

* Erohal

**Functional Specifications and Methodology**

**Project Advisor:**

Mr. Muhammad Ishaq Raza

**Group Members:**

Muhammad Arsil Khan 16L-4167

Ali Iqbal 16L-4134

Usama Azeem 16L-4351

Abdul Akhir Ihsan 16L-4265

National University of Computer and Emerging Sciences

Department of Computer Science

Lahore, Pakistan

Table of Contents

[List of Tables iv](#_Toc86365656)

[List of Figures vi](#_Toc86365657)

[1. Introduction 1](#_Toc86365658)

[1.1 Purpose of this Document 1](#_Toc86365659)

[1.2 Intended Audience 1](#_Toc86365660)

[1.3 Definitions, Acronyms, and Abbreviations 1](#_Toc86365661)

[2. General Description 3](#_Toc86365662)

[2.1 User Characteristics 3](#_Toc86365663)

[2.2 Domain Overview 3](#_Toc86365664)

[3. Functionality 4](#_Toc86365665)

[3.1 Functional Requirements 4](#_Toc86365666)

[3.1.1 Functional requirements for Users 4](#_Toc86365667)

[3.2 Non-Functional Requirements 4](#_Toc86365668)

[3.2.1 Availability 4](#_Toc86365669)

[3.2.2 Usability 4](#_Toc86365670)

[3.2.3 Reliability 4](#_Toc86365671)

[3.2.4 Scalability 4](#_Toc86365672)

[3.2.5 Data Integrity 4](#_Toc86365673)

[3.3 Assumptions 5](#_Toc86365674)

[4. System Architecture 6](#_Toc86365675)

[5. Use Cases 8](#_Toc86365676)

[5.1 Login Account 8](#_Toc86365677)

[5.2 Create Account 8](#_Toc86365678)

[5.3 Inspection Details 8](#_Toc86365679)

[5.4 View Issues List 9](#_Toc86365680)

[5.5 Issues Details 9](#_Toc86365681)

[5.6 Edit Issue 10](#_Toc86365682)

[5.7 Create Inspection Template 10](#_Toc86365683)

[5.8 Update Inspection Template Type 11](#_Toc86365684)

[5.9 Add Item to Existing Template 11](#_Toc86365685)

[5.10 Remove Inspection Template Item 12](#_Toc86365686)

[5.11 Update Inspection Item Type 12](#_Toc86365687)

[5.12 Create Issue 13](#_Toc86365688)

[5.13 Ignore Inspection 14](#_Toc86365689)

[5.14 Merge Issues 15](#_Toc86365690)

[5.15 View all Service Requests 16](#_Toc86365691)

[5.16 Create new Service Request 16](#_Toc86365692)

[5.17 Examine Service Request 17](#_Toc86365693)

[5.18 View Work Orders 17](#_Toc86365694)

[5.19 Examine Work Order 18](#_Toc86365695)

[5.20 Create new Work Order 18](#_Toc86365696)

[5.21 Select Inspection Template 19](#_Toc86365697)

[5.22 Creating New Inspection 19](#_Toc86365698)

[5.23 Add Issue in Work Order 20](#_Toc86365699)

[5.24 User Assignment 20](#_Toc86365700)

[5.25 Asset Information 21](#_Toc86365701)

[5.26 Work Order Assignment 21](#_Toc86365702)

[5.27 Timing 22](#_Toc86365703)

[5.28 Parts and Cost 22](#_Toc86365704)

[5.29 View Assets 23](#_Toc86365705)

[5.30 Create Asset 23](#_Toc86365706)

[5.31 Select Asset Category 24](#_Toc86365707)

[5.32 Add Asset Image 25](#_Toc86365708)

[5.33 Add an assignment 25](#_Toc86365709)

[5.34 Delete an assignment 26](#_Toc86365710)

[5.35 View Usages 27](#_Toc86365711)

[5.36 View filtered usages of an asset 27](#_Toc86365712)

[5.37 View Consumptions 28](#_Toc86365713)

[5.38 View filtered consumptions of an asset 28](#_Toc86365714)

[5.39 Edit Profile 29](#_Toc86365715)

[5.40 View User Profile 29](#_Toc86365716)

[5.41 View Assets Assignment History 30](#_Toc86365717)

[5.42 Change Password 30](#_Toc86365718)

[5.43 View User Management Page 31](#_Toc86365719)

[5.44 Add a New Role 31](#_Toc86365720)

[5.45 Delete a Role 32](#_Toc86365721)

[5.46 Update a Role 33](#_Toc86365722)

[5.47 View All User groups 34](#_Toc86365723)

[5.48 View a User group 34](#_Toc86365724)

[5.49 Remove a Member from a User Group 35](#_Toc86365725)

[5.50 Add a Member to a User Group 36](#_Toc86365726)

[5.51 Create a User Group 36](#_Toc86365727)

[5.52 Create a User 37](#_Toc86365728)

[5.53 View a User’s Personal Information Page 37](#_Toc86365729)

[5.54 Edit User’s Personal Information 38](#_Toc86365730)

[6. Graphical User Interfaces 40](#_Toc86365731)

[7. Database Design 60](#_Toc86365732)

[7.1 ER Diagram 60](#_Toc86365733)

[7.2 Data Dictionary 64](#_Toc86365734)

[7.2.1 Image Voices 64](#_Toc86365735)

[7.2.2 Inspection Template 64](#_Toc86365736)

[7.2.3 Inspection Items 65](#_Toc86365737)

[7.2.4 Inspection Item Category 65](#_Toc86365738)

[7.2.5 Inspection Report 66](#_Toc86365739)

[7.2.6 Inspection Report Category 66](#_Toc86365740)

[7.2.7 Inspection Report Field 67](#_Toc86365741)

[7.2.8 Issue Reporting 68](#_Toc86365742)

[7.2.9 Assign 69](#_Toc86365743)

[7.2.10 Time 69](#_Toc86365744)

[7.2.11 Work Order 70](#_Toc86365745)

[7.2.12 Plannings 71](#_Toc86365746)

[7.2.13 Costs 71](#_Toc86365747)

[7.2.14 Issue Resolutions 72](#_Toc86365748)

[7.2.15 Part Used 72](#_Toc86365749)

[7.2.16 Tracking Report 73](#_Toc86365750)

[7.2.17 User Action 73](#_Toc86365751)

[7.2.18 Asset 74](#_Toc86365752)

[7.2.19 Consumption 75](#_Toc86365753)

[7.2.20 Usage 76](#_Toc86365754)

[7.2.21 Asset Field 77](#_Toc86365755)

[7.2.22 Asset Images 77](#_Toc86365756)

[7.2.23 Attachments 77](#_Toc86365757)

[7.2.24 Category 77](#_Toc86365758)

[7.2.25 Field Template 78](#_Toc86365759)

[7.2.26 Field 78](#_Toc86365760)

[7.2.27 Company 79](#_Toc86365761)

[7.2.28 Company Field 79](#_Toc86365762)

[7.2.29 Employee Category 80](#_Toc86365763)

[7.2.30 User 80](#_Toc86365764)

[7.2.31 User Field 81](#_Toc86365765)

[7.2.32 User Preferences 81](#_Toc86365766)

[7.2.33 User Group Members 82](#_Toc86365767)

[7.2.34 User Groups 82](#_Toc86365768)

[7.2.35 Role 82](#_Toc86365769)

[7.2.36 Permission 83](#_Toc86365770)

[7.2.37 Notifications 83](#_Toc86365771)

[7.2.38 Assignment Record 84](#_Toc86365772)

[7.2.39 Assignment Record Archive 84](#_Toc86365773)

[7.2.40 Reminder 85](#_Toc86365774)

[7.2.41 Firebase User Tokens 85](#_Toc86365775)

[8. Risk Analysis 86](#_Toc86365776)

[9. System Requirements 87](#_Toc86365777)

[9.1 Hardware Requirements 87](#_Toc86365778)

[9.2 Software Requirements 87](#_Toc86365779)

[10. References 88](#_Toc86365780)

List of Tables

Table 1:[[Login Account 9](#_Toc19700198)](#_Login_Account)

[[Table 2: View Inspections List 9](#_Toc19700198)](#_Toc35691363)

[[Table 3: Inspection Details 10](#_Toc19700198)](#_Toc35691364)

[[Table 4: View Issues List 10](#_Toc19700198)](#_Toc35691365)

[[Table 5: Issues Details 11](#_Toc19700198)](#_Toc35691366)

[[Table 6: Edit Issue 11](#_Toc19700198)](#_Toc35691367)

[[Table 7: Create Inspection Template 12](#_Toc19700198)](#_Toc35691368)

[[Table 8: Update Inspection Template Type 12](#_Toc19700198)](#_Toc35691369)

[[Table 9: Add Item to Existing Template 13](#_Toc19700198)](#_Toc35691370)

[[Table 10: Remove Inspection Template Item 13](#_Toc19700198)](#_Toc35691371)

[[Table 11: Update Inspection Item Type 14](#_Toc19700198)](#_Toc35691372)

[[Table 12: Create Issue 15](#_Toc19700198)](#_Toc35691373)

[[Table 13: Ignore Inspection 16](#_Toc19700198)](#_Toc35691374)

[[Table 14: Merge Issues 17](#_Toc19700198)](#_Toc35691375)

[[Table 15: View all Service Requests 17](#_Toc19700198)](#_Toc35691376)

[[Table 16: Create new Service Request 18](#_Toc19700198)](#_Toc35691377)

[[Table 17: Examine Service Request 18](#_Toc19700198)](#_Toc35691378)

[[Table 18: View Work Orders 19](#_Toc19700198)](#_Toc35691379)

[[Table 19: Examine Work Order 19](#_Toc19700198)](#_Toc35691380)

[[Table 20: Create new Work Order 20](#_Toc19700198)](#_Toc35691381)

[[Table 21: Select Inspection Template 20](#_Toc19700198)](#_Toc35691382)

[[Table 22: Creating New Inspection 21](#_Toc19700198)](#_Toc35691383)

[[Table 23: Add Issue in Work Order 21](#_Toc19700198)](#_Toc35691384)

[[Table 24: User Assignment 22](#_Toc19700198)](#_Toc35691385)

[[Table 25: Asset Information 22](#_Toc19700198)](#_Toc35691386)

[[Table 26: Work Order Assignment 23](#_Toc19700198)](#_Toc35691387)

[[Table 27: Timing 23](#_Toc19700198)](#_Toc35691388)

[[Table 28: Parts and Cost 24](#_Toc19700198)](#_Toc35691389)

[[Table 29: View Assets 24](#_Toc19700198)](#_Toc35691390)

[[Table 30: Create Asset 25](#_Toc19700198)](#_Toc35691391)

[[Table 31: Select Asset Category 26](#_Toc19700198)](#_Toc35691392)

[[Table 32: Add Asset Image 26](#_Toc19700198)](#_Toc35691393)

[[Table 33: Add an assignment 27](#_Toc19700198)](#_Toc35691394)

[[Table 34: Delete an assignment 28](#_Toc19700198)](#_Toc35691395)

[[Table 35: View Usages 28](#_Toc19700198)](#_Toc35691396)

[[Table 36: View filtered usages of an asset 29](#_Toc19700198)](#_Toc35691397)

[[Table 37: View Consumptions 29](#_Toc19700198)](#_Toc35691398)

[[Table 38: View filtered consumptions of an asset 30](#_Toc19700198)](#_Toc35691399)

[[Table 39: Edit Profile 30](#_Toc19700198)](#_Toc35691400)

[[Table 40: View User Profile 31](#_Toc19700198)](#_Toc35691401)

[[Table 41: View Assets Assignment History 31](#_Toc19700198)](#_Toc35691402)

[[Table 42: Change Password 32](#_Toc19700198)](#_Toc35691403)

[[Table 43: View User Management Page 32](#_Toc19700198)](#_Toc35691404)

[[Table 44: Add a New Role 33](#_Toc19700198)](#_Toc35691405)

[[Table 45: Delete a Role 34](#_Toc19700198)](#_Toc35691406)

[[Table 46: Update a Role 35](#_Toc19700198)](#_Toc35691407)

[[Table 47: View All User groups 35](#_Toc19700198)](#_Toc35691408)

[[Table 48: View a User group 36](#_Toc19700198)](#_Toc35691409)

[[Table 49: Remove a Member from a User Group 37](#_Toc19700198)](#_Toc35691410)

[[Table 50: Add a Member to a User Group 38](#_Toc19700198)](#_Toc35691411)

[[Table 51: Create a User Group 38](#_Toc19700198)](#_Toc35691412)

[[Table 52: Create a User 39](#_Toc19700198)](#_Toc35691413)

[[Table 53: View a User’s Personal Information Page 39](#_Toc19700198)](#_Toc35691414)

[[Table 54: Edit User’s Personal Information 40](#_Toc19700198)](#_Toc35691415)

[Table 55: Image Voices 64](#_Toc35706568)

[Table 56: Inspection Template 64](#_Toc35706569)

[Table 57: Inspection Items 65](#_Toc35706570)

[Table 58: Inspection Item Category 65](#_Toc35706571)

[Table 59: Inspection Report 66](#_Toc35706572)

[Table 60: Inspection Report Category 66](#_Toc35706573)

[Table 61: Inspection Report Field 67](#_Toc35706574)

[Table 62: Issue Reporting 68](#_Toc35706575)

[Table 63: Assign 69](#_Toc35706576)

[Table 64: Time 69](#_Toc35706577)

[Table 65: Work Order 70](#_Toc35706578)

[Table 66: Plannings 71](#_Toc35706579)

[Table 67: Costs 71](#_Toc35706580)

[Table 68: Issue Resolutions 72](#_Toc35706581)

[Table 69: Part Used 72](#_Toc35706582)

Table 70: [Tracking Report 73](#_Toc35706583)

[Table 71: User Action 73](#_Toc35706584)

[Table 72: Asset 74](#_Toc35706585)

[Table 73: Consumption 75](#_Toc35706586)

[Table 74: Usage 76](#_Toc35706587)

[Table 75: Asset Field 77](#_Toc35706588)

[Table 76: Asset Images 77](#_Toc35706589)

[Table 77: Attachments 77](#_Toc35706590)

[Table 78: Category 77](#_Toc35706591)

[Table 79: Field Template 78](#_Toc35706592)

[Table 80: Field 78](#_Toc35706593)

[Table 81: Company 79](#_Toc35706594)

[Table 82: Company Field 79](#_Toc35706595)

[Table 83: Employee Category 80](#_Toc35706596)

[Table 84: User 80](#_Toc35706597)

[Table 85: User Field 81](#_Toc35706598)

[Table 86: User Preferences 81](#_Toc35706599)

[Table 87: User Group Members 82](#_Toc35706600)

[Table 88: User Groups 82](#_Toc35706601)

[Table 89: Role 82](#_Toc35706602)

[Table 90: Permission 83](#_Toc35706603)

[Table 91: Notifications 83](#_Toc35706604)

[Table 92: Assignment Record 84](#_Toc35706605)

[Table 93: Assignment Record Archive 84](#_Toc35706606)

[Table 94: Reminder 85](#_Toc35706607)

[Table 95: Firebase User Tokens 8](#_Firebase_User_Tokens)5

List of Figures

Figure 1: Level 1 System Architecture 8

Figure 2: Level 2 System Architecture 9

Figure 3: Login 42

Figure 4: Dashboard 42

Figure 5: Inspections List 43

Figure 6: Inspection Details 43

Figure 7: Issues List 44

Figure 8: Issue Details 1 44

Figure 9: Issue Details 2 45

Figure 10: Inspection Template Creation 45

Figure 11: Issue Creation 46

Figure 12: Work Orders 46

Figure 13: Creating Work Order 1 47

Figure 14: Creating Work Order 2 47

Figure 15: View Work Orders 48

Figure 16: Work Order Details 48

Figure 17: Assigning Work Orders 49

Figure 18: Work Orders Timing 49

Figure 19: Parts and Cost 50

Figure 20: Assets 50

Figure 21: View Asset 1 51

Figure 22: View Asset 2 51

Figure 23: View Asset 3 52

Figure 24: View Asset 4 52

Figure 25: View Asset 5 53

Figure 26: Create New Asset 53

Figure 27: View Usage 54

Figure 28: View Consumption 54

Figure 29: Profile Overview 55

Figure 30: Asset Assignment History 55

Figure 31: Change Password 56

Figure 32: Change Other's Password 56

Figure 33: Notifications 1 57

Figure 34: Notifications 2 57

Figure 35: User Management 58

Figure 36: Roles and Permissions 58

Figure 37: User Groups 59

Figure 38: Create User Group 59

Figure 39: Create User 60

Figure 40: Personal Information 60

Figure 41: User Group Profile 61

Figure 42: ER Diagram 1 62

Figure 43: ER Diagram 2 63

Figure 44: ER Diagram 3 64

Figure 45: ER Diagram 4 65

# Introduction

The manual inspection of your asset can be tricky and inaccurate at times and when it contains multiple sources then handling each of them at the same time becomes quite impossible. This leads to less maintenance which increases the risk of your asset failure and ultimately leads to costly repairs, which are approximately 30% of the life-cycle costs of a vehicle [1]. With the advancement in technology it is questionable to not have a system that can provide automated timely inspection reports with predictive failure analysis and potentially enable mechanisms to avoid or counteract failure, or recommend maintenance of asset prior to failure.

The purpose of this project is to create an automated system that can handle huge amount of data coming from various Teltonika IOT devices i.e., FMB 920 and FMB 630, and then using Artificial Intelligence and Data Science, to optimize and analyze it in such a way that it can predict precise components failure and avoid the unnecessary inspection and provide with a maintenance schedule that is based on the usual condition of the vehicle in its operating conditions [2].

Our main focus will be on vehicle components live inspection, maintenance and failure. The system will be gathering huge data of different components of a vehicle from installed IOT devices which will contain geographical location, trip data and crash detection etc. to train our predictive analysis model that will eventually help us in detection of failure points. We will also be maintaining a record of all previous maintenance and failures of components of each asset to further enhance our predictions with time.

## Purpose of this Document

The purpose of this document is to give a detailed overview of the project. The document would be working as a guideline for the development team on what is to be actually implemented. Moreover, this would explain all the functional/ non-functional requirements of the project. The document would be providing a detailed description of the use cases and the system architecture. Furthermore, some insight on what the actual system would look like in its final form. The database describes what data would be stored in order to work on the system. Also, the hardware/ software requirements are also described in this document.

## Intended Audience

The intended audience of Erohal are the managers and owners of large organizations. These large organizations have numeric complex assets in abundance which they have to manage, maintain and analyze, and hence have a very tedious and complicated process for it.

## Definitions, Acronyms, and Abbreviations

Erohal: Name of the product

Asset: A valuable item having a complexity that makes it tedious to maintain

Inspection: A detailed observation of the components of an asset.

Issue: Problematic components identified during inspections

Service Request: Asset or its components maintenance and servicing.

Work order: Record of work done for resolving issues.

Preventive Maintenance: Ability to predict maintenance schedules for assets and its components.

Anomaly detection: Ability to automatically identify problematic components and then earlier in time.

# General Description

## User Characteristics

The End Users are classified into two groups namely the Administrators and General Users. The General Users will be the caretakers, drivers, maintainers, repairmen etc. who will be providing initial and routine information of Assets, conducting inspections, reporting issues, carry out and update work orders, etc. The Administrators will be the managers of an organization/department which will be responsible for managing roles and permissions, Asset assignments and their Inspection templates, manage issues and create and plan work orders for their resolution etc.

## Domain Overview

Erohal aims to ease the management and maintenance of Assets throughout the lifetime of an Asset in an organization, the Asset gets registered in Erohal, then it is assigned to various general users throughout its lifetime, the Asset is then monitored and periodically inspected by Users who then report issues and service requests against the Asset, the Administrators then analyze and marge the related issues for planning and creating a work order, assigning the Work orders to relevant repairmen which in turn carry out inspections before, after and during work orders. IOT plays an intelligent role in Erohal in terms of continuous monitoring of vital components of an Asset to identify anomalies and malfunctions and then eventually leading to planning and creating preventative maintenance measures and schedules for the Asset.

# 

# Functionality

## Functional Requirements

### Functional requirements for Users

* System will allow user to login by authenticating user login credentials.
* System will allow user to create a new account by providing required information.
* System will allow user to reset password via email.
* System will allow user to logout.
* System will allow user to edit profile.
* System will display clothing products based on user’s profile.
* System will allow user to filter out products based on Type.
* System will allow user to filter out products based on Sizes.
* System will allow user to filter out products based on Colors.
* System will allow user to filter out products based on Waists.
* System will allow user to filter out products based on Price Range.
* System will allow user to filter out products based on Discount factor.
* System will allow user to view selected product details.
* System will allow user to buy product by redirecting to respected page.
* System will allow user to add products to favorites.

## Non-Functional Requirements

### Availability

System will be available for the users for at-least 160 hours a week.

### Usability

System will have an intuitive design.

* Our system interface will be easy to learn and user interactive. A rookie will require at-most 5 seconds to learn the interface.

### Reliability

System will show the result of the desired filter within 3 seconds.

### Scalability

System will ensure there will be no significant performance degrade for at-most 100 users.

### Data Integrity

System will ensure user’s personal data is not tempered by unauthorized source.

## Assumptions

* Internet facility is available.
* Intended users are educated enough to operate and understand the internet.

# 

# System Architecture

Following is the diagram representing the system architecture of our project:

Level 1:

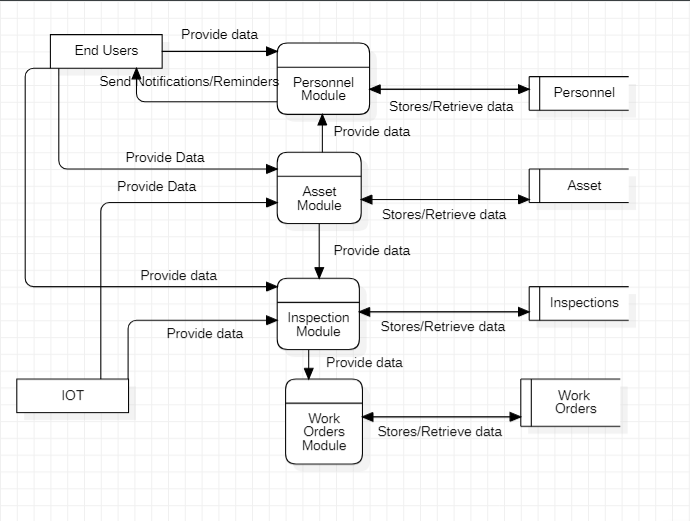


Figure : Level 1 System Architecture

This is the level 1 diagram of system architecture

Level 2:

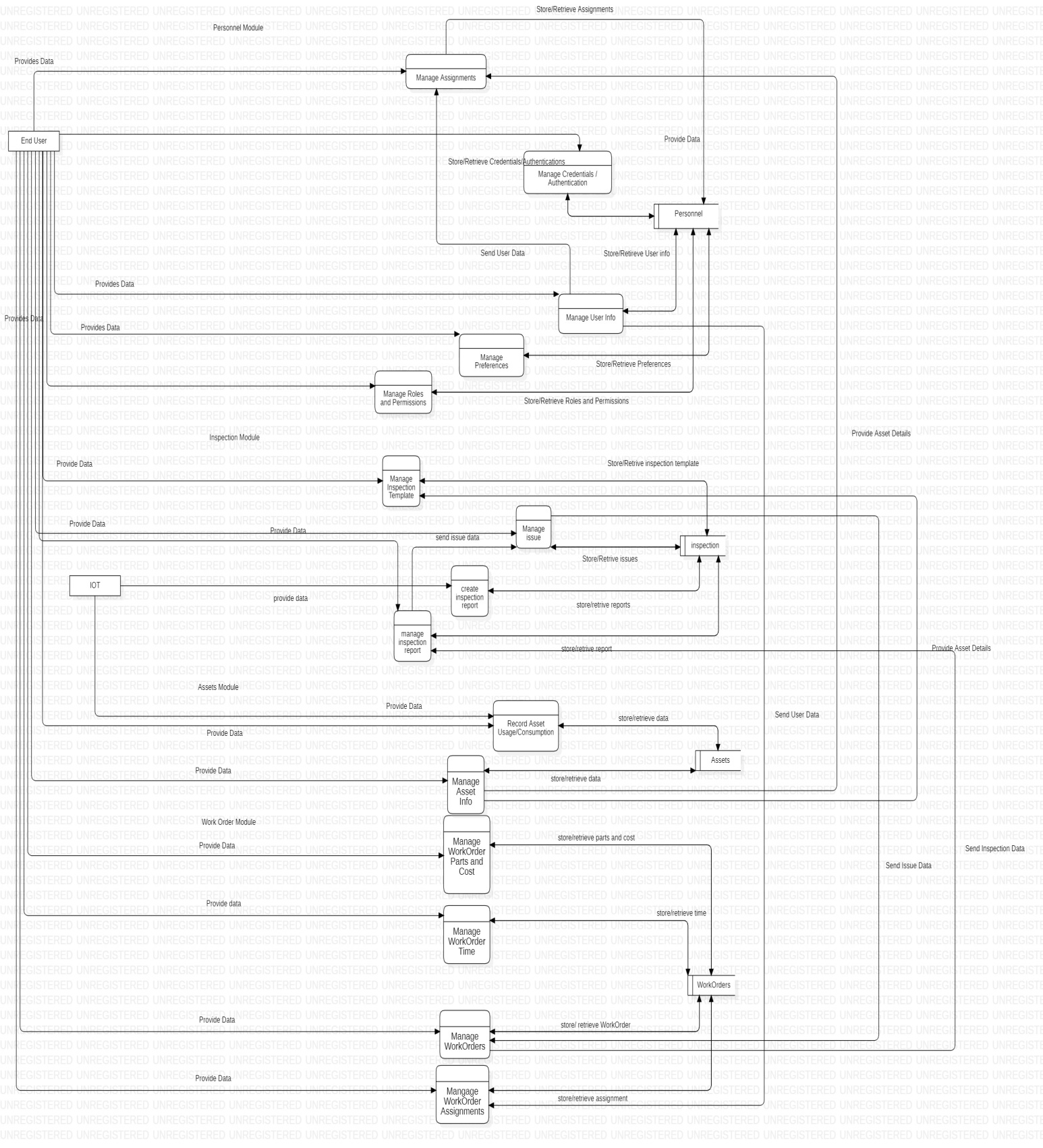


Figure : Level 2 System Architecture

This is the level 2 diagram of system architecture

# Use Cases

## Login Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Login Account | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to login into account. | | |
| **Pre-Conditions** | | User should be registered in the system.  User should not already be logged in. | | |
| **Post-Conditions** | | User shall be logged in successfully. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User opens the login page. | | 2 | System displays Login page asking for username and password. |
| 3 | User enters valid login credentials and press login button. | | 4 | System verifies the user credentials, establishes session and redirects the user to the home page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid login credentials and presses the login button. | | 4-A | System prompts the error message: *Incorrect username or password entered.* |

## Create Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Create Account | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to create a new account by providing the required information. | | |
| **Pre-Conditions** | | User clicked on Sign-up button from the login page. | | |
| **Post-Conditions** | | User’s account shall be created successfully. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the Sign-up button from the Login page. | | 2 | System displays Sign-up page asking for required information. |
| 3 | User enters valid information and presses submit button. | | 4 | System verifies the information, creates a new account and redirects the user to the login page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid information and presses submit button. | | 4-A | System prompts the error message: *Incorrect/Missing required information* |

## Reset Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Reset Password | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to reset his/her account password. | | |
| **Pre-Conditions** | | User should have an existing account.  User clicked on “Forget Password?” button from Login page. | | |
| **Post-Conditions** | | User shall receive reset password request on his/her email. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Forget Password?” button from the Login page. | | 2 | System displays a prompt asking for user’s username. |
| 3 | User enters username and presses submit button. | | 4 | System verifies the username, sends an email to the respective user account and redirects the user to the login page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid username and presses submit button. | | 4-A | System prompts the error message: *Username not found* |

## Logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Logout | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to logout his/her account. | | |
| **Pre-Conditions** | | User should have logged in.  User clicked on “Logout” button from Home page. | | |
| **Post-Conditions** | | User shall be logged out. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Logout?” button from the home page. | | 2 | System will logout the user and redirect the user to Login page. |
| **No Alternative Flow** | | | | |

## Edit Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Edit Profile | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to edit his/her profile. | | |
| **Pre-Conditions** | | User should have logged in.  User clicked on “Edit Profile” button from Home page. | | |
| **Post-Conditions** | | User’s profile shall be updated. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Edit Profile” button from the Home page. | | 2 | System displays a new page containing user’s current information. |
| 3 | User edits his/her profile by providing valid information and presses update button. | | 4 | System validates the information, updates his/her profile and redirects the user to the Home page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid profile information and presses update button. | | 4-A | System prompts the error message: *Information provided is incorrect.* |

## View Product Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View Product Details | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to view the details of the selected product. | | |
| **Pre-Conditions** | | User should have logged in.  User should have clicked on a “Details” button. | | |
| **Post-Conditions** | | Users shall be to see the product details. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Details” button from the Home page. | | 2 | System displays a pop-up containing product details. |
| **No Alternative Flow** | | | | |

## Buy Product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Buy a Product | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to buy a product from the respected brand’s website. | | |
| **Pre-Conditions** | | User must be logged in.  User must have clicked on “Detail” button for the selected product. | | |
| **Post-Conditions** | | System will redirect user to the respective brand website. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Buy” button located in the product details. | | 2 | System redirects the user to the respected link of the product. |
| **No Alternative Flow** | | | | |

## 

## Add to Favorites

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Add to favorites | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to add a product to his/her favorites list. | | |
| **Pre-Conditions** | | User must be logged in.  Selected product should not be in the favorites list | | |
| **Post-Conditions** | | System will update the favorites list by adding a product. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Favorites” icon from the selected product. | | 2 | System adds the product to his/her favorites list. |
| **No Alternative Flow** | | | | |

## Remove from favorites

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Remove from favorites | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to remove a product from his/her favorites list. | | |
| **Pre-Conditions** | | User must be logged in.  Selected product should be in the favorites list | | |
| **Post-Conditions** | | System will update the favorites list by removing a product. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Favorites” icon from the selected product. | | 2 | System will remove the product from his/her favorites list. |
| **No Alternative Flow** | | | | |

## View favorites list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View favorites list | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to view favorites list. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User shall be able to view his/her favorites list. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “View favorites” button from the home page. | | 2 | System redirects the user to the new page containing a list of user’s favorite items. If there are no products in the favorites list then an empty list will be displayed. |
| **No Alternative Flow** | | | | |

## Filter Product by Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by type | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product type. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Type” filter from the dropdown menu. | | 4 | System displays a pop-up of types for clothing products e.g. (Men’s T-Shirts, Men’s Polo Shirts, Men’s Trousers). |
| 5 | User selects a type from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Size

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by size | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product sizes. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Size” filter from the dropdown menu. | | 4 | System displays a pop-up of sizes for clothing products e.g. (Small, Large). |
| 5 | User selects a size from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Color

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by color | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product color. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Color” filter from the dropdown menu. | | 4 | System displays a pop-up of colors for clothing products e.g. (Red, Green). |
| 5 | User selects a color from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Waist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by waist | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product waist. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Waist” filter from the dropdown menu. | | 4 | System displays a pop-up of waists for clothing products e.g. (30-inch, 32-inch). |
| 5 | User selects a waist from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Price Range

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by price range. | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product price range. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Price” filter from the dropdown menu. | | 4 | System displays a pop-up of price range for clothing products e.g. (Rs. 1200-1800, Rs. 750-1000). |
| 5 | User selects a price range from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Filter Product by Discount Factor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Filter product by discount factor | | |
| **Actors** | | Client | | |
| **Summary** | | User will be able to filter out products based on product discount factor. | | |
| **Pre-Conditions** | | User must be logged in. | | |
| **Post-Conditions** | | User will be able to view filtered product list. | | |
| **Special Requirements** | | Reliability (filtering takes less than 3 seconds). | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the “Filter” button from the home page. | | 2 | System displays a dropdown menu containing different types of filters. |
| 3 | User selects the “Discounted” filter from the dropdown menu. | | 4 | System displays a pop-up of Discount range for clothing products e.g. (20% - 30% off). |
| 5 | User selects a Discount range from the pop-up window. | | 6 | System reloads the home page and displays the products based on the user’s filtering criteria. |
| **No Alternative Flow** | | | | |

## Examine Service Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View individual service request. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to see service request of particular person. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also on service request section. | | |
| **Post-Conditions** | | Admin should be able to examine each service request. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on left side search button. | | 2 | Another page open with all the information of that service request will open. |
| **No Alternative Flow** | | | | |

## View Work Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View work orders | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to see all work orders. | | |
| **Pre-Conditions** | | Admin should be login in the system. | | |
| **Post-Conditions** | | Admin should see all work orders successfully | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on work orders from left side menu. | | 2 | All work orders will be displayed. |
| **No Alternative Flow** | | | | |

## Examine Work Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View individual work order. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be work order of particular issue. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also on work order section. | | |
| **Post-Conditions** | | Admin should be able to examine that work order. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on left side search button. | | 2 | Another page open with all the information of that work order will open. |
| **No Alternative Flow** | | | | |

## Create new Work Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Create new work order | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will create new service request. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also on the work order section or examining the issue or can also create work order in issue section by selecting a particular issue. | | |
| **Post-Conditions** | | Admin should be able to create new work order of a selected issue section successfully. | | |
| **Special Requirements** | | There should be an issue of asset. | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on create new in work order. | | 2 | Redirect the admin in issue section where all issues all visible. |
| 3 | Select a particular issue. | | 4 | Issue will be selected. |
| 5 | Admin then again click on create. | | 6 | System will show the admin to create work order for existing work order or create new. |
| 7 | Admin select one new option. | | 8 | A new work order will be created. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on create work order button. | | 2-A | If there remain one requirement unfilled then system will show the admin that these fields are required. |

## Select Inspection Template

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Select Inspection Template. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will select existing inspection template in creating new work order. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also creating new work order. | | |
| **Post-Conditions** | | System will add new inspection template in creating work order. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on left side add existing inspection template button. | | 2 | Another page open asking to admin what type of inspection you want i.e. asset, asset category, work order. |
| 3 | Admin select one of category. | | 4 | A new inspection template will add. |
| **No Alternative Flow** | | | | |

## Creating New Inspection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Creating new Inspection. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will create new inspection for work order. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also creating new work order. | | |
| **Post-Conditions** | | System will add new inspection for creating work order. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on left side ‘+’ button. | | 2 | Another page open asking to admin what type of inspection you want. |
| 3 | Admin select one of category. | | 4 | Category will add in inspection. |
| 5 | Admin can also add more fields about inspection for work order. | | 6 | All this will be stored in inspection template. |
| **No Alternative Flow** | | | | |

## Add Issue in Work Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Add issue in work order. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will add issue in work order. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also creating new work order. | | |
| **Post-Conditions** | | System will add new issue in that work order. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin click on add issue button which is at the bottom. | | 2 | System will ask the admin for issue. |
| 3 | Admin will select category. | | 4 | An issue category will be added. |
| **No Alternative Flow** | | | | |

## User Assignment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | User Assignment. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will assign work order to user or group. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also creating new work order. | | |
| **Post-Conditions** | |  | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin user assignment. | | 2 | System will ask admin for user or group. |
| 3 | Admin will select one option. | | 4 | Work order will be assigned. . |
| **No Alternative Flow** | | | | |

## Asset Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Asset Information. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will see asset information in work order. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also creating new work order. | | |
| **Post-Conditions** | | Admin will see asset information. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin will click on asset information. | | 2 | System will show the information about the asset. |
| **No Alternative Flow** | | | | |

## Work Order Assignment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Work Order Assignment. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will see that whom the work order was assigned. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also examining work order. | | |
| **Post-Conditions** | | Admin will see work order assignment. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin will click on assignment. | | 2 | System will show information of assigned work order. |
| **No Alternative Flow** | | | | |

## Timing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Timing. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will see of timing of work order and can also add time. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also examining work order. | | |
| **Post-Conditions** | | Admin will see timing of work order. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin will click on timing. | | 2 | System will show the timing. |
| 3 | Admin click on add time. | | 4 | System will ask time and work order category. |
| 5 | Admin will write all information and timing | | 6 | System will add new time. |
| **No Alternative Flow** | | | | |

## Parts and Cost

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Parts and Cost. | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will see parts and cost. He can also add parts and cost in work order. | | |
| **Pre-Conditions** | | Admin should be registered in the system and also examining work order. | | |
| **Post-Conditions** | | Admin will see added parts of work order and also add parts and their cost. | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Admin will click on parts and costs. | | 2 | System will show added parts and cost in work order. |
| 3 | Admin click on add parts. | | 4 | System will ask parts information. |
| 5 | Admin will click add costs. | | 6 | System will ask category and cost. |
| 7 | Admin will click Add button. | | 8 | New parts and cost will be added. |
| **No Alternative Flow** | | | | |

## View Assets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View Assets | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to view his/her assets | | |
| **Pre-Conditions** | | User should be logged in. | | |
| **Post-Conditions** | | User successfully viewed his/her assets | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
|  | **Actor Action** | | **System Response** | |
| 1 | User clicks on the assets from Dashboard. | | 2 | System loads the assets page. |
| **Alternative Flow** | | | | |
| 1 | User clicks on the assets from the side menu. | | 2-A | System loads the assets page. |

## Create Asset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Create Asset | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to add assets | | |
| **Pre-Conditions** | | User should be on Assets page | | |
| **Post-Conditions** | | Admin and User added assets successfully | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on Create Asset. | | 2 | Create New Asset page is displayed, showing asset detail, asset description, asset category, gallery and attachments needed to be filled. |
| 3 | User enters the details and press save. | | 4 | System checks the necessary fields needed to be filled and create that asset.  System prompts the success message: *Asset added successfully.* Redirects the user back to Assets page. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User miss important fields and press save. | | 4-A | System highlighted the required fields in red and prompts the error message:  *Please fill required fields!* |

## Select Asset Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Select Asset Category | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to select the asset category from dropdown while creating new asset | | |
| **Pre-Conditions** | | User should be on Create Asset page. | | |
| **Post-Conditions** | | User successfully selected an asset category | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
|  | **Actor Action** | | **System Response** | |
| 1 | User clicks on the dropdown menu on Asset Category section | | 2 | System pops the dropdown of already added asset categories. |
| 3 | User clicks on the desired asset category. | | 4 | System highlighted the selected category. |
| **Alternative Flow** | | | | |
| 1 | User unable to click on the dropdown menu on Asset Category section | | 2-A | System does not have any added category. |

## Add Asset Image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Add Asset Image | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to add an asset image while creating new asset | | |
| **Pre-Conditions** | | User should be on Create Asset page. | | |
| **Post-Conditions** | | User successfully added an asset image | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
|  | **Actor Action** | | **System Response** | |
| 1 | User selects the gallery tab on Create New Asset page. | | 2 | System displays the gallery tab with a big plus icon on it. |
| 3 | User clicks on the plus icon. | | 4 | System pops the open window showing files stored on your local machine. |
| 5 | User selects the image and clicks on open. | | 6 | System inserts the image on the gallery tab. |
| **Alternative Flow** | | | | |
| 5 | User selects file other than that of image and clicks on open. | | 6-A | System does not add other files. |

## Add an assignment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Add an assignment | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to add assignments of an asset | | |
| **Pre-Conditions** | | User should be on view Asset page | | |
| **Post-Conditions** | | Admin and User added an assignment successfully | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User selects the Assignment tab. | | 2 | System displays the assignment tab. |
| 3 | User clicks on the add button. | | 4 | System pops a small window showing dropdown list of available users. |
| 5 | User clicks on dropdown menu of the available users. | | 6 | System shows the list of the available users. |
| 7 | User selects the desired user. | | 8 | System displays his/her name on the dropdown. |
| 9 | User clicks on the assign button. | | 10 | System assigned the user and prompts the successful message: *Asset Assigned to user Successfully*. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 9 | User clicks on the assign button after selecting an already assigned user. | | 10-A | System does nothing and prompts the error message:  *Please assign another user!* |

## Delete an assignment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Delete an assignment | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to delete assignments of an asset | | |
| **Pre-Conditions** | | User should be on view Asset page | | |
| **Post-Conditions** | | Admin and User deleted an assignment successfully | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User selects the Assignment tab. | | 2 | System displays the assignment tab. |
| 3 | User clicks on the delete icon. | | 4 | System pops a small window showing warning whether you user want to delete it or not. |
| 5 | User selects I’m sure. | | 6 | System delete the assigned user and prompts the successful message: *User Un-assigned Successfully*. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 5 | User No instead of I’m sure. | | 6-A | System returns to the previous page. |

## View Usages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View Usages | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to view his/her asset usages | | |
| **Pre-Conditions** | | User should be logged in. | | |
| **Post-Conditions** | | User successfully viewed his/her assets usages | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
|  | **Actor Action** | | **System Response** | |
| 1 | User clicks on the usage from the side menu. | | 2 | System loads the usages page. |

## View filtered usages of an asset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View filtered usages of an asset | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to view filtered usages of an asset | | |
| **Pre-Conditions** | | User should be on Usages page | | |
| **Post-Conditions** | | Admin and User viewed filtered usages of his/her asset successfully | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the dropdown menu in usages page. | | 2 | System displays the list of all his/her assets. |
| 3 | User selects the desired asset. | | 4 | System selects that asset. |
| 5 | User selects start date and end date and then clicks on filter button. | | 6 | System shows all the usages of the selected asset from start date till end date. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 5 | User clicks on filter button without selecting start date and end date. | | 6-A | System displays all the usages of selected asset from the day it was created till now. |

## View Consumptions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View Consumptions | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to view his/her assets consumptions | | |
| **Pre-Conditions** | | User should be logged in. | | |
| **Post-Conditions** | | User successfully viewed his/her assets consumptions | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
|  | **Actor Action** | | **System Response** | |
| 1 | User clicks on the Consumption from the side menu. | | 2 | System loads the consumptions page. |

## View filtered consumptions of an asset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View filtered consumptions of an asset | | |
| **Actors** | | Admin, User | | |
| **Summary** | | User will be able to view filtered consumptions of an asset | | |
| **Pre-Conditions** | | User should be on Consumptions page | | |
| **Post-Conditions** | | Admin and User viewed filtered consumptions of his/her asset successfully | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the dropdown menu in consumptions page. | | 2 | System displays the list of all his/her assets. |
| 3 | User selects the desired asset. | | 4 | System selects that asset. |
| 5 | User selects start date and end date and then clicks on filter button. | | 6 | System shows all the consumptions of the selected asset from start date till end date. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 5 | User clicks on filter button without selecting start date and end date. | | 6-A | System displays all the consumptions of selected asset from the day it was created till now. |

## Edit Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Edit profile | | |
| **Actors** | | Admin/User | | |
| **Summary** | | User will be able to edit his profile details | | |
| **Pre-Conditions** | | User profile page is displayed | | |
| **Post-Conditions** | | User successfully edits and updates his profile details | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the edit button. | | 2 | Edit profile page is displayed, showing all of user’s editable fields. |
| 3 | User edits his desirable fields and clicks the update button. | | 4 | System successfully updates the user’s profile details. |
| **Alternative Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 3 | User enters invalid details or leaves it empty. | | 4-A | System displays an error message prompting the user to enter details in the correct format. |

## View User Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View user profile | | |
| **Actors** | | Admin/User | | |
| **Summary** | | User will be able to view his or her profile | | |
| **Pre-Conditions** | | User is logged in to the system | | |
| **Post-Conditions** | | User profile is displayed | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the profile icon. | | 2 | System shows a pop up with options to view profile or log out. |
| 3 | Users clicks on the profile option. | | 4 | System displays user profile page. |
| **Alternative Flow** | | | | |
| 1 | User clicks on the settings tab. | | 2-A | Settings page is displayed with four options; *Profile, Mange Templates, User Management, User Preferences* |

## View Assets Assignment History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View Assets Assignment History | | |
| **Actors** | | Admin | | |
| **Summary** | | User will be able to see all the assets and their total assignments | | |
| **Pre-Conditions** | | User profile page is displayed | | |
| **Post-Conditions** | | Assets assignment history page is displayed | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | User clicks on the asset assignment history tab. | | 2 | Asset assignment history page is displayed to the user. |
| **No Alternative Flow** | | | | |

## Change Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Change Password | | |
| **Actors** | | Admin/User | | |
| **Summary** | | User will be able to change his password | | |
| **Pre-Conditions** | | User profile page is displayed | | |
| **Post-Conditions** | | Password is successfully updated | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
|  | **Actor Action** | | **System Response** | |
| 1 | User clicks on the change your password tab. | | 2 | System displays the change password page and asks for old and new password. |
| 3 | Enter valid credentials and update. | | 4 | System verifies credentials, updates it and shows the “password updated successfully” popup message. |
| **No Alternative Flow** | | | | |

## View User Management Page

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View User Management page | | |
| **Actors** | | Admin | | |
| **Summary** | | User will be able to view his user management page | | |
| **Pre-Conditions** | | User is logged in to the system | | |
| **Post-Conditions** | | User management page is displayed | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | click on the settings tab. | | 2 | Settings page is displayed with four options; *Profile, Mange Templates, User Management, User Preferences.* |
| 3 | Click on the user management option. | | 4 | System displays user management page. |
| **No Alternative Flow** | | | | |

## Add a New Role

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | Add a New Role | | | | |
| **Actors** | | | Admin | | | | |
| **Summary** | | | Admin will be able to add a new role | | | | |
| **Pre-Conditions** | | | User Management page is displayed | | | | |
| **Post-Conditions** | | | New role is added successfully | | | | |
| **Special Requirements** | | | None | | | | |
| **Basic Flow** | | | | | | | |
| **Actor Action** | | | | | **System Response** | | |
| 1 | Click on the manage roles and permissions tab. | | | | 2 | | System display *manage roles and permissions* page including icons to add, delete and update role details. |
| 3 | Click on the add icon. | | | | 4 | | System asks for *Role Name* and *Role Description.* |
| 5 | Enter valid details and click create. | | | | 6 | | System successfully creates a new role. |
| **Alternative Flow** | | | | | | | |
| **Actor Action** | | | | **System Response** | | | |
| 5 | | User leaves a field empty. | | 4-A | | System displays an error message and prompts the user to fill up the field. | |
| 5 | | User clicks cancel | | 4-B | | System goes back to the previous step. | |

## Delete a Role

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | Delete a Role | | | | |
| **Actors** | | | Admin | | | | |
| **Summary** | | | Admin will be able to delete a role | | | | |
| **Pre-Conditions** | | | User Management page is displayed | | | | |
| **Post-Conditions** | | | The role is deleted successfully | | | | |
| **Special Requirements** | | | None | | | | |
| **Basic Flow** | | | | | | | |
| **Actor Action** | | | | | **System Response** | | |
| 1 | Click on the manage roles and permissions tab. | | | | 2 | | System displays *manage roles and permissions* page including icons to add, delete and update role details. |
| 3 | Select a role and click on the delete icon. | | | | 4 | | System asks for a confirmation to delete the role. |
| 5 | Click ‘yes’. | | | | 6 | | System successfully deletes the role. |
| **Alternative Flow** | | | | | | | |
| **Actor Action** | | | | **System Response** | | | |
| 3 | | User did not select a role. | | 4-A | | System displays an error message and prompts the user to select a role first. | |
| 5 | | User clicks ‘no’ | | 6-A | | System goes back to the previous step. | |

## Update a Role

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | Update a role | | | | |
| **Actors** | | | Admin | | | | |
| **Summary** | | | Admin will be able to update a role’s details | | | | |
| **Pre-Conditions** | | | User Management page is displayed | | | | |
| **Post-Conditions** | | | The role is details are successfully updated | | | | |
| **Special Requirements** | | | None | | | | |
| **Basic Flow** | | | | | | | |
| **Actor Action** | | | | | **System Response** | | |
| 1 | Click on the manage roles and permissions tab. | | | | 2 | | System displays *manage roles and permissions* page including icons to add, delete and update role details. |
| 3 | Select a role and click on the update icon. | | | | 4 | | System shows *Role Name* and *Role Description* fields to update. |
| 5 | Edit desirable fields and click ‘update’ | | | | 6 | | System successfully updates the role. |
| **Alternative Flow** | | | | | | | |
| **Actor Action** | | | | **System Response** | | | |
| 3 | | User did not select a role. | | 4-A | | System displays an error message and prompts the user to select a role first. | |
| 5 | | User clicks ‘cancel’ | | 6-A | | System goes back to the previous step. | |

## View All User groups

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View All User Groups | | |
| **Actors** | | Admin | | |
| **Summary** | | User will be able to view all the user groups | | |
| **Pre-Conditions** | | User Management page is displayed | | |
| **Post-Conditions** | | User groups page is displayed | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Click on the user groups tab. | | 2 | System displays all the user groups and their basic details such as Group Names, Group Manager, Total Members and Created by. |
| **No Alternative Flow** | | | | |

## View a User group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View a User Group | | |
| **Actors** | | Admin | | |
| **Summary** | | User will be able to view a single user group | | |
| **Pre-Conditions** | | User Groups page is displayed | | |
| **Post-Conditions** | | The selected user group page and its details are displayed | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Click the view icon of the desired user group. | | 2 | System displays group details such as the names and profile pictures of all the group members and the manager, with options to directly remove members or to edit group users. |
| **No Alternative Flow** | | | | |

## Remove a Member from a User Group

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | | Remove a Member from a User Group | | | | |
| **Actors** | | | Admin | | | | |
| **Summary** | | | Admin will be able to remove a member from a user group | | | | |
| **Pre-Conditions** | | | The selected user group page and its details are displayed | | | | |
| **Post-Conditions** | | | Successfully removed a member from the user group | | | | |
| **Special Requirements** | | | None | | | | |
| **Basic Flow** | | | | | | | |
| **Actor Action** | | | | | **System Response** | | |
| 1 | Click on the edit group users tab. | | | | 2 | | System displays both the members that are in the group (chosen), which can be removed, or are available to be added in the group (choices). |
| 3 | Select the members that are to be removed from the chosen column and click on the remove icon. | | | | 4 | | System moves the group members from the chosen column to the choices column. |
| 5 | Click save changes. | | | | 6 | | System removes the group members and shows the feedback pop-up message that *user group successfully updated.* |
| **Alternative Flow** | | | | | | | |
| **Actor Action** | | | | **System Response** | | | |
| 1-A | | Click delete icon from the profile information of the desired member to be removed from the group. | | 6 | | System removes the group members and shows the feedback pop-up message that *user group successfully updated.* | |

## Add a Member to a User Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Add a Member to a User Group | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to add a member to a user group | | |
| **Pre-Conditions** | | The selected user group page and its details are displayed | | |
| **Post-Conditions** | | Successfully added a member to the user group | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Click on the edit group users tab. | | 2 | System displays both the members that are in the group (chosen), which can be removed, or are available to be added in the group (choices). |
| 3 | Select the members that are to be added to the group and click on the add icon. | | 4 | System moves the group members from the choices column to the choices column. |
| 5 | Click save changes. | | 6 | System adds the group members and shows the feedback pop-up message that *user group successfully updated.* |
| **No Alternative Flow** | | | | |

## Create a User Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Create a User Group | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to create a user group | | |
| **Pre-Conditions** | | User groups page is displayed | | |
| **Post-Conditions** | | Successfully created a user group | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Click on ‘create user group’. | | 2 | System opens the create user group page to enter the group name and select the group members. |
| 3 | Enter the group name, select group members and click on the move button | | 4 | System moves the group members from the choices column to the chosen column. |
| 5 | Click on the manager icon to give manager rights to one of the members and click create user group. | | 6 | System successfully creates the user group. |
| **No Alternative Flow** | | | | |

## Create a User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Create a User | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin will be able to create a user | | |
| **Pre-Conditions** | | User management page is displayed | | |
| **Post-Conditions** | | Successfully created a new user | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Click on ‘create user’ button. | | 2 | System opens the create user page for the user to enter his personal information details. |
| 3 | Enter the personal information details and click the drop-down button of ‘select role’. | | 4 | System shows available roles to choose from. |
| 5 | Select a role and then click the create button. | | 6 | System successfully creates a new user. |
| **No Alternative Flow** | | | | |

## View a User’s Personal Information Page

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | View a User’s Personal Information Page | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin/User will be able to View a user’s personal information | | |
| **Pre-Conditions** | | User management page is displayed | | |
| **Post-Conditions** | | User personal information page is opened | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Click on the view icon of a user in the user management page. | | 2 | System opens the personal information page of the user |
| **No Alternative Flow** | | | | |

## Edit User’s Personal Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | Edit User’s Personal Information Page | | |
| **Actors** | | Admin | | |
| **Summary** | | Admin/User will be able to edit a user’s details | | |
| **Pre-Conditions** | | User personal information page is displayed | | |
| **Post-Conditions** | | Successfully updated user details | | |
| **Special Requirements** | | None | | |
| **Basic Flow** | | | | |
| **Actor Action** | | | **System Response** | |
| 1 | Enter the new details in the desired field and click the update button. | | 2 | System successfully updates the user details. |
| **No Alternative Flow** | | | | |

# Graphical User Interfaces

The following screens are for the admin end:



Figure 3: Login

This GUI shows LOGIN page for an admin

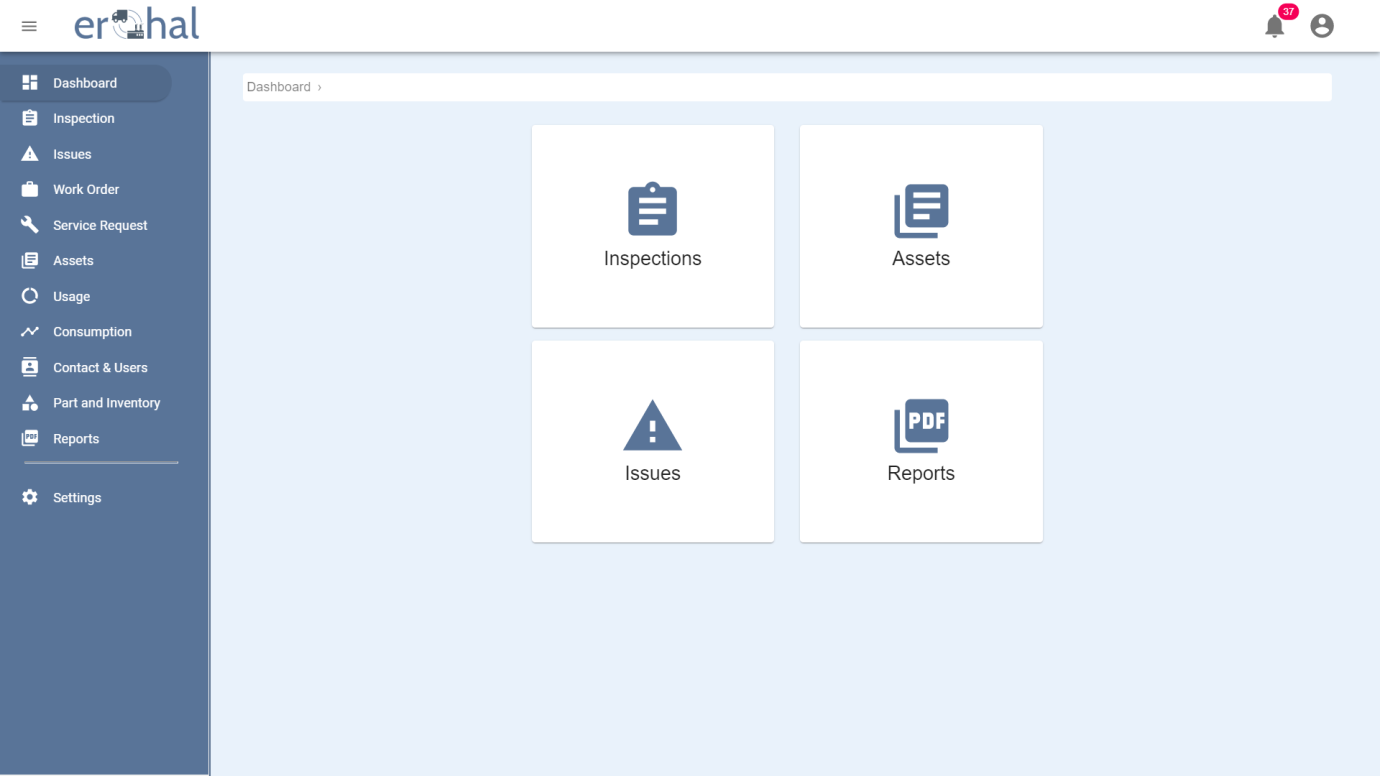


Figure 4: Dashboard

This GUI shows the dashboard of the application.

