**Software Engineering**

**Milestone-2: Design and Implementation**

**Course number:** CMPS310

**Submission date:** 23-11-2024

**Theory Class section:** L52

* **Effort distribution of the student:**

**SID:** 202206265 **STUDENT NAME:** Lana Ahl **Effort is given:** 25%

**SID:** 202107324 **STUDENT NAME:** Habiba Khattab **Effort is given:** 25%

**SID:** 202110167 **STUDENT NAME:** Zunaira Hamid **Effort is given:** 25%

**SID:** 202103014 **STUDENT NAME:** Alreem Zainal **Effort is given:** 25%

* **DECLARATION:** We hereby certify that no part of this project or product has been copied from any other student’s work or from any other sources except where due acknowledgment is made in the project. No part of this project/product has been written/produced for us by any other person.
* **Task 1 (Two sequence diagrams, normal scenarios & class diagram):**

**A diagram of a project

Description automatically generated**

**A diagram of a diagram

Description automatically generated**

A diagram of a computer network

Description automatically generated

A diagram of a diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC03 | **Accident Reporting** | |
| **Brief Description** | Allow the owner of a vehicle involved in an accident to report it online, provided no dispute exists. | |
| **Primary actors** | Vehicle Owner (Offending Vehicle), Insurance company | |
| **Trigger** | The vehicle owner wants to report an accident. | |
| **Preconditions:**   * Both vehicles must be registered in the system. * Both parties must agree on their responsibility for the accident. | | |
| **Post-conditions:**   * An accident report is created and shared with insurance companies. * A unique accident case number is generated. | | |
| **Main Success Scenario** | | |
| **Actor Action** | | **System Response** |
| 1. Vehicle owners enter VINs for both vehicles. | | 2. Retrieves vehicle details. (See 2.a. for alternative flow) |
|  | | 3. provide the date, time, location, and a brief |
| 4. The vehicle owner provides accident details (date, location, etc.) and accident description. | | 5. find the insurance policy of the offending vehicle and create an accident report with a unique case number. (See 5.a. for alternative flow) |
| 6.TheoffendingVehicle owner confirms that they are responsible. | | 7. sends the accident report to the insurance company |
| 8. The insurance company acknowledges the receipt. | | 9. Store accident report. |
| **Alternative flows:**  2.a. Invalid VIN: If the VIN does not match any vehicle, the system displays "Incorrect vehicle information."  5. a. Dispute Exists: If a dispute exists, the system terminates the session and instructs both parties to visit a police station. | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** UC04 | **Pay Fines and Invoices** | |
| **Brief Description** | Enables vehicle owners to pay unpaid fines and invoices using a credit card. | |
| **Primary actors** | Vehicle Owner, QPay | |
| **Trigger** | The vehicle owner initiates the payment process. | |
| **Preconditions:**   * The vehicle must have unpaid fines or invoices. * The owner must have valid credit card details. | | |
| **Post-conditions:**   * The fines or invoices are marked as paid. * A payment receipt is generated. | | |
| **Main Success Scenario** | | |
| **Actor Action** | | **System Response** |
| 1. Vehicle owner enters the VIN and credit card details | | 2. System retrieves vehicle details and lists unpaid fines/invoices. |
| 3. The vehicle owner selects invoices to pay. | | 4. The system calculates the total amount. |
| 5. Vehicle owner enters credit card details. | | 6. System sends details to qPay. |
| 7. qPay verifies the card and approves the payment. | | 8. The system marks fines as paid and generates a receipt. (See 8.a. for alternative flow) |
|  | | 9. Displays receipt and updates the unpaid invoices list. |
| **Alternative flows:**  8.a. If the credit card is invalid, qPay sends an error message, and the system requests valid card details. | | |

* **Task 2 (Constraints and quality properties):**

1. **Scalability:**

Description:

* The software should be able to support 30 million registrations within 10 years without degrading performance.

1. **Security:**

Description:

* The software must protect sensitive information through encryption.

1. **Performance:**

Description:

* The software must respond to user interaction within an average of 2 seconds under normal or peak conditions.

