

DriverPass App
System Design Document (SDD)
Matthew Pool
Southern New Hampshire University
Version 0.1
08/15/2023

Table of Contents

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
 2. System Overview.....	 4
2.1 System Functionality (UX).....	4
2.2 User Functionality.....	7
2.3 UML Use Case Diagram.....	9
 3. Design Details.....	 10
3.1 Architectural Design.....	10
3.2 User Interface (UI).....	11
3.3 UML Class Diagram.....	12
3.4 UML Activity Diagram (Contact Us).....	13
3.5 UML Activity Diagram (Log In / Log Out).....	14
3.6 UML Sequence Diagram (Contact Us).....	15
 4. Requirements.....	 16
3.1 Technical Requirements.....	16
3.2 Nonfunctional Requirements.....	17

1. Introduction

The System Design Document (SDD) describes how the functional and nonfunctional requirements listed in the Business Requirements Document (BRD) for the DriverPass app transform into more technical system design specifications from which the system is built, showing the high-level system design and low-level detailed design specifications. The SDD provides a high-level overview of the system architecture and data structure of the DriverPass system, as well as user experience and interface design and operational scenarios. This is a brief (incomplete) SDD and only provides limited design detail for system components and development, as more detail will be created to provide a more comprehensive design. If approved, further elaboration on the SDD and related documents will be made.

1.1 Purpose

The DriverPass app has well-defined functional and nonfunctional requirements, including User Interface (UI) and User Experience (UX), as detailed in the DriverPass Business Requirements Document (BRD), also located at https://github.com/matthew-pool/systems_dev_SDLC, alongside this document. This document (SDD) provides greater technical detail into the design required for development and implementation of the DriverPass app, including technical diagrams using the Unified Modeling Language (UML). Classes will be shown in detail, including all the users of the system and other significant objects and entities, as well as some common processes, functionality, and use cases.

1.2. Scope

This SDD and BRD should provide enough information to understand the functionality and nonfunctional requirements of at least the first major production release of the DriverPass app. Further UI design details and low-level diagrams will be provided upon further iterations of the system. The DriverPass app will be a three-tier system, using a web/app server and a database server, alongside the user's web browser (desktop or mobile). All major browsers will be compatible and work with any of the following up-to-date operating platforms: Microsoft Windows, Apple macOS, Linux, Apple iOS, or Google Android. The DriverPass app must meet all requirements listed in the BRD and SDD by May 8th, 2023 (tentative deadline).

2. System Overview

2.1 System Functionality

- **Customer UX**

- *Register Account:* Upon clicking the “Sign-Up” link (at the top-right of all pages if user not logged in) to view the sign-up form, provide requested information consisting of first name, last name, address, city, state, zip, phone number, email address, username, and password (twice), and click the “Submit” button. A pop-up message will confirm account registration or notify user of any issues, including trying to register an existing username or failing to follow all password rules appropriately.
- *User Login:* A “Log In” (or “Log Out”) link will appear underneath the “Sign Up” link and upon clicking will display the login form. After entering a username and password, the typed password is hashed and compared to the hashed password in the database matching that username (primary key). A pop-up message will confirm a successful login or notify the user of any issues.
- *Input Form:* Click “Pick a Package” to open Input Form page and view the details of the available packages, click the appropriate package’s radio button, and click the desired date from the pop-up calendar, and click the “Submit” button. A pop-up message will confirm appointment or notify user of any issues.
- *View/Change Package:* Upon clicking “Appointments” at the top-left of the *Details* page, appointments will be shown, and the user can click any

appointment to see its details. The user can then click “Cancel Appointment” (which will display terms of cancelation and require a confirmation from the user), “Change Appointment” (confirmation required), or click “Done” to hide the information. If the appointment is canceled, confirmation of the cancelation will be sent to the user’s email address on file.

