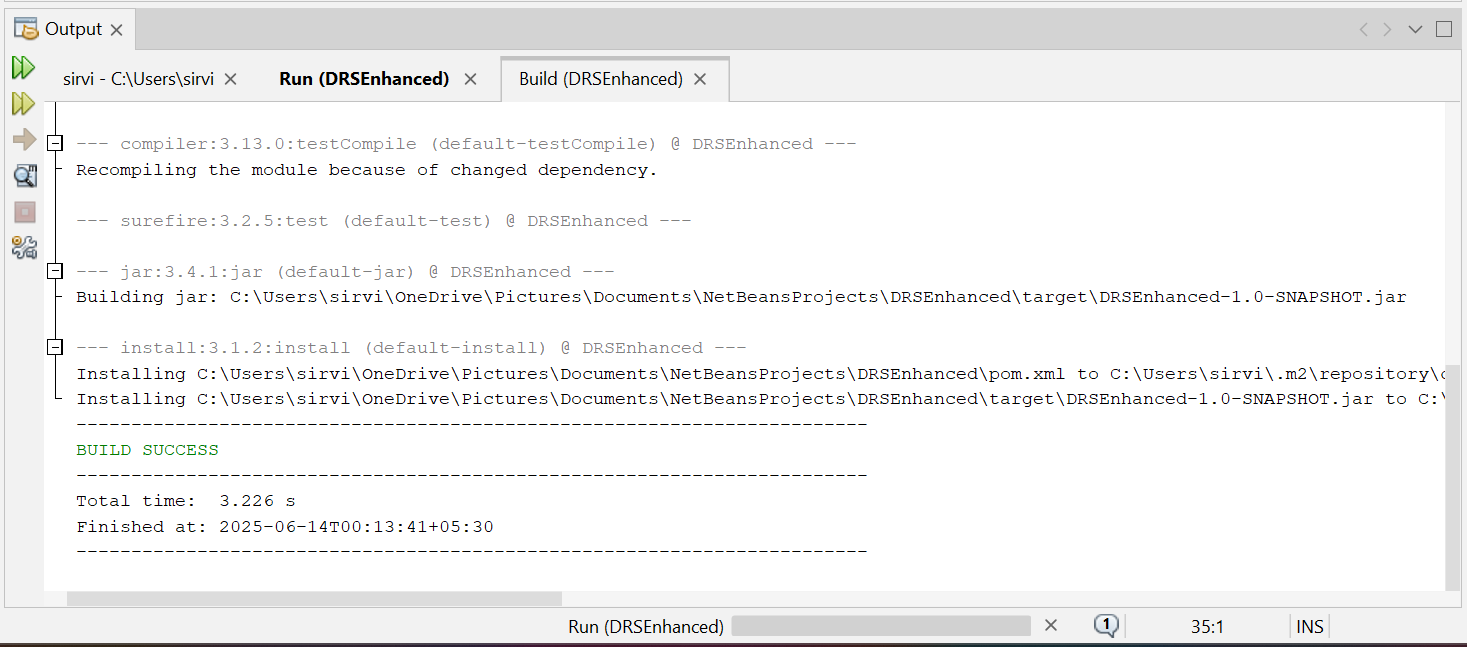
Figure 5:Er diagram

# **4. Test Plan**

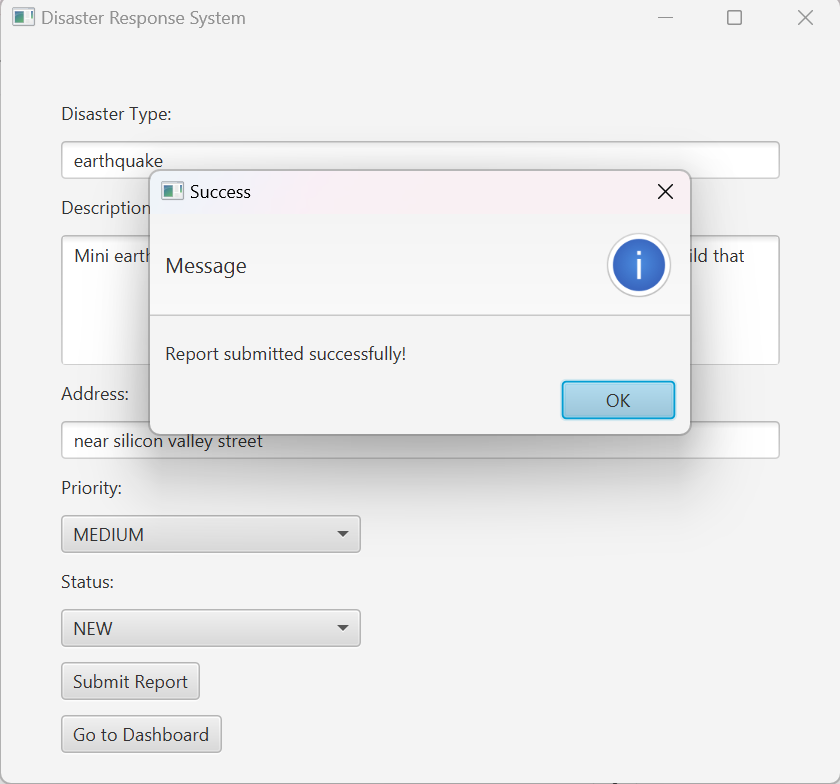
| **Test ID** | **Description** | **Input** | **Expected Output** | **Result** |
| --- | --- | --- | --- | --- |
| **TC01** | Submit valid report | All fields filled | Data saved, success alert shown | pass |
| **TC02** | Submit empty report | All fields empty | Error alert shown | pass |
