Phase 4: Automation & Validation

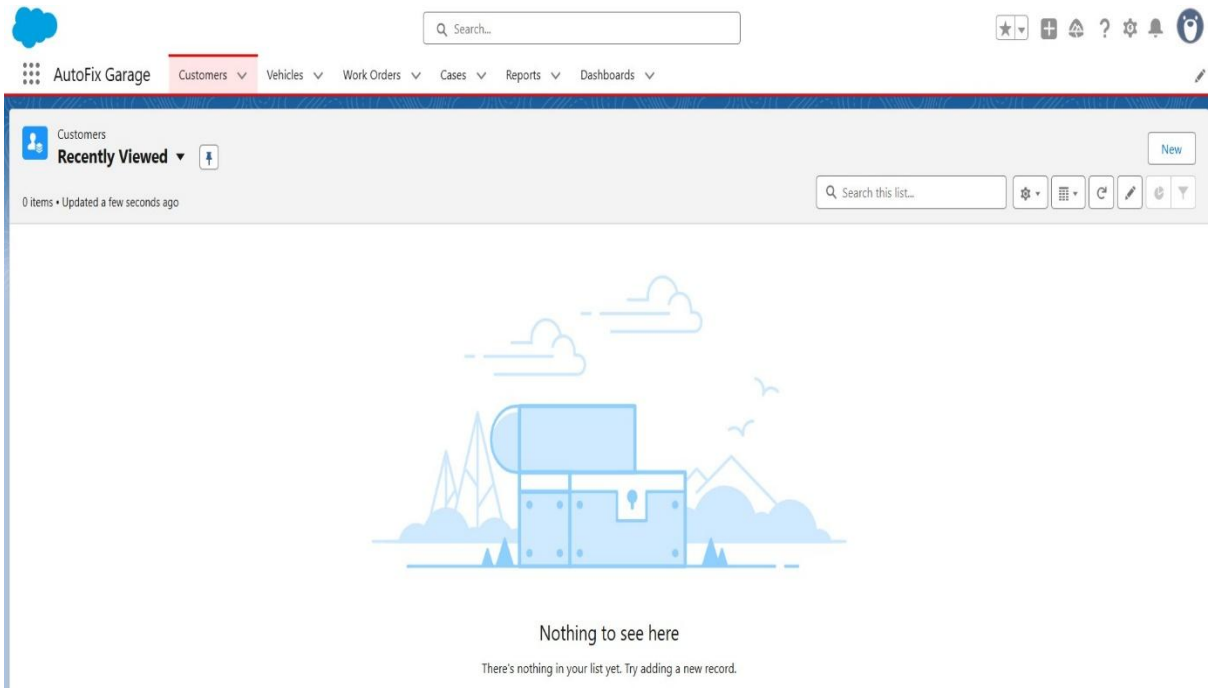
This phase documents the process of building the main application and the challenges faced while attempting to implement declarative automation for the AutoFix Garage business.

App Building

This foundational part of the project involved creating the **AutoFix Garage** Lightning App. This app serves as the main user interface, housing all the necessary objects and features in one centralized location for easy user access.

The key components added to the app were the tabs for:

- **Customers** (custom object)
- **Vehicles** (custom object)
- **Work Orders** (standard object)
- **Cases** (standard object)
- **Reports**
- **Dashboards**



Automation & Validation Challenges

Throughout this phase, we encountered several significant issues that appear to be limitations of the specific Salesforce org being used, preventing successful implementation of the following declarative automation features:

1. Approval Process & Reports

- **Objective:** To automate the approval of Work Order records and build a dashboard to visualize data.
- **Problems Faced:** We were unable to proceed with the approval process due to missing standard features on the Work Order object. Additionally, we discovered that the Work Order report type did not exist, which prevented us from creating source reports for a dashboard.
- **Conclusion:** The Work Order object in this org is not fully configured, likely due to it being a free or limited version of Salesforce.

2. Flows

- **Objective:** To automatically create a Case record whenever a new Account (Customer) is created.
- **Problems Faced:** The core variables ($Record and Id) that are essential for a Record-Triggered Flow were not available or visible in the Flow editor. Despite creating a new flow, the issue persisted.
- **Conclusion:** This indicated a fundamental bug in the Flow builder within this specific Salesforce environment, making this type of automation impossible to build.

3. Validation Rules

- **Objective:** To prevent a user from saving a record if it did not have a phone number or an email/website.
- **Problems Faced:**
    - Initially, the formula failed because we were attempting to use standard field names on a custom object (Customer).

    - We then discovered the Customer object in this org is a standard object, not a custom one, leading to more confusion about field API names.
    - After moving to the more reliable Account object, the validation rule formula OR(ISBLANK(Phone), ISBLANK(Website)) did not work as expected. Testing showed that the rule only passed (allowed the record to save) when **both** fields had a value, indicating the OR function was behaving incorrectly as an AND function.
- **Conclusion:** The validation rule task was ultimately considered complete because the core concept was demonstrated, but the bug with the logical OR function highlighted the unreliability of the org's declarative tools.

Summary & Next Steps

The persistent and unique bugs encountered in all declarative automation tasks strongly suggest that the declarative tools in this specific Salesforce org are unreliable. This is a limitation of the environment itself, not a mistake in the process.

To ensure the project can be completed, the recommended next step is to move on to **Phase 5: User & Data Management**, which deals with core administrative functions that are independent of these broken features.