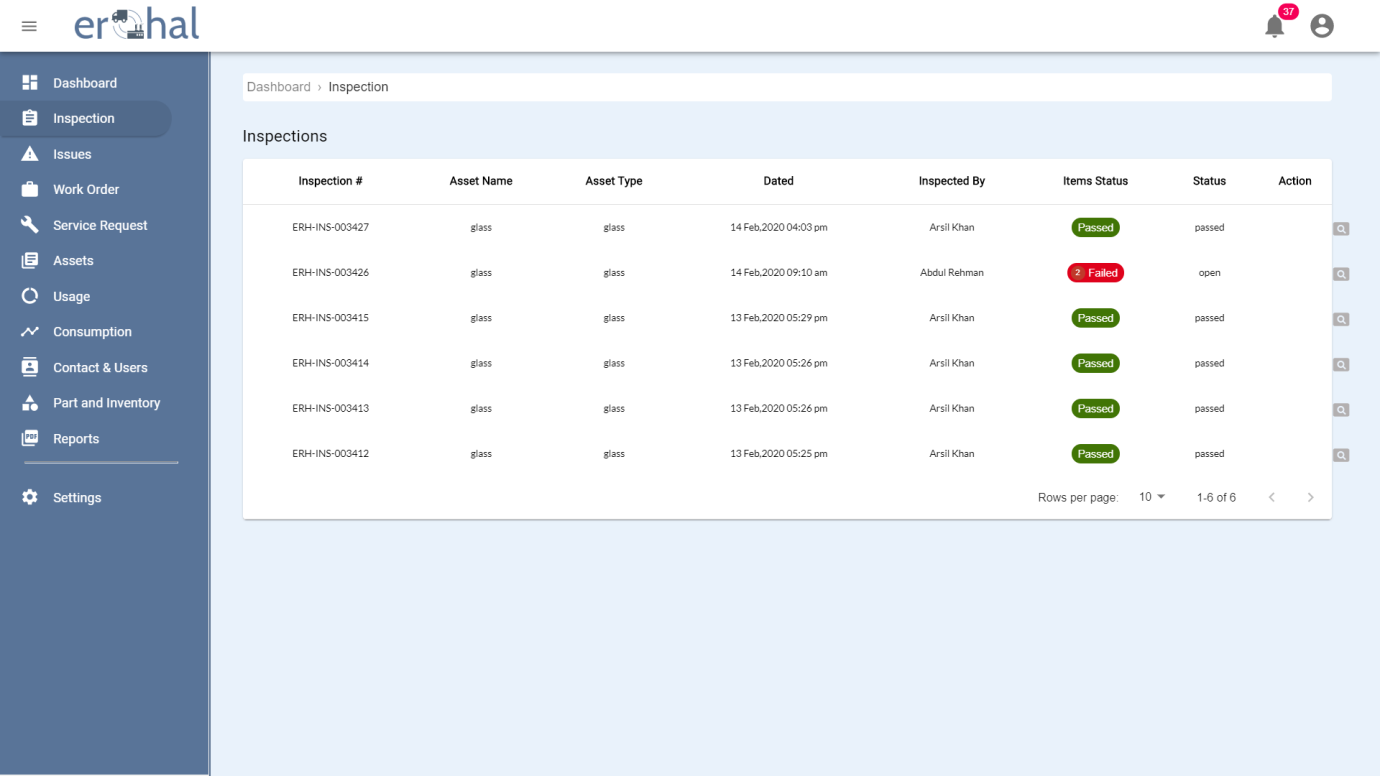


Figure 5: Inspections List

This GUI shows list of all inspections done by users.

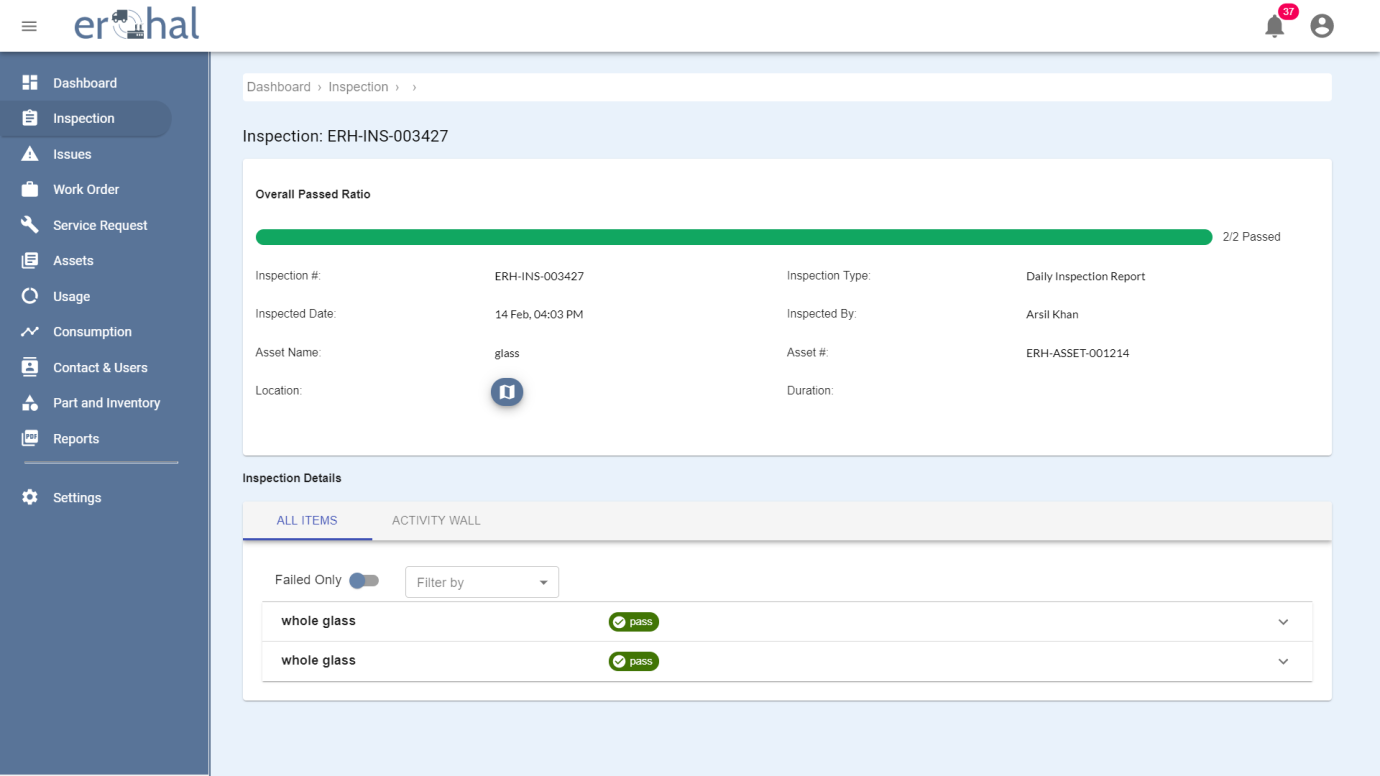


Figure 6: Inspection Details

This GUI shows the details of an inspection.

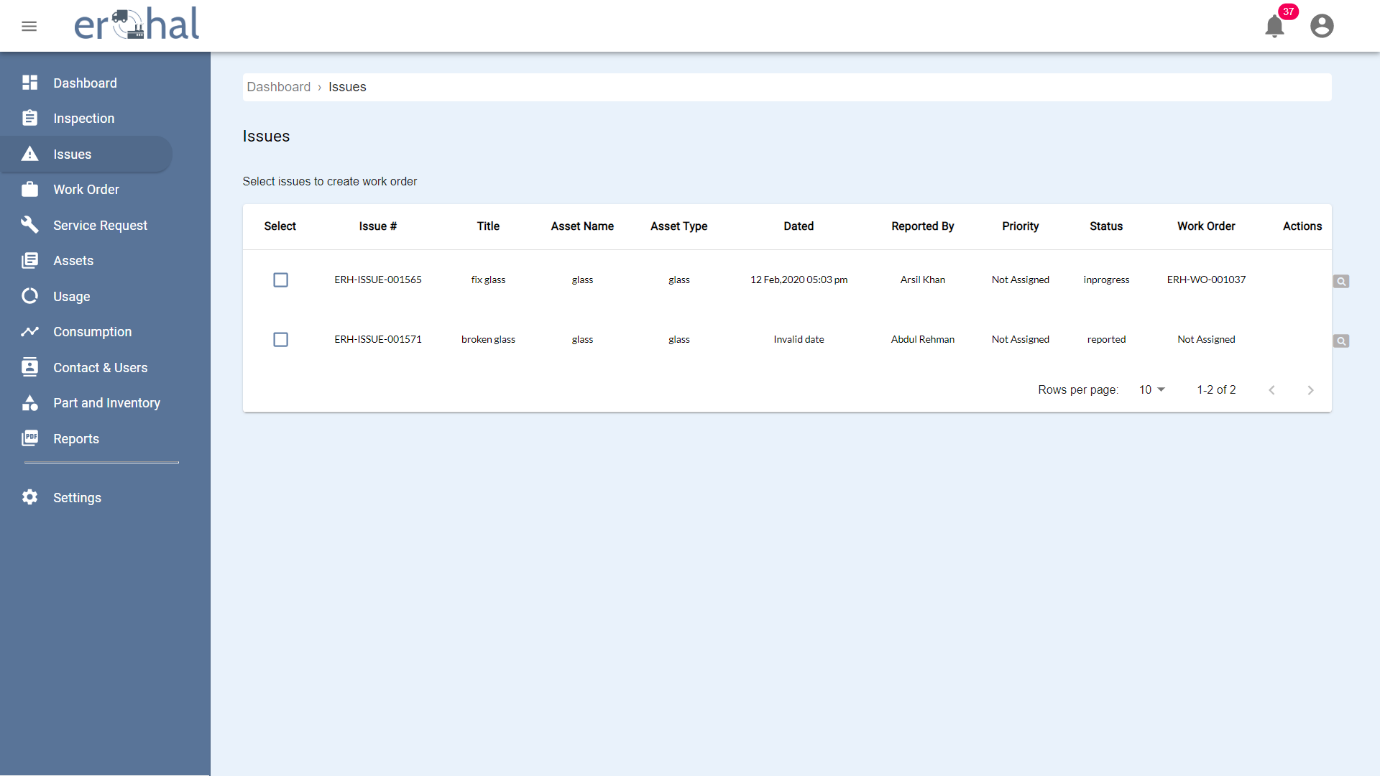


Figure 7: Issues List

This GUI shows the list of all the issues appeared in assets.

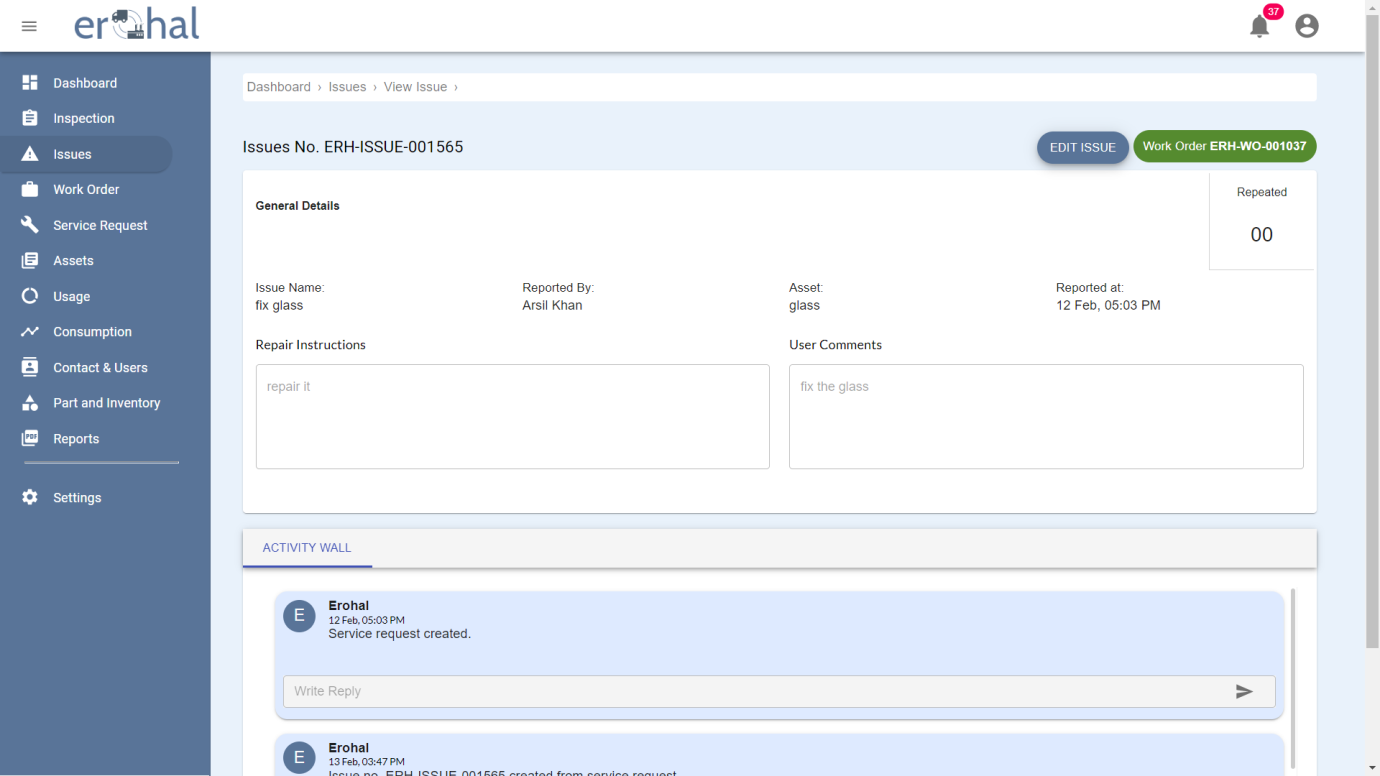


Figure 8: Issue Details 1

This GUI shows the details of an issue and allows to edit that issue.

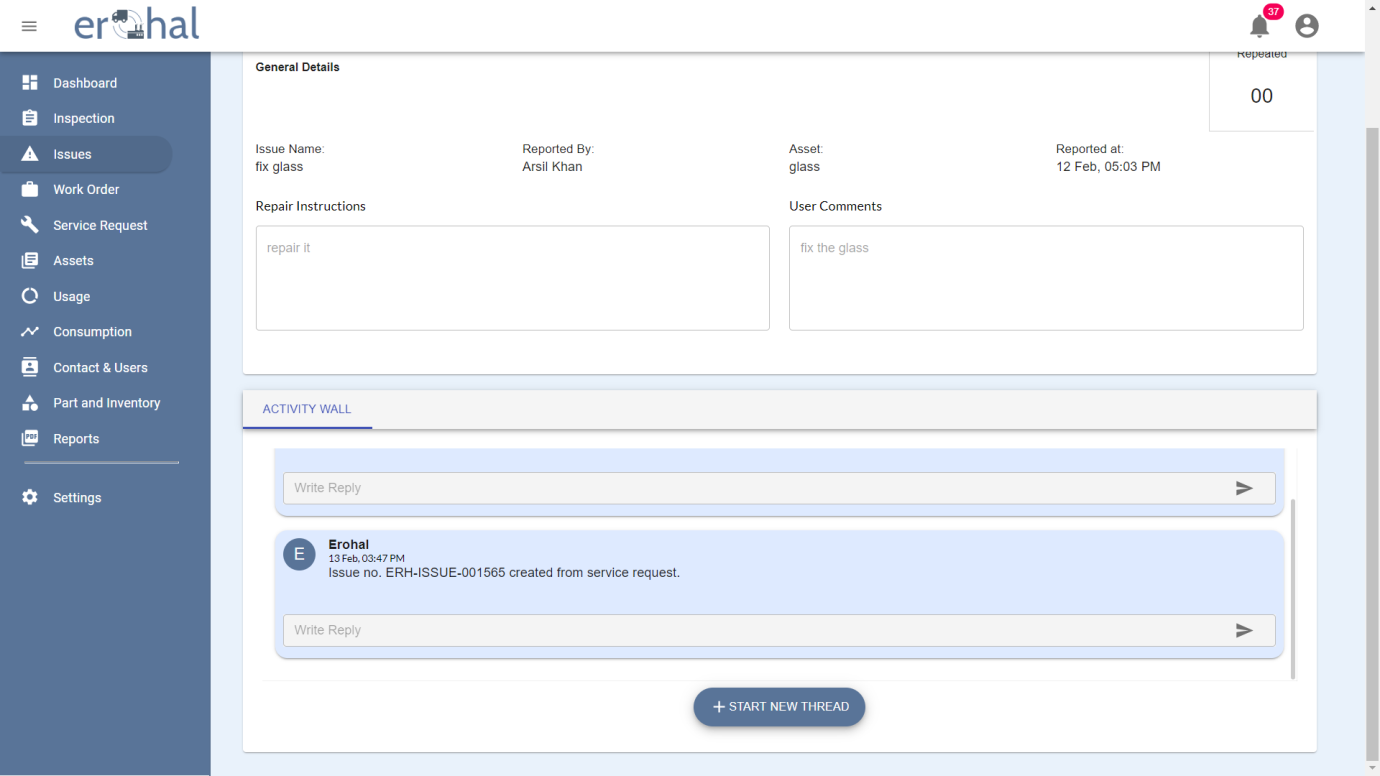


Figure 9: Issue Details 2

This GUI is the remaining part of issue details which shows the activity wall for that issue.

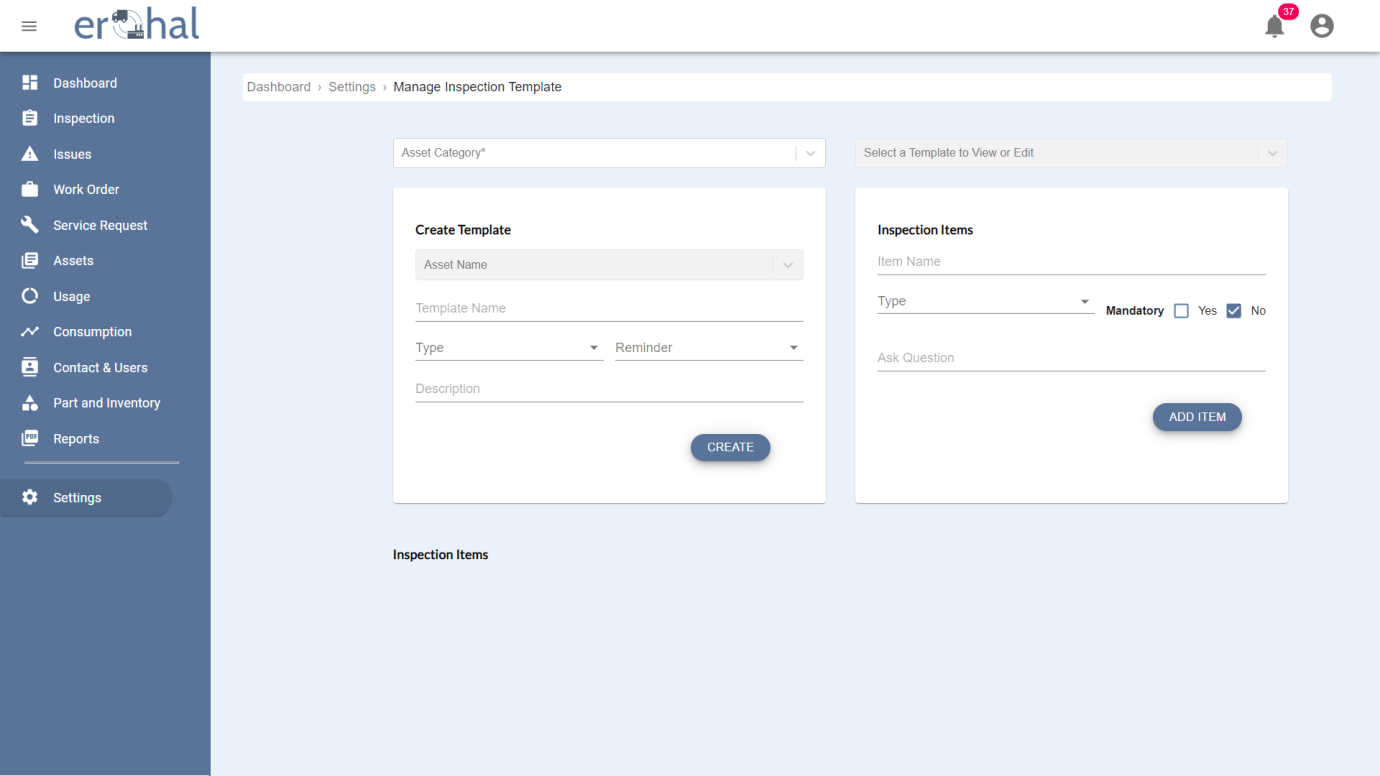


Figure 10: Inspection Template Creation

This GUI shows the view of creation of an inspection template and add an item in that template.

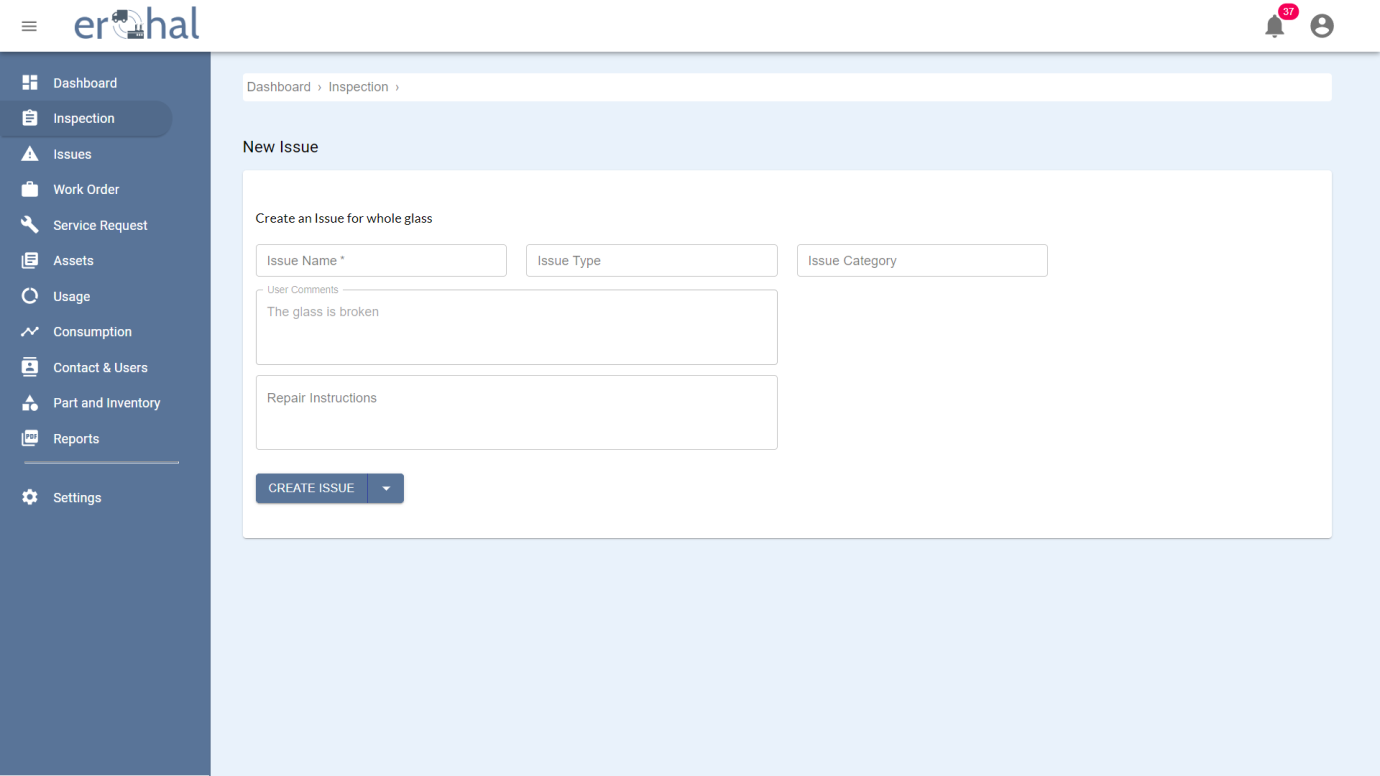


Figure 11: Issue Creation

This GUI shows the form used to create an issue.

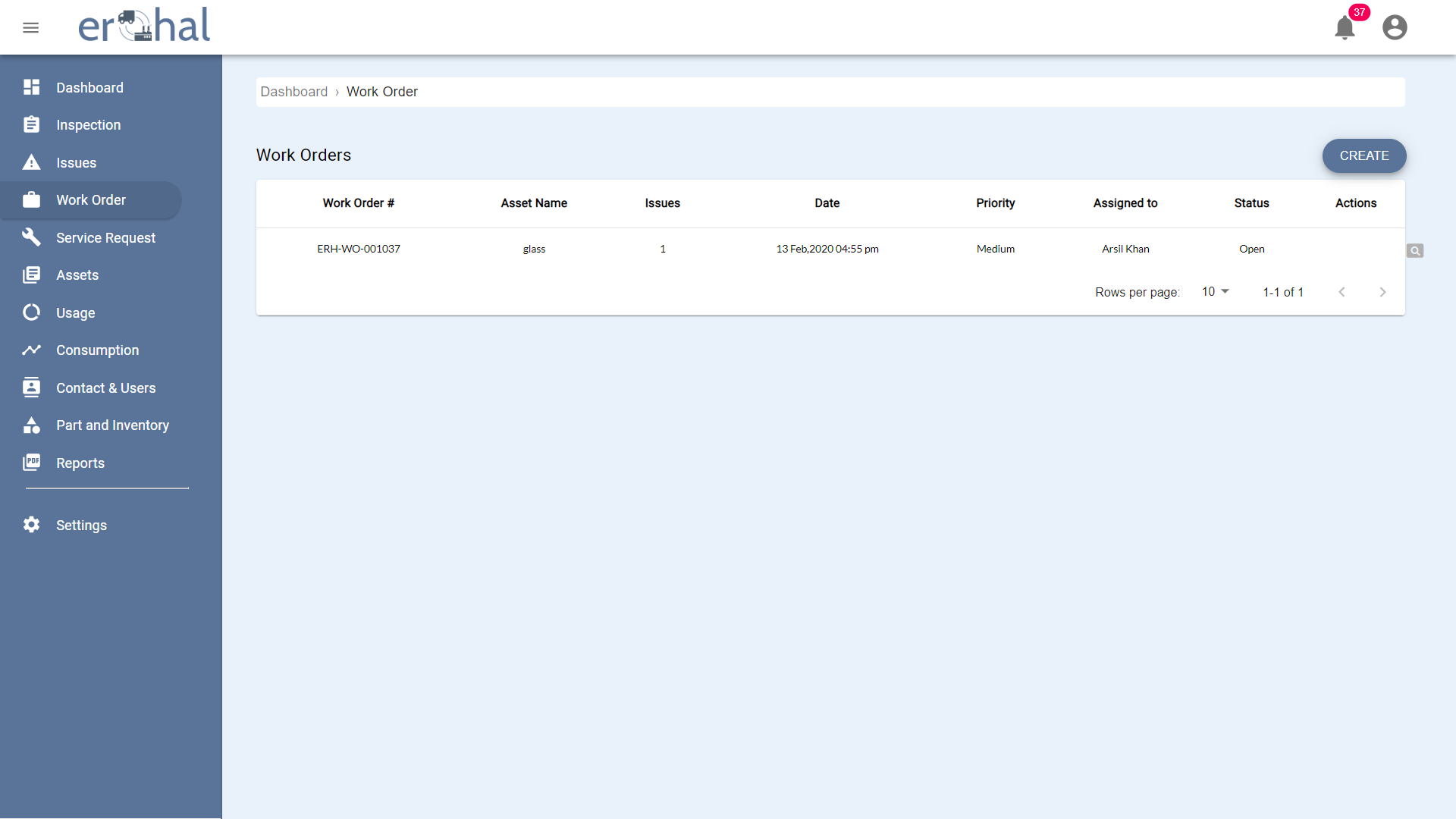


Figure 12: Work Orders

This GUI shows the list of all service requests.

