

---

# Unemployed Elves

— Mariia L, Greg Terrell, Dipesh  
Jha, Sameer Baig —

---

# Elevator Pitch

- Problem
  - Contracted companies hired by Amazon have no streamlined form of driver documentation at end of shift
- Solution
  - Internal system that logs driver checklist
  - Closed Database
    - Records all checklists
    - Contains all current drivers

# Overview

Log in to submit an issue:

First Name

Last Name

Driver ID

Phone Number

Verify

# First Page

- User Story

- As an Amazon delivery driver, I want the form to display error messages for invalid inputs, so I can ensure all required information is entered correctly.
- As an Amazon delivery driver, I want to provide my Driver ID along with my personal details, so the system can uniquely identify me.

- Requirement

- Input validation for all required fields.
- Integration with backend API /login.
- Clear UI and error messages for invalid inputs.

# First Page - Design

Sequence Diagram:



# First Page Implementation

**Log in to submit an issue:**

First Name  
Name

Last Name  
Last Name

Driver ID  
A123456

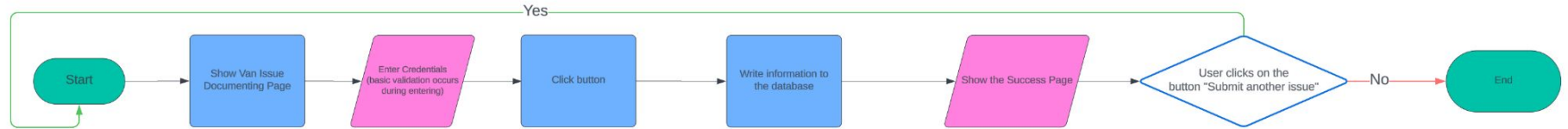
Phone Number  
1111111111

Verify

# Second Page

- User Story
  - As an Amazon delivery driver, I want to document issues with vans without using hand-written notes.
- Requirement
  - The second page shall consist of a data collection form prompting the user to input the following: date and time, VIN, license plate, license state, issue region checklist, and issue description

# Second Page Design





# Second Page Implementation

### Van Issue Documentation Form

**Date and Time \***

**VIN \***

17 characters (e.g., 4Y1SL6S848Z411439)

**License Plate \***

1-7 characters (e.g., 1ABC234)

**License State \***

Select your state

**Issue Region Checklist \***

- ☐ I. Front Side
- ☐ II. Passenger Side
- ☐ III. Back Side
- ☐ IV. Driver Side
- ☐ V. In Cab

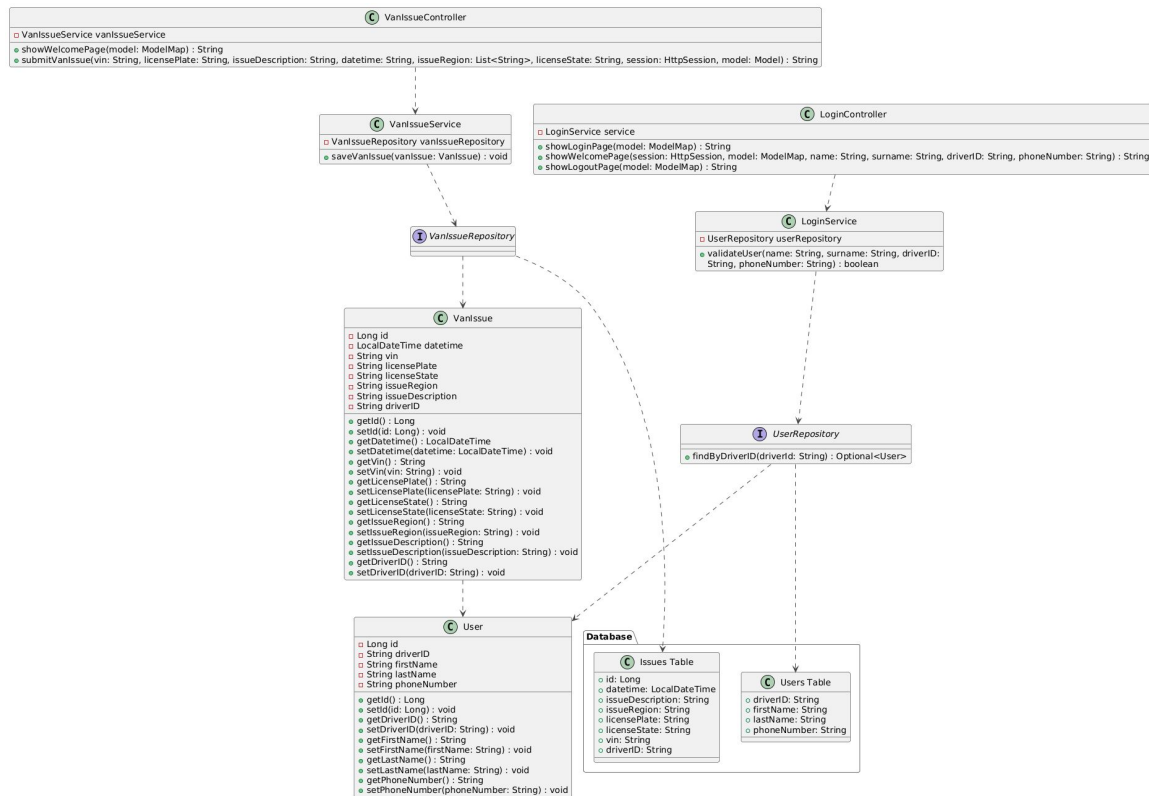
**Description \***

Submit

# Database Setup

- User Story
  - As an Amazon delivery driver, I don't want to repeatedly type in my credentials.
- Requirement
  - Records in the database shall contain driver\_id from the first page to uniquely identify user.
  - The provided information shall be verified against the pre-loaded database of user details.