- *View Driver:* Clicking the driver photo on the *Details* page will pull up the driver’s first name, last name, phone number, and employee ID. A “Done” button can then be clicked to hide this information again.
- *Password Reset:* The Student Information section of the *Details* page will provide a “Password Reset” link that can be clicked to allow for creating a new password (to be confirmed twice) via a link to the customer’s email address or mobile phone. The password must comply with the previously stated password rules. The user will be redirected to the *Details* page, upon successfully resetting their password.
- *Contact Us:* Underneath the “Sign Up/Log In/Log Out” link, there will be a “Contact Us” link that the user can click, which will bring up the *Contact Us* page. The user must type in their preferred email address, name, and message, and then click “Submit”. A pop-up will confirm successful submission, and a “Done” button will be presented that the user can click to go back to the previous page.
- *View Details:* User may click any of the following links to view more details (“Done” button will be displayed to go back):

- test progress, driver notes, student information, special needs, driver photo, student photo, view statistics/graphs

- **Client (DriverPass) UX**

- **Admin Interface:** Once authorized via a username and password, administrators (users belonging to the “Admin” group) will be brought to an Admin page with various links and information:
 - *Manage Users:* Opens a page listing all usernames, each of which can be clicked to open the details of that user. Options include “Delete User”, “Modify Account”, “Block User”, “Contact Student”, “Reset Password”, and “Done”, in which an admin may change user account information or delete or block the account altogether. A pop-up confirmation will be required to process any changes.
 - *View Report:* Opens a page of various reports of data pulled from the database, including “User Statistics”, “Active Users”, “Accounting Data”, “Appointments”, and “User Activity”. Each has a “Print” button next to the report name to allow printing any of the reports.
 - *Modify System:* Includes advanced admin options for changing the code or layout of the system, removing, adding, or modifying packages, updating, and performing other various administrative tasks like S.E.O.

2.2 User Functionality

Users of the system must have the following functionality:

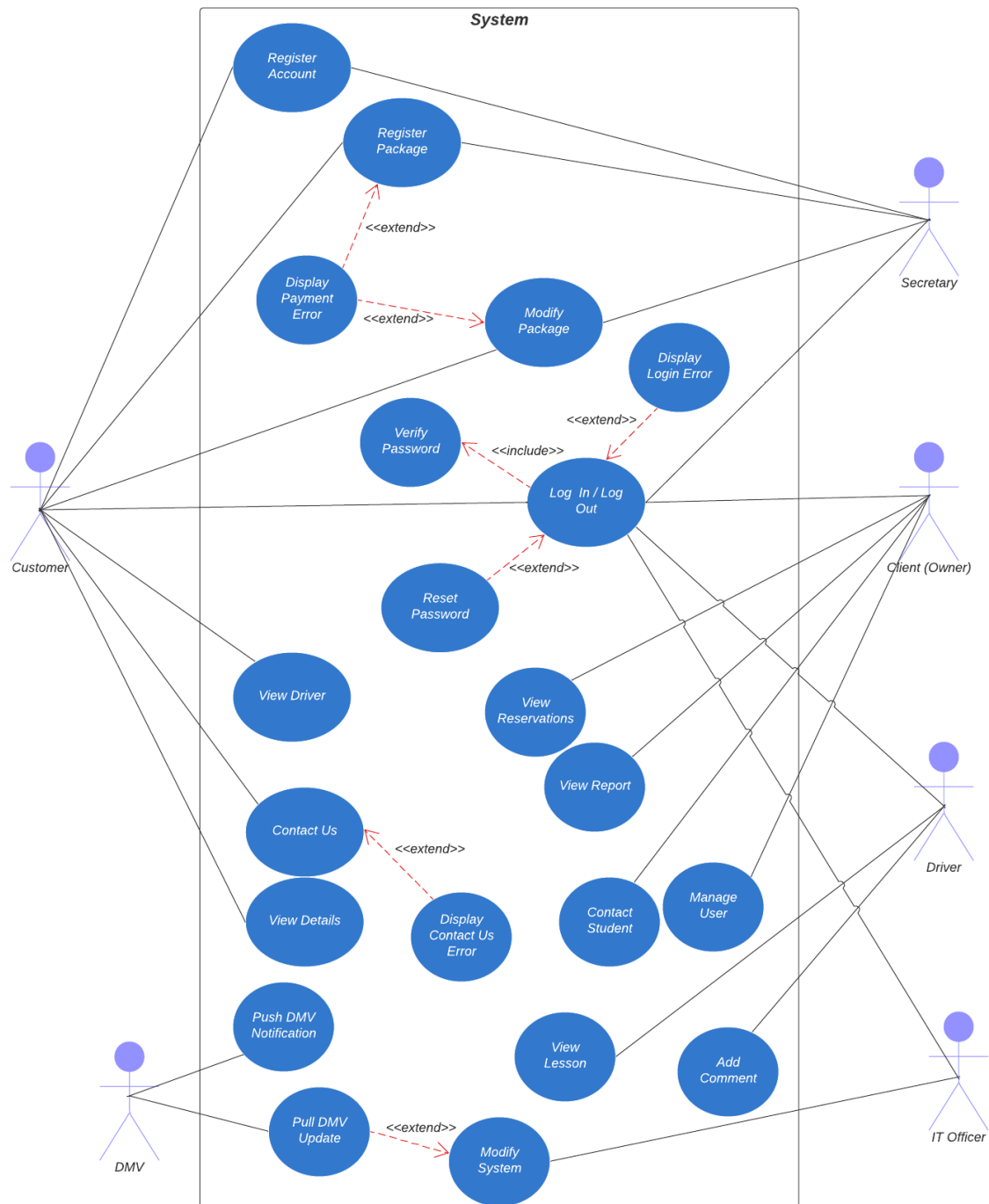
- **Customer Functionality**
 - register or log in to / log out of account
 - register for one of three packages, specifying date and time
 - cancel or modify appointments online
 - see driver's identification information (total of 10 cars with 1 driver each)
 - password reset online (for forgotten account password)
 - contact DriverPass
 - view online test progress, driver notes, student information, special needs, driver photo, and student photo
- **Secretary Functionality**
 - register or log in to / log out (password reset) of account, on behalf of Customer
 - Register or modify package ,on behalf of Customer
- **Client (Owner) Functionality**
 - log in to / log out of account
 - download reports and some information to work on at home
 - full access to all accounts to reset password or block user access from former employees
 - track who registers/cancels reservations and who modifies each last
 - print activity reports of user activity
 - contact student
- **IT Officer Functionality**

- log in to / log out of account
- modify system, including updating via pull request to DMV
- **DMV Functionality**
 - sends update notification to system when available

Additional system functionality:

- **Statistics & Graphs**
 - amount of people who completed on time
 - average test time
 - amount of people who pass/fail
 - mean/average grades of students

2.3 UML Use Case Diagram



3. Design Details

3.1 Architectural Design

Layer 1: Presentation Tier - Front End (Web Browser UI)

- *Input Form* page for customer (or secretary) registration, package selection, scheduling
- *Reservation* page to cancel, modify, view appointments online
- *Contact Us* page to contact DriverPass
- *Password Reset* page for forgotten passwords
- *Details* page showing driver and student information, test progress, special needs, photos
- *Admin Page*: client-access page for reports, user account controls, reservation tracking, modification, and IT Officer interface for maintaining and modifying the system

Layer 2: Application Tier - Web App / Server (Business Logic)

- DriverPass requires a web-based (cloud) system to cover security, backup, and other technical platforms
- Cloud server handles communications and processes information between the data layer and user's device
- Cloud provider serves web pages to customers and other users and handles web application logic