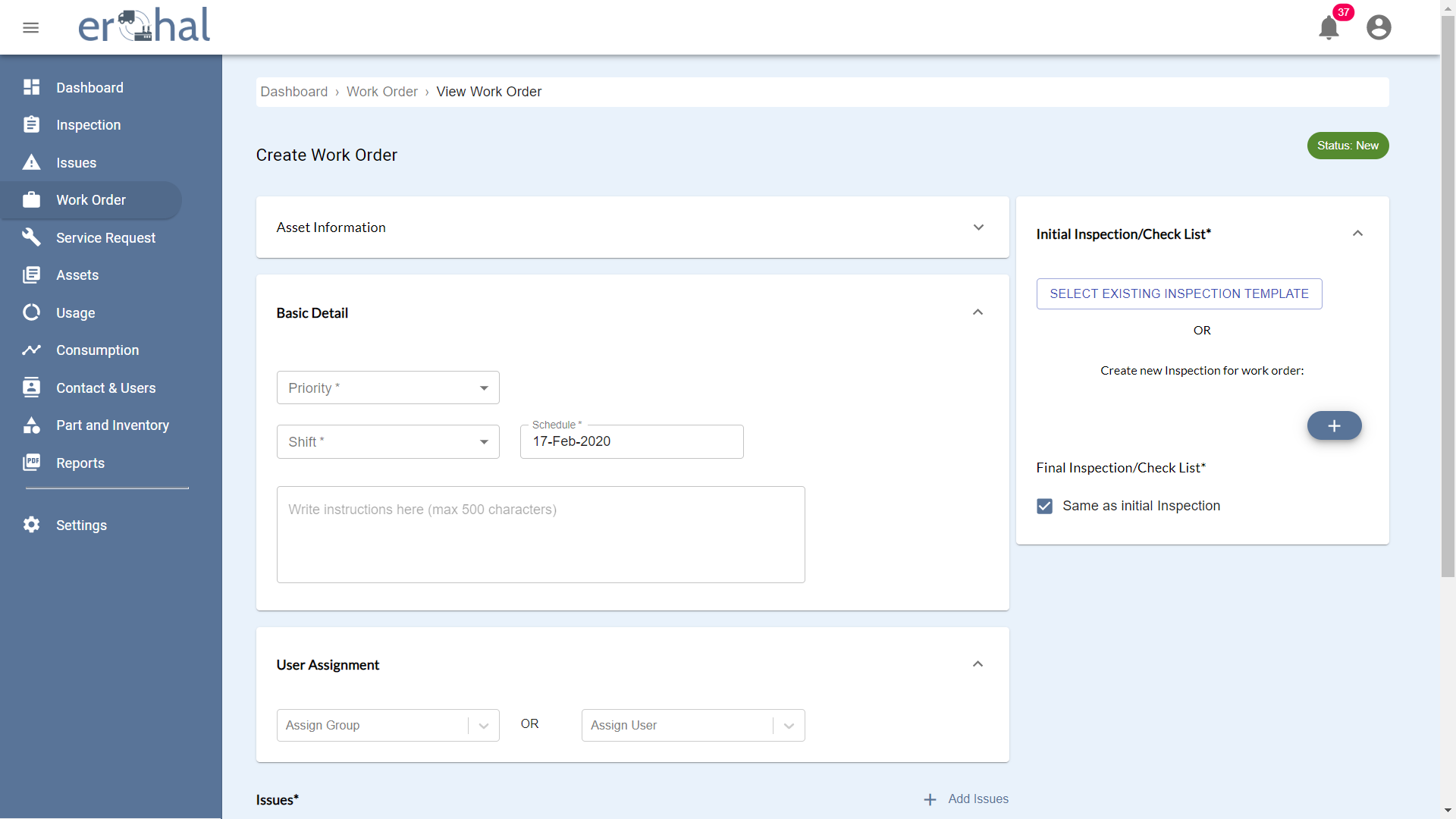


Figure 13: Creating Work Order 1

This GUI shows the requirements of creating work order.

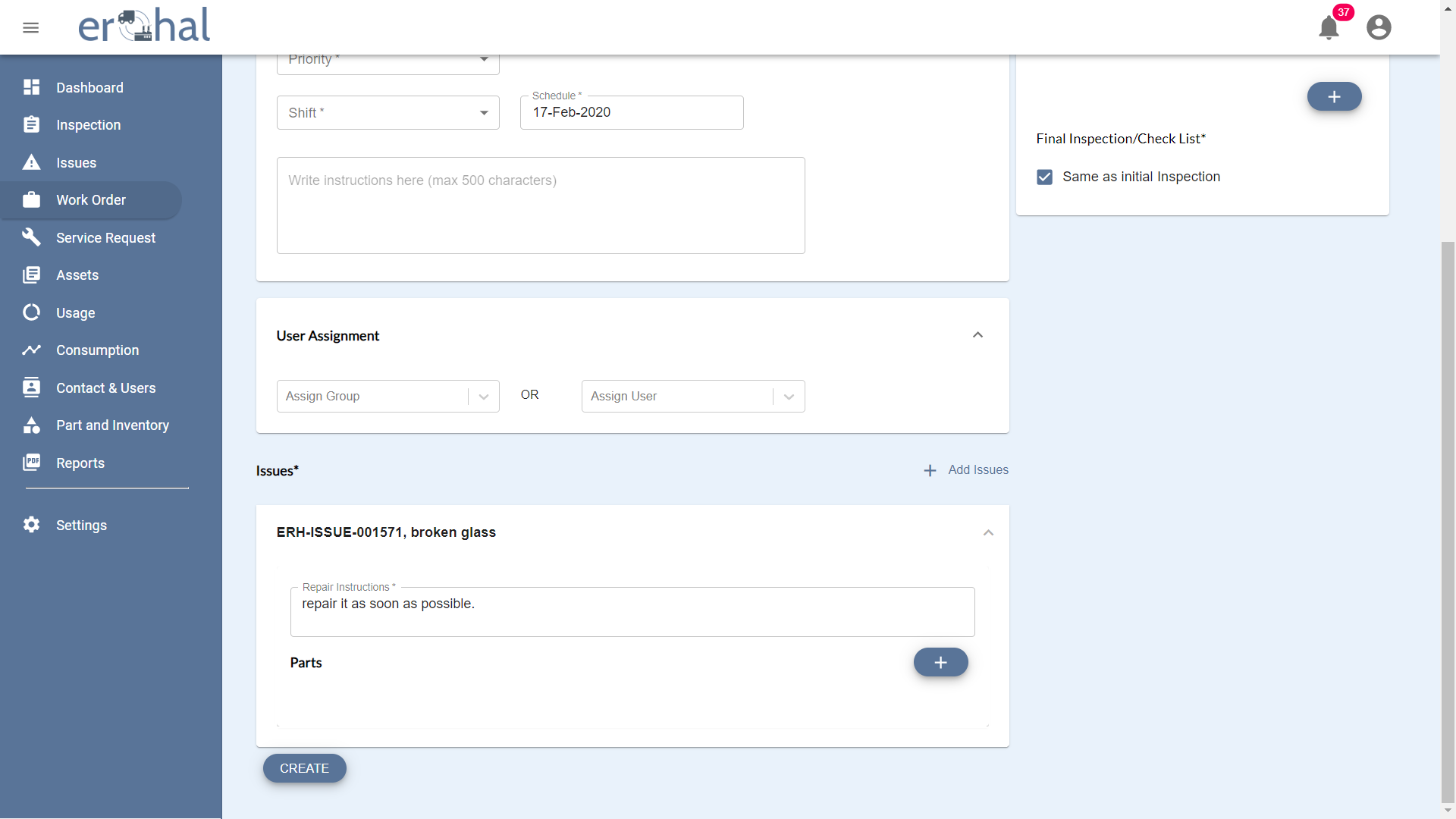


Figure 14: Creating Work Order 2

This GUI shows the requirements of creating work order.

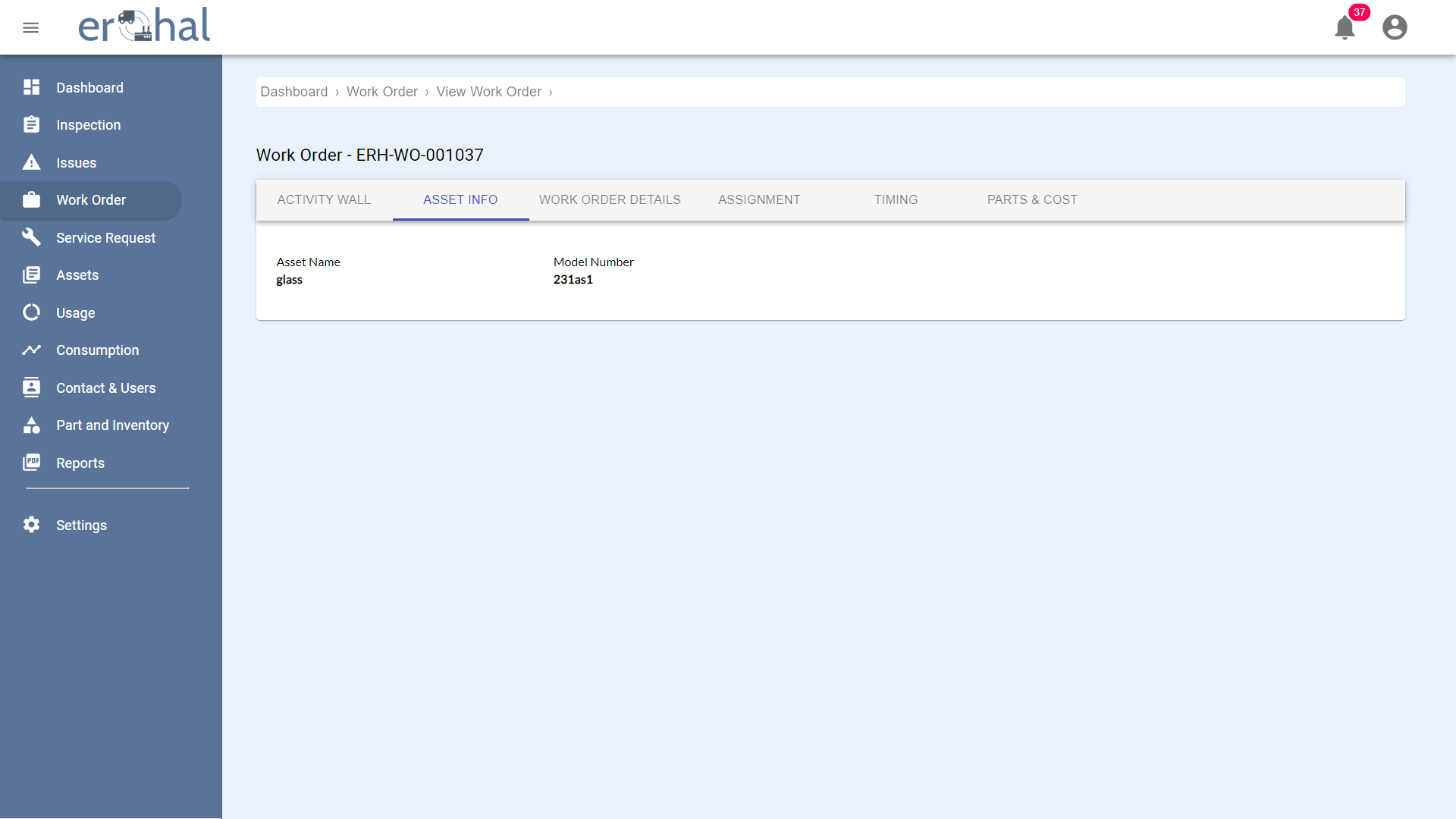


Figure 15: View Work Orders

This GUI shows the asset info of work orders.

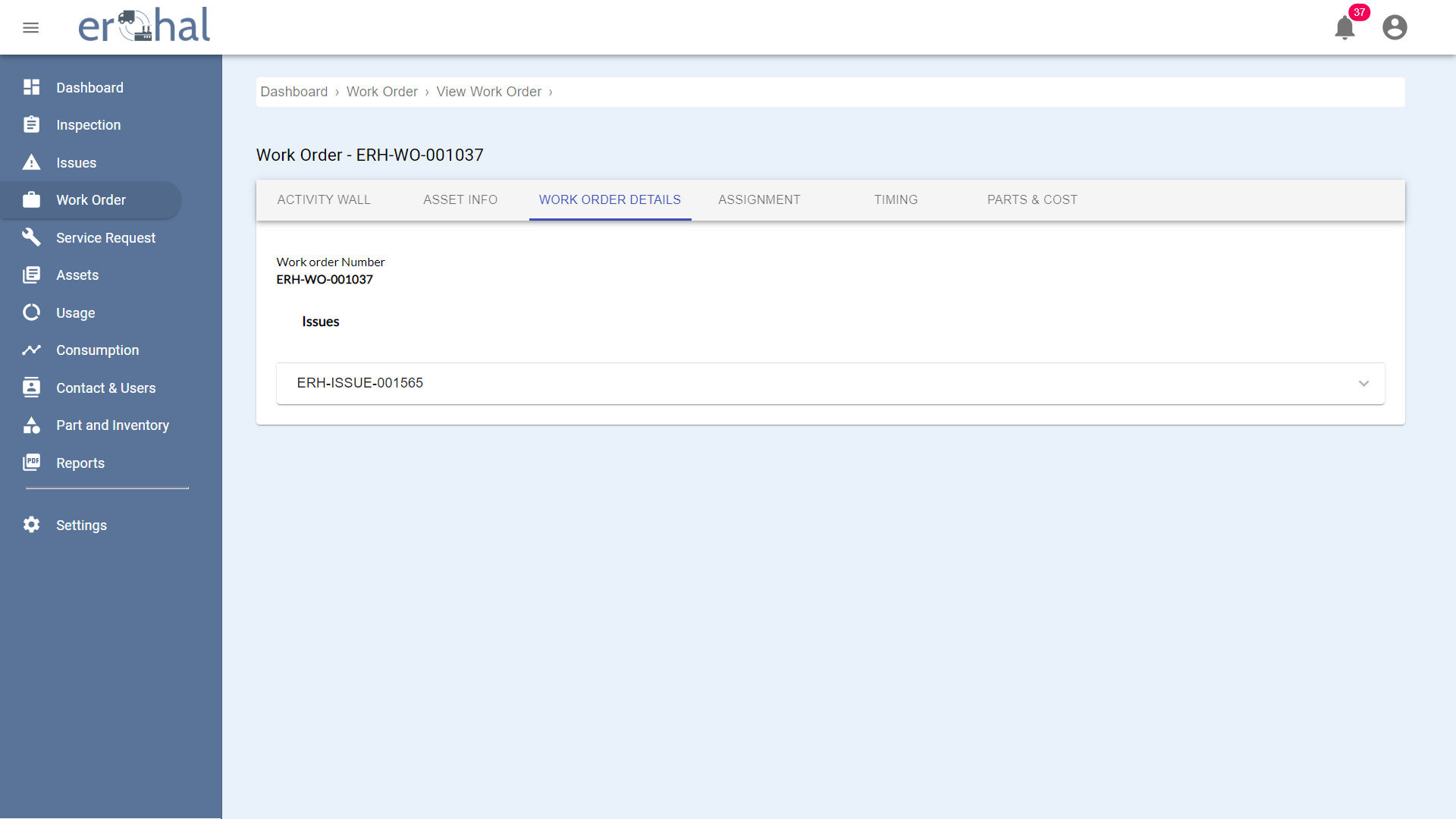


Figure 16: Work Order Details

This GUI shows work order details of work order.

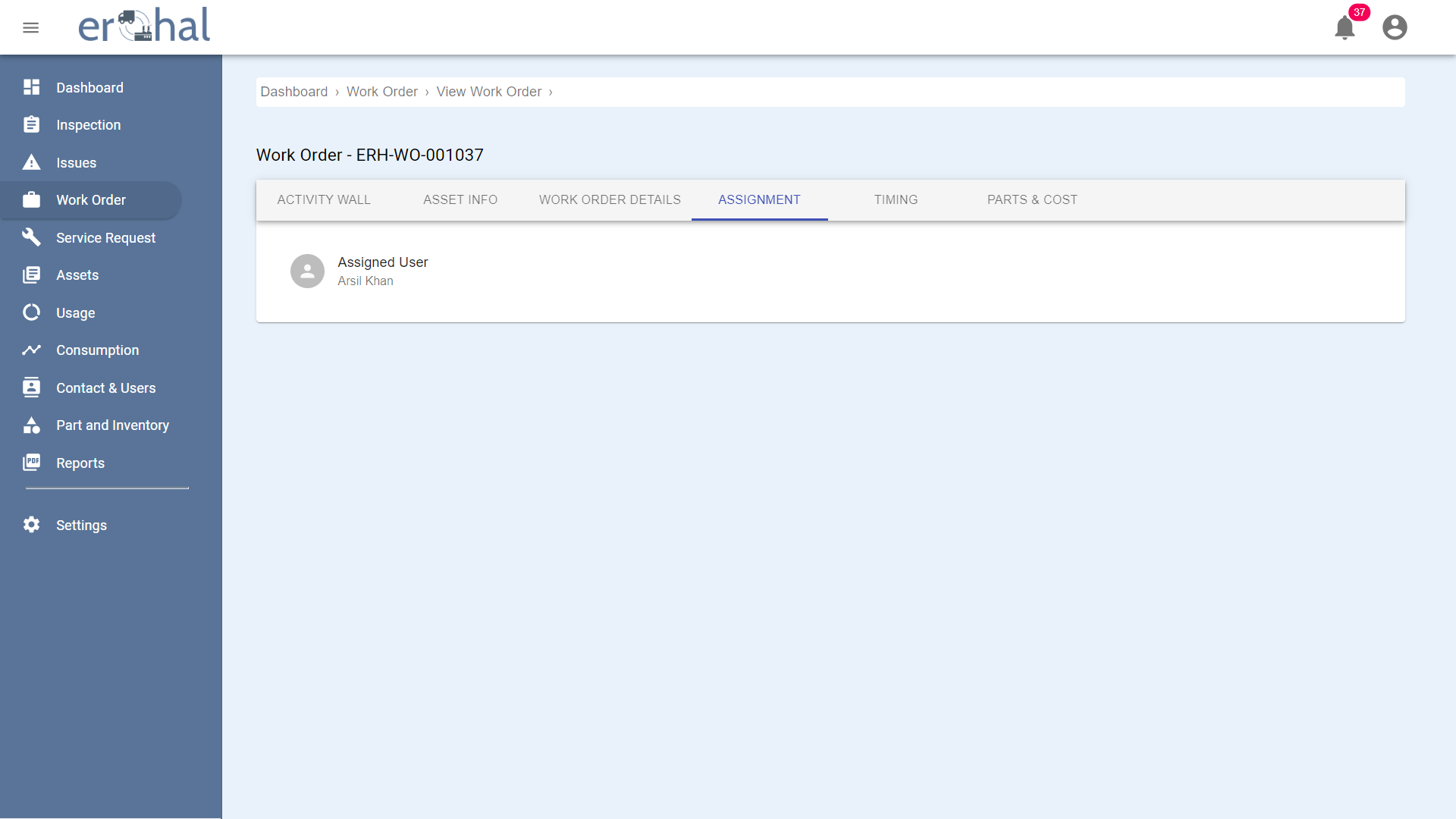


Figure 17: Assigning Work Orders

This GUI shows assigned user of a work order.



Figure 18: Work Orders Timing

This GUI shows the timing of work order.

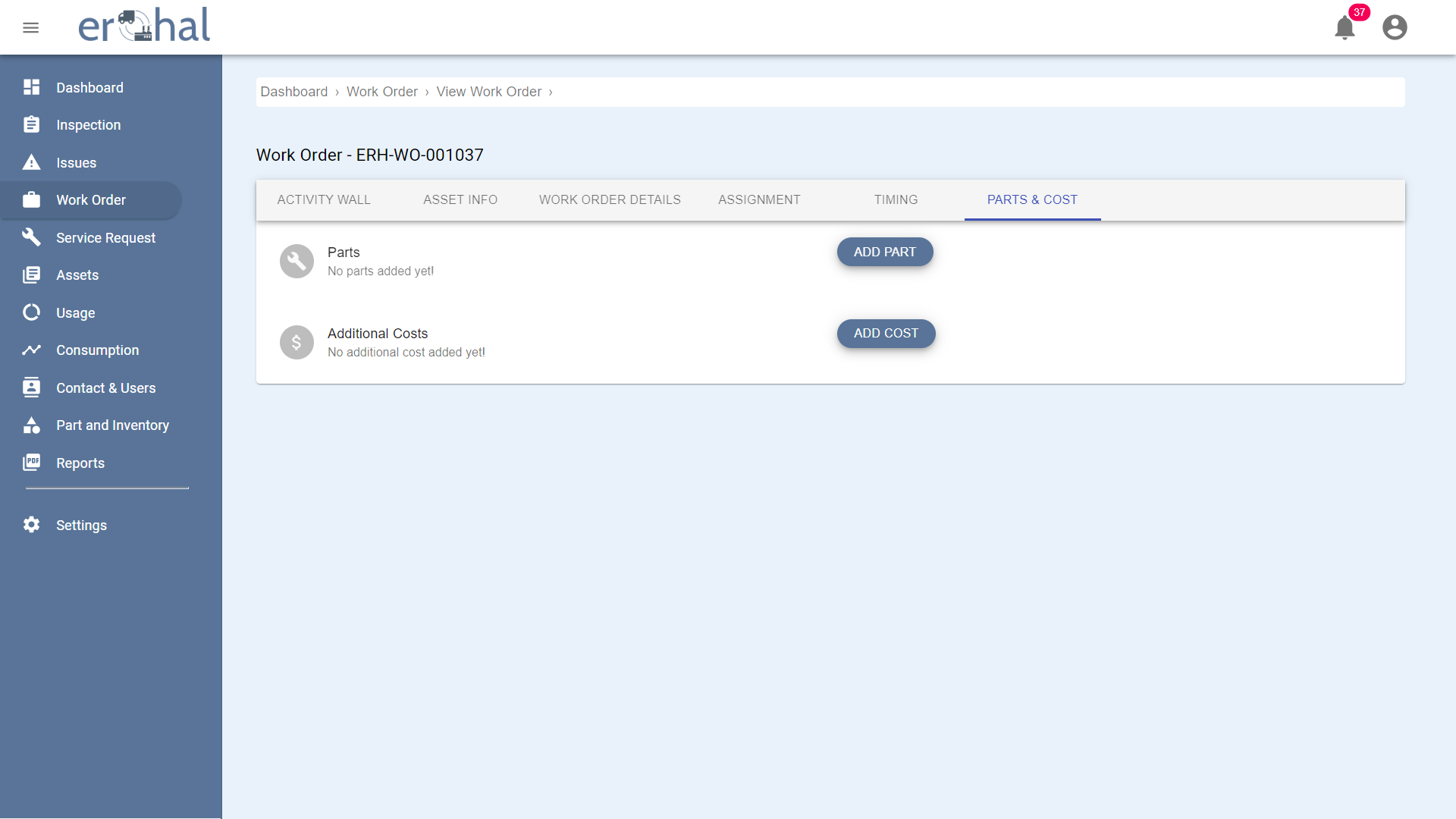


Figure 19: Parts and Cost

This GUI shows the parts and their costs of work order.

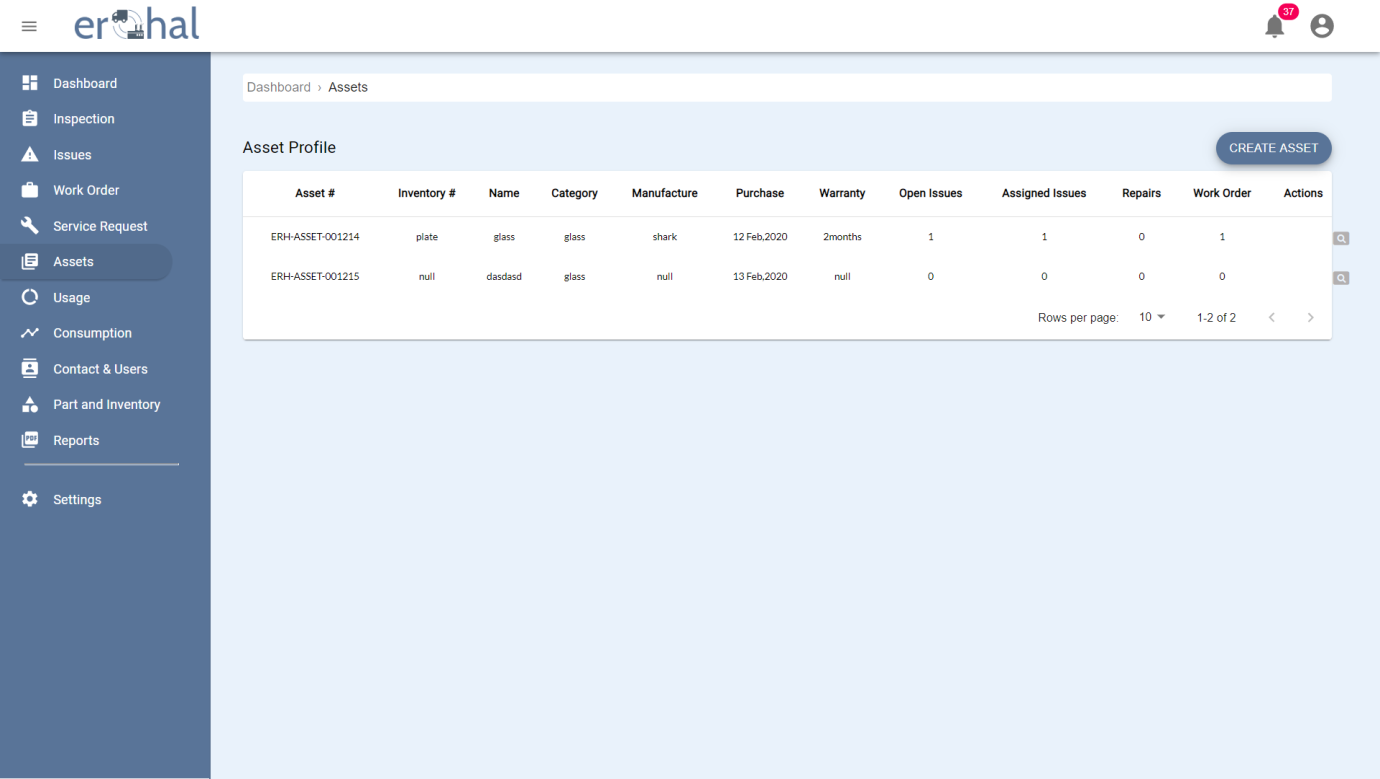


Figure 20: Assets

This GUI shows list of all assets of user which can be further viewed in detail. User can also create an asset from this page.