# Database UML Diagram



# Database Implementation

### Van Issue Documentation Form

Date and Time \*

VIN \*

17 characters (e.g., 4Y1SL65848Z411439)

License Plate \*

1-7 characters (e.g., 1ABC234)

License State \*

Select your state ▾

Issue Region Checklist \*

- ☐ I. Front Side
- ☐ II. Passenger Side
- ☐ III. Back Side
- ☐ IV. Driver Side
- ☐ V. In Cab

Description \*

Submit

## R.3 Testing of User Login Validation (First Page)

As a testing developer, I want to test the user authorization form for different user inputs so I can be sure that only amazon delivery drivers can log in.

### Selenium Integration Testing of the first Page

1. User cannot type any invalid keys to the form. (R.3.3-R.3.6)
2. User cannot access be redirected to the next page if credentials are not in the database, even is basic validation performed well. (R.3.1-R.3.2)

```
1.
public void testInvalidName ()
public void testInvalidSurname ()
public void testInvalidDriverID ()
public void testInvalidPhoneNumber ()
```

```
2.
public void testCorrectCredentials()
public void testIncorrectCredentials()
```

R.3.1: When valid credentials are entered, the user shall be redirected to the next (van issue documenting) page.

R.3.2: An error message shall be displayed indicating that the credentials do not match the pre-loaded database when incorrect credentials are entered (e.g., incorrect driver ID or other invalid fields).

```
[INFO] Running com.example.LoginFormTest
```

```
09:58:18.162 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Using msedge driver 131.0.2903.70 (resolved driver for Edge 131)
09:58:18.203 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Exporting webdriver.edge.driver as C:\Users\maria\.cache\selenium\msedge\win64\131.0.2903.70\msedge.driver.exe
09:58:22.167 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Using msedge driver 131.0.2903.70 (resolved driver for Edge 131)
09:58:22.173 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Exporting webdriver.edge.driver as C:\Users\maria\.cache\selenium\msedge\win64\131.0.2903.70\msedge.driver.exe
09:58:25.259 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Using msedge driver 131.0.2903.70 (resolved driver for Edge 131)
09:58:25.269 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Exporting webdriver.edge.driver as C:\Users\maria\.cache\selenium\msedge\win64\131.0.2903.70\msedge.driver.exe
09:58:28.114 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Using msedge driver 131.0.2903.70 (resolved driver for Edge 131)
09:58:28.122 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Exporting webdriver.edge.driver as C:\Users\maria\.cache\selenium\msedge\win64\131.0.2903.70\msedge.driver.exe
09:58:31.675 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Using msedge driver 131.0.2903.70 (resolved driver for Edge 131)
09:58:31.679 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Exporting webdriver.edge.driver as C:\Users\maria\.cache\selenium\msedge\win64\131.0.2903.70\msedge.driver.exe
09:58:34.611 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Using msedge driver 131.0.2903.70 (resolved driver for Edge 131)
09:58:34.617 [main] INFO io.github.bonigarcia.wdm.WebDriverManager -- Exporting webdriver.edge.driver as C:\Users\maria\.cache\selenium\msedge\win64\131.0.2903.70\msedge.driver.exe
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 19.99 s -- in com.example.LoginFormTest
```

### Unit testing of validateUser service function

1. When we perform validation of user credentials, the credentials are not case-sensitive. (R.3.7)
2. If something does not match what we have in the database (actually mocking here), the function returns false.

```
void testValidateUser_caseInsensitivity ()

void testValidateUser incorrectDriverID ()
void testValidateUser incorrectPhoneNumber ()
void testValidateUser incorrectLastName ()
void testValidateUser_correctData ()
```

```
[INFO] Running com.example.service.LoginServiceTest
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0,
Time elapsed: 0.219 s -- in com.example.service.LoginServiceTest
```

```
[INFO] Results:
[INFO]
[INFO] Tests run: 16, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 35.573 s
[INFO] Finished at: 2024-12-01T09:58:46-05:00
[INFO] -----
```

# Conclusion

- What it accomplishes for the company
  - Adds new loop of communication regarding fleet
  - Cost Benefits
    - Allows contracted company to increase efficiency with vehicle repairs
- What other features could be added in the future
  - Pictures
  - Create new driver credentials
  - Manager can access information
- How this class went overall
  - Guest lectures were an insightful look into how different perspectives in same frameworks