Layer 3: Data Tier – Database Storage Server

- *ACID-compliant SQL Server*: processes query commands related to stored data
- *Customer data*: first name, last name, address, phone number, email address
- *Driver data*: first name, last name, address, phone number, employee ID
- *Appointment data*: customer name, driver name, appointment date and time
- *Encrypted data*: login credentials, access permissions, billing information, security code
- *DMV data*: compliance rules, policies, sample questions

3.2 User Interface (UI)

- **Compatibility & Controls**

- HTML/CSS, such as buttons, text fields, radio buttons, etc., will be used to ensure compatibility with updates and modern platforms (Windows, macOS, Linux, iOS, or Android with Chrome, Safari, Firefox, Edge, or Opera)
- JavaScript will be utilized to allow for interactive controls and widgets, such as a pop-up calendar for scheduling appointments, dialogs, and other dynamic behavioral elements

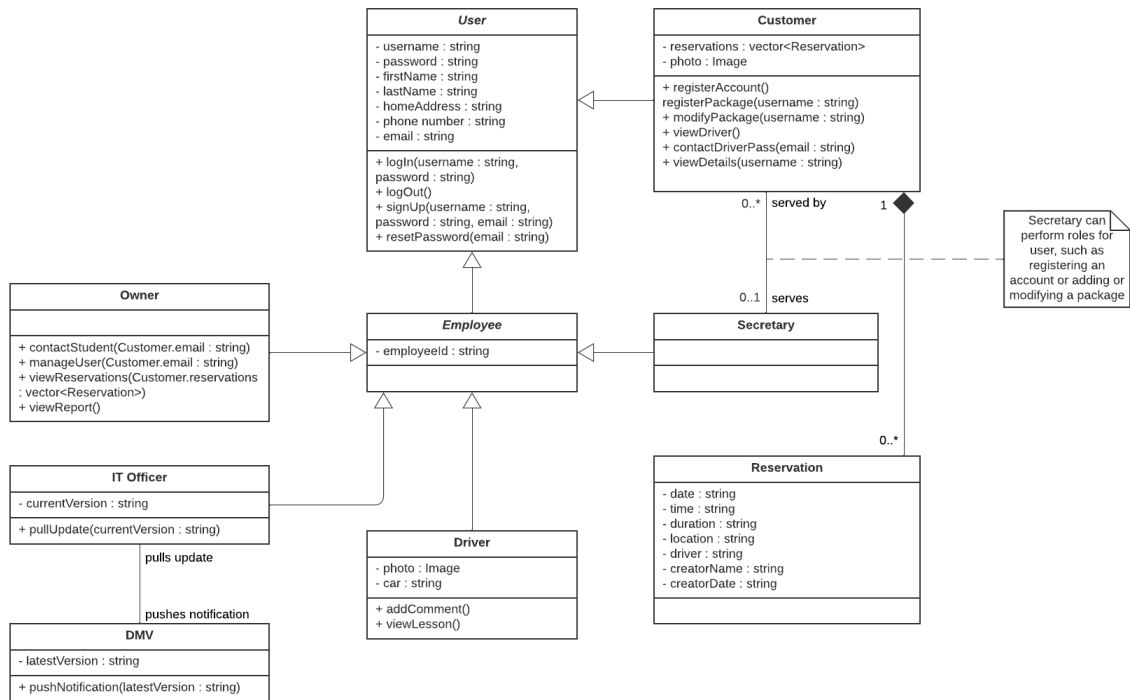
- **Design & Layout**

- *All Pages*: must include a centered company logo at the top of each page
- *User Interface*: Intuitive and consistent interface to be used by students (customers) or secretary and administrative users (IT Officer and owner)
- *Online Test Progress*: customer's available tests, tests in progress, completed tests
 - Fields: "Test Name", "Time Taken", "Score", "Status"
 - "Status" States: NOT TAKEN, IN PROGRESS, FAILED, PASSED
- *Driver Notes*:
 - Fields: "Lesson Time", "Start Hour", "End Hour", "Driver Comments"

