# CS 255 Business Requirements Document Template

## Version 1.0: System Components and Design

## 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?*

* The client, DriverPass, is an educational service that offers various packages for the licensing exam administered at DMVs. They would like to offer the ability to online classes, practice tests, and on-the-road training.

### 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?*

* The owner of DriverPasss noticed first-time takers of the licensing exam administered at DMVs often fail. Thus, there must be an opportunity to offer more comprehensive training to new drivers.
* DriverPass wants to offer a cloud-based online learning environment that offers three comprehensive learning packages.
* Based on tier, users can schedule multiple 2-hour driving appointments and have access to online courses and practice tests.
* The System should have the following users:
  + Administrator – Full access and ability to control who sees what content
  + Secretary – Able to make and adjust appointments for students.
  + User (Student) – Can make and adjust appointments and track progress.
  + Drivers – Can log notes based on driving lessons with users.

### 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?*

* What the system will be able to do?
  + Automate Scheduling
  + Track users’ statuses.
  + Track drivers and cars available for driving lessons.
  + Notify the administrator when DMV regulations change.
* Measurable Tasks:
  + Scheduling services for driving lessons
  + Online classes
  + Online Tests/Quizzes
  + Online signatures showing all changes made, by whom, and when.
  + The ability to schedule driving appointments online or via phone (with secretary).
  + The ability to buy packages online or through the phone (with secretary).

## 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?*

* The system needs to run on various systems. It should be accessible from desktops/laptops, phones, tablets, etc., ensuring access to a broad customer base.
* The system should run as fast as possible, as most websites are abandoned if they are unable to load in 1-2 seconds. Data shows that 87% of websites are abandoned after 3 seconds (Anderson, 2024).
* System Updates:
  + Security and System: The system must be updated regularly to handle new security threats and ensure functionality.
  + Compliance Updates: DriverPass wishes to stay up-to-date on DMV rules and regulation changes.
  + Feature Updates: DriverPass wishes the ability to update packages. The code to do this should be well commented and easily maintained to facilitate potential changes.

#### 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?*

* The system should be developed on a cloud-based server. This will remove the responsibility of maintaining a server and allow frameworks like microservices to be implemented.
* Back End Tools:
  + Databases—A relational database should be used to store customer information, such as name, address, payment information, username, password, and tier purchased.
  + API – An API that can handle HTTP requests must be implemented.
  + Authentication – a service must be bought or built to handle user authentication.
  + Cloud Platform – A service to host the application.

#### 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?*

* Usernames/Emails are typically not case-sensitive. However, passwords shall be case-sensitive for more secure access.
* The System shall inform the admin of the following problems:
  + Security Breaches – If there is attempted access to accounts outside of the standard threshold (3 failed logins) or if there is unusual/suspicious accessing of secure information.
  + Failures – If the system or any components suffer a failure of some type, such as denial of access.
  + Maintenance – The admin shall be notified if the system undergoes expected/unexpected maintenance.
* The system will implement role-based access based on the theory of least access.
* The Roles shall include:
  + Admin – Full access and the ability to create and delete roles.
  + IT officer – Access to technical settings and system maintenance.
  + Secretary – Access to user information, ability to schedule, modify, and cancel appointments for all users.
  + User – Access to personal information and class progress. They can schedule, modify, or cancel appointments for themself.

#### 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?*

* The Admin and Secretary shall be able to make changes to a user (add/remove/modify) without modifications to the code.
* The system shall run platform-independently and notify the admin if the cloud server has expected maintenance.
* The IT officer will need access to:
  + System Configurations – Including configurations for authentication and performance parameters.
  + Maintenance – The ability to implement system updates and maintenance to fix potential errors or update the system.
  + System Logs – To identify and adjust any system failures or redundancies.

#### 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?*

* The user shall have a unique name and password to access the system.
* Securing the Data Exchange:
  + Cloud-based servers offer robust encryption that will help secure client-server communication.
  + Role-based access controls can limit the amount of damage done by hacked accounts.
  + Require strong passwords.
* Accounts shall be locked after three failed attempts and require 15 minutes or admin intervention before accounts are unlocked.
* CAPTCHA can help combat automated attacks.
* If the user forgets the password, the user shall be sent a password reset link to either mobile or email.

### 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 allow the purchase and access to one of three tiers.
* The system shall allow the user or secretary to schedule a two-hour driving lesson and track when/who appointments are adjusted.
* The system shall display the user's progress in courses.
* The system shall allow the administrator to download read-only reports for offline/remote work.
* The system shall require a unique username and password to access content.
* The system shall allow the disablement of specific tiers.
* The system shall encrypt sensitive data.
* The system shall process payments and monitor the amount owed.

### 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.)?*

* Needs of Interface:
  + User-friendly design
  + Able to function on multiple devices (PC, mobile, tablet, etc…)
  + Secure login/authentication process
  + Compliant with accessibility standards
* Users:
  + Administrator
    - * Manage roles and permissions
      * Generate, view, and download reports
      * Disable package tiers
* IT Officer
  + - Maintain and update system
    - Monitor performance and security
* Secretary
  + - Schedule, modify, and cancel appointments
    - Process payments on the customer’s behalf
* User
  + - Schedule, modify, and cancel appointments
    - Complete online classes/quizzes
    - Process payments
* Driver
* Make notes based on the performance of students during driving lessons.
* View and manage scheduled driving lessons
* Update availability

### 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?*

* The user has internet access.
* Both users and admins have a level of familiarity with similar online systems.
* DriverPass will use a pay-per-use cloud service.

### 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?*

* Lesson scheduling depends on users’ geographical location. Travel time between lessons and state-specific DMV regulations must be adhered to.
* Adding tiers requires developer intervention. DriverPass wishes to be able to add new tiers as the program evolves; however, this will require a knowledgeable developer’s intervention.
* Cyber threats are ever-evolving; thus, the system must be constantly monitored for potential weaknesses.
* The system must be robust and scalable to offer real-time updates regarding progress and schedule updates.

### 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

Description automatically generated