Southern New Hampshire University

CS 255 Business Requirements Document Template

Tania Boursiquot

Complete this template by replacing the bracketed text with the relevant information.

This template lays out all the different sections that you need to complete for Project One. Each section has guiding questions to prompt your thinking. These questions are meant to guide your initial responses to each area. You are encouraged to go beyond these questions using what you have learned in your readings. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client's needs. There is no required length for the final document. Instead, the goal is to complete each section based on your client's needs.

Tip: You should respond in a bulleted list for each section. This will make your thoughts easier to reference when you move into the design phase for Project Two. One starter bullet has been provided for you in each section, but you will need to add more.

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 develop a web-based system for DriverPass, a company that helps students prepare for their DMV driving tests through online practice exams and in-person driving lessons.
- The client, DriverPass (represented by Liam and Ian), wants a system that allows students to register, schedule appointments, access study materials, and take practice exams online.
- The system should also allow employees (IT admin, secretary, instructors) to manage user accounts, schedule lessons, track progress, and generate reports.
- The system needs to be cloud-based, secure, and accessible across various devices (PCs, tablets, mobile phones).
- The goal is to reduce failure rates among students by offering comprehensive, accessible, and flexible training tools.

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 wants to address the high failure rate among driving test applicants by offering better training through online practice exams and in-person driving sessions.
- The problem they aim to solve is the lack of practical, accessible tools that adequately prepare students for the DMV test.

- The system should allow users to register, select a training package, book appointments, and access training materials.
- Customers should be able to view their progress, including completed tests and test scores, and reschedule or cancel appointments if needed.
- Admins and employees should be able to manage user accounts, appointments, driver assignments, and activity logs.
- The system must track who made or modified each reservation for accountability.
- A secure login system with role-based access is required for different users (admins, IT, secretaries, instructors, and students).
- Integration with DMV data is necessary to keep training content and rules up to date.

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 allow students to create accounts reset passwords, and security log in.
- The system shall allow students to register for one of three training packages, which include varying levels of on-the-road instructions, in-person DMV instructions, and online practice exams.
- The system shall support online appointment booking, modification, and cancellation for driving lessons.
- The system shall allow staff (secretary) to schedule appointments manually over the phone or in person.
- The system shall allow customers to input contacts details, pickup/drop-off location, and payment information securely.
- The system shall assign each lesson to a specific instructor, vehicle, time, and customer.
- The system shall track changes to appointments and maintain logs of who created, modified, or canceled each reservation.
- The system shall allow instructors to leave lesson notes, times, and feedback for each student.
- The system shall allow students to view progress on practice exams (test name, time taken, score, and status).
- The system shall generate downloadable reports and activity logs for management.
- The system shall be adaptable for enabling/disabling training packages.
- The system shall integrate with DMV data to receive updates and send alerts when policies or content change.

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 shall be web-based and hosted in the cloud, accessible via standard internet browsers.
- The system shall support responsive design for use on desktops, laptops, tablets, and mobile phones.
- The system shall be available 24/7 with minimal downtime (target 99.9% uptime).
- The system shall load user pages (dashboard, booking, test history) within 2 seconds under normal network conditions.
- The system shall support concurrent access by multiple users without performance degradation.
- The system shall regularly check for DMV content updates (e.g., daily or weekly, based on integration availability).
- System logs and data should be backed up automatically and stored securely.

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 shall be platform-independent from the user's perspective, running on any OS that supports modern web browsers (Windows, macOS, Linux, iOS, Android).
- The system shall be hosted on a cloud platform such as AWS, Azure, or Google Cloud, to handle scalability, security, and backups.
- The system shall use a relational database (e.g., MySQL, PostgreSQL) for storing user data, appointments, training progress, and system logs.
- The back end shall be built using industry-standard technologies such as Node.js, Python, or Java.
- The front end shall be accessible via browsers like Chrome, Firefox, Safari, and Edge.
- The system shall not require any client-side installations; access shall be entirely browser-based.

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 shall distinguish users by unique usernames (typically email addresses) and role-based access control (e.g., student, secretary, admin, IT officer, instructor).
- All login inputs (usernames and passwords) shall be case-sensitive.
- Data entry fields (e.g., names, addresses, phone numbers, credit card details) shall be validated in real time to prevent formatting errors and incomplete submissions.
- Scheduling conflicts (e.g., double-booked cars or instructors) shall trigger immediate system alerts and prevent submission.
- The system shall generate alerts/log entries when errors occur, such as failed login attempts, scheduling conflicts, or failed payments.

