# CS 255 1-3 Assignment: Collecting Customer Requirements

Michael Wood

01/12/2025

Instructor: Christopher Eppenger

## System Components and Design

### Purpose

*What is the purpose of this project? Who is the client and what do they want their system to be able to do?*

* DriverPass hopes to capitalize on a market void by teaching students for their driving exam at their local department of motor vehicles (DMV).
* By offering online courses, practice exams, and on-road instruction, Liam, the proprietor of DriverPass, hopes to improve driving instruction.

### System Background

*What does DriverPass want the system to do? What is the problem they want to fix? What are the different components needed for this system?*

* Liam wants access to data from anywhere including any computer or mobile device and be able to download info and reports.
* Ian, the IT officer, has administrative access to grant complete authority over accounts if a user's password needs to be changed and to prevent access if someone is fired.
* Monitoring of made, canceled, and updated reservations. The owner of DriverPass has access to tracking information and can print activity reports.
* The ability to schedule two-hour driving lessons. Reservations for the time of day and date should be possible for users via their account. Keep track of the time, automobile, and user that is paired with a driver.
* Make reservations for three packages. It would be nice to have the ability to add and remove packages in the future. the ability to disable packages if no users sign up.
* Online appointment scheduling, cancellation, and modification are available to users.
* When registering, users enter their first and last names, address, phone number, state, credit card number, security code, and expiration date. Add the locations for pickup and drop-off. If a user forgets their password, it can be immediately reset.
* Current modifications with DMV. To give updated updates for new laws, policies, or sample questions, DriverPass is connected to the DMV.
* A cloud-based web interface.
* Online test progress is part of the web interface. Displays what the customer is working on, what needs to be done and what has been finished.

### Objectives and Goals

*What should this system be able to do when it is completed? What measurable tasks need to be included in the system design to achieve this?*

* After completing DriverPass, clients will be able to select from three driving packages:

1. Package 1: Driving with a trainer for six hours
2. Package 2: Includes an in-person lesson where we go over DMV policies and procedures and eight hours in a car with a trainer.
3. Package 3: Includes a twelve-hour vehicle ride with a trainer, an in-person instruction session where we go over DMV regulations and policies, and access to our online course with all the information and resources. There are also practice exams in the online course.

* Each driving session lasts two hours; the sessions will be spaced out throughout time in two-hour blocks. Appointments may be altered, canceled, or modified by customers.
* The DriverPass interface will display the customer's online progress. Name, time spent, score, and status are all included.
* Owner Liam, IT Officer Ian, and the secretary all have administrative access.
* Owner has all access
* Ian has access to accounts, maintenance and system modifications.
* Secretary has access to clients and calendar for appointment and contact purposes.

## Requirements

### Nonfunctional Requirements

*In this section, you will detail the different nonfunctional requirements for the DriverPass system. You will need to think about the different things that the system needs to function properly.*

#### Performance Requirements

*What environments (web-based, application, etc.) does this system need to run in? How fast should the system run? How often should the system be updated?*

* To provide speedy performance and real-time updates, the system should run in a web-based, cloud environment. In addition to avoiding duplicate appointments, users should be able to easily check available appointments and teachers. Regular system updates are required, and test results should be displayed as soon as they are done. In subsequent upgrades, DriverPass must be able to adjust the driving packages and guarantee that any modifications are implemented as soon as possible to remain in accordance with DMV requirements.

#### Platform Constraints

*What platforms (Windows, Unix, etc.) should the system run on? Does the back end require any tools, such as a database, to support this application?*

* To ensure that a broad spectrum of people may utilize the system, it should be built to run on several platforms, such as Windows, macOS, and Unix-based systems. It should work with mobile devices like iOS and Android as well as web-based settings for even more versatility.
* A dependable database is required for the back end to store and manage user data, appointment schedules, test results, and payment details. A development framework and a web server are a couple of additional components that will support and assure the system's operation.

#### Accuracy and Precision

*How will you distinguish between different users?* *Is the input case-sensitive? When should the system inform the admin of a problem?*

* The system will need unique login credentials to differentiate amongst the various user types—students, teachers, and administrators. These credentials may include passwords and email addresses or usernames. The system will balance strictness and convenience by making passwords case-sensitive but not usernames or emails. This will increase security and usability.
* Any problems, including schedule conflicts, unsuccessful payment attempts, or inaccurate user data updates, should be immediately reported to the administrator by the system. Automated alerts or notifications will enable administrators to promptly address issues, guaranteeing user satisfaction and seamless system operations.