Lesson Time	Start Hour	End Hour	Driver Comments

3.3 UML Class Diagram

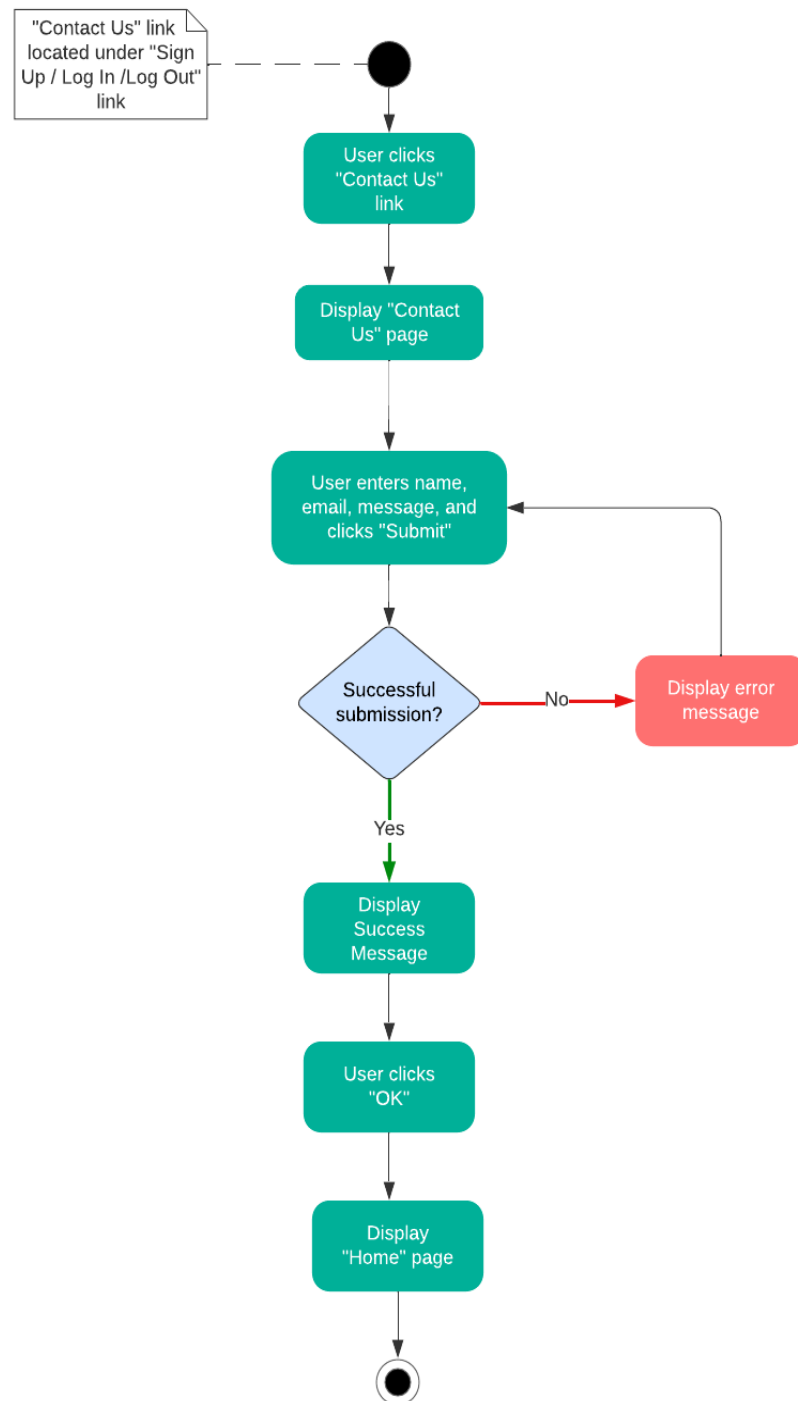
UML Class Diagram
Matthew Pool | August 10, 2023