1. **Maintainability:**

Description:

* The software should be easy to maintain, allowing new features to be added or existing ones modified without affecting other components.

1. **Usability:**

Description:

* The software must have a clear and intuitive user interface on any device such as desktop, laptop, or mobile.

1. **Portability:**

Description:

* The software should be able to run on any device such as a desktop, laptop, or mobile.

1. **Availability:**

Description:

* The software should remain 24/7 with minimal downtime and robust mechanisms to recover from hardware or software failures.

1. **Modifiability:**

Description:

* The software should support adding new functional such as support for electric vehicle registration or automated online traffic processing.

1. **Data Integrity:**

Description:

* Data stored or processed by the system must be accurate, consistent and free from corruption or unauthorized modification.
* **Task 3 (Implementation of one-use case with a nice interface):**

**Documentation for the Multi-System JavaFX Application**

**1. Overview**

This JavaFX application integrates two systems:

* **Accident Reporting**: Allows users to log details of an accident.
* **Vehicle Transfer Management**: Facilitates the process of transferring a vehicle from one owner to another.

The application includes a main menu for navigation between the two systems, each implemented as separate views. The application ensures a responsive and user-friendly interface, proper error handling, and modular code for maintainability.

**2. Components**

**a. Main Menu**

* **FXML File:** MainView.fxml
* **Controller:** MainController
* **Description:** Acts as the central navigation menu. Contains buttons to navigate to the Accident Reporting and Vehicle Transfer Management systems.

**b. Accident Reporting System**

* **FXML File:** AccidentView.fxml
* **Controller:** AccidentController
* **Model:** Accident
* **Description:** Handles input and management of accident details, generates a report, and allows users to review or close the application.

**c. Vehicle Transfer Management System**

* **FXML File:** TransferView.fxml
* **Controller:** TransferController
* **Model:** Owner, Vehicle
* **Description:** Manages the process of transferring ownership of a vehicle by recording relevant details.

**3. Implementation Detail**

* **Main Menu**

1. **FXML File:** MainView.fxmlUses a VBox to align two buttons vertically, each leading to a specific system. The layout is simple and responsive.
2. **Controller Class:** MainController

Handles navigation between views. Buttons load the respective FXML files and replace the current scene.

* **Accident Reporting System**

1. **FXML File: AccidentView.fxml**

* The form is divided into three panes:
  1. Vehicle Information Input (using a GridPane).
  2. Accident Details Input (hidden initially, using another GridPane).
  3. Accident Report Display (hidden initially, using a VBox).

1. **Controller Class: AccidentController**

* Handles the retrieval of vehicle data, input validation, accident report generation, and pane visibility toggling.
* **Vehicle Transfer System**

1. **FXML File: TransferView.fxml**

The UI for the Vehicle Transfer System uses a GridPane layout to organize input fields and buttons in a structured format.

* **Structure**:
  1. Input fields for vehicle and owner details.
  2. Input fields for new owner details.
  3. A button to confirm the transfer.
  4. An area of the details entered and the payment field
  5. Two buttons “pay” and “cancel”

1. **Controller Class: TransferController**

This class handles the business logic for the vehicle transfer system.

* **Responsibilities**:
  1. Validate user input.
  2. Simulate transferring ownership of the vehicle.
  3. Display confirmation or error messages