#### Adaptability

*Can you make changes to the user (add/remove/modify) without changing code? How will the system adapt to platform updates? What type of access does the IT admin need?*

* Without changing the code, users can be added, removed, or modified via a user management interface. This makes it possible for administrators to immediately alter the system, guaranteeing its adaptability and usability.
* Version control and modular development methods should be used to ensure that the system is designed to adapt to platform updates. Frequent updates and maintenance will prevent interruptions and guarantee compatibility with new platform versions.
* Complete access to the backend system is necessary for the IT administrator to update the database, manage user accounts, and troubleshoot issues. Administrator access is necessary to apply updates, track system performance, and resolve technical problems.

#### Security

*What is required for the user to log in? How can you secure the connection or the data exchange between the client and the server? What should happen to the account if there is a “brute force” hacking attempt? What happens if the user forgets their password?*

* Users need to enter a secure password and a unique username or email address to log in. An additional layer of account security can be added by using multi-factor authentication (MFA).
* The client-server connection and data transmission should be secured with HTTPS and SSL/TLS encryption. To protect sensitive information, data stored in databases should be encrypted.
* A secure password recovery option, like sending a reset link to the user's registered email address, should be provided by the system in case the user forgets their password. Verification steps should be included in the reset process to ensure the user's identity.
* After several unsuccessful attempts to log in, the system should temporarily freeze the account in the event of a brute-force hacking attempt and alert the administrator and user to any unusual activity. Automated attacks can also be avoided via CAPTCHAs.

### Functional Requirements

*Using the information from the scenario, think about the different functions the system needs to provide. Each of your bullets should start with “The system shall . . .” For example, one functional requirement might be, “The system shall validate user credentials when logging in.”*

* The system shall validate user login credentials.
* The system shall allow scheduling of driving lessons without conflict.
* The system shall provide three driving package options.
* The system shall allow instructors to provide feedback on student driving.
* The system shall show test history and status (passed, failed, in progress).
* The system shall let users update personal and payment information with tracking.
* The system shall show real-time instructor and car availability.
* The system shall display test scores immediately after completion.
* The system shall notify users of scheduling issues.
* The system shall connect to the local DMV for compliance updates.
* The system shall allow admin remote access to data.

### User Interface

*What are the needs of the interface? Who are the different users for this interface? What will each user need to be able to do through the interface? How will the user interact with the interface (mobile, browser, etc.)?*

* The interface must be user-friendly, responsive, and accessible via mobile devices and web browsers. Students, professors, and administrators are examples of many user categories. Students should be able to register, take practice tests, schedule classes, review their scores, and update their personal information. Instructors must manage schedules, provide feedback, and change availability. Administrators will oversee monitoring user behavior, resolving scheduling issues, and updating driver packages.
* Users will use web browsers and mobile apps to interact with the interface; on computers, they will use traditional navigation, while on mobile devices, they will use touch gestures. All users should be able to easily access the required features thanks to a simple design that can be adjusted to fit various screen sizes.

### Assumptions

*What things were not specifically addressed in your design above? What assumptions are you making in your design about the users or the technology they have?*

* Some components, such as comprehensive safety measures, offline functionality, and user analytics for tracking user behavior, were not explicitly integrated into the above design. Furthermore, there was no emphasis on specific accessibility elements for people with disabilities.
* To utilize the web-based system, users must have a reliable internet connection and be familiar with web browsers and mobile applications. Users are assumed to have basic technical knowledge and be able to use an app or website without much assistance. Furthermore, the design assumes that the cloud infrastructure and associated technology will be reliable and secure enough to handle real-time updates and data storage.

### Limitations

*Any system you build will naturally have limitations. What limitations do you see in your system design? What limitations do you have as far as resources, time, budget, or technology?*

* There are certain difficulties with the system design. The dependence on a steady internet connection is a significant drawback, which may cause issues for users in places with inadequate access. The system's capacity to manage many users concurrently is another issue, as this could cause performance issues. Continuous updates and maintenance will also be necessary to guarantee continued security and adjust to emerging threats.
* Furthermore, there are resource constraints, such as financial, time, and technological ones. Tight timelines may make it more difficult to improve the system or provide adequate user assistance, while low budgets may prevent the inclusion of sophisticated features. Some users may struggle to use the system effectively due to limited training choices and compatibility concerns with older devices. The expectation is that the cloud infrastructure and related technology will be reliable and secure enough to handle real-time updates and data storage.

### Gantt Chart

*Please include a screenshot of the GANTT chart that you created with Lucidchart. Be sure to check that it meets the plan described by the characters in the interview.*

A screenshot of a computer

AI-generated content may be incorrect.