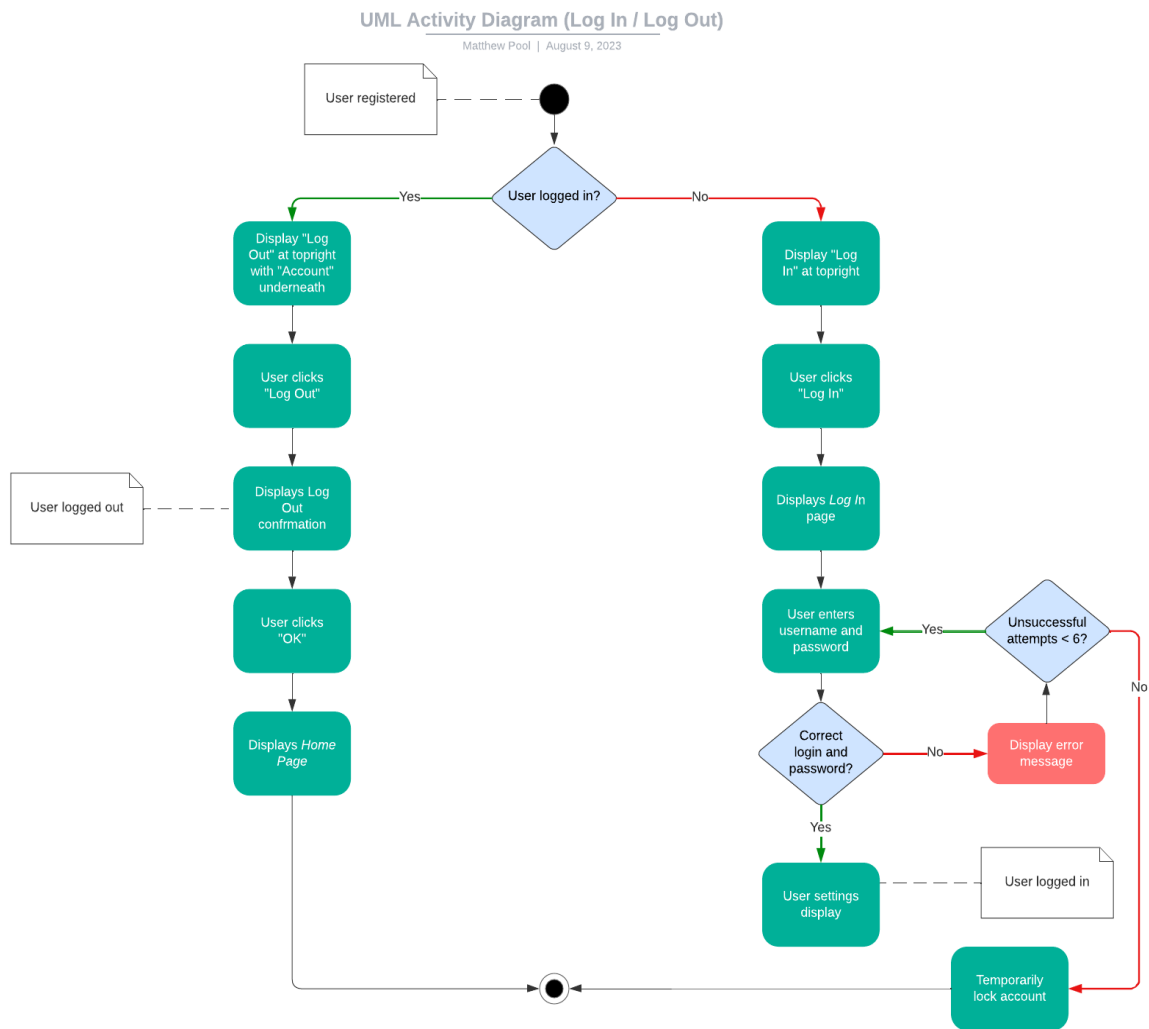
3.4 UML Activity Diagram (Contact Us)

UML Activity Diagram (Contact Us)