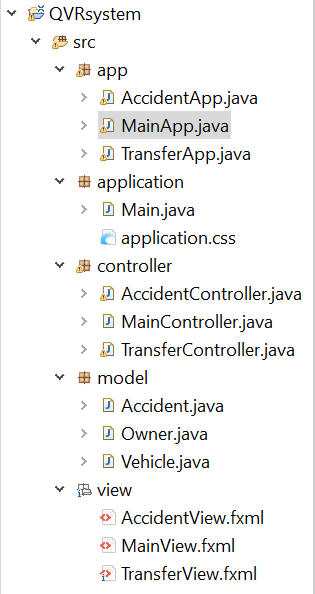
**4. Features**

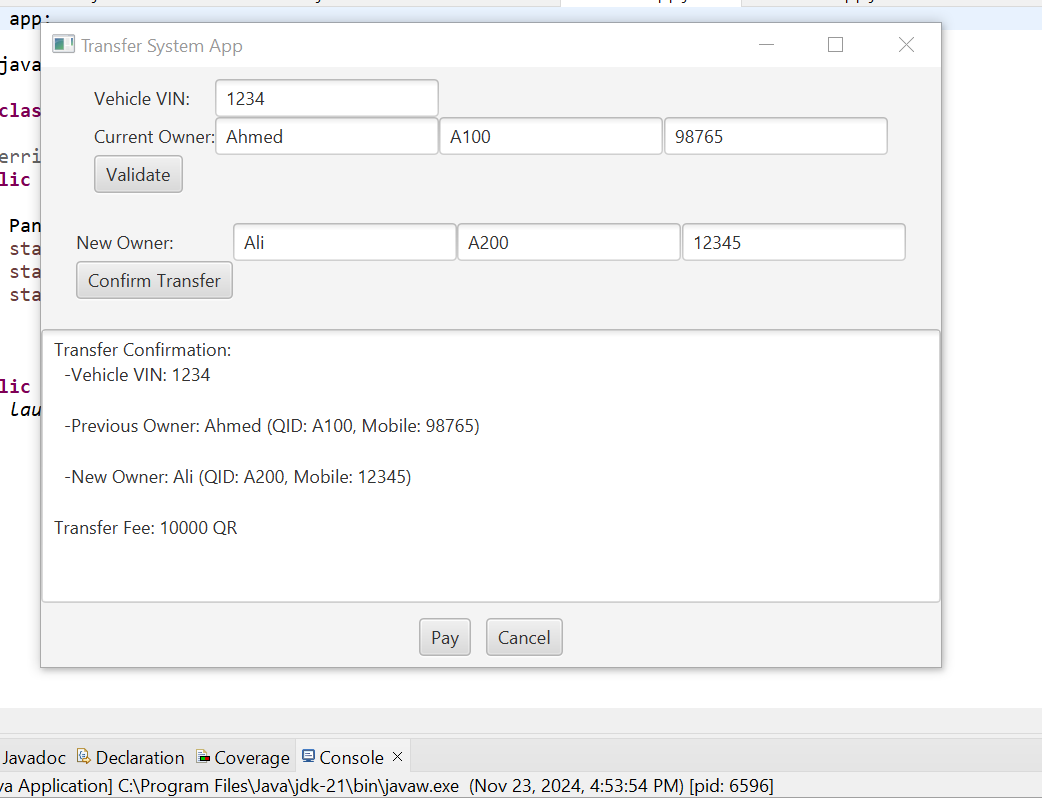
* Modular FXML files ensure clear separation of views and functionality.
* Responsive layouts (VBox and GridPane) for better user experience.
* Simple navigation through the main menu.
* Proper error handling with user feedback.