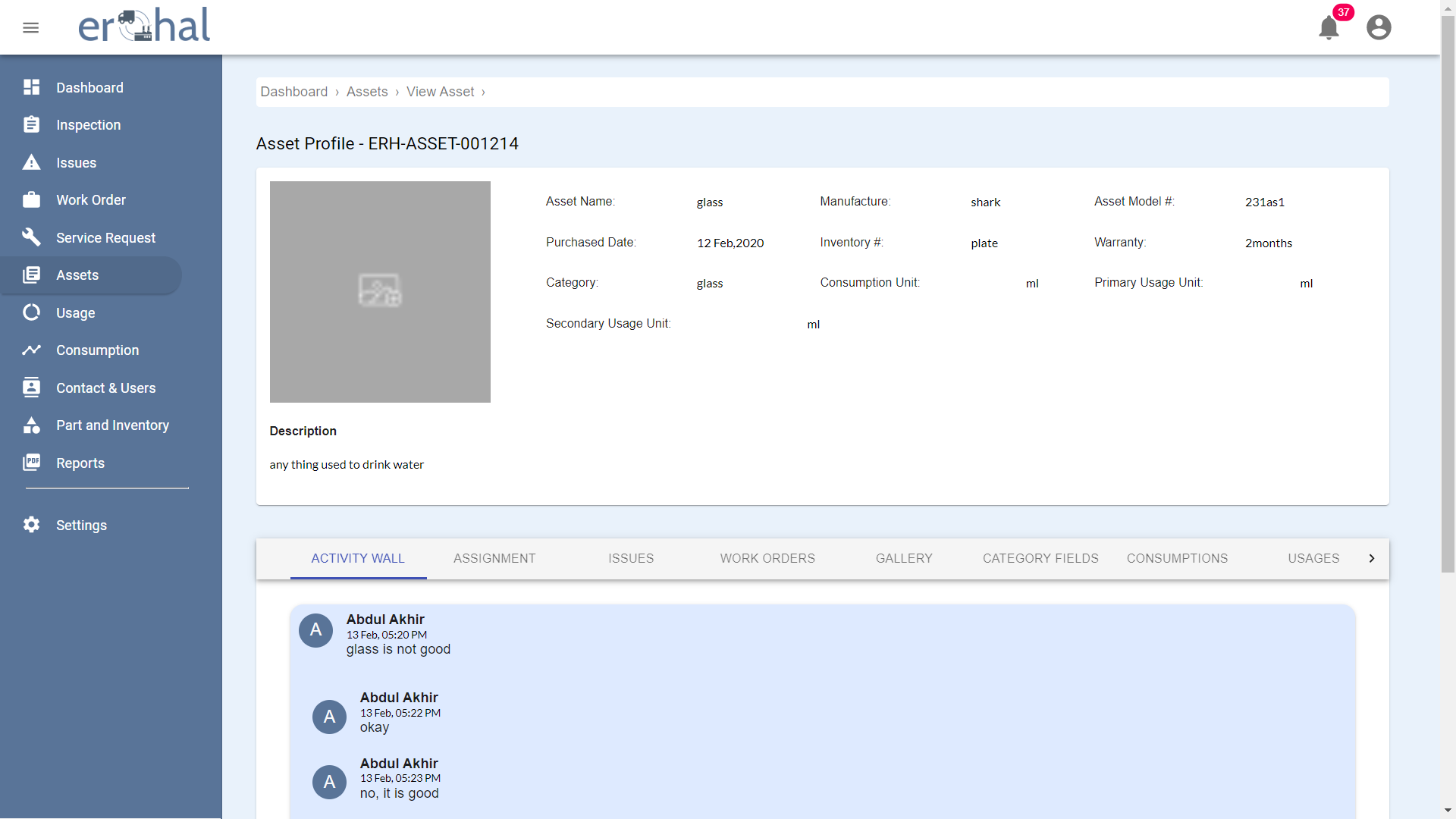


Figure 21: View Asset 1

This GUI shows details of an asset.

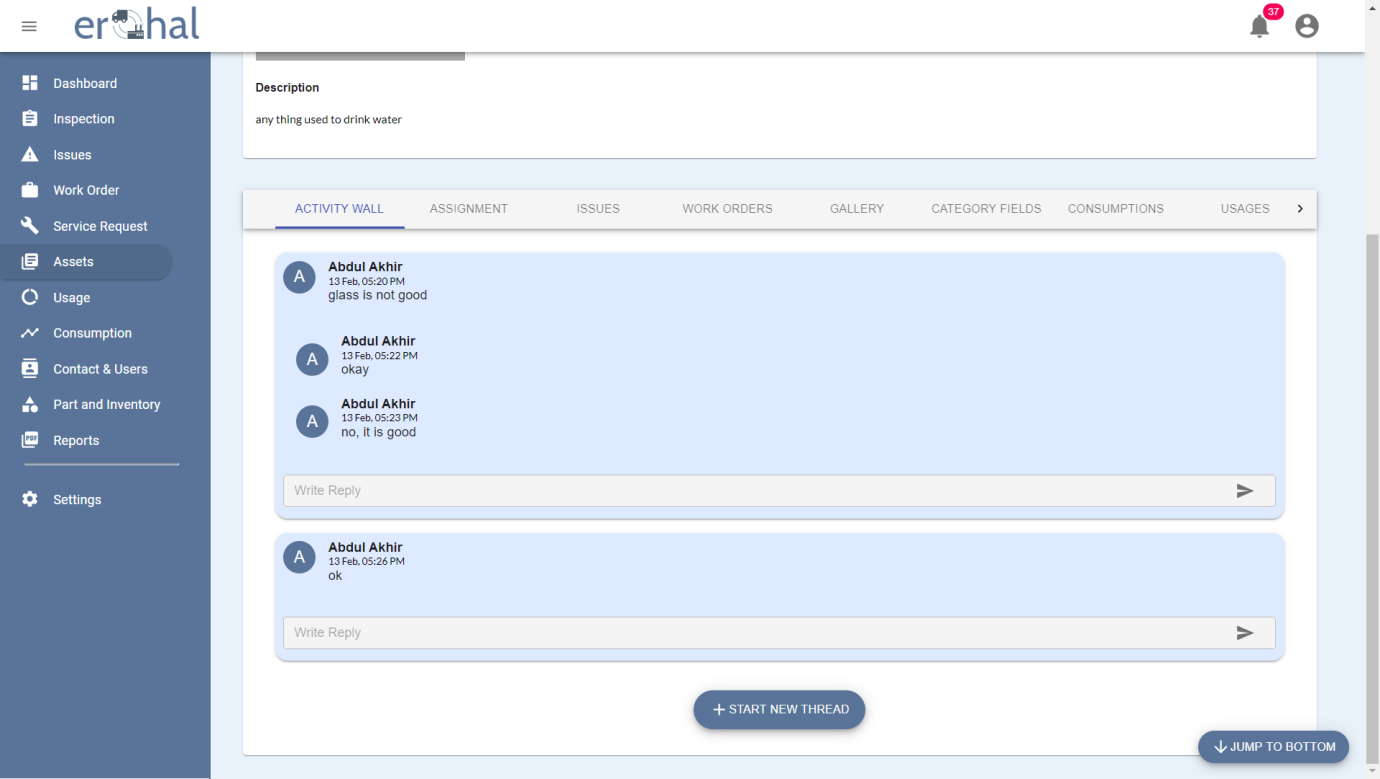


Figure 22: View Asset 2

This GUI shows the message threads of an asset.

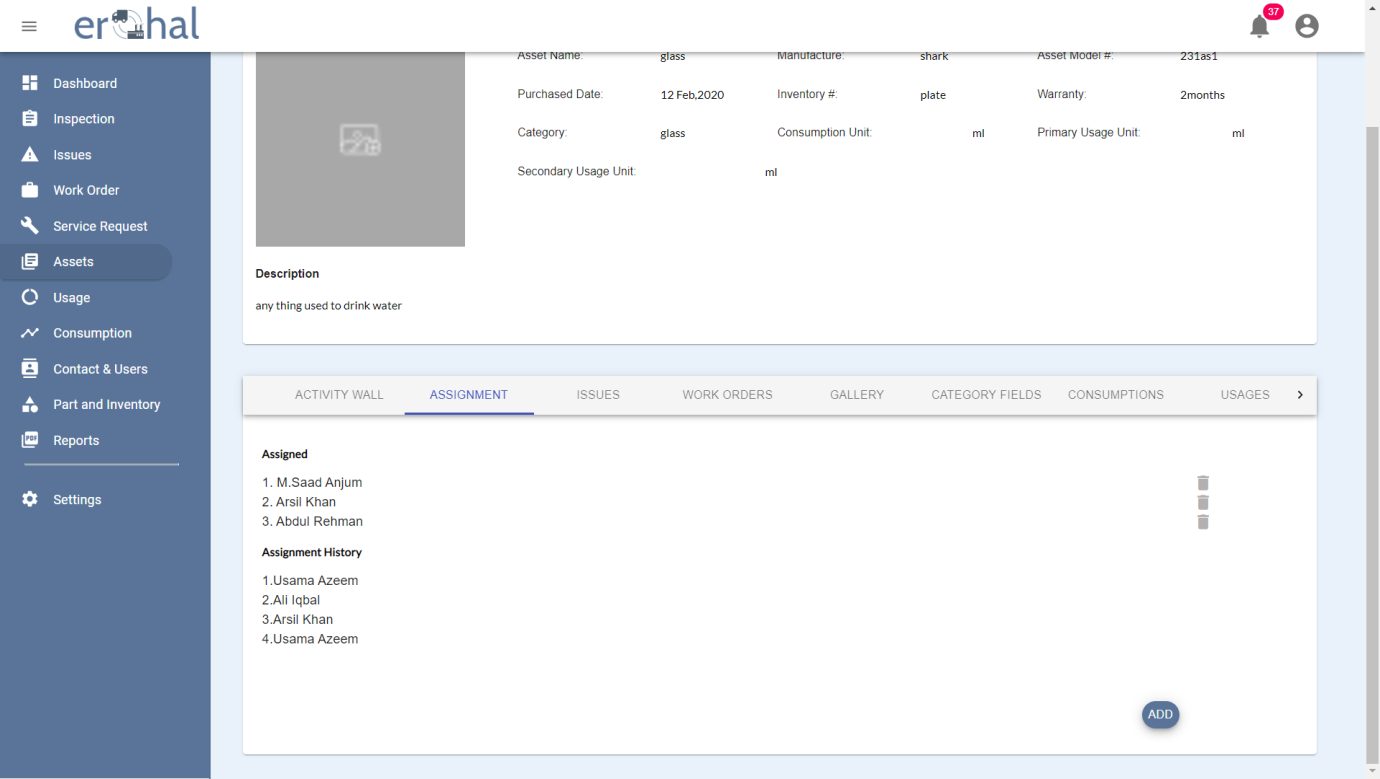


Figure 23: View Asset 3

This GUI shows all assignments of an asset.

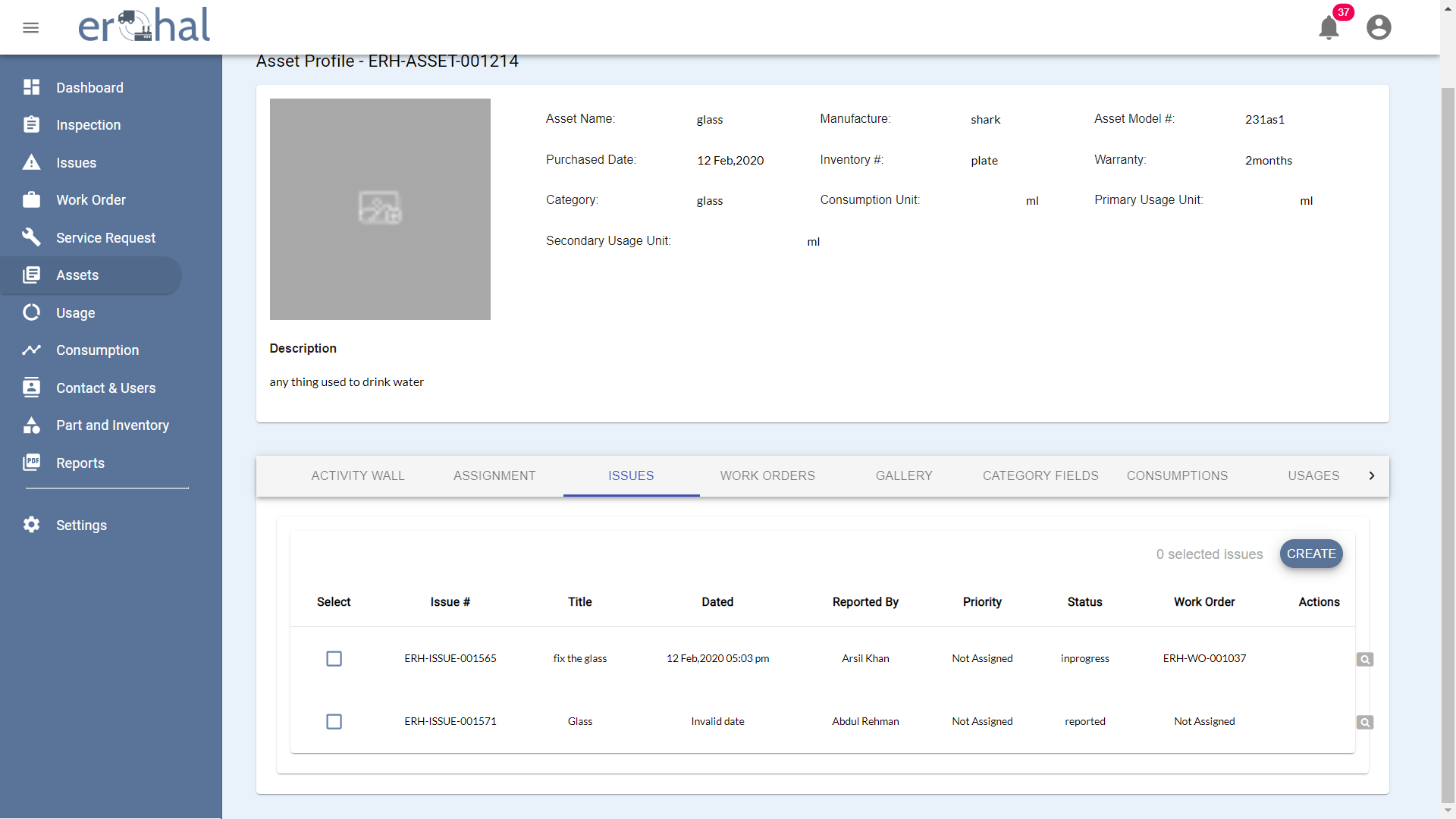


Figure 24: View Asset 4

This GUI shows list of all issues of an asset.

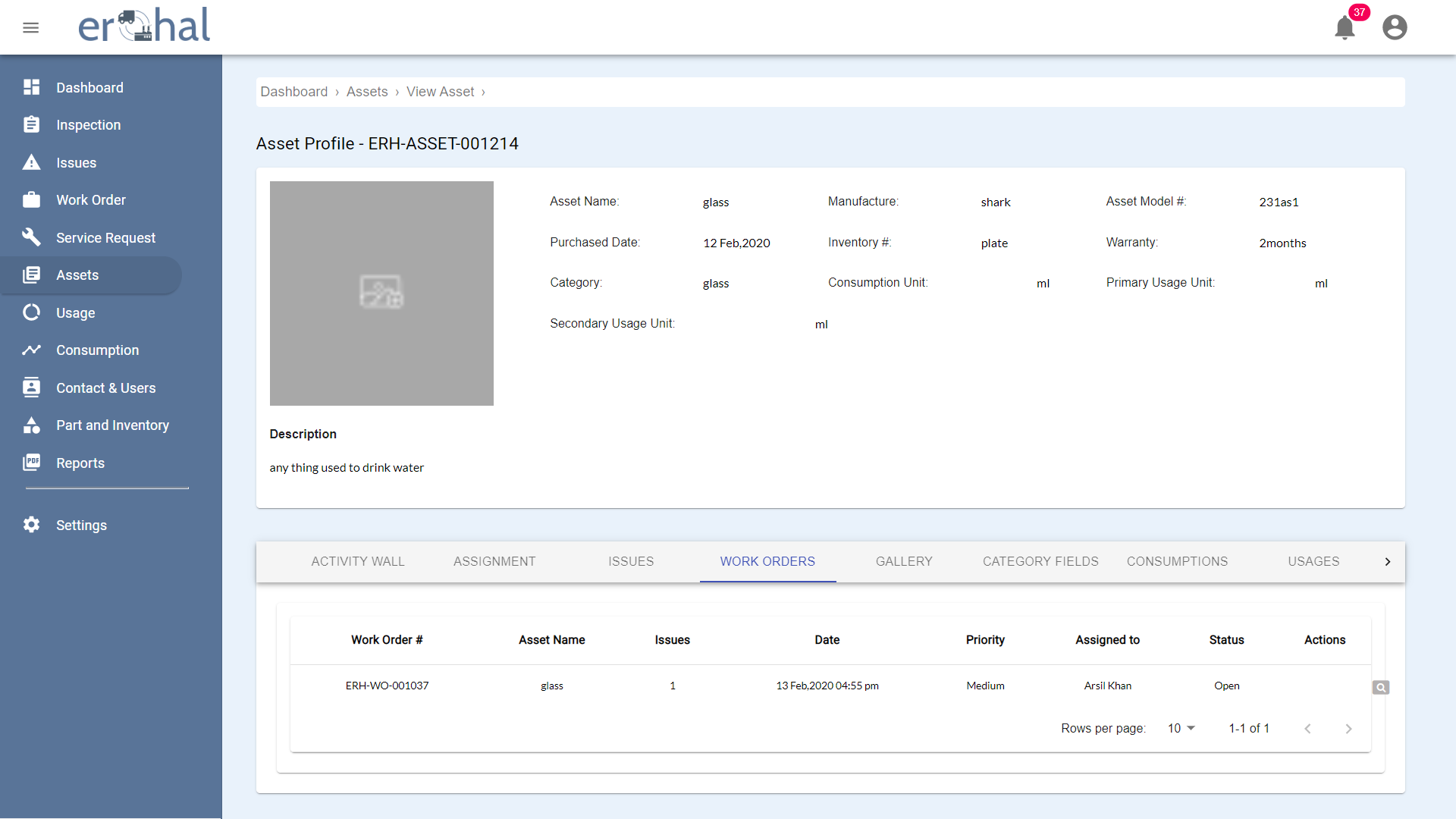


Figure 25: View Asset 5

This GUI shows all work orders of an asset.

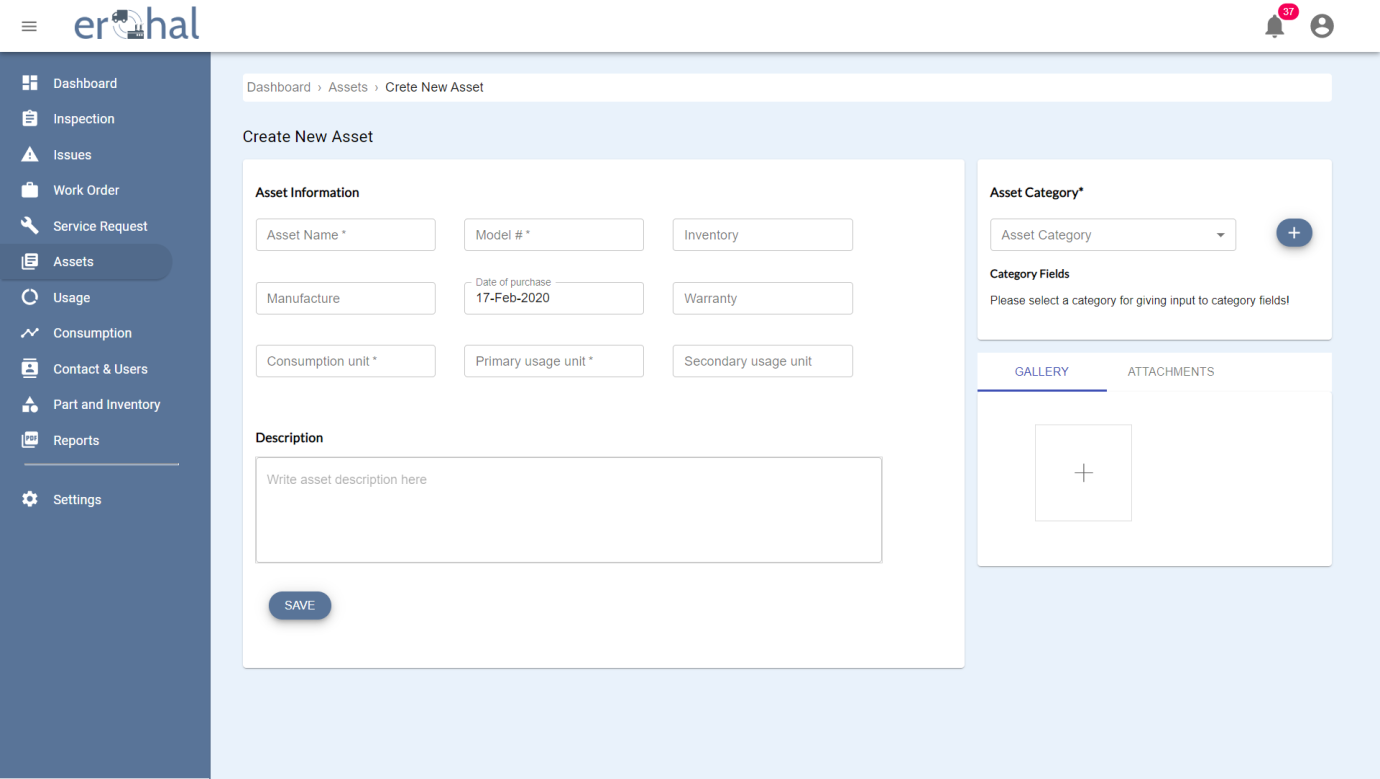


Figure 26: Create New Asset

User can create new asset on this GUI.

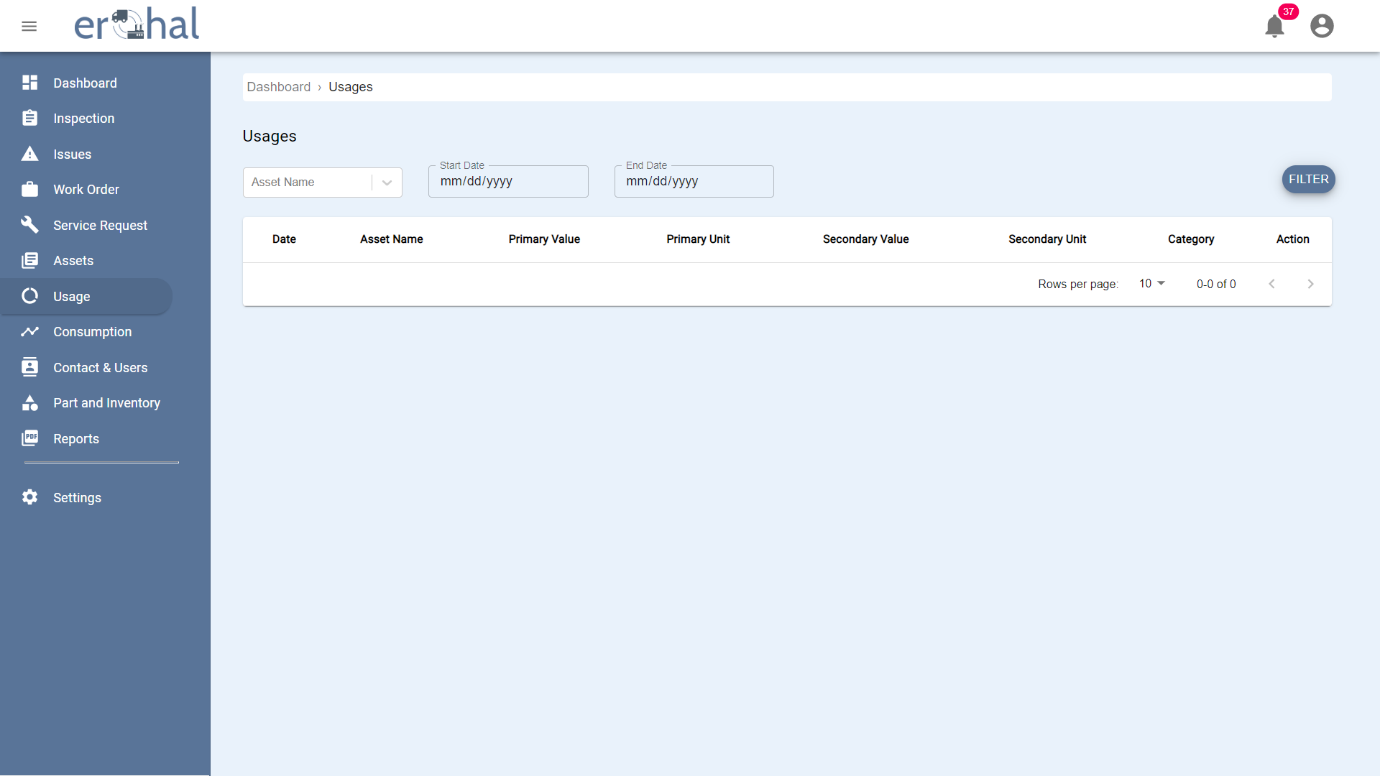


Figure 27: View Usage

This GUI shows all usages of an asset.