- Admins shall receive notifications for critical system errors or repeated access issues, such as brute-force login attempts or database connection failures.
- Data displayed to users (e.g., test results, lesson times) shall be precise and updated in real time upon any change.

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 shall allow authorized admin users to add, remove, or modify user accounts through a secure administrative interface without requiring code changes.
- The system shall allow enabling or disabling specific training packages through the admin panel, allowing flexibility without modifying core code.
- The system shall be hosted on a cloud platform, ensuring it stays compatible with browser and platform updates through regular automated maintenance and deployment cycles.
- The IT admin shall have full system access, including the ability to:

  • Reset user passwords

  • Lock or deactivate user accounts

  • Configure permissions for other roles

  • View full audit logs and activity reports

- The system shall be built using modular architecture to allow for easier future upgrades or feature additions.

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 shall be required to log in using a unique username (email) and a strong password.
- All data exchanged between the client and the server shall be encrypted using HTTPS/SSL protocols.
- Sensitive information (e.g., passwords, credit card data) shall be securely encrypted and stored using hashing algorithms and PCI-compliant methods.
- The system shall automatically lock an account after a set number of failed login attempts (e.g., 5), triggering an alert to the IT admin.
- Users shall be able to reset their passwords through a secure, automated password recovery process via email.
- The IT admin shall have the ability to manually reset passwords and unlock accounts if necessary.

- User sessions shall expire after a period of inactivity to prevent unauthorized access.

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 users to create and manage their own accounts.
- The system shall validate user credentials when logging in.
- The system shall allow users to reset their passwords through a secure process.
- The system shall allow students to view, book, reschedule, or cancel driving lessons.
- The system shall allow secretaries to schedule lessons manually over the phone or in person.
- The system shall assign driving sessions to available instructors and cars, avoiding scheduling conflicts.
- The system shall provide students with access to online training material and practice exams.
- The system shall track student progress on exams, displaying test names, time taken, score, and status.
- The system shall allow instructors to enter lesson times and notes after each session.
- The system shall provide managers and IT staff with access to downloadable reports and activity logs.
- The system shall send alerts or updates when DMV content changes are available.
- The system shall allow admins to enable or disable training packages.
- The system shall log user actions such as booking, modifying, or canceling appointments.
- The system shall support secure payment input for package registration.

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 system interface shall be web-based, accessible via modern internet browsers on desktops, tablets, and smartphones.
- The interface shall be intuitive and user-friendly, with clear navigation and mobile responsiveness.
- • Students will use the interface to:
    - Create and manage accounts
    - View and select training packages
    - Schedule, modify, or cancel driving lessons
    - Take online practice exams and track progress
    - View instructor feedback and lesson history

- Secretaries will use the interface to:
  - Create or edit student records
  - Book appointments manually for students who call or visit
  - View instructor availability and scheduling conflicts
- Instructors will use the interface to:
  - View their student records
  - Enter lesson times and notes for each session
- IT Admins will use the interface to:
  - Manage user accounts and permissions
  - Reset passwords and unlock accounts
  - Monitor logs and system activity
- Managers will use the interface to:
  - Download reports and monitor business performance
  - Review student and instructor activity

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 (students, staff, instructors) will have access to a device with internet capability and a modern web browser.
- • It is assumed that users will have basic digital literacy to navigate the web interface and perform actions like scheduling or taking tests.
- • It is assumed that secure cloud infrastructure and hosting will be provided and maintained by a third-party cloud service provider.
- • It is assumed that DMV will provide access to up-to-date rules, sample test questions, and policy updates in a format that can be integrated or parsed by the system.
- • It is assumed that sensitive information such as credit card data will be processed through a secure, PCI-compliant third-party payment processor.
- • It is assumed that training packages will not frequently change and that enabling/disabling a package will meet current business needs without requiring structural changes.
- • It is assumed that there will be adequate staff to manage appointment scheduling, customer service, and user support.