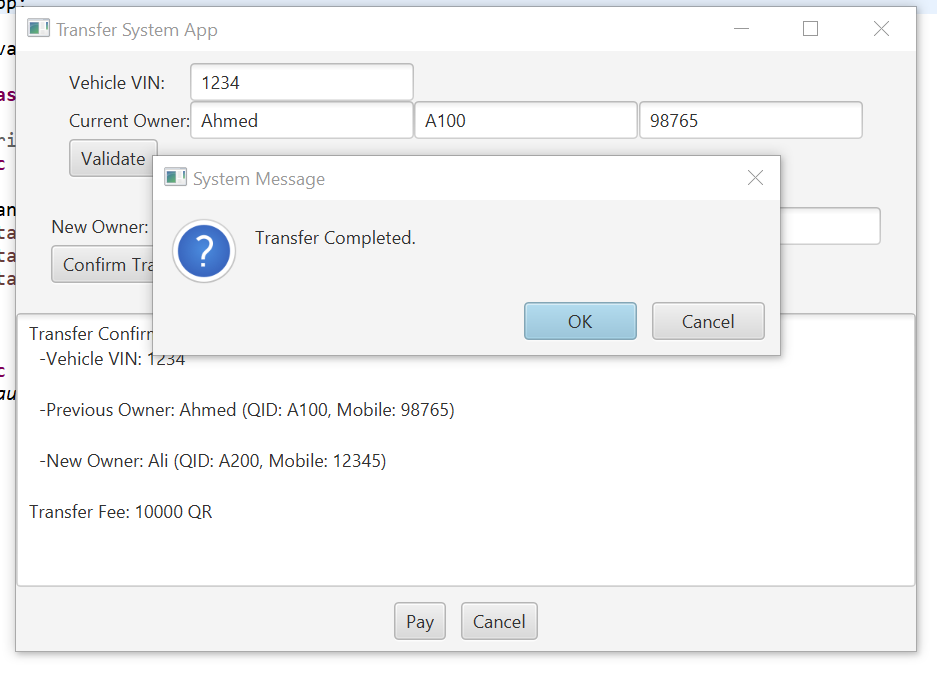
**5. Improvements**

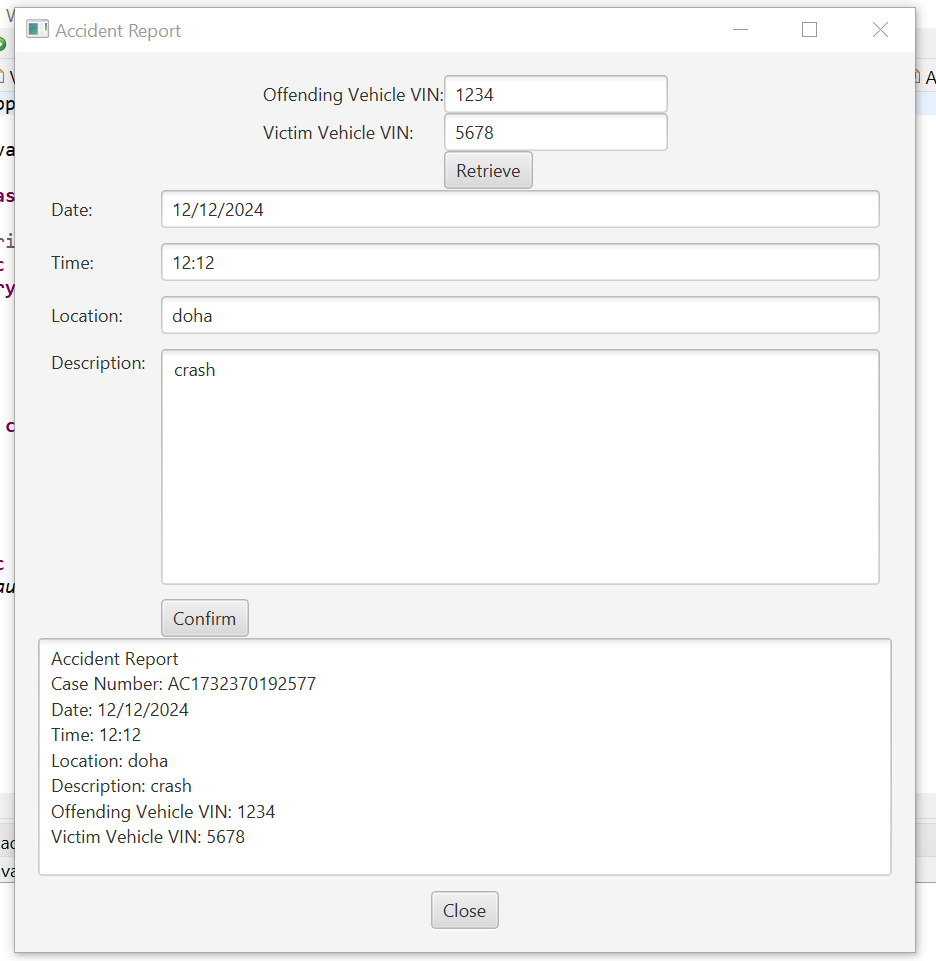
* Integrate a database for real-world VIN validation.
* Implement logging for error tracking.