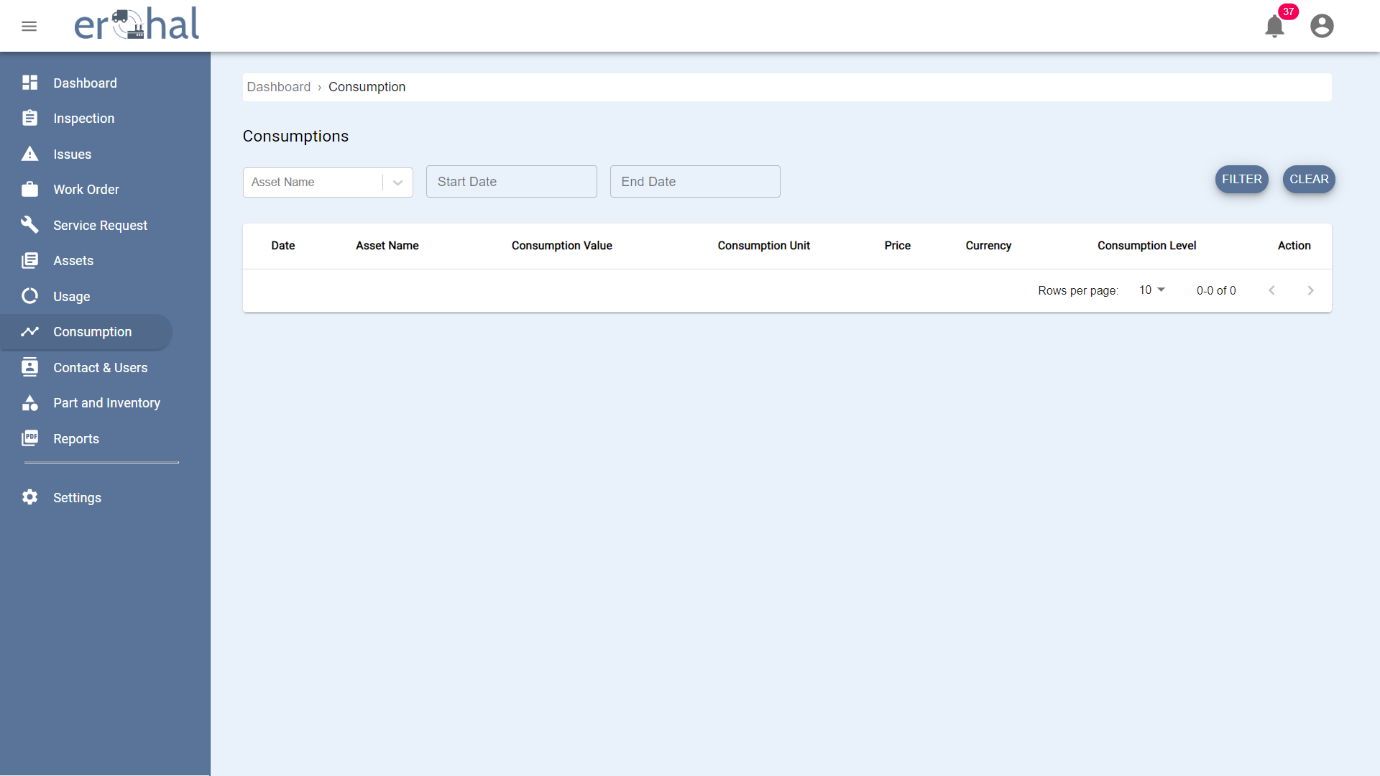


Figure 28: View Consumption

This GUI shows all consumptions of an asset.

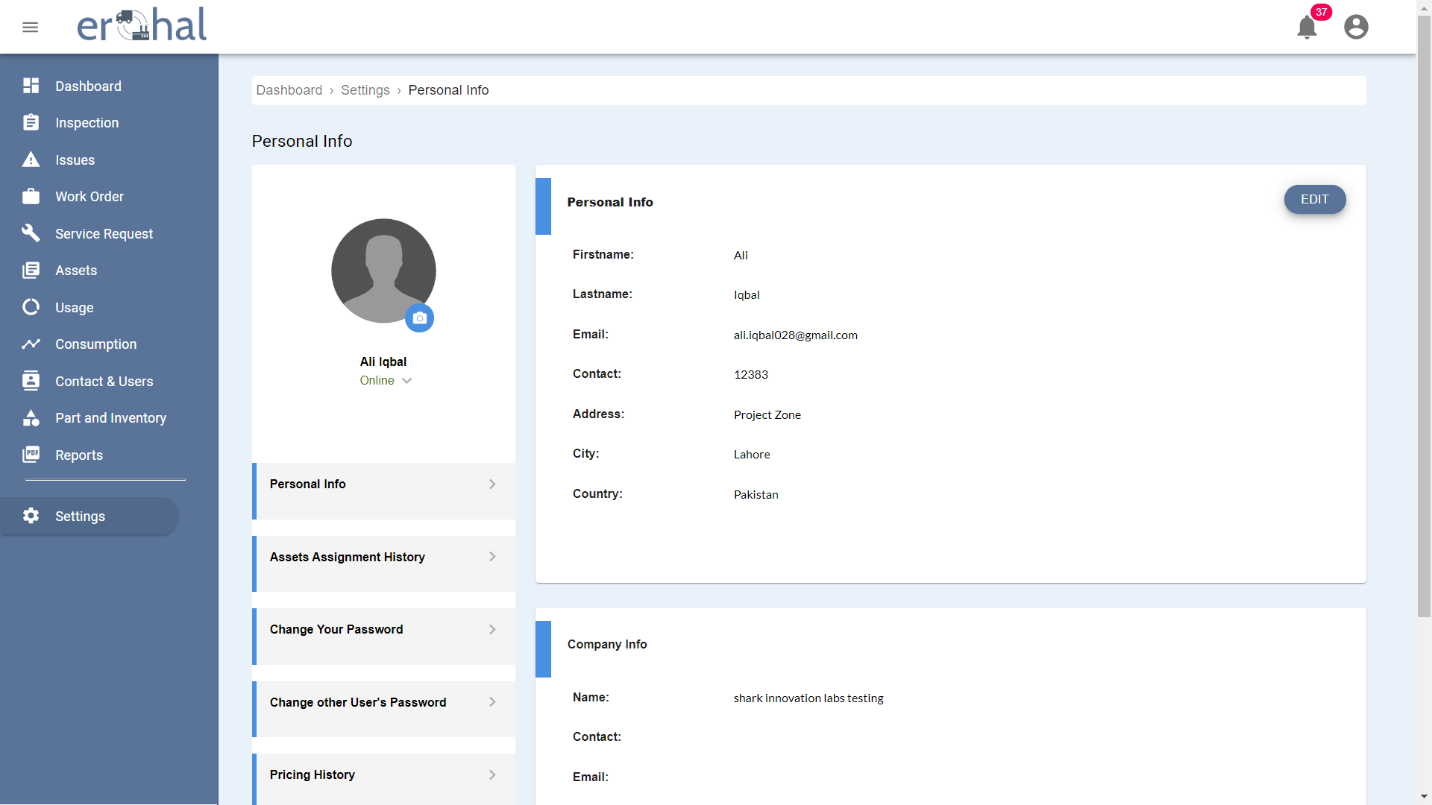


Figure 29: Profile Overview

This GUI show personal and company’s information of a profile.

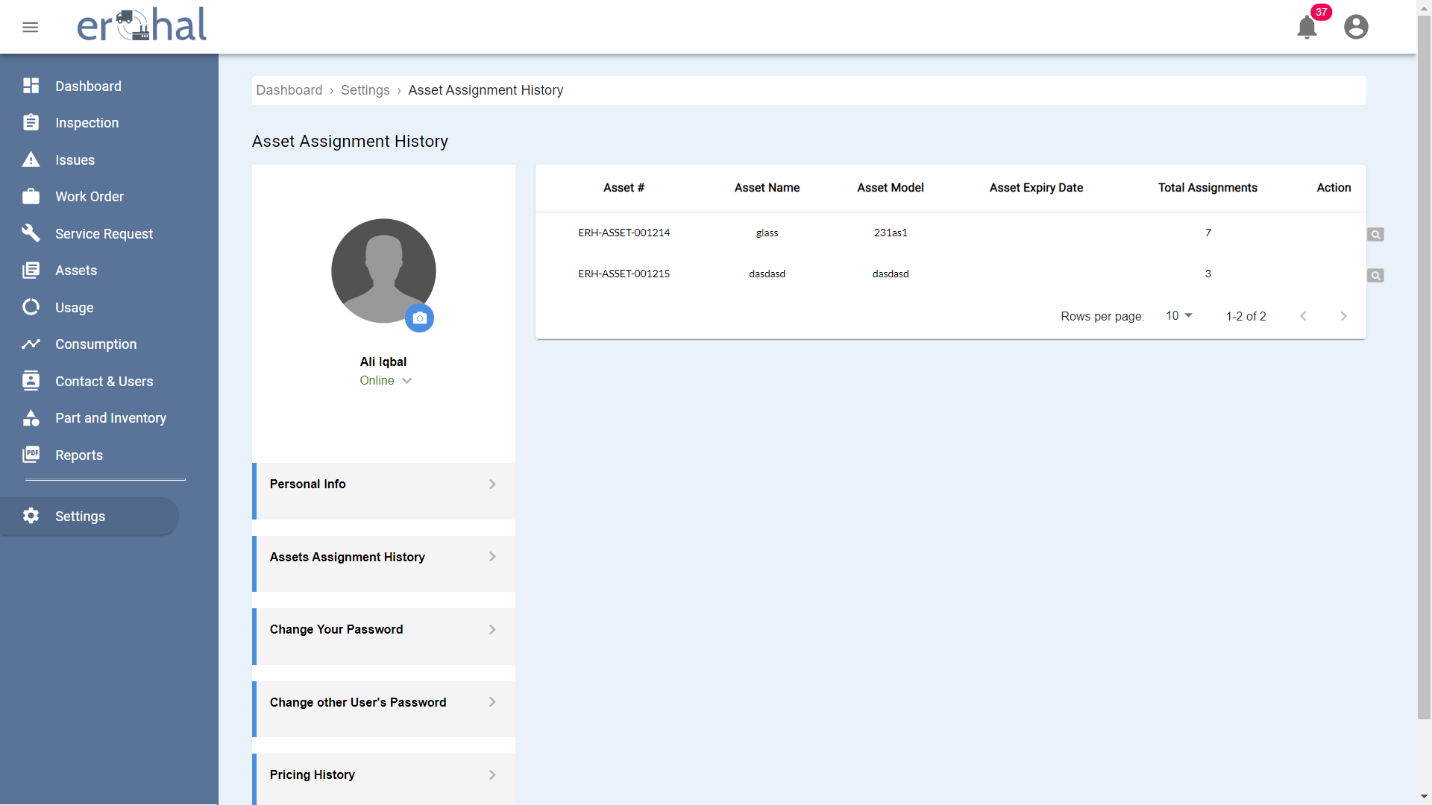


Figure 30: Asset Assignment History

This GUI shows the history of assets assignment.

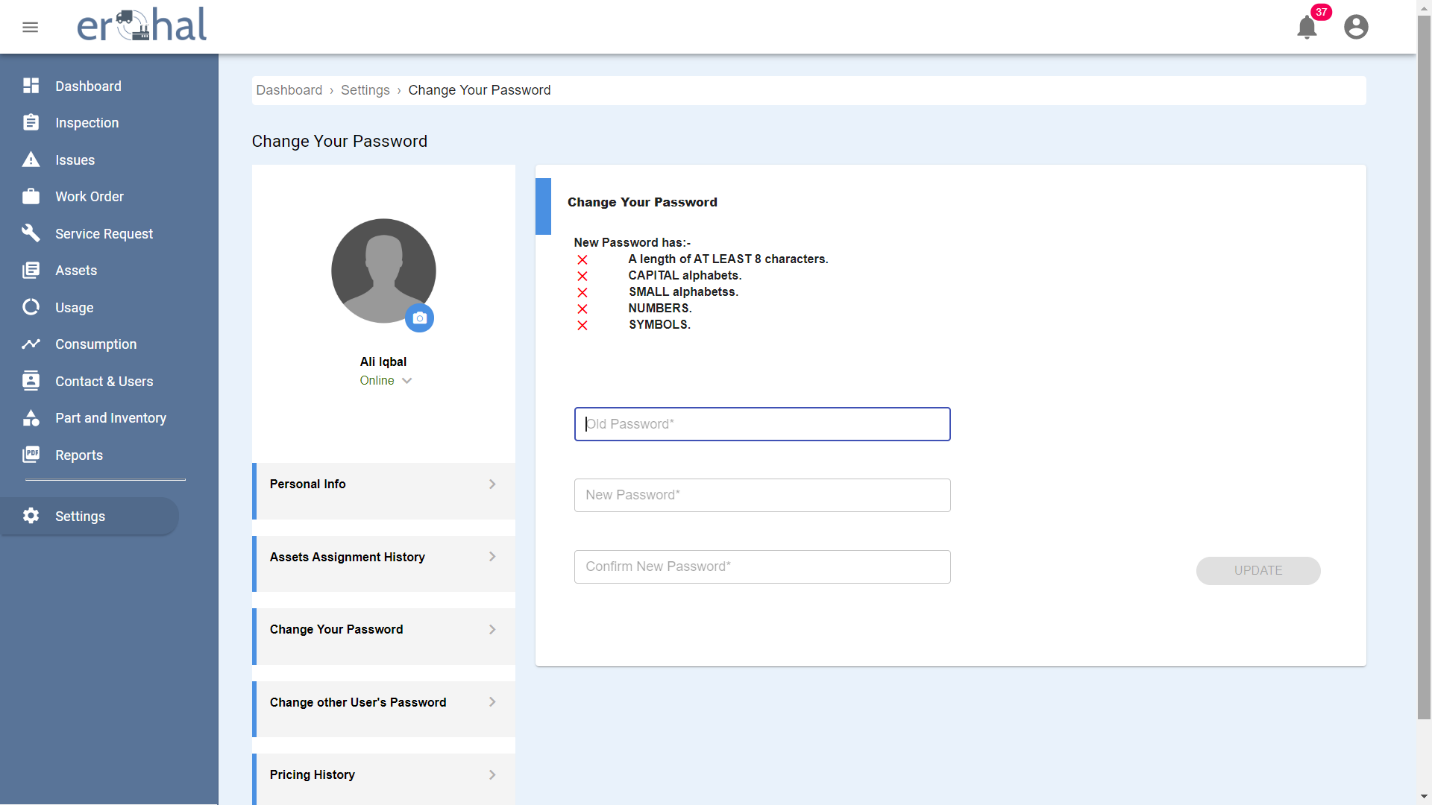


Figure 31: Change Password

This GUI shows how to change your password.

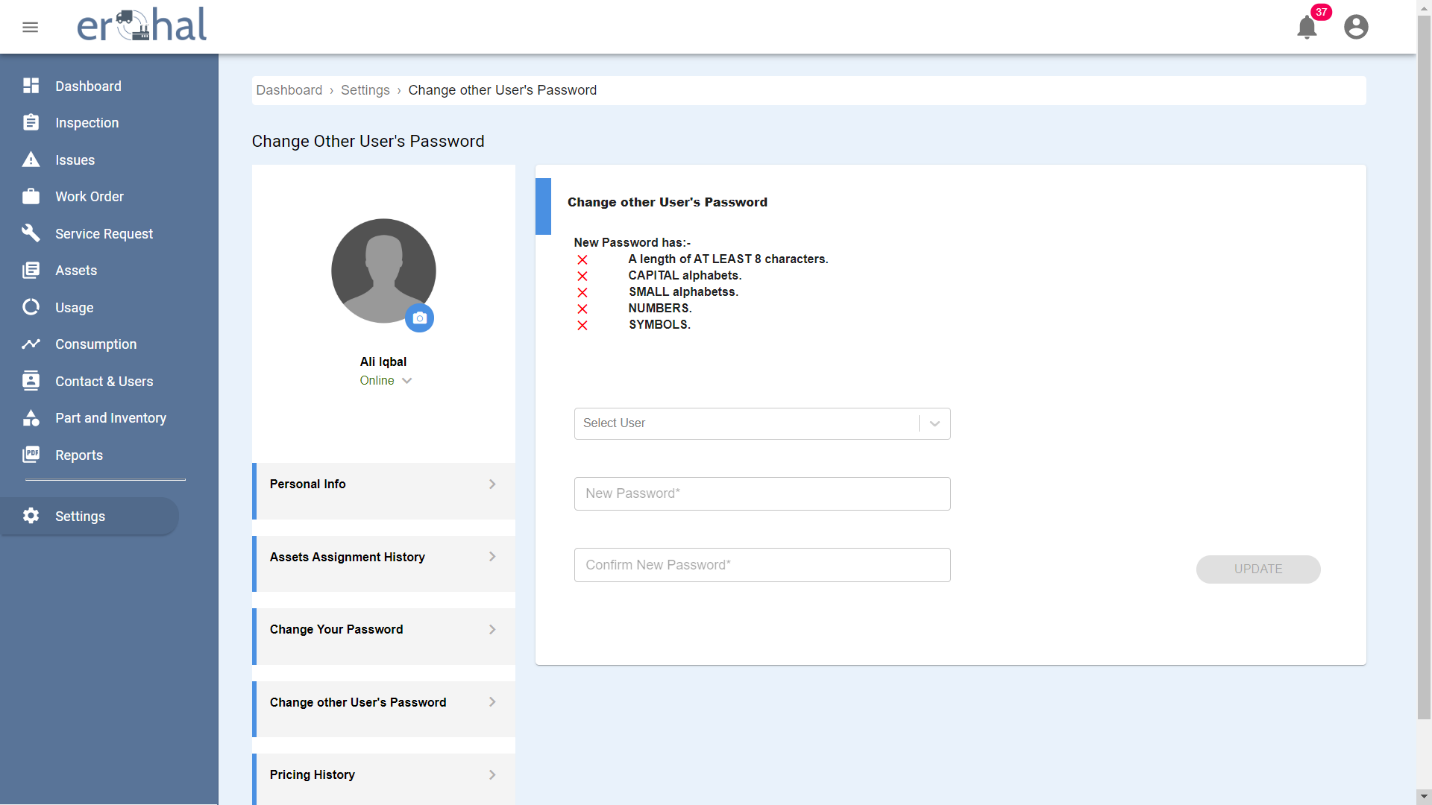


Figure 32: Change Other's Password

This GUI shows how to change another user’s password

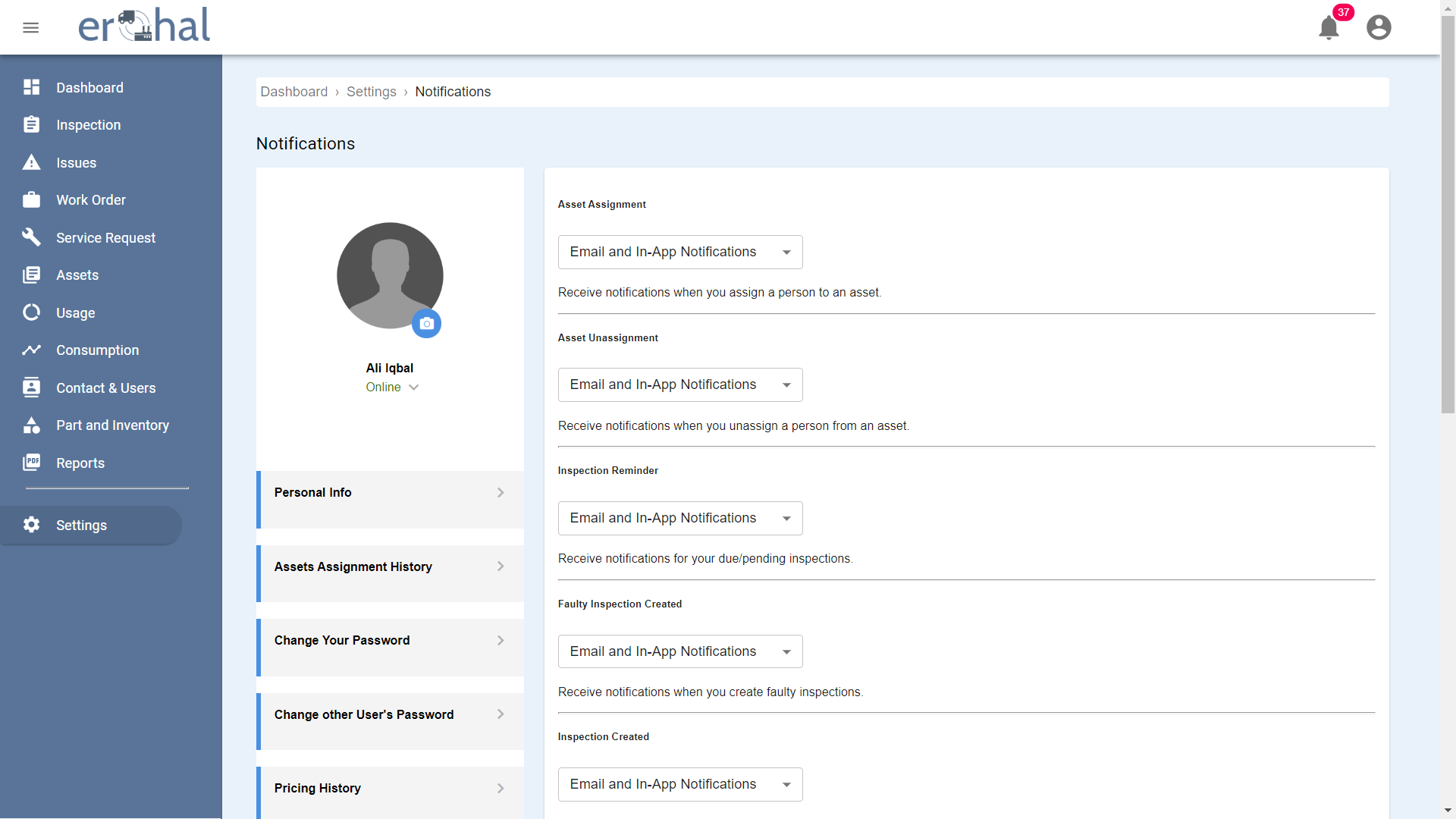


Figure 33: Notifications 1

This GUI shows categories for notifications.

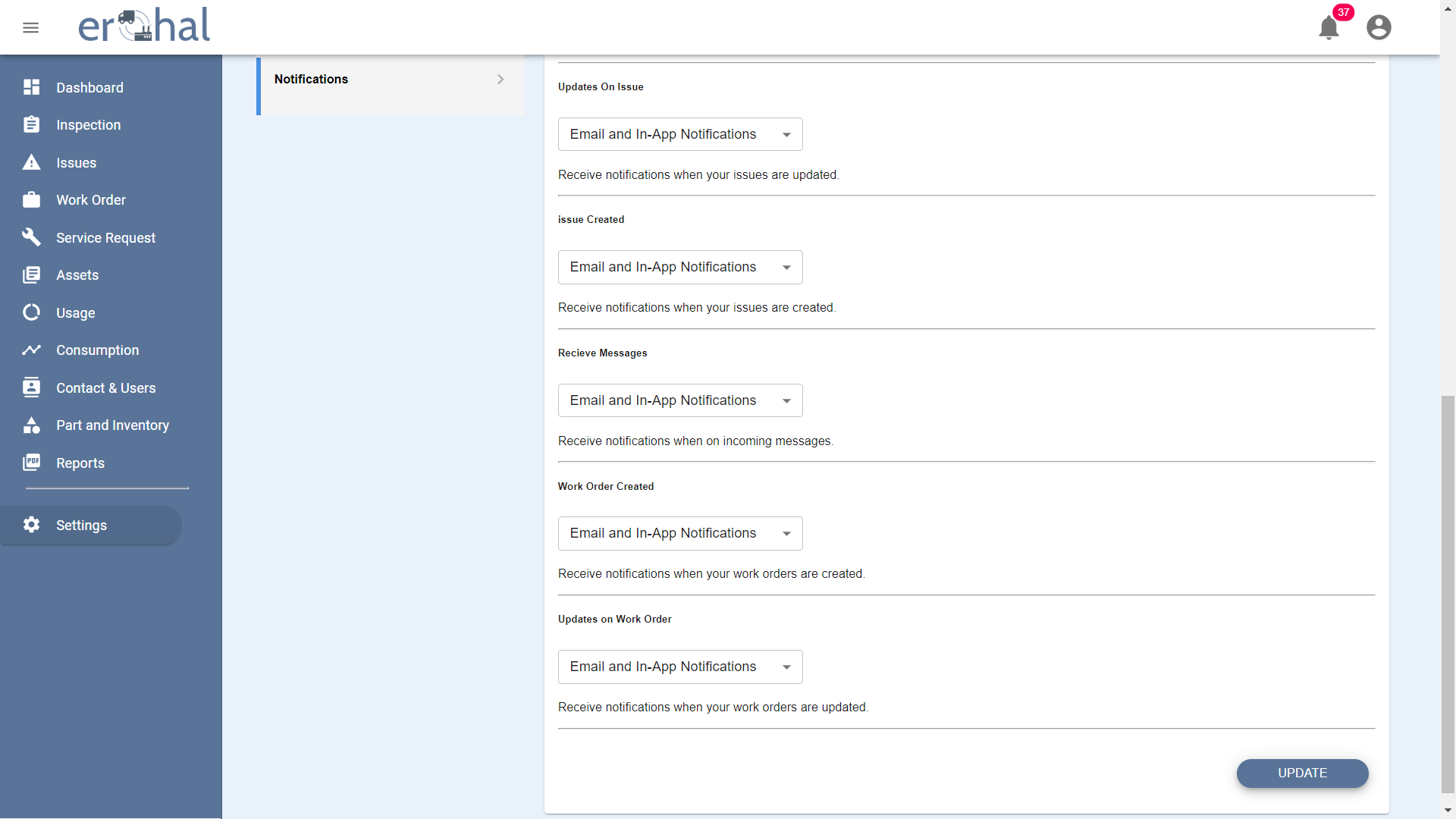


Figure 34: Notifications 2

This GUI shows categories for notifications with options to update their type.

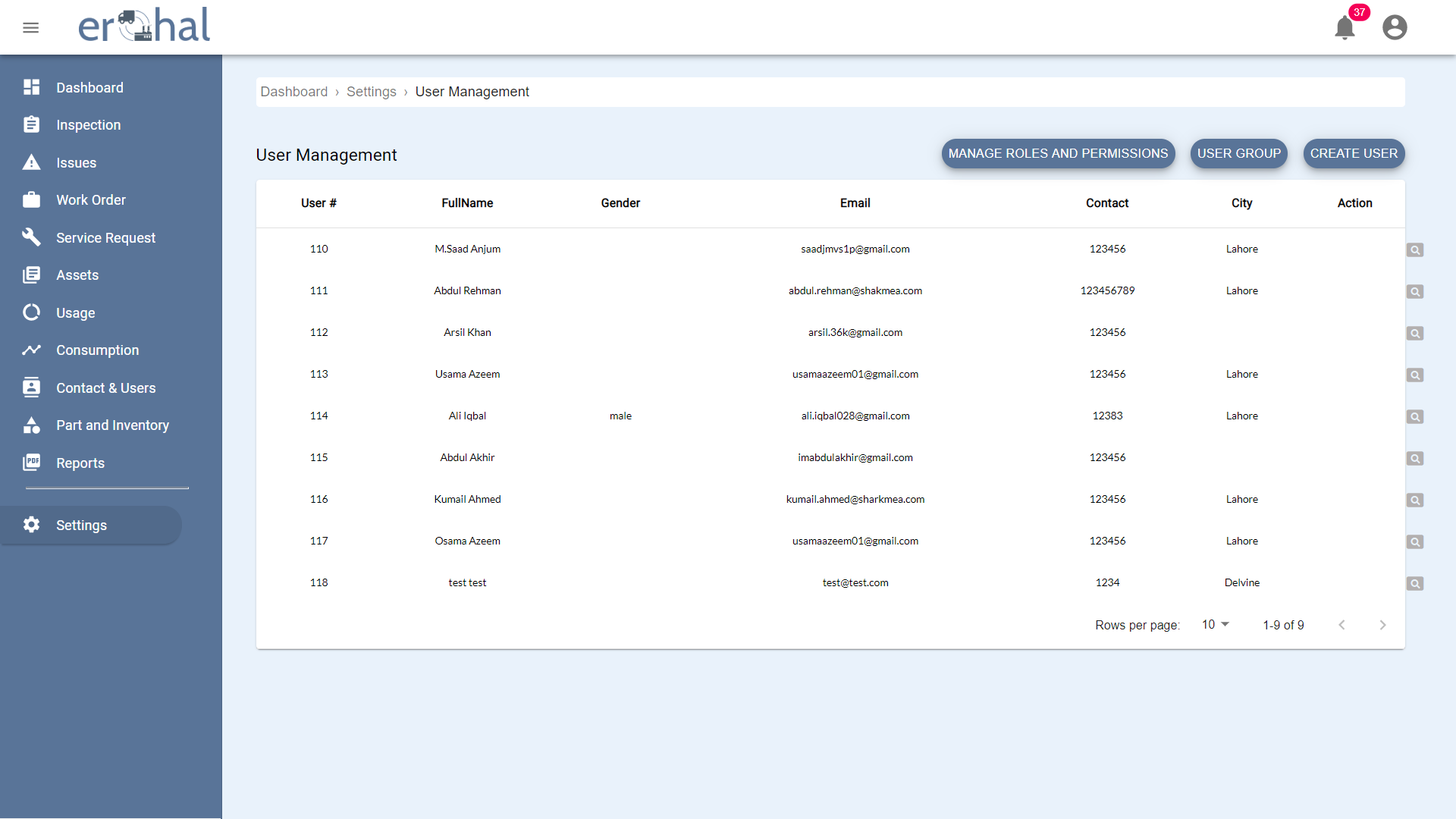


Figure 35: User Management

This GUI shows user management page with all the users.