| **TC03** | View reports in dashboard | Click "Load Reports" | Table shows all saved reports | pass |
| **TC04** | Apply filter | Priority = HIGH | Only high-priority reports shown | pass |
| **TC05** | Auto-priority suggestion | Description: “Explosion” | Priority auto-set to CRITICAL | pass |

# **5. Evidence of Testing**

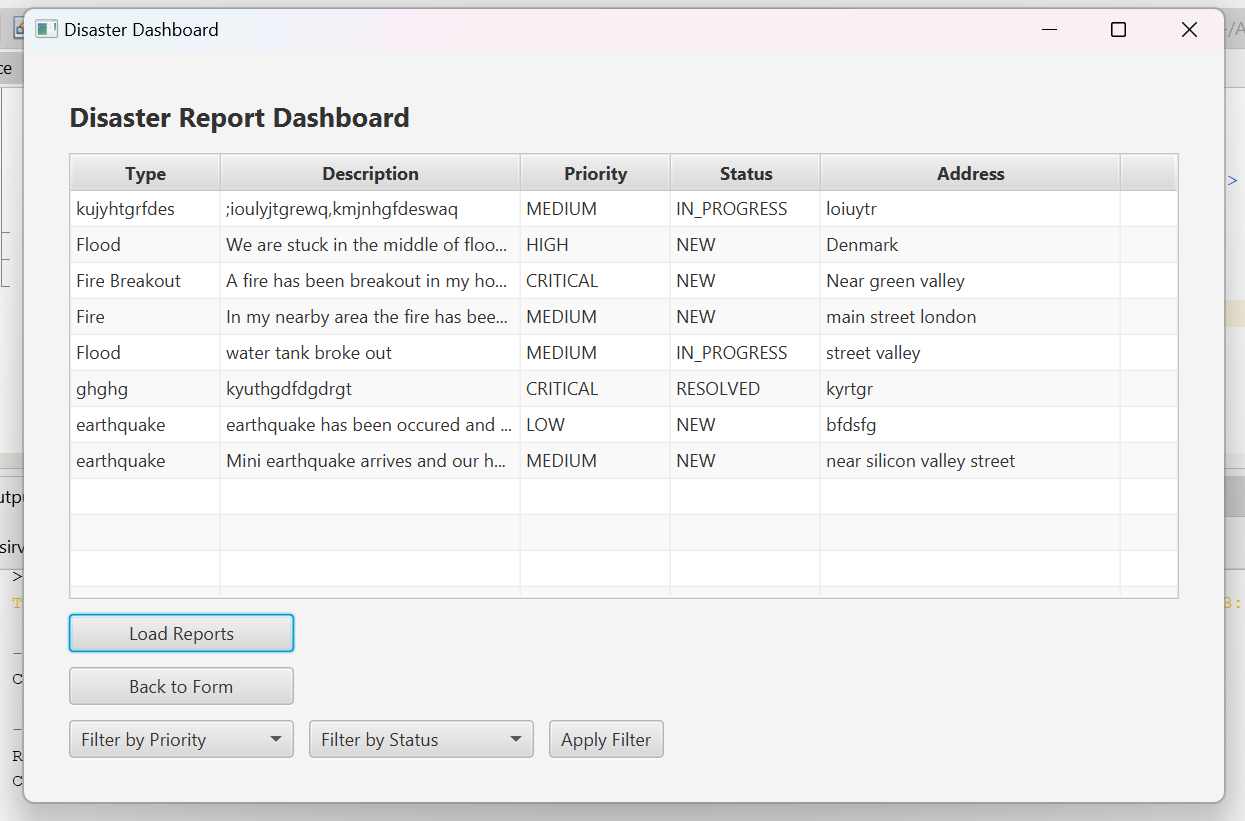
## **Application launch screen in NetBeans**