Limitations

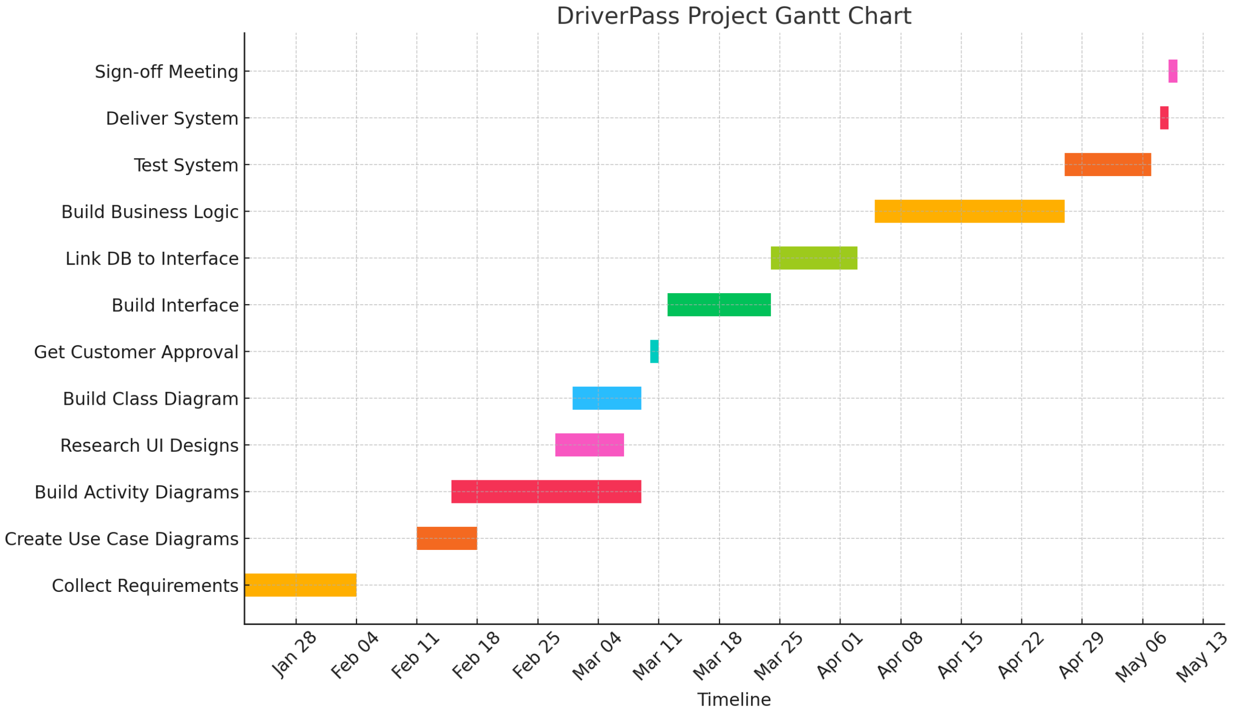
Any system you build will naturally have limitations. What limitations do you see in your system design? What limitations do you have as far as resources, time, budget, or technology?

- The system will not support full offline functionality—users must be online to book or modify appointments, access lessons, or take tests.
- • Any major changes to training packages (such as adding or removing one) will require developer involvement and cannot be done through the admin interface.
- • Integration with the DMV depends on the availability and consistency of their update format and may require manual updates if automation is not feasible.

- • Budget constraints may limit advanced features such as AI-based scheduling or predictive performance tracking in the initial release.
- • Time constraints limit the inclusion of future features or expansions (e.g., mobile apps, instructor certification tracking) during this development cycle.
- • Resource limitations may restrict real-time customer support features, such as live chat or in-app messaging, during initial deployment.
- • There may be limited customization of UI themes or branding until future updates are considered.

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.


DriverPass Project Gantt Chart
Sign-off Meeting
Deliver System
Test System
Build Business Logic
Link DB to Interface
Build Interface
Get Customer Approval
Build Class Diagram
Research UI Designs
Build Activity Diagrams
Create Use Case Diagrams
Collect Requirements
Jan 28
Feb 04
Feb 11
Feb 18
Feb 25
Mar 04
Mar 11
Mar 18
Mar 25
Apr 01
Apr 08
Apr 15
Apr 22
Apr 29
May 06
May 13
Timeline