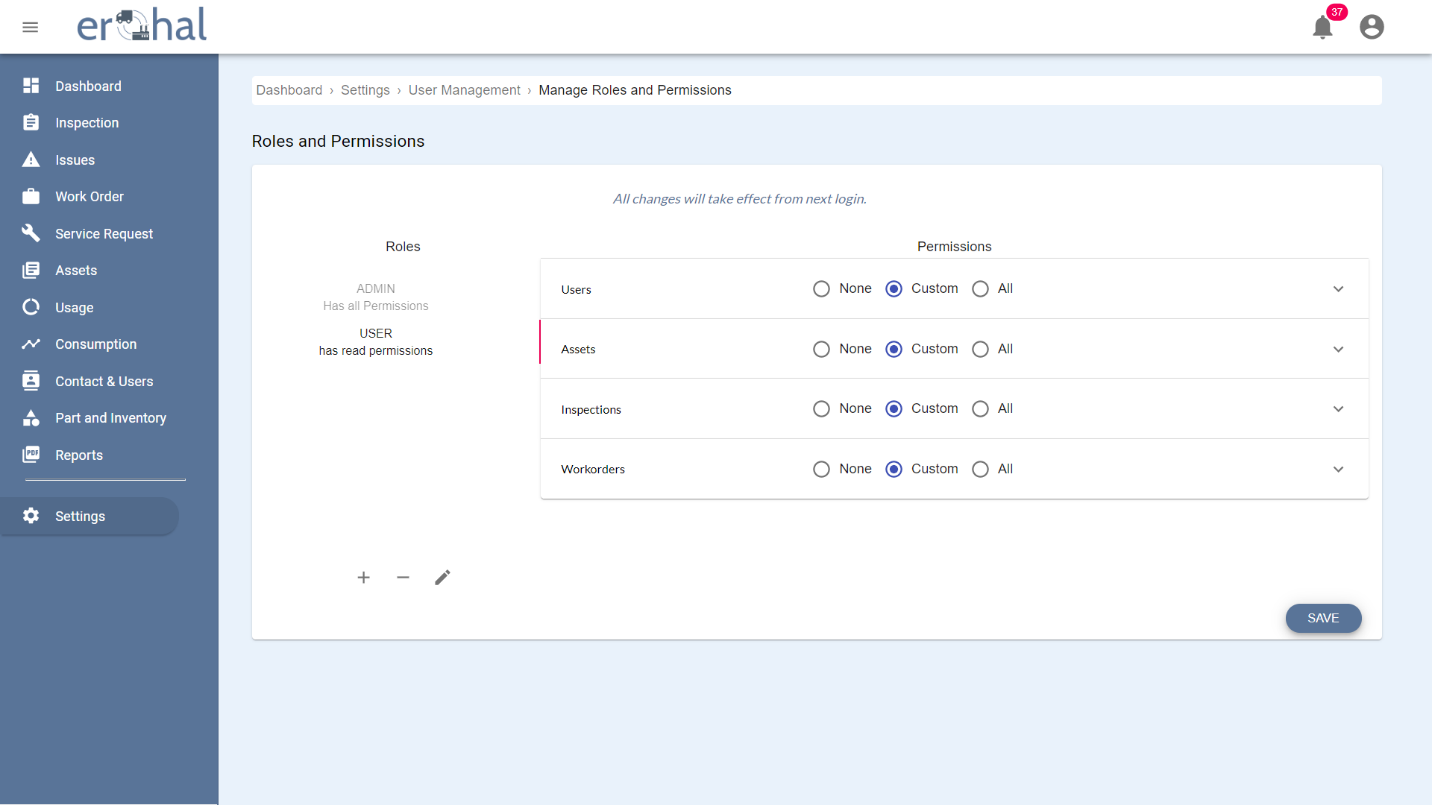


Figure 36: Roles and Permissions

This GUI shows the page to manage roles and permissions of a user.



Figure 37: User Groups

This GUI shows all user groups.

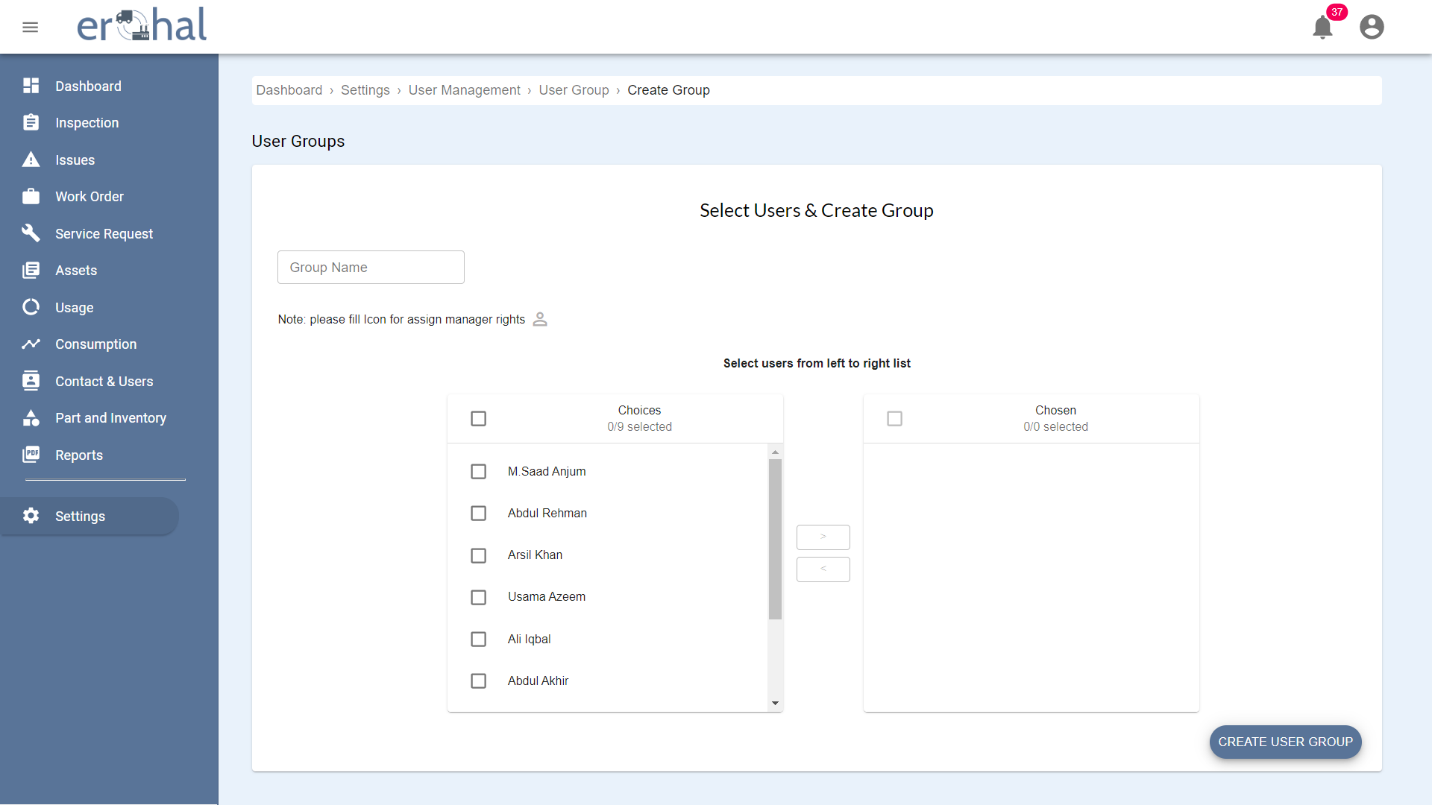


Figure 38: Create User Group

This GUI shows how to create a new user group.

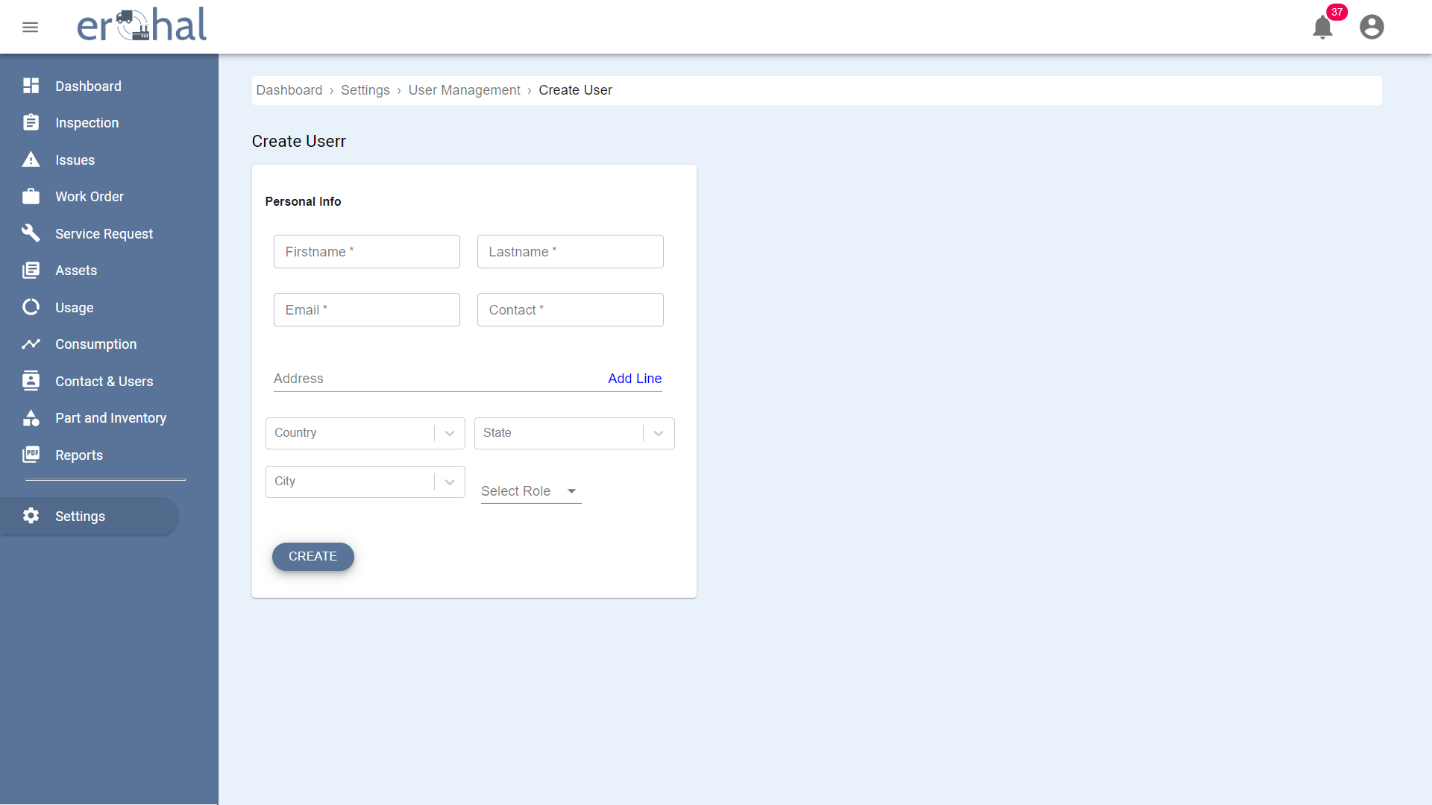


Figure 39: Create User

This GUI shows how to create a new user.

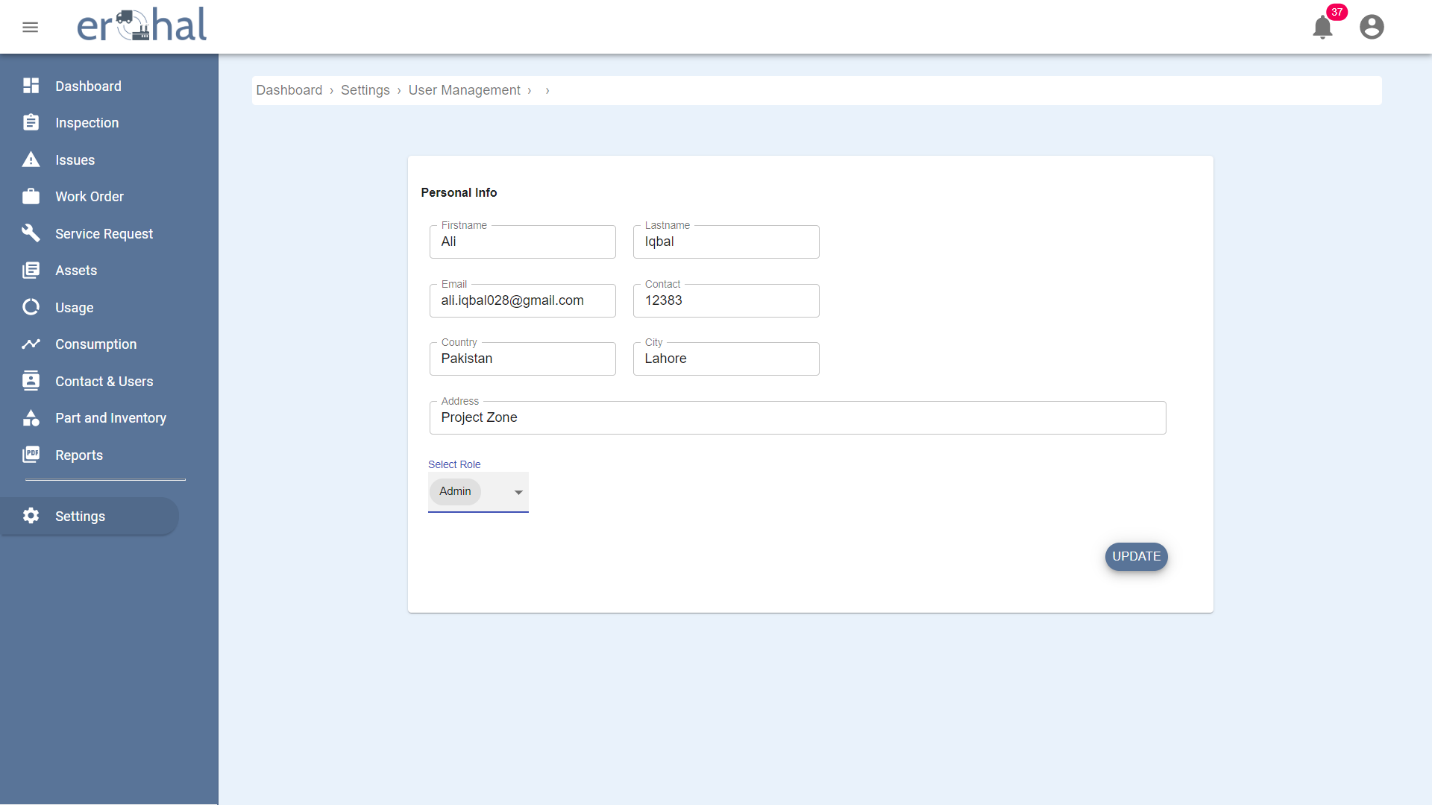


Figure 40: Personal Information

This GUI shows the personal information and the role of a user with an option to update the details.

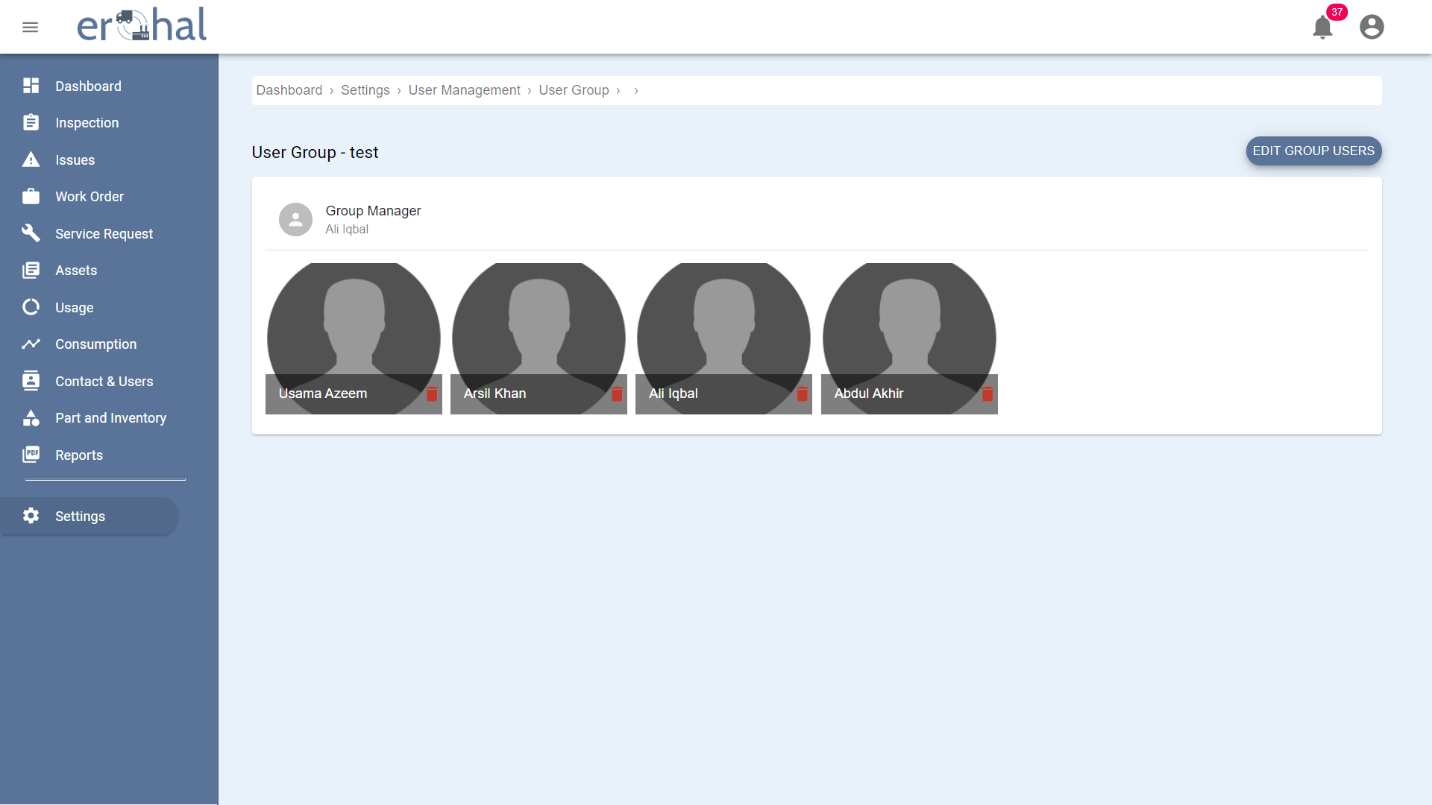


Figure 41: User Group Profile

This GUI shows the details of a single user group.

