Southern New Hampshire University

CS 255 Business Requirements Document

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?

- Deliver a cloud-hosted, web-based platform for DriverPass to manage both online driver education and behind-the-wheel training.
- Give students and staff secure, cross-device compatible access to all system features, such as registration, scheduling, coursework, reporting, and the ability to export data for offline use.
- Streamline the full training lifecycle: account setup, package selection, lesson booking, progress tracking, and activity reporting.
- Equip DriverPass leadership (owner, IT head) with robust administrative controls and dashboards for monitoring system usage and enforcing security policies.

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 exists to tackle the high DMV test-failure rate by unifying exam prep, rules instruction, and practical driving sessions in a single solution.
- Existing platforms are disjointed, and students must jump between separate tools for lesson content, scheduling, and performance feedback, leaving gaps in their preparation.
- Essential Functionality:
    - **User & Role Management**: Owner, IT officer, secretary, student, and instructor. Each with tailored and appropriate permissions.
    - **Reservation Engine**: Match students with cars and trainers in two-hour slots, across three initial package tiers.
    - **Online Learning Hub**: Host course materials, practice exams, and a progress dashboard (status: not started, in progress, failed, passed).
    - **Audit & Reporting**: Capture every change (who, what, when) and generate filterable, printable reports.
    - **Enrollment & Payment**: Intake form with personal details, pickup/drop-off locations, and secure-credit card fields.
    - **DMV Integration**: Automated feed for new rules or question-bank updates, with in-system notifications.
    - **Responsive UI**: Based on the provided sketch, featuring test progress, student profiles, driver notes, special needs flags, and photo panels.

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?

- Implement role-based access control covering at least five distinct user types, with precise and appropriate create/read/update/delete rights.
- Enable self service student registration, online booking/modification/cancellation of driving lessons, and selection among three starter packages.
- Provide an administrative tool to immediately disable any training package, preventing further enrollments without code changes.
- Build a testing module that logs each exam's name, duration, score, and completion status, and records progress in a dashboard.
- Record a full audit trail for all data operations and offer exportable activity logs with the ability to filter by user, date range, or action.
- Allow offline export of key reports (compatible with Excel) while enforcing online-only data updates to avoid synchronization conflicts.
- Integrate with the DMV's update service to pull in new content and alert the IT officer within 24 hours of any changes.
- Deploy to a secure, managed cloud environment with built-in backups and compliant handling of payment information.

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?

- [Insert text]

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?

- [Insert text]

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?

- [Insert text]

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?

- [Insert text]

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?

- [Insert text]

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."

- [Insert text]

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.)?

- [Insert text]

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?

- [Insert text]

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?

- [Insert text]

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.

[Insert chart]