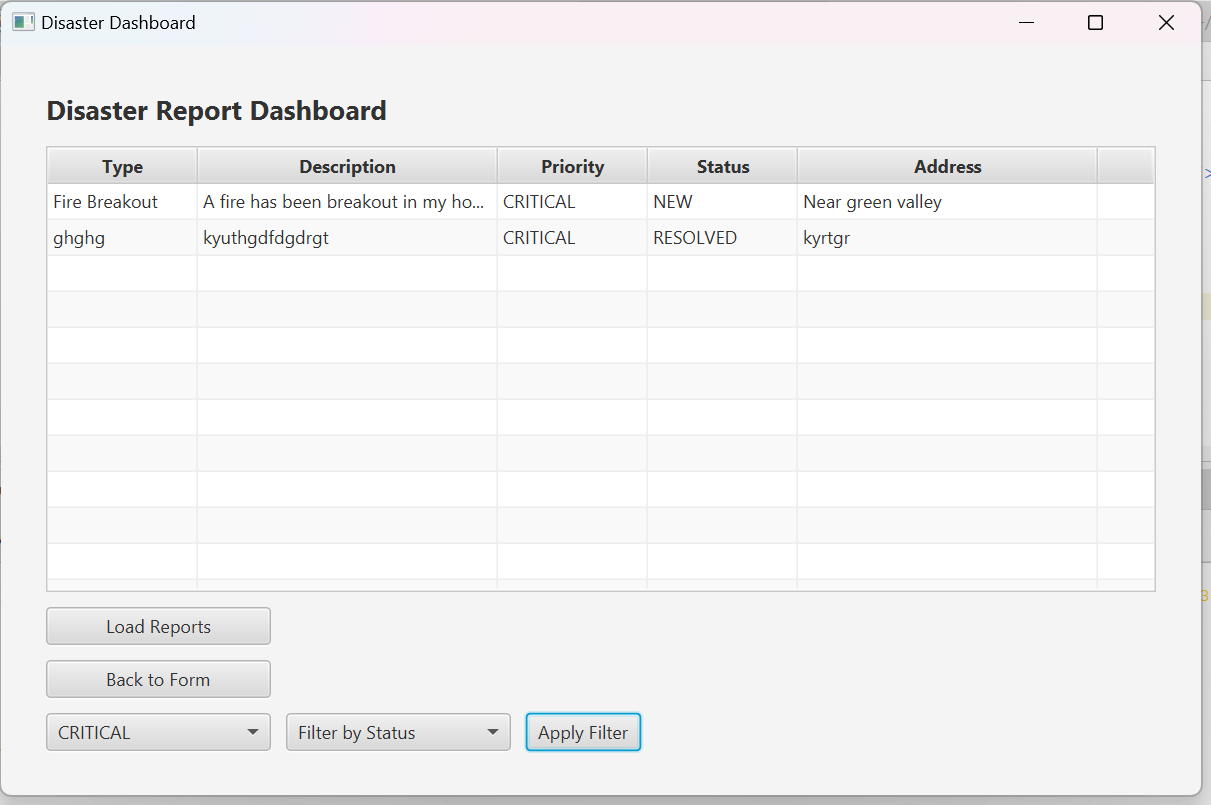
## **Filled report form with "Submit Report" clicked**