# 

# Database Design

## ER Diagram

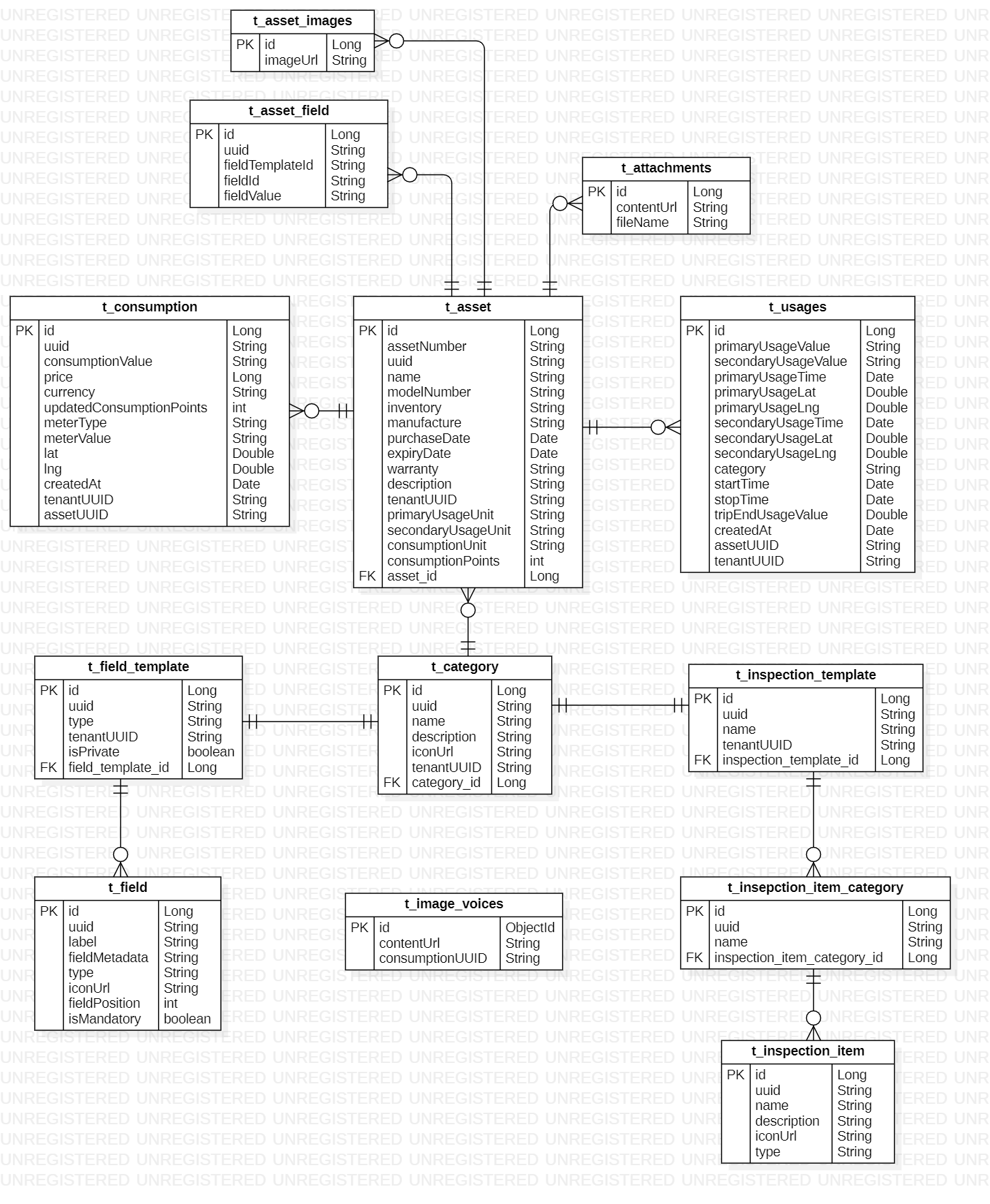


Figure 42: ER Diagram 1

This shows the ER Diagram of Asset Management Service

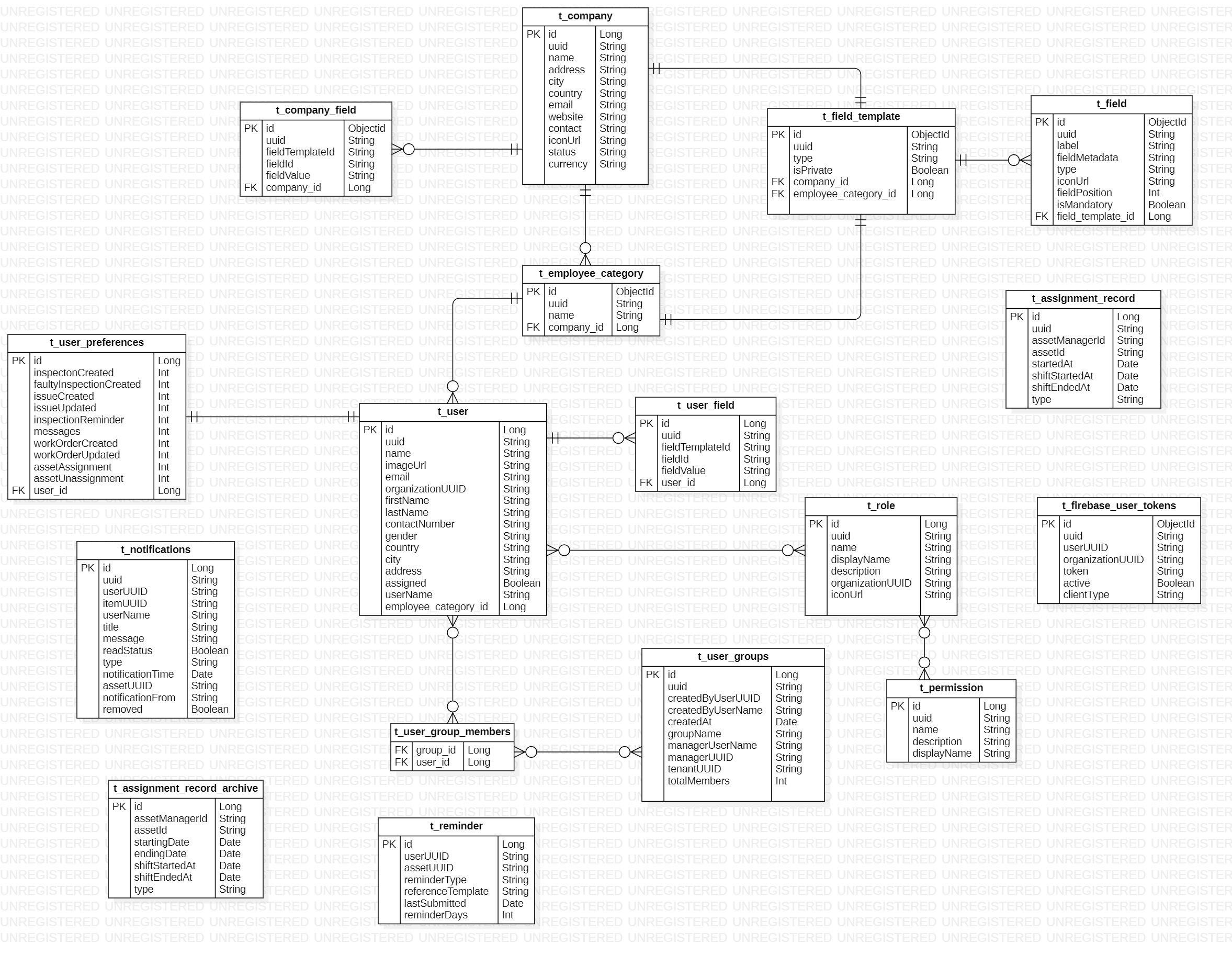


Figure 43: ER Diagram 2

This shows the ER Diagram of Asset Personnel Service

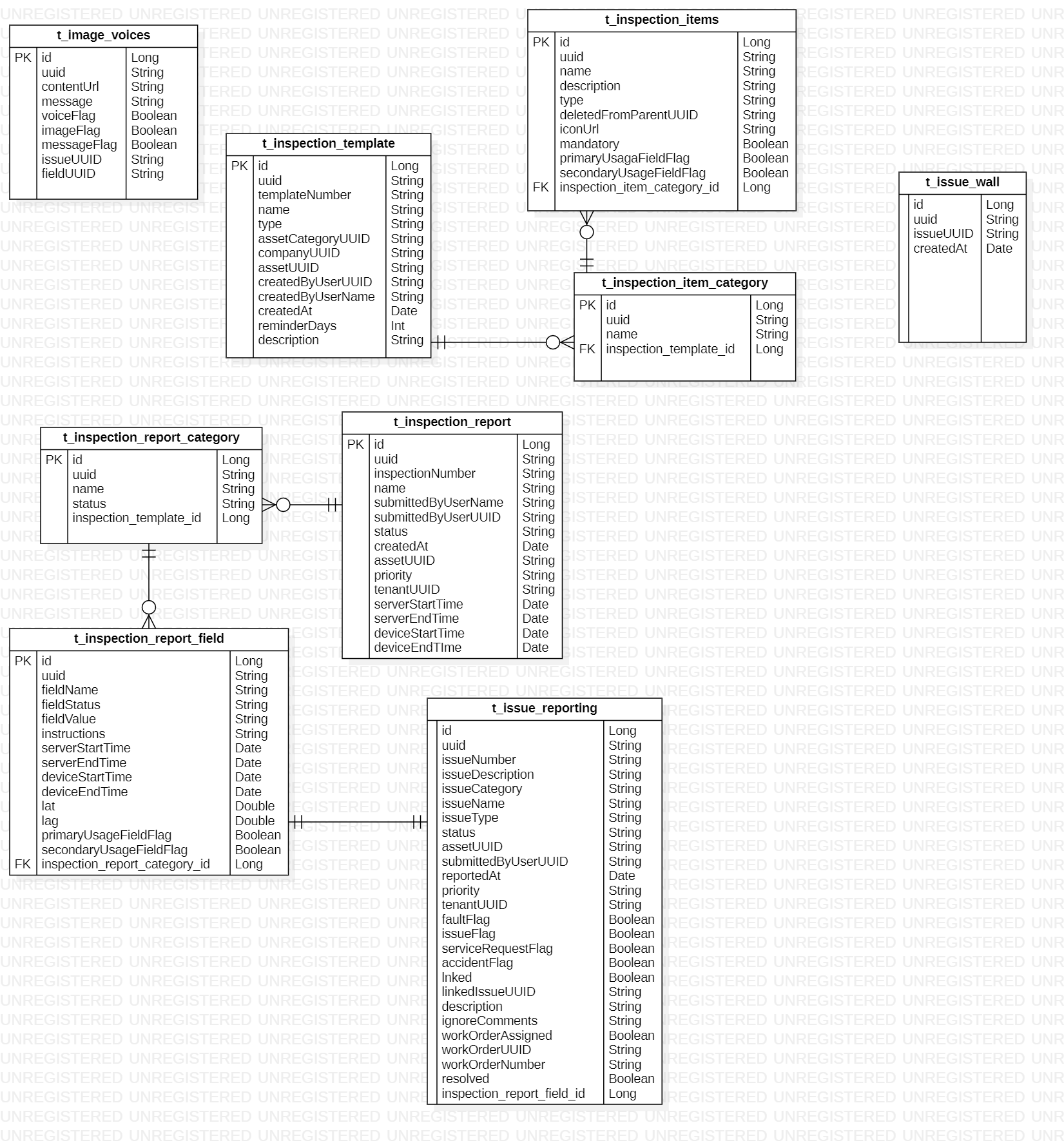


Figure 44: ER Diagram 3

This shows the ER Diagram of Inspection Service

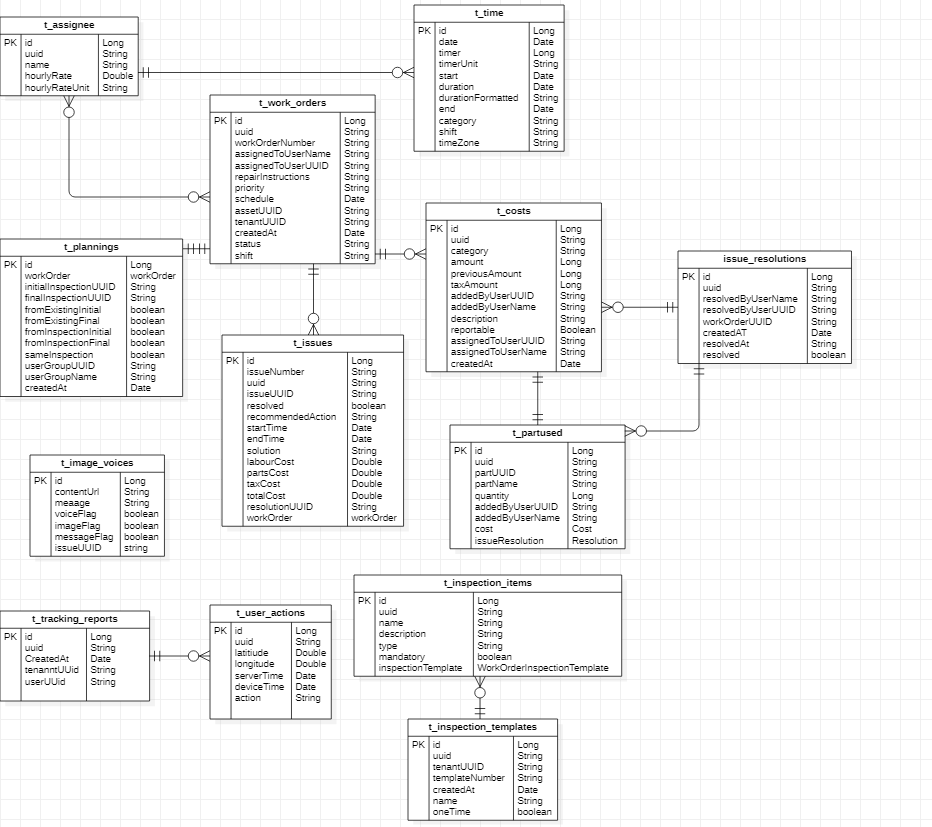


Figure 45: ER Diagram 4

This is the ER Diagram of Work Order Service

## Data Dictionary

### Image Voices

Table 55: Image Voices Data Dictionary

This is the data dictionary of image voices.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| Id | Long | 123 |
| Uuid | String | 1a2b31231wso |
| ContentUrl | String | [https://somename.com](https://somename.com/) |
| Message | String | Some message |
| VoiceFlag | Boolean | True |
| ImageFlag | Boolean | False |
| MessageFlag | Boolean | True |
| IssueUUID | String | 1a2b31231wso |
| FieldUUID | String | 1a2b31231wso |

### Inspection Template

Table 56: Inspection Template Data Dictionary

This is the data dictionary of inspection template.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| Id | Long | 3231 |
| Uuid | String | 1a2b31231wso |
| TemplateNumber | String | ERH-INS-003378 |
| Name | String | AC inspection template |
| Type | String | Daily inspection |
| AssetCategoryUUID | String | 1a2b31231wso |
| CompanyUUID | String | 1a2b31231wso-2 |
| AssetUUID | String | 1a2b3%1231wso |
| CreatedByUserUUID | String | 1a2b3123#2@1wso |
| CreatedByUserName | String | someusername |
| CreatedAt | Date | 12-03-2020 12:35:12 |
| ReminderDays | Int | 4 |
| Description | String | Any description can fit in |

### Inspection Items

Table 57: Inspection Items Data Dictionary

This is the data dictionary of inspection items.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| Id | Long | 23 |
| Uuid | String | 1a2b31231wso |
| Name | String | Car |
| Description | String | Any description |
| Type | String | Car |
| DeletedFromParentUUID | String | 1a2b3123#2@1wso |
| IconUrl | String | [https://somename.com](https://somename.com/) |
| Mandatory | Boolean | False |
| PrimaryUsageFieldFlag | Boolean | True |
| SecondaryUsageFieldFlag | Boolean | True |
| inspection\_item\_category\_id | Long | 3 |

### Inspection Item Category

Table 58: Inspection Item Category Data Dictionary

This is the data dictionary of inspection item category

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Types** | **Examples** |
| Id | Long | 23 |
| Uuid | String | 1a2b3123#2@1wso |
| Name | String | Car |
| Inspection\_template\_id | Long | 5 |

### Inspection Report

Table 59: Inspection Report Data Dictionary

This is the data dictionary of inspection report.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Types** | **Examples** |
| Id | Long | 7 |
| Uuid | String | 1a2b3123#2@1wso |
| InspectionNumber | String | ERH-INS-003378 |
| Name | String | Daily inspection report |
| SubmittedByUserName | String | someusername |
| SubmittedByUserUUID | String | 1a2b3123#2@1wso |
| Status | String | passed |
| CreatedAt | Date | 12-03-2020 12:35:12 |
| AssetUUID | String | 1a2b3123#2@1wso |
| Priority | String | medium |
| TenantUUID | String | 1a2b3123#2@1wso |
| ServerStartTime | Date | 12-03-2020 12:35:12 |
| ServerEndTime | Date | 12-03-2020 12:35:12 |
| DeviceStartTime | Date | 12-03-2020 12:35:12 |
| DeviceEndTime | Date | 12-03-2020 12:35:12 |

### Inspection Report Category

Table 60: Inspection Report Category

This is the data dictionary of inspection report category.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| Id | Long | 23 |
| Uuid | String | 1a2b3123#2@1wso |
| Name | String | Daily inspection report |
| Status | String | pending |
| inspection\_template\_id | Long | 12 |

### Inspection Report Field

Table 61: Inspection Report Field

This is the data dictionary of inspection report field.

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Types** | **Examples** |
| Id | Long | 3 |
| Uuid | String | 1a2b3123#2@1wso |
| FieldName | String | Bike inspection report |
| FieldStatus | String | Pass |
| FieldValue | String | Any measured value |
| Instructions | String | Some instructions |
| ServerStartTime | Date | 12-03-2020 12:35:12 |
| ServerEndTime | Date | 12-03-2020 12:35:12 |
| DeviceStartTime | Date | 12-03-2020 12:35:12 |
| DeviceEndTime | Date | 12-03-2020 12:35:12 |
| Lat | Double | 233.232321 |
| Lag | Double | 0.23123131 |
| PrimaryUsageFieldFlag | Boolean | True |
| SecondaryUsageFieldFlag | Boolean | False |
| Inspection\_report\_category\_id | Long | 6 |

### Issue Reporting

Table 62: Issue Reporting

This is the data dictionary of issue reporting.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Types** | **Examples** |
| Id | Long | 12 |
| Uuid | String | 1a2b3123#2@1wso |
| IssueNumber | String | ERH-ISSUE-001555 |
| IssueDescription | String | Some description about issue |
| IssueName | String | cleanliness |
| IssueType | String | cleanliness |
| Status | String | new |
| AssetUUID | String | 1a2b3123#2@1wso |
| SubmittedByUserUUID | String | 1a2b3123#2@1wso |
| ReportedAt | Date | 12-03-2020 12:35:12 |
| Priority | String | High |
| TenantUUID | String | 1a2b3123#2@1wso |
| FaultFlag | Boolean | False |
| IssueFlag | Boolean | False |
| ServiceRequestFlag | Boolean | True |
| AccidentFlag | Boolean | False |
| Linked | Boolean | True |
| LinkedIssueUUID | String | 1a2b3123#2@1wso |
| Description | String | Any description |
| IgnoreComments | String | Ignore comments |
| WorkOrderAssigned | Boolean | True |
| WorkOrderUUID | String | 1a2b3123#2@1wso |
| WorkOrderNumber | String | ERH-WO-001036 |
| Resolved | Boolean | True |
| inspection\_report\_field\_id | Long | 1 |

### Assign

Table 63: Assign Data Dictionary

This is the data dictionary of the assign table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (PK) | Long | 12 |
| uuid | String | 1a2b3123#2@1wso |
| name | String | someusername |
| hourlyRate | Double | 2123442 |
| chourlRateUnit | Date | 02/11/2020 |
| totalWorkedTime | String | 12 hours |
| totalWorkedTimeFormatted | String | 12 hours |

### Time

Table 64: Time Data Dictionary

This is the data dictionary of the time table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Datatype** | **Example** |
| id (Pk) | Long | 12 |
| date | Date | 02/11/2020 |
| timer | Long | 233 |
| timerUnit | String | second |
| start | Date | 02/11/2020 |
| duration | Date | 02/11/2020 |
| durationFormmated | String | 23 hours |
| end | Date | 02/11/2020 |
| category | String | GPS |
| shift | String | 23 hours |

### Work Order

Table 65: Work Order Data Dictionary

This is the data dictionary of work order table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 12 |
| uuid | String | 1a2b3123#2@1wso |
| workOrderNumber | String | 1a2b3123#2@1wso |
| assignedToUserName | String | someusername |
| assignedToUserUUID | String | 1a2b3123#2@1wso |
| repairInstruction | String | Some instructions |
| Priority | String | high |
| Schedule | Date | 02/11/2020 |
| assetUUID | String | 1a2b3123#2@1wso |
| tenantUUID | String | 1a2b3123#2@1wso |
| createdAt | Date | 02/11/2020 |
| Status | String | pending |
| Shift | String | day |
| estimatedTime | Date | 02/11/2020 |
| elapsedTime | Date | 02/11/2020 |
| slapsedTimeFormatted | String | 23 hours |
| startTime | Date | 02/11/2020 |
| endTime | Date | 02/11/2020 |
| startUsageType | String | type |
| startUsageValue | String | 23 km |
| startUsageAdded | boolean | true |
| endUsageType | String | type |
| endUsageValue | String | 24 km |
| endUsageAdded | boolean | False/true |

### Plannings

Table 66: Plannings Data Dictionary

This is the data dictionary of the plannings table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 12 |
| initialInspectionUUID | String | 1a2b3123#2@1wso |
| finalInspectionUUID | String | 1a2b3123#2@1wso |
| fromExistingInitial | Boolean | false |
| fromExistingFinal | Boolean | salse |
| fromInspectionInitial | Boolean | true |
| fromInspecionFinal | Boolean | false |
| sameInspection | Boolean | true |
| userGroupUUID | String | 1a2b3123#2@1wso |
| userGroupName | String | 1a2b3123#2@1wso |

### Costs

Table 67: Costs Data Dictionary

This is the data dictionary of the costs table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 12 |
| uuid | String | 1a2b3123#2@1wso |
| Category | String | Glass |
| Amount | Long | 320000 |
| previousAmount | Long | 233 |
| taxAmount | Long | 323323 |
| addedByUserUUID | String | 1a2b3123#2@1wso |
| addedByUserName | String | someusername |
| Description | String | something |
| Reportable | Boolean | true |
| assignedToUserUUID | String | 1a2b3123#2@1wso |
| aasignedToUserName | String | someusername |
| createdAt | Date | 2 am |

### Issue Resolutions

Table 68: Issue Resolutions Data Dictionary

This is the data dictionary of the issue resolutions table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 12 |
| uuid | String | 1a2b3123#2@1wso |
| resolvedByUserName | String | someusername |
| resolvedByUserUUID | String | 1a2b3123#2@1wso |
| workOrderUUID | String | 1a2b3123#2@1wso |
| createdAt | Date | Date and time |
| resolvedAt | String | date |
| Resolved | boolean | True/False |

### Part Used

Table 69: Part Used Data Dictionary

This is the data dictionary of the part used table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 12 |
| uuid | String | 1a2b3123#2@1wso |
| partUUID | String | 1a2b3123#2@1wso |
| partName | String | shaft |
| Quantity | Long | 232332 |
| addedByUserUUID | String | 1a2b3123#2@1wso |
| addedByUserName | String | someusername |

### Tracking Report

Table 70: Tracking Report Data Dictionary

This is the data dictionary of tracking report.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id(PK) | Long | 12 |
| uuid | String | 1a2b3123#2@1wso |
| createdAt | Date | Date and time |
| tenantUUID | String | 1a2b3123#2@1wso |
| userUUID | String | 1a2b3123#2@1wso |

### User Action

Table 71: User Action Data Dictionary

This is the data dictionary of user action.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id(PK) | Long | 12 |
| uuid | String | 1a2b3123#2@1wso |
| latitude | Double | 7732873283723 |
| longitude | Double | 326732238 |
| serverTime | Date | 02/11/2020 |
| deviceTime | Date | 02/11/2020 |
| action | String | Something to say |

### Asset

Table 72: Asset Data Dictionary

This is the data dictionary of the asset.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (PK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| assetNumber | String | 13A76t |
| uuid | String | 1a2b31231wso |
| name | String | Akhir |
| modelNumber | String | 2008 |
| inventory | String | cars |
| manufacture | String | Honda |
| purchaseDate | Date | 02/11/2020 |
| expiryDate | Date | 05/11/2020 |
| warranty | String | 3 months |
| description | String | Asset detail |
| tenantUUID | String | 1a2b31231wso |
| primaryUsageUnit | String | kilometer |
| secondaryUsageUnit | String | mile |
| consumptionUnit | String | meter |
| consumptionPoints | int | 12 |
| asset\_id (FK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |

### Consumption

Table 73: Consumption Data Dictionary

This is the data dictionary of the consumption.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (PK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| uuid | String | 1a2b31231wso |
| consumptionValue | String | 128316 |
| price | Long | 32999 |
| currency | String | Dollars |
| updatedConsumptionPoints | Int | 13981 |
| meterType | String | mileage |
| meterValue | String | 12890 |
| lat | Double | 13.899348883 |
| lng | Double | 13.899348883 |
| createdAt | Date | 02/11/2020 |
| tenantUUID | String | 1a2b31231wso |
| assetUUID | String | 1a2b31231wso |

### Usage

Table 74: Usage Data Dictionary

This is the data dictionary of the usage.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (PK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| primaryUsageValue | String | “19231242535” |
| secondaryUsageValue | String | “19231242535” |
| primaryUsageTime | Date | 14:00 |
| primaryUsageLat | Double | 13.899348883 |
| primaryUsageLng | Double | 13.918836542 |
| secondaryUsageTime | Date | 15:00 |
| secondaryUsageLat | Double | 13.899348883 |
| secondaryUsageLng | Double | 13.918836542 |
| category | String | “category name” |
| startTime | Date | 12:00 |
| stopTime | Date | 13:00 |
| tripEndUsageValue | Double | 3000 |
| createdAt | Date | 02/11/2020 |
| assetUUID | String | 1a2b31231wso |
| tenantUUID | String | 1a2b31231wso |

### Asset Field

Table 75: Asset Field Data Dictionary

This is the data dictionary of the asset field.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (Pk) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| uuid | String | 1a2b31231wso |
| fieldTemplateId | String | 1a2b31231wso |
| fieldId | String | 1a2b31231wso |
| fieldValue | String | “2000” |

### Asset Images

Table 76: Asset Images Data Dictionary

This is the data dictionary of the asset images.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (Pk) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| imageUrl | String | [https://imageurl.com](https://imageurl.com/) |

### Attachments

Table 77: Attachments Data Dictionary

This is the data dictionary of the attachments.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (Pk) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| contentUrl | String | [https://attachments.com](https://attachments.com/) |
| fileName | String | Erohal |

### Category

Table 78: Category Data Dictionary

This is the data dictionary of the category.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (PK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| uuid | String | 1a2b31231wso |
| name | String | category1 |
| description | String | “category of the asset” |
| iconUrl | String | [https://iconurl.com](https://iconurl.com/) |
| tenantUUID | String | 1a2b31231wso |
| category\_id (FK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |

### Field Template

Table 79: Field Template Data Dictionary

This is the data dictionary of the field template.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (PK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| uuid | String | 1a2b31231wso |
| type | String | Primary |
| tenantUUID | String | 1a2b31231wso |
| isPrivate | Boolean | True |
| field\_template\_id (FK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |

### Field

Table 80: Field Data Dictionary

This is the data dictionary of the field.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Data Type** | **Example** |
| id (PK) | Long | Long(“578f6fa2df35c7fbdbaed8d3”) |
| uuid | String | 1a2b31231wso |
| label | String | vehicles |
| fieldMetadata | String | “type, name, model” |
| type | String | secondary |
| iconUrl | String | [https://iconurl.com](https://iconurl.com/) |
| fieldPosition | Int | 1 |
| isMandatory | Boolean | True |

### Company

Table 81: Company Data Dictionary

This is the data dictionary of the company table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (PK) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| name | String | Ali Iqbal |
| address | String | House: 123, Street: 1, Block: D |
| city | String | Lahore |
| country | String | Pakistan |
| email | String | [abc@gmail.com](mailto:abc@gmail.com) |
| website | String | [https://somename.com](https://somename.com/) |
| contact | String | 03001234567 |
| iconUrl | String | [https://somename.com](https://somename.com/) |
| status | String | current |
| currency | String | dollar |

### Company Field

Table 82: Company Field Data Dictionary

This is the data dictionary of the company field table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **Datatype** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| fieldTemplateId | String | 123 |
| fieldId | String | 123 |
| fieldValue | String | 123 |
| company\_id (FK) | Long | 123 |

### Employee Category

Table 83: Employee Category Data Dictionary

This is the data dictionary of employee category table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| name | String | Ali iqbal |
| company\_id (FK) | Long | 123 |

### User

Table 84: User Data Dictionary

This is the data dictionary of the user table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| name | String | Ali iqbal |
| imageUrl | String | [https://somename.com](https://somename.com/) |
| email | String | ali@gmail.com |
| organizationUUID | String | 129sf42mkn |
| firstName | String | Ali |
| lastName | String | Iqbal |
| contactNumber | String | 03001234567 |
| gender | String | m |
| country | String | Pakistan |
| city | String | Lahore |
| address | String | House: 123, Street: 1, Block: D |
| assigned | Boolean | true |
| userName | String | Ali028 |
| employee\_category\_id (FK) | Long | 123 |

### User Field

Table 85: User Field Data Dictionary

This is the data dictionary of the user field table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| fieldTemplateId | String | 123 |
| fieldId | String | 123 |
| fieldValue | String | 123 |
| user\_id (FK) | Long | 123 |

### User Preferences

Table 86: User Preferences Data Dictionary

This is the data dictionary of the user preferences table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| inspectonCreated | Int | 1 |
| faultyInspectionCreated | Int | 1 |
| issueCreated | Int | 1 |
| issueUpdated | Int | 1 |
| inspectionReminder | Int | 1 |
| Messages | Int | 1 |
| workOrderCreated | Int | 1 |
| workOrderUpdated | Int | 1 |
| assetAssignment | Int | 1 |
| assetUnassignment | Int | 1 |
| user\_id (FK) | Long | 123 |

### User Group Members

Table 87: User Group Members Data Dictionary

This is the data dictionary of the user group members table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| group\_id (Fk) | Long | 123 |
| user\_id (Fk) | Long | 123 |

### User Groups

Table 88: User Groups Data Dictionary

This is the data dictionary of the user groups table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| createdByUserUUID | String | 12h4g1375f |
| createdByUserName | String | Ali Iqbal |
| createdAt | Date | 02/02/2020 |
| groupName | String | Innovation labs |
| managerUserName | String | Ali028 |
| managerUUID | String | 91spke84nr |
| tenantUUID | String | 437hfc2oq |
| totalMembers | Int | 5 |

### Role

Table 89: Role Data Dictionary

This is the data dictionary of the role table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| name | String | Ali iqbal |
| displayName | String | Ali028 |
| description | String | This is a description |
| organizationUUID | String | 18734c02ld |
| iconUrl | String | [https://somename.com](https://somename.com/) |

### Permission

Table 90: Permission Data Dictionary

This is the data dictionary of the permission table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| name | String | Ali iqbal |
| description | String | This is a description |
| displayName | String | Ali028 |

### Notifications

Table 91: Notifications Data Dictionary

This is the data dictionary of the notifications table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| userUUID | String | 9834nfxd00 |
| username | String | Ali028 |
| title | String | Issue detected |
| message | String | This is a message |
| readStatus | Boolean | True |
| type | String | primary |
| notificationTime | Date | 02/02/2020 |
| assetUUID | String | N2c93470x08u |
| notificationFrom | String | Shark innovation labs |
| Removed | Boolean | False |

### Assignment Record

Table 92: Assignment Record Data Dictionary

This is the data dictionary of the assignment record table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| uuid | String | 1a287dcsjh7 |
| assetManagerId | String | 123 |
| assetId | String | 123 |
| startedAt | Date | 02/02/2020 |
| shiftStartedAt | Date | 02/02/2020 |
| shiftEndedAt | Date | 02/02/2020 |
| type | String | primary |

### Assignment Record Archive

Table 93: Assignment Record Archive Data Dictionary

This is the data dictionary of the assignment record archive table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| assetManagerId | String | 123 |
| assetId | String | 123 |
| startingDate | Date | 02/02/2020 |
| endingDate | Date | 02/02/2020 |
| shiftStartedAt | Date | 02/02/2020 |
| shiftEndedAt | Date | 02/02/2020 |
| type | String | primary |

### Reminder

Table 94: Reminder Data Dictionary

This is the data dictionary of the reminder table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| userUUID | String | 1394ncxp99u |
| assetUUID | String | 394qvytpp0w |
| reminderType | String | type |
| referenceTemplate | String | reference |
| lastSubmitted | Date | 02/02/2020 |
| reminderDays | Int | 29 |

### Firebase User Tokens

Table 95: Firebase User Tokens Data Dictionary

This is the data dictionary of the firebase user tokens table.

|  |  |  |
| --- | --- | --- |
| **Fields** | **DataType** | **Example** |
| id (Pk) | Long | 123 |
| userUUID | String | 945nco48cn02 |
| assetUUID | String | 2c948nvp409 |
| reminderType | String | type |
| referenceTemplate | String | reference |
| lastSubmitted | Date | 02/02/2020 |
| reminderDays | Int | 29 |

# 

# Risk Analysis

Predictive maintenance, even if it is fairly accurate, can pose certain different problems with some risks attached to them.

* The trained model can never be 100 percent accurate and will only be giving an estimated average lifetime remaining of a component. If the model predicts a component’s complete worn-out in X days, so, scheduling a replacement in X days would still be a gamble. Accurately predicting scheduling replacements still doesn’t completely eliminated the chances of an early breakdown of a component.
* Replacing a component too late may further damage more components and add to costly machine repairs and downtime, but changing it too early is like wasting the remaining lifespan of a component. So, finding a sweet spot for scheduling replacement is difficult [3].
* Although, the IOT devices come with inbuilt batteries that can be used as a power source in emergency, but can never be completely relied upon as their standby time would not be much given their low capacity.
* Malfunctioning of an IOT device can send inaccurate data to the server and can contribute in erroneous prediction of the trained model. For instance, gyroscope drift [4] can cause the readings to keep changing even when the device is still, in which case the device me need re-calibration of gyroscope.
* It is a possibility that there might be more service requests than normal limit which cannot be handled at a time.
* There is a possibility that internet might not be available in some areas which may add to the delay in data transmission to the servers.
* Service requests may not be entertained due to public holidays and scheduled replacements may be postponed to a further date, essentially risking the failure of a component.
* If the IOT device is not properly fastened to the vehicle, it can cause disconnection or damage to the device.

# 

# System Requirements

## Hardware Requirements

* Internet speed of at least 3Mbps required.

## Software Requirements

* Any browser that supports notification permissions.

# References

[1] Romain Bosquet, Pierre-Olivier Vandanjon, Alex Coiret, and Tristan Lorino, Model of High-Speed Train Energy Consumption, World Academy of Science, Engineering and Technology, 2013.

[2] Gopalakrishna Palem, Condition-Based Maintenance using Sensor Arrays and Telematics, International Journal of Mobile Network Communications & Telematics, 3(3):19-28, 2013.

[3] A. Desmet, "The three dangers of predictive maintenance models," 2017.

[4] S. Seshan and A. Seshan, "Using the Gyro Sensor," 2015.