# CS 255 Business Requirements Document Template

## Aero Berry

10/04/25

## 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 the project is to create an online and in-person training platform that improves the success rate of driver’s license applicants called the DriverPass system. The system is intended to provide students with access to online practice exams, instructional materials, and the ability to schedule on-the-road lessons with certified instructors. DriverPass is aiming to address the 65% failure rate of students taking the driving test by providing comprehensive preparation through digital and real-world instruction. The system will serve multiple user groups like students, instructors, administrators, and IT personnel by integrating scheduling, testing, reporting, and user management features into a secure, easy-to-use platform. The ultimate goal of the system is to enhance student performance, streamline lesson management, and ensure data accuracy and accessibility for all their stakeholders.

### 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 identified a gap in the market for an integrated platform that would support both digital and practical driving education. The new DriverPass system will allow students to register, pay for packages, take online practice exams, track progress, and book on-the-road training sessions. Instructors will manage schedules and student records through the system, while administrators will monitor operations, generate reports, and manage system maintenance. The system will be hosted on a secure web platform with scalable infrastructure to support future growth and mobile access. The project is being designed as a combatant to the high failure rate of students that take the driving test.

### 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?*

 Improve the pass rate for driver’s license tests through structured practice exams and training resources.

 Offer an online system that is accessible 24/7 to accommodate students’ schedules and locations.

 Create a seamless scheduling system that allows students to book lessons with certified instructors.

Provide management tools to oversee student progress, instructor performance, and system analytics.

 Protect user information and ensure secure handling of payments, test data, and personal details.

 Design a system architecture that allows for future feature expansion and easy maintenance.

Enable communication between students, instructors, and support staff through internal messaging and notifications.

## 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 must support multiple concurrent users without degradation in response time. Pages should be loaded in a timely manner under normal traffic conditions. The backend infrastructure should be capable of scaling to support increased usage as DriverPass expands to new regions. I think the DriverPass system could follow a quarterly maintenance and update schedule, with potential patches released monthly as needed. To address any potential bugs, improve usability, or make any small security patches without interrupting service.

#### 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 platform as requested must be web-based, accessible on all modern browsers, and compatible with both desktop and mobile devices. The system should integrate with external APIs, like Google Maps for location-based lesson scheduling, and a payment gateway for secure transactions. The system will require a **relational database** to manage user data, test results, and lesson schedules. A Java-based framework can manage business logic, while using APIs can handle communication between the client and server.

#### 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 DriverPass system will maintain accuracy by validating user inputs, ensuring reliable calculations, and clearly defining system responses to errors. Different user roles like students, instructors, and administrators will be distinguished through secure, role-based login credentials. Each role grants specific access permissions, such as instructors managing training schedules and students accessing practice exams. Usernames and emails will not be case-sensitive to prevent login confusion, while passwords will remain case-sensitive to ensure stronger security. I feel this approach balances usability with data protection. The system will automatically inform administrators when a significant issue occurs, such as repeated failed logins, data synchronization errors, or payment processing failures. Alerts will appear on the administrator dashboard and can be sent via email for timely review.

#### 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 system will allow administrators to add, remove, or modify user accounts directly through an administrative dashboard without changing any source code. This functionality will ensure efficient user management and reduces dependency on IT staff for minor adjustments.  
  To adapt to platform updates, the system will use modular components and version-controlled APIs. This approach allows for compatibility with browser, OS, or server updates without requiring full redeployment. The IT administrator will also have full backend access to manage user data, oversee performance, apply patches, and perform backups through secure administrative tools.

#### 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 will be required to log in using a unique username and password combination. Multifactor authentication could also be added for higher security.
* All data exchanges between the client and server will be secured using HTTPS with SSL/TLS encryption. Sensitive information, such as passwords, will be hashed and salted in the database to prevent exposure.
* If a “brute force” hacking attempt is detected, the system will temporarily lock the account and alert the administrator.
* If a user forgets their password, they will be able to reset it through a secure email or SMS-based verification process to regain access safely.

### 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 when logging in.
* The system shall allow students to take online practice exams and receive immediate feedback.
* The system shall enable instructors to schedule and manage on-the-road training sessions.
* The system shall track student progress and store exam history in the database.
* The system shall allow administrators to manage user accounts, permissions, and reports.
* The system shall notify users via email or dashboard updates of upcoming lessons, completed tests, or system alerts.

### 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 intuitive, responsive, and accessible via web browsers and mobile devices.  
The main user groups are students, instructors, and administrators.

* **Students** will need to register, log in, take practice tests, schedule driving sessions, and review performance results.
* **Instructors** will need to view student progress, manage schedules, and record training results.
* **Administrators** will need to manage users, monitor activity, and review system metrics.  
  Users will interact through a browser-based interface designed with clear navigation menus, dashboards, and feedback messages.

### 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?*

* It is assumed that all users have access to reliable internet and a modern web browser.
* It is also assumed that instructors and administrators have basic computer literacy and can navigate a dashboard interface.
* The design assumes that the system will be hosted in a secure cloud environment capable of handling expected user traffic.

### 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 may be limited by available development resources and the project timeline outlined in the schedule.
* Performance may depend on users’ internet speeds, and server maintenance windows could cause brief periods of downtime.
* Additionally, the system’s accuracy in scheduling may depend on external API reliability from calendar or payment services.

### 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 diagram with multiple colored squares

AI-generated content may be incorrect.