A screenshot of a computer

Description automatically generated**6. Screenshots:**







A screenshot of a computer

Description automatically generated

* **Task 4 (Testing the program developed in Task (3)):**

Case1: Transfer ownership:

|  |
| --- |
| **Project Name: iQVR App** |
| **Test Case Template** |
| **Test Case ID: 1 Test Designed by: Alreem zainal** |
| **Test Priority (Low/Medium/High): High Test Designed date: 15 November 2024** |
| **Module Name: Transfer Vehicle Screen Test Executed by: Alreem zainal** |
| **Test Title: Test Execution date: 15 November 2024** |
| **Description: Test the iQVR Transfer page** |
|  |
|  |
| **Pre-conditions: the registration already exists** |
| **Dependencies: None** |
|  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Step | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
| 1 | Navigate to transfer page |  | Page should open | As expected | pass |  |
| 2 | Provide valid VIN | 1234 | Credential can be entered | As expected | pass |  |
| 3 | Provide current owner name | Ahmed | Credential can be entered | As expected | pass |  |
| 4 | Provide current owner QID | A100 | Credential can be entered | As expected | pass |  |
| 5 | Provide current owner Mobile number | 98765 | Credential can be entered | As expected | pass |  |
| 6 | Click on validate button |  | information validated | As expected | pass |  |
| 7 | Provide new owner name | Sara | Credential can be entered | As expected | pass |  |
| 8 | Provide new owner QID | S100 | Credential can be entered | As expected | pass |  |
| 9 | Provide new owner Mobile number | 565656 | Credential can be entered | As expected | pass |  |
| 10 | Click on confirm transfer Button |  | Transfer made | As expected | pass |  |
| **Post conditions:**  Ownership changed, new registration sticker created, invoice was created. | | | | | | |

Case2: Accident Report:

|  |
| --- |
| **Project Name: iQVR App** |
| **Test Case Template** |
| **Test Case ID: 2 Test Designed by: Alreem zainal** |
| **Test Priority (Low/Medium/High): High Test Designed date: 15 November 2024** |
| **Module Name: Report Accident Screen Test Executed by: Alreem zainal** |
| **Test Title: Test Execution date: 15 November 2024** |
| **Description: Test the iQVR Report Accident page** |
|  |
|  |
| **Pre-conditions: The registration already exists for both vehicles.**  **The accident only involves two vehicles.** |
| **Dependencies: None** |
|  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Test Step | Test Data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
| 1 | Navigate to report accident page |  | Page should open | As expected | pass |  |
| 2 | Provide valid VIN of offending vehicle | 1234 | Credential can be entered | As expected | pass |  |
| 3 | Provide valid VIN for victim vehicle | 5678 | Credential can be entered | As expected | pass |  |
| 4 | Click on retrieve button |  | Details text fields appear | As expected | pass |  |
| 5 | Provide accident date | 12/12/2020 | Credential can be entered | As expected | pass |  |
| 6 | Provide accident location | Doha | Credential can be entered | As expected | pass |  |
| 7 | Provide accident description | Accident rear end | Credential can be entered | As expected | pass |  |
| 8 | Click on confirm transfer Button |  | Accident reported | As expected | pass |  |
| **Post conditions:**  Offending vehicle insurance company was informed with the accident details. | | | | | | |

**-How the program was tested:**

The test was done by entering correct or wrong or empty data into the GUI and preserving if the results were as expected.

* **Task 5 (Testing two NFRs):**

**Portability:**

The software components may run on different machines:

Runs on different Oss:

A screenshot of a computer

Description automatically generatedWe tested it on both systems, and it works the same way on macOS and Windows:

macOS

**A screenshot of a computer

Description automatically generated**

Windows

**Reliability:**

-system handles wrong input without crashing:

When wrong or missing input it displays a message.

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

**Modifiability:**

The system can handle new tools:

After creating the transfer and report accident GUI we added the main GUI.

A screenshot of a computer

Description automatically generated