An Asset Management System for Increased Efficiency and Accountability

Authors: Matthew Bolding, Joey Flores, Zyler Niece, Emma Sanders
Advisors: Dr. Krishna Kadiyala and Dr. Bingyang Wei

Project Goal

Provide an asset management system, suitable for a wide range of users, so that Chalk Mountain Services of Texas, LLC. (“the company”) may execute day-to-day business operations optimally and efficiently.

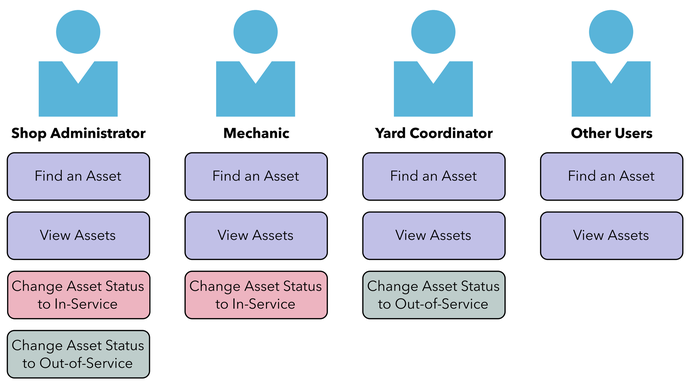
Problem Motivation

The company’s assets, composed of tractors and trailers, need to have the correct state assigned to them–in-service or out-of-service. The current solution–one that’s not mobile friendly–allows for any user to make modifications to any asset, the consequences of which cascades throughout the company; other systems rely on data maintained by this system. To remediate these problems, the new system:

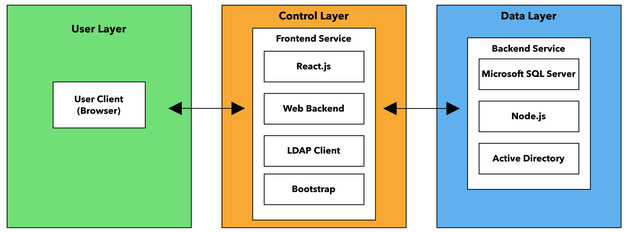
- implements role-based authentication;
- provides a mobile-friendly interface.

Through these changes, the company workflow is more efficient and each asset status change tags the appropriate user with the action.

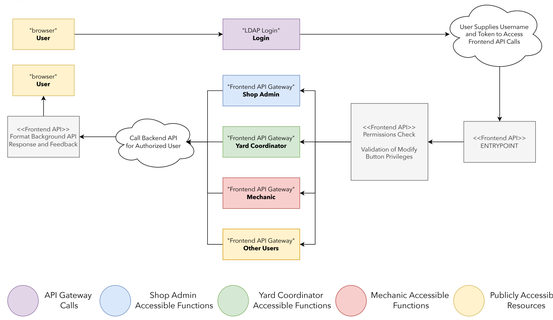
Use Case Diagram



System Architecture



Front End



Back End

Tables

Equipment Table	Request Table	Note Table
smallint UNITID	uniqueidentifier REQUEST_ID	uniqueidentifier NOTE_ID
nvarchar(15) UNITNUMBER	nvarchar(50) USER	nvarchar(15) UNITNUMBER
smallint COSTCTCODE	nvarchar(50) TIME	nvarchar(max) NOTES
nvarchar(20) LOCATION	nvarchar(15) UNITNUMBER	nvarchar(50) TIME
nvarchar(20) TYPE	bit STATUS	

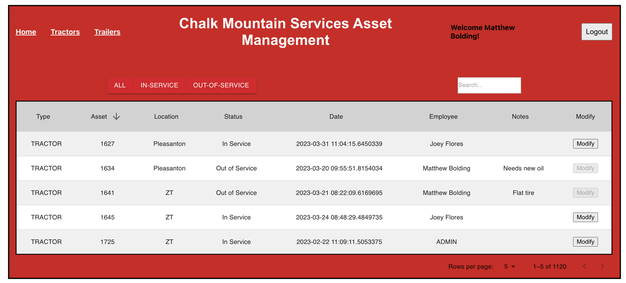
SelectedAssets

SelectedAssets
nvarchar(20) TYPE
nvarchar(15) UNITNUMBER
nvarchar(20) LOCATION
bit STATUS
nvarchar(max) NOTES
nvarchar(50) USER

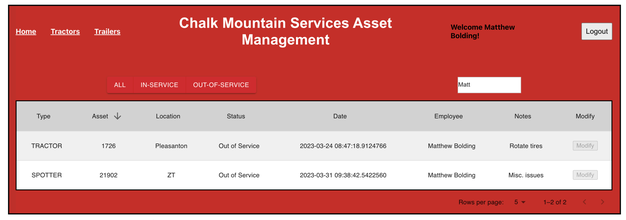
Stored Procedures

dbProcViewAssets	dbProcViewTrailers	dbProcViewTractors	dbProcGetAssetStatus	dbProcGetEquipmentCount
varchar(50) SortColumn	varchar(50) SortColumn	varchar(50) SortColumn	varchar(15) UNITNUMBER	**returns int**
varchar(4) SortOrder	varchar(4) SortOrder	varchar(4) SortOrder	**returns bit**	
int PageSize	int PageSize	int PageSize		
int PageNumber	int PageNumber	int PageNumber		
bit StatusBit	bit StatusBit	bit StatusBit		
varchar(128) SearchText	varchar(128) SearchText	varchar(128) SearchText		
returns List<SelectedAssets>	**returns List<SelectedAssets>**	**returns List<SelectedAssets>**		

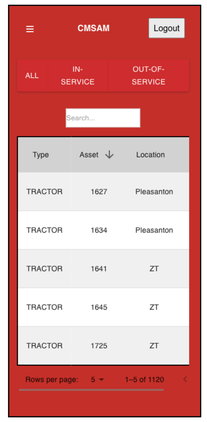
Screenshots



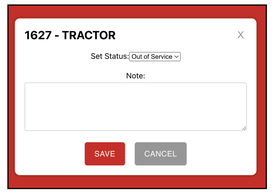
Role-Based Authentication



Search Functionality



Mobile-Friendly Interface



Modifying Asset Status

Sign in Page for Role-Based Access

Technologies Used

React, Microsoft SQL Server, Node.js

Challenges

- Configuring Lightweight Directory Access Protocol to interface with Active Directory.
- Constrained to working in a remote desktop environment.
- Implementing role-based access with security groups.
- Learning how to work with Microsoft SQL Server.

Acknowledgements

The team would like to thank the following individuals:

- **Dr. Krishna Kadiyala, Dr. Bingyang Wei, Dorian Dhamo:** for providing excellent feedback throughout the iterative design process, encouraging us to keep our sights on the highest priority goals, and answering any question we might have.