# CS 255 Business Requirements Document Template

## 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?*

* This project is for one of our clients, DriverPass, a student driver’s information and services company, their purpose being to make a web-based system that offers booking, rescheduling, and cancellation of student driver sessions in what they claim is a void of such material. They want their system to provide access to a library of materials that include videos, available online and for download to view offline.

### 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?*

* DriverPass noticed that too many people were failing their driving tests with the DMV and believe that these failures are a sign of poor education and training.
* DriverPass looks to offer better and more easily accessible as well as flexible education and training to young drivers as a solution to this issue
* This education requires a system that allows people to purchase and schedule classes on their own time.
* A cloud provider or a webserver is required for this system to store data online and offsite from their company.

### 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?*

* system as a whole should receive updates from the DMV with the latest materials required.
* The system as a whole should provide end users with a way to automatically reset their passwords.
* The system should provide management with the following functions
  + Add new administrators
  + Remove administrators
  + Turn off packages no longer offered
  + Reset passwords for administrators
  + Disable packages no longer desired
* The system should provide administrator users with the following functions
  + Reset user passwords
  + Disable packages no longer desired
  + Generate and print activity reports on individual accounts
  + Schedule training (or re-schedule)
  + Cancel training
  + Update online materials
* The system should provide end users with the following functions
  + Purchase a package
  + Schedule training (or re-schedule)
  + Cancel training
  + Access online materials including practice test history

## 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?*

* This project will be web-based and must be compatible with all major internet browsers.
* The application must operate on any operating system, including Mac, Windows, and Linux, through the most popular browsers.
* Response times should be under 30 seconds.
* The system should be updated as needed, particularly when the DMV provides new materials.

Rationale: The client requested an online booking system and online materials for students who purchase packages. To achieve this, developing a web-based project, such as a website or web app, is the simplest solution. This approach avoids the need for installation on various operating systems, ensuring compatibility with all standard OS types, as the end users' OS cannot be predetermined. A maximum response time of 30 seconds is reasonable due to the complexity of some systems, though faster responses are preferable. Updates should be implemented as required, with mandatory updates occurring when new information is released by the DMV.

#### 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 run on a Windows operating system.
* The back end requires at least one database.
* Ideally, this database will store course materials, user tables and profiles, relevant DMV data, billing information, and appointment schedules.
* If managing everything in a single database becomes too complex, separate databases can be created for different item types, such as one for user tables and profiles and another for course materials and test materials.

Rationale: The system needs to support many users simultaneously and will operate as a server-client setup. The website will host a large amount of content, including online classes, materials, user login information, and billing details for paying users. Keeping all this information on the front end is impractical and poses significant security risks. Therefore, at least one database is essential for managing this data securely and efficiently.

#### 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 distinguish between different types of users by unique user ID numbers, login names, and passwords.
* The ID number should be assigned by the system and should not be changeable, but the login name and password should be set by the user and changeable if needed.
* The input for login name (username) should not be case-sensitive, but the input for the password shall be case-sensitive.
* In the event of a problem with the system, an IT admin should receive an email notification within 5 minutes of the event.

Rationale: Two different major types of users are needed, admin and end users. Therefore, the system needs to be able to distinguish and offer the functionality associated with the account type to the user logging in. The ID number should be assigned by the system to make it easily searchable and to avoid any duplicates. The username should be unique and non-case-sensitive to avoid mix-ups like someone having username JaneJones and another having the username janejones. Passwords should be case-sensitive for added security. A five-minute response time in the event of a problem is reasonable if the problem has not resolved itself. The IT admins don’t need an email every time the internet fluctuates for a minute or two or the server goes down and boots right back up.

#### 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?*

* Here’s a reiterated version of your text:
* The system should enable admin users to add, remove, and modify users without altering the source code.
* The system must adapt to Microsoft updates, with the server capable of downloading and installing these updates without issues.
* The IT admin should have full administrative access, including visibility of all user and administrator accounts. They should also have the ability to hide course packages offered by the company, but they must not have read or written access to the source code.
* Rationale: Admin users need the capability to manage user accounts, whether for end users or other admins, to handle situations like customers discontinuing their accounts or admin staff changes. Relying on developers for each account modification would be impractical. The system should seamlessly handle Microsoft updates, requiring a system reboot to apply them. The client also specified that admin users should be able to hide inactive packages from customer view without accessing the source code, preventing unauthorized changes that could potentially crash the system.

#### 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, whether admin or end users, must log in on the website by providing their username and password.
* Secure connections are established using SSL encryption, which employs a TLS handshake to generate session keys that encrypt all data within the session.
* To protect against brute force hacking attempts, the system should temporarily lock an end user out for 10 minutes after five unsuccessful login attempts. For admin users, the lockout should occur after three unsuccessful attempts.
* If either user type has 10 or more failed login attempts, their account should remain locked until a company representative unlocks it, requiring the user to call in and validate their identity.
* If a user forgets their password, they should have access to a "forgot password" link that allows them to reset it. This link should prompt them to provide their username and the email address linked to their account to receive a reset link.
* Rationale: Requiring a username and password to log in is the simplest and most effective way to validate users. While two-factor authentication is an option, it may not be necessary for this project. SSL encryption is widely understood and used, providing secure connections by encrypting data during the session, which ensures user confidence and protects company data from unauthorized access. Brute force attacks can be mitigated with temporary lockouts and prevented by locking the account after multiple failed attempts, requiring manual unlocking. This approach makes brute force attacks impractical.