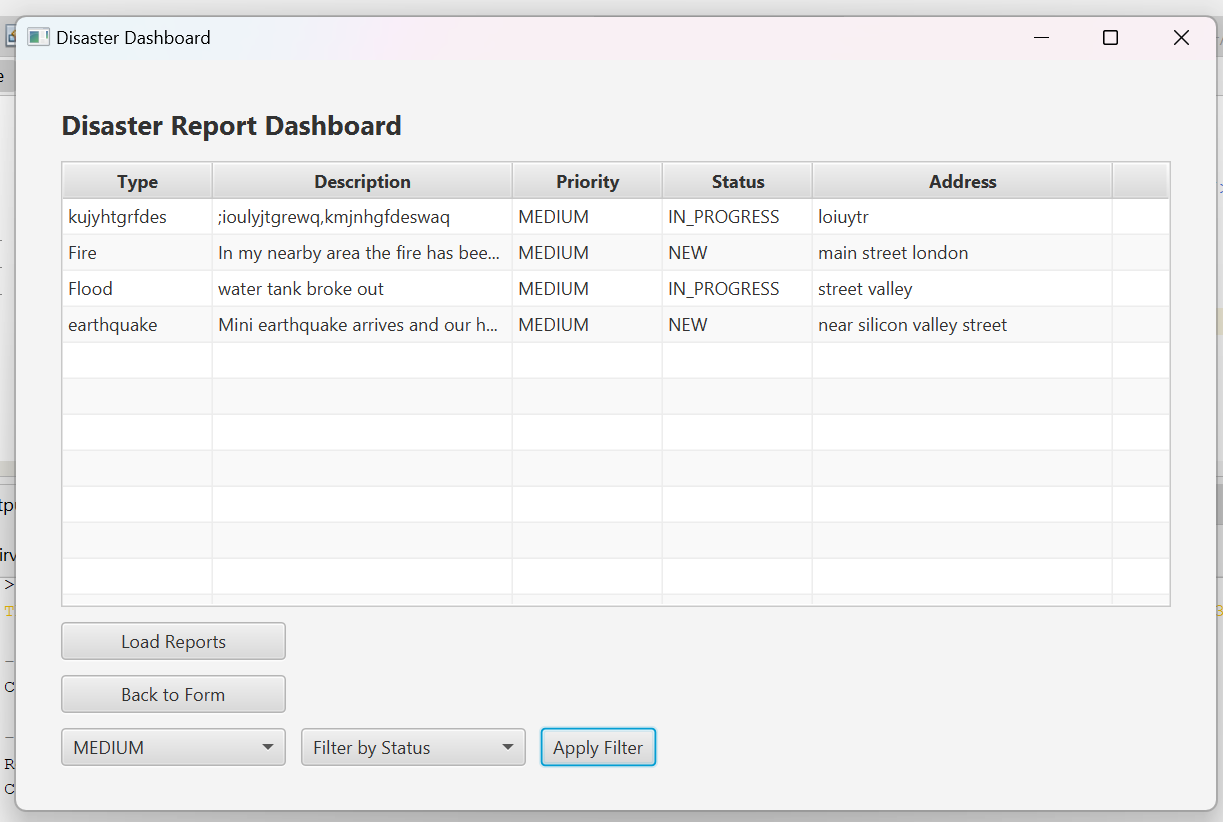
****