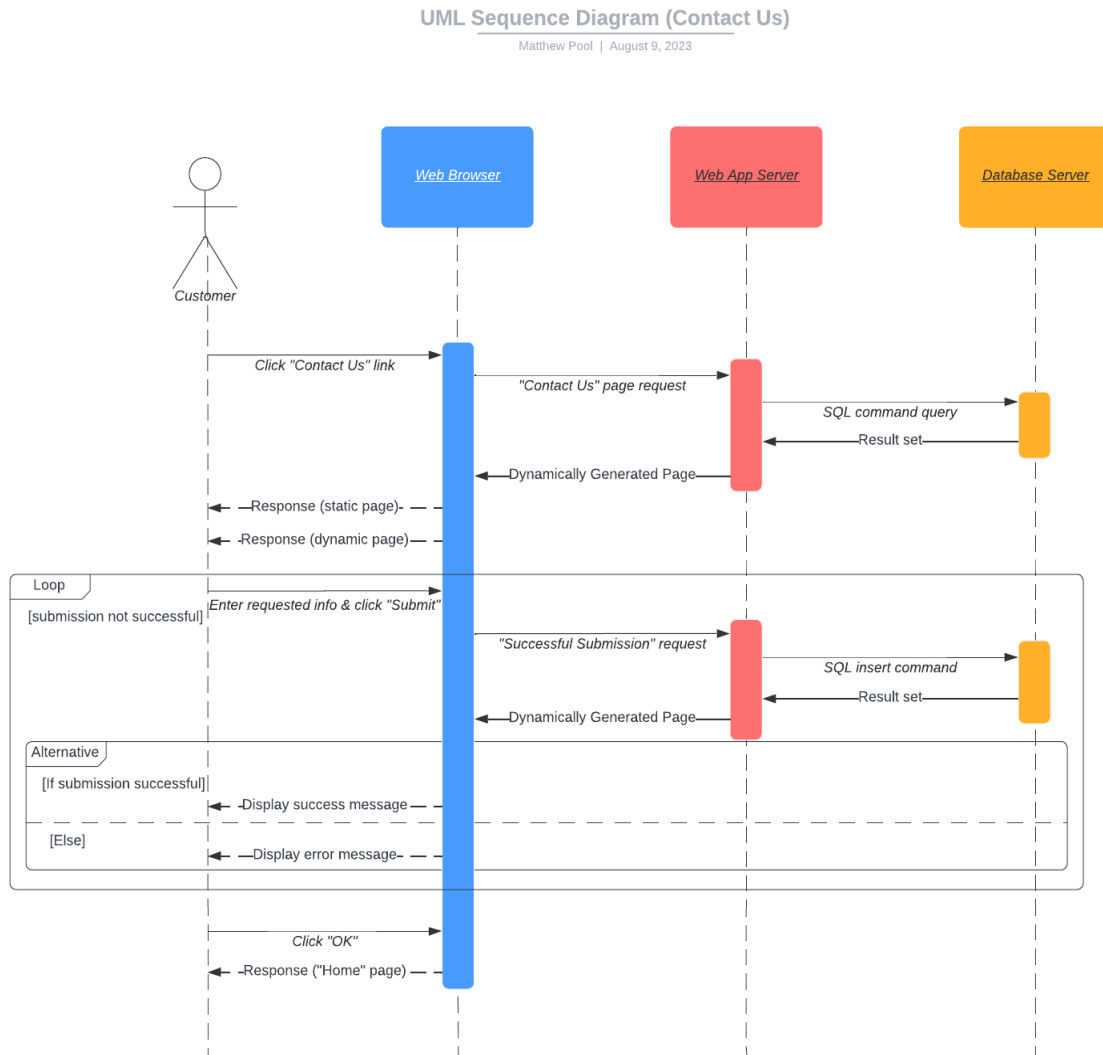
Matthew Pool | August 9, 2023



3.5 UML Activity Diagram (Log In / Log Out)



3.6 UML Sequence Diagram (Contact Us)



4. Requirements

4.1 Technical Requirements

- *Layer 1: Presentation Tier – Front End (Web Browser UI)*
 - client device must have an up-to-date desktop or mobile OS, such as Microsoft Windows, Apple macOS, Linux, Apple iOS, or Google Android
 - up-to-date web browser, including Google Chrome, Apple Safari, Mozilla Firefox, Microsoft Edge, or Opera, on user's device
 - standard keyboard and mouse, or touchscreen, input with standard monitor output
 - user should have a stable internet connection of at least 5 Mbps download and 1 Mbps upload with latency less than 100ms recommended for best experience
 - font size/type and site colors must follow DriverPass' predetermined and established brand format
- *Layer 2: Application Tier – Web / App Server (Business Logic)*
 - cloud-based provider platform to serve dynamic web page content as response to client's HTTP/HTTPS requests
 - server handles web application (business) logic and increases resources as demand grows
 - cloud provider must provide industry-standard security and backup services
 - cloud server handles communications and processes information between data layer and user (presentation) layer
- *Layer 3: Data Tier – Database Storage Server*
 - ACID-compliant SQL cloud-based server to process query commands related to data
 - database stores encrypted user information, system data, DMV data, and other information pertinent to the system
 - database server should utilize same cloud provider as application server to ensure consistency with performance and usability

TOOLS

- **Development:** HTML, CSS, JavaScript with Visual Studio Code IDE, running on either Microsoft Windows, Apple macOS, or Linux desktop platform, running up-to-date and robust security software

SCALABILITY

- CDNs and load balancers can be implemented if scaling the app out to a larger geographical proximity or if number of users increases significantly

4.2 Nonfunctional Requirements

Performance Requirements

- System should run as a web-based application accessible via desktop or mobile devices
- Response time for loading pages and processing user actions should be less than 2 seconds
- Appointment availability should be updated in “real-time” to show the latest driving session availability

Platform Constraints

- System should run on all major desktop and mobile platforms, including Windows, macOS, Linux, iOS, and Android
