Southern
New Hampshire
University

CS 255 Business Requirements Document Template

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?

- This proposal is intended for our client, DriverPass, whose mission is to help students prepare for the DMV driving test through a program they have established.
- DriverPass plans to provide a service with online assessments, classroom lessons, and multiple packages for on-road driver training.
- The objective of the project is to implement a web-based application that provides registration, scheduling, testing that tracks students, and system administration for DriverPass.
- The application will allow remote access to data, exportable reports, and will support an efficient process for DriverPass.

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 was created to address a societal challenge: the high failure rate among students taking the DMV driving test.
- DriverPass will improve student success rates by offering:
  - Online lessons and customizable practice tests.
  - On-road driving lessons with licensed, authorized drivers.
  - Numerous value-flexible packages for training (6, 8, or 12 hours on the road with optional addons).
- The application must support:
  - User types for students, IT admin, secretary, and management.
  - Online scheduling with system-level tracking of appointments.
  - Assignment of lessons and drivers to selected packages.
  - Secure login access with user activity tracking.
  - Generation of reports and notifications about available lessons.

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 application shall allow users (students) to:
    - Register accounts and select their packages for training.
    - Schedule, cancel and/or reschedule on-line.
    - Assess their progress and scores on their on-line testing (e.g., cleared, not cleared, in progress).
- The secretary will have the ability to:
    - Give students a chance to enter their information manually (by phone or walk-in).
    - Assign drivers with a vehicle and time slots for lessons.
    - View and modify any appointments, if required.
- Management shall be able to:
    - Reset or disable user accounts.
    - Track and audit user activity (who reserved, who modified, who cancelled).
    - Generate printable reports from the activity and logs of lessons.

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 cloud-based and responsive, offering access from any internet-connected device.
- It should support concurrent users, including students, staff, and admin, with minimal latency.
- Real-time updates (e.g.: test results, scheduling changes) should sync without needing manual refresh.
- The system should be maintained regularly with updates synchronized with DMV rule changes.

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 will run on modern web browsers (Chrome, Firefox, Safari) on desktop, tablet, and mobile.
- Backend requirements include a secure relational database (e.g.: PostgreSQL) and a cloud-hosted environment with automatic backup support.
- Frontend should be device-responsive and accessible from any OS with internet access.

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 require user authentication with role-based access control (admin, secretary, student).
- It must track changes (e.g.: who created/modified/canceled appointments) for accountability.
- Data validation should prevent scheduling conflicts, ensure credit card fields are securely handled, and maintain consistency across reports.

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?

- Admins (like Ian) shall be able to disable or enable packages as needed without code changes.
- The system should allow for scalable additions like new packages or DMV-compliance modules in future iterations.
- Permission should be editable by IT personnel without vendor assistance.

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?

- All logins must be authenticated using secure credentials and password reset flows.
- Data transmission between client and server must be encrypted (e.g.: HTTPS, SSL).
- After five failed login attempts, accounts should be temporarily locked to prevent brute-force attacks.
- Users must be able to reset passwords throughout an automated email system.

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 register, log in, and select a training package.
- The system shall allow users to schedule, cancel, or modify driving lessons online.
- The system shall allow the secretary to input student info received via phone or walk-in.
- The system shall allow the admin to assign a driver, vehicle, and time slot for each appointment.
- The system shall validate credit card input and store data securely.
- The system shall allow students to view their online test progress and scores.
- The system shall provide a notification system for DMV updates and appointment reminders.
- The system shall allow the IT admin to reset user passwords, block access, and generate activity reports.

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.)?

- Users: Student, Secretary, IT Admin, Management
- Student Interface:
  - Register and login
  - Book/reschedule appointments
  - Take and track practice tests
  - View training schedule and progress
- Secretary Interface:
  - Add/edit appointments
  - Input student data from calls/walk-ins
  - Assign trainers manually
- Admin Interface:
  - View/edit all users
  - Reset passwords, disable packages
  - View audit logs, generate reports
- Accessible through any browser or mobile device.

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 will have access to stable internet and modern devices.
- Secretaries are trained to operate the scheduling module.
- The system will initially launch in English and only serve one region or DMV jurisdiction.
- DriverPass will provide the required DMV-compliant content and updates when needed.

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?

- Customizing packages (e.g.: adding new modules) will require developer support iniitally.
- DMV integration is limited to receiving updates; full automation may not be in the first release.
- Budget and time may restrict features like chatbots, instructor mobile apps, or real-time location tracking.
- User feedback loops and feature refinement will take place post-launch.

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 Gantt Chart

Task	Start	End	January Week 1	Week 2	Week 3	Week 4	February Week 1	Week 2	Week 3	Week 4	March Week 1	Week 2	Week 3	Week 4	April Week 1	Week 2	Week 3	Week 4	May Week 1	Week 2	Week 3	Week 4
Collect Requirements	Jan 22	Feb 4				▬																
Create Use Case Diagrams	Feb 11	Feb 18						▬														
Build Activity Diagrams	Feb 15	Mar 9							▬													
Research UI Designs	Feb 27	Mar 7								▬												
Build Class Diagram	Mar 1	Mar 9									▬											
Get Customer Approval	Mar 10	Mar 11										▬										
Build Interface	Mar 12	Mar 24										▬										
Link DB to Interface	Mar 24	Apr 3												▬								
Build Business Logic	Apr 5	Apr 27														▬						
Test System	Apr 27	May 7																▬				
Deliver System	May 8	May 9																		▬		
Sign-off Meeting	May 9	May 10																		▬		