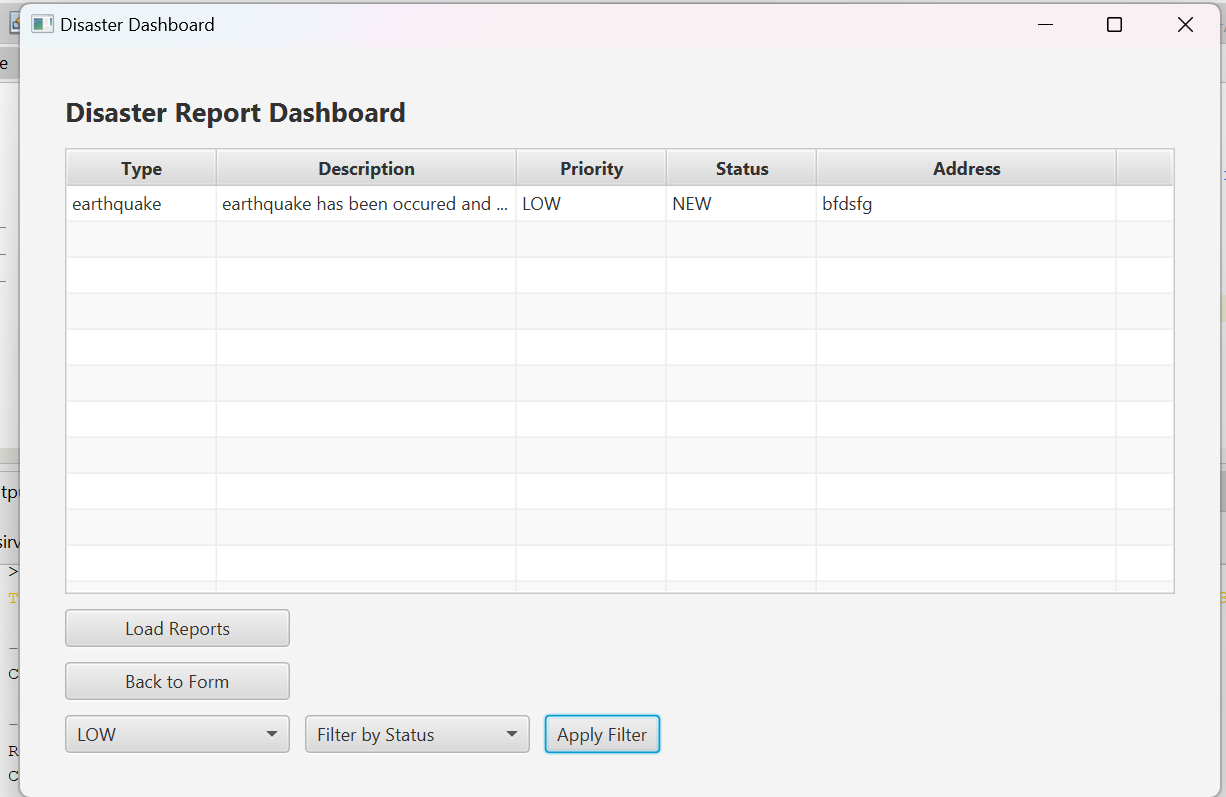
## **Dashboard view showing all reports**