- System should be accessible on all major web browsers, including Chrome, Safari, Firefox, Edge, and Opera
- Back-end requires a cloud-based database to store customer information, driver details, appointments, and other data securely
- Back-end requires a web/app server to serve pages to user’s browser/device

Accuracy and Precision

- User accounts will require unique usernames and passwords with rules, including mixed capitalization, at least one special character and numeric digit, and must be at least 8 characters long.
- Authenticated users will have access to different functionality based on associated authorization rules
- The username input will not be case-sensitive, but the password input will be case sensitive to allow for greater security
- The system should notify the admin of any critical issues, such as security breaches or technical issues like server crashes or other relevant concerns

Adaptability

- System should allow administrative users to add, remove, or modify user accounts without changing underlying code
- System should be designed with a modular structure to accommodate platform updates and future changes, without causing severe downtime
- IT admin should have full access to system settings, user accounts, and database access
- Future updates will allow customized packages with functionality to add or remove modules by the developer or systems analyst
- DMV connected to update new compliance rules, policies, and sample questions and to trigger notification for the client
- System will be automatically backed up on a regular basis to a secondary cloud-storage to allow full system recoverability in the chance of system data loss or failure

Security

- User will need to use their associated username and password, which must consist of mixed capitalization, at least one special character and numeric digit, and be at least 8 characters long
- Typed passwords will be hashed (and salted) and compared to the stored hashed password in the database
- Forgotten passwords can be reset by clicking the “Forgot Password” link underneath the password entry textbox
- Connection between client and server should utilize SSL/TLS encryption to protect data exchange via HTTP/HTTPS, respectively
- After 6 unsuccessful login attempts, the account will be locked, and a password reset will be required, to prevent potential hacking attacks like brute force
- Advanced Encryption Standard (AES) will be used to encrypt the database data at rest