### 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 use login credentials to determine the user ID and information.
* The system shall offer the following options for end users: review packages, purchase a package, schedule/cancel/modify in-person training, schedule/cancel/modify behind-the-wheel training, review progress/profile, and log out.
* Based on the subscribed package, end users will have additional options: access online classes, take practice tests, review online content, and download content for offline work.
* The system shall identify the role of a logged-in user.
* The system shall display different options for admin users, including: add/modify student profiles, add/modify admin profiles, delete users, reset user passwords, validate users with security questions, schedule/cancel/modify in-person appointments, schedule/cancel/modify behind-the-wheel training, print activity reports, download content for offline viewing, and disable purchasable packages.
* The system shall email receipts to customers when they purchase packages.
* The system shall notify instructors via email when a session is booked for training or behind-the-wheel instruction.
* The system shall validate users attempting to access restricted areas of the site, such as online classes restricted by user level.
* The system shall provide technical reports to admin users upon request via their on-screen menu.
* Rationale: The first step is validating the login to correctly identify the user, which determines the display throughout the system. Once logged in, the system distinguishes between end users and admin users. For end users, the system must display the specified features, further determined by the purchased package, based on DriverPass specifications. Admin users need the outlined functionalities to perform tasks like resetting passwords without developer assistance and disabling unavailable packages. Confirmation emails for booked sessions ensure clear communication between instructors and users. Validating users accessing certain site areas prevents unauthorized access for security reasons. Finally, providing reports to admin users allows them to track account activities, resolve disputes, and evaluate the software's effectiveness.

### 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.)?*

* Many possible physical user interfaces exist for this system. All share at least some features in common.
  + The user interface shall have a display screen.
  + The user interface shall have a keyboard or simulated keyboard.
  + The user interface shall have a mouse or touchpad.
* The users will interact with the interface using a browser, whether a mobile browser or computer (desktop/laptop) browser.
* The different users for this interface are customers (or end users) and administrators (or admins).
* The user interface for customers consists of basic messages including:
  + Enter Username
  + Enter Password
  + Select Package
  + Enter Payment Information
  + View/Edit Profile
  + Change Password
  + Schedule/Cancel/Modify Appointment
  + Access Online Training Materials
  + Take Practice Tests
  + Download Content
* The user interface for admins consists of basic messages including:
  + Add/Modify/Delete User
  + Change Password
  + Print Reports
  + Hide Packages
  + Schedule/Cancel/Modify Appointment
  + Download Content

### 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?*

* All users have reliable, high-speed internet access.
* Users have computers or phones equipped with proper input devices (keyboard and mouse or touchscreen).
* The server providers can support the bandwidth needed for any number of users accessing the service simultaneously.
* Users have sufficient electricity or battery power for their needs on the system.
* All users are using up-to-date web browsers to access the content.

* The reports requested are the only essential reports in the current design.
* Rationale: These assumptions are made for specific reasons. Reliable internet access and electricity are fundamental for any online system to function. The assumption about server provider bandwidth is based on the lack of information regarding expected user numbers and server load in discussions. The web browser assumption acknowledges that developers should not be expected to support obsolete browsers like Netscape Navigator or AOL, which are rarely used today. The assumption about reports stems from the necessity for the client to specify their needs; additional reports will not be generated unless requested explicitly.

### 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?*

* The system cannot function without power.
* The system cannot function usefully without internet access.
* The system relies on Microsoft Windows. Therefore, if a major problem affecting all Windows servers were to come up, the system could be compromised until a patch becomes available.
* The timeline is short for producing the system which could impact desired functionality.
* The system would have limited real-time notification ability because it is browser-based. If a user is not logged in and active, they may miss a cancelled appointment or something similar.
* The system has no way to prevent unauthorized republication of course materials since the client requested download access for said materials.
* Rationale: The rationale for each of these is different, except the power and internet bullet points. The power and internet limitations are for any web-based system. A web-based system needs power and internet access at the server side and client side in order to function as designed. The Windows limitation is because Microsoft is a third party whose products we are relying on. If their product fails, so does ours. The timeline limitation is because it is less than five months. That seems like a quick turnaround for a system with so much going on. The real-time limitation is common to many web applications too since you must be in an application to get the notifications from it. This could be mitigated with email, but that assumes reliable internet coverage as well. The final limitation regarding republication is serious because people could have one person pay for the classes, download all the materials, and then share it with potentially thousands of others. This should be addressed in a future design document to mitigate that threat.

### 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.*