****

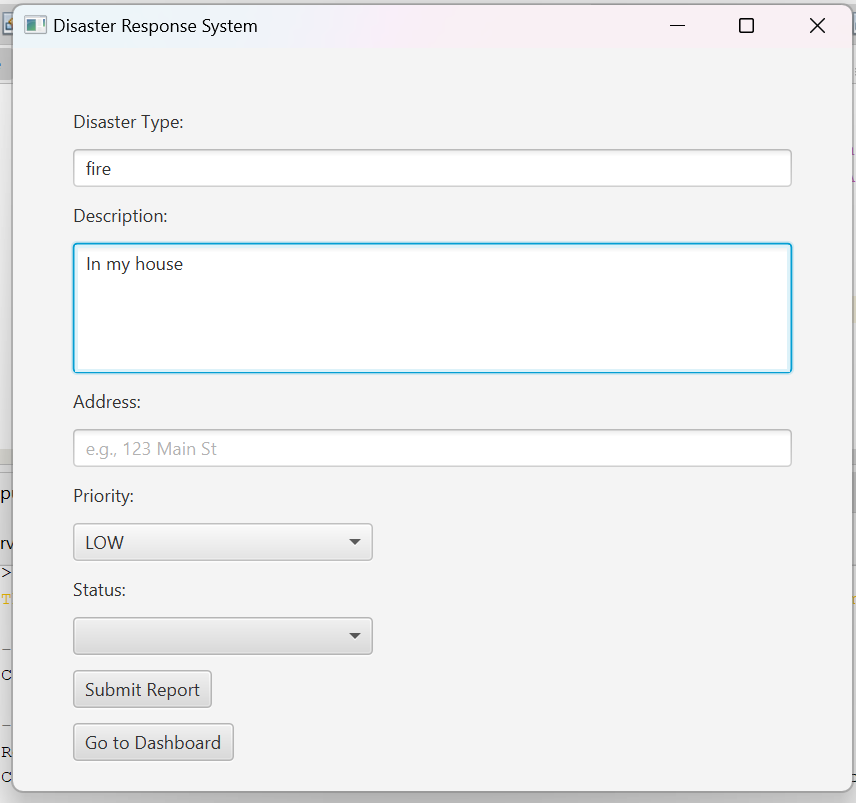
## **ComboBox filter applied (with different result shown)**

****

****

****

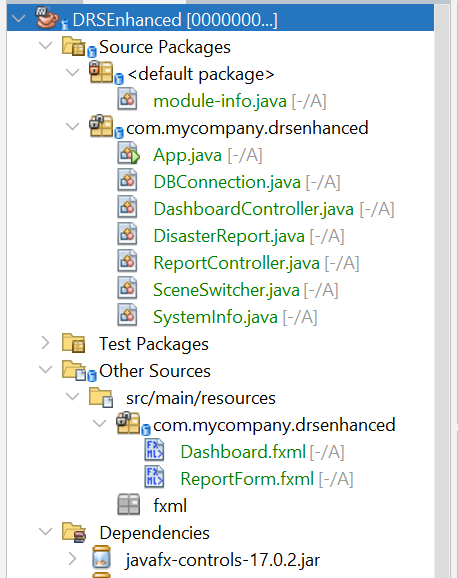
## **Auto-priority suggestion after typing e.g., “fire”**

****

# **6. Folder Structure**

The project follows a standard Maven JavaFX structure with clear separation between business logic and UI components:

1. com.mycompany.drsenhanced contains all controller and utility classes:
   1. ReportController.java: Handles form submission and navigation
   2. DashboardController.java: Manages dashboard logic, filtering, and data binding
   3. DisasterReport.java: Model class for disaster report objects
   4. DBConnection.java: Provides static method to connect to MySQL
   5. SceneSwitcher.java: Utility for FXML scene transitions
2. src/main/resources contains:
   1. ReportForm.fxml: The user-facing report submission interface
   2. Dashboard.fxml: The admin dashboard for managing reports

****

# **7. How to Run the Application**

**Step-by-Step Guide:**

1. Open the DRSEnhanced project in NetBeans
2. Ensure JavaFX and MySQL are installed and configured
3. Import MySQL table using .sql script below
4. Modify DBConnection.java to match your MySQL username/password
5. Run App.java from NetBeans

**MySQL Table Creation (SQL Script):**

CREATE TABLE disaster\_reports (

id INT AUTO\_INCREMENT PRIMARY KEY,

type VARCHAR(50),

description TEXT,

priority VARCHAR(20),

status VARCHAR(20),

address VARCHAR(100)

);