# 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?*

The purpose of this project is to design and implement an online system for DriverPass, an organization that offers a comprehensive driving education platform.

The client, DriverPass, wants a system that allows users to sign up for driving courses, schedule lessons, access study materials, and track their progress.

The system must be accessible online, cater to multiple user roles (students, instructors, administrators), and support automated scheduling and communication.

### 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 currently handles most of its scheduling and instruction manually, leading to inefficiencies and miscommunications.

The main problem to fix is the lack of a centralized, user-friendly system for students and instructors to manage lessons and materials.

Required components:

* A responsive web application interface
* A secure user management system
* A database to store user data, lesson schedules, and course materials
* Integration with calendar and email systems for reminders and notifications

### 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?*

* The system should:
  + Automate student sign-up, course selection, and scheduling
  + Provide secure login and user-role-based access
  + Track and display progress for students and instructors
  + Notify users of upcoming lessons and changes
  + Be scalable and maintainable for future features
* Measurable tasks:
  + Minimum 95% system uptime
  + Lesson booking confirmation emails delivered within 5 minutes
  + Account setup process under 3 minutes for a new user

## ⚙️Requirements

### Nonfunctional Requirements

#### 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 must be fully web-based and accessible through major browsers (Chrome, Firefox, Safari, Edge).

Page load times must be under 2 seconds for 90% of requests.

The system must support concurrent usage by at least 500 users with no degradation in performance.

Monthly updates should be scheduled to fix bugs and introduce minor enhancements.

#### 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 application must run on Windows and Unix servers.

Backend must be supported by a relational database (PostgreSQL or MySQL).

Cloud-based deployment (AWS or Azure) is preferred for scalability and redundancy

#### 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?*

Users will be uniquely identified by email; roles (admin, student, instructor) are assigned at registration.

Inputs such as usernames and passwords will be case-sensitive.

The system should automatically flag scheduling conflicts and notify administrators in real time.

#### 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?*

User account management (add/remove/modify) should be accessible via admin panel, not hardcoded.

The system must support modular code design to allow updates and extensions without major overhauls.

IT admins must have full backend access with audit trail and logging capabilities.

#### 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 must authenticate via username/email and strong password; two-factor authentication (2FA) optional.

All communication must use HTTPS/SSL encryption.

After five failed login attempts, the account will be locked for 30 minutes and admin will be alerted.

Password recovery via secure tokenized email link.

### 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 credentials upon login.

The system shall allow students to register for an account and select a driving package.

The system shall enable instructors to view, accept, or decline lesson requests.

The system shall display upcoming lessons on a calendar interface for both students and instructors.

The system should send email notifications for new bookings, cancellations, and reminders.

The system should allow administrators to add or modify driving courses and assign instructors.

The system shall track student progress and display milestones.

### 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 will be browser-based and responsive (mobile and desktop friendly).
* **User roles and needs:**
  + **Students**: Register, login, book lessons, view schedule, access course materials
  + **Instructors**: View student bookings, accept/reject sessions, leave notes, track student progress
  + **Administrators**: Manage users, set lesson prices, edit course materials, assign instructors
* Interaction: Standard browser input (mouse, keyboard, touch), calendar pop-ups, secure login/logout

### 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?*

* Users have access to internet and modern devices (smartphones, tablets, desktops).
* Instructors and students are familiar with online booking systems.
* The database and cloud services will be provisioned and available before launch.
* 24/7 support and server maintenance are outsourced or pre-negotiated.

### 

### 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?*

Budget constraints may limit the inclusion of advanced AI features or predictive analytics in version 1.0.

Timeline limits deep integration with third-party APIs (e.g., driving record lookup).

Resource constraints restrict simultaneous development of a full-featured mobile app.

The system does not include payment processing integration in phase one (handled offline for now).

### Gantt Chart