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?

- The purpose of this project is to create a centralized, user-friendly system for DriverPass to improve scheduling, communication, and tracking student progress.
- The client, DriverPass, wants to help students navigate the driver's education process more easily by offering online booking, automated reminders, and visibility into their progress.
- The system should support students, instructors, and administrators with different access levels and capabilities.

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 relies on disconnected tools, which causes confusion and delays for students and instructors.
- Students are unsure of next steps, testing dates, and scheduling.
- Instructors struggle to manage calendars and keep track of student progress manually.
- The new system should centralize scheduling, communication, and progress tracking in one platform.
- Components needed include:
- Student portal
- Instructor interface
- Admin dashboard
- Notification system
- Calendar and scheduling integration

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?

- Enable students to:
- Register and log in to the system
- View their current progress and next steps
- Book or reschedule lessons and exams online
- Enable instructors to:
- Manage schedules
- Access student progress
- Provide automated reminders for lessons, exams, and progress updates
- Improve communication between students and instructors
- Reduce manual errors and increase scheduling efficiency
- Measurable outcomes:
- 100% online scheduling
- 90% reduction in missed appointments due to reminders
- Real-time progress tracking for all students

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 be accessible through a web browser and mobile devices.
- Pages should load in under 3 seconds.
- The system should support 99.9% uptime and be updated quarterly for feature and security 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 system should be compatible with Windows, macOS, Android, and iOS platforms via web browsers.
- It will use a cloud-hosted database to store user and scheduling data securely.
- Backend technology may include PHP, Python, or JavaScript with database integration (e.g., MySQL or PostgreSQL).

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?

- User accounts must be unique, identified by email or username.

- Input fields must validate case-sensitive data where applicable (e.g., passwords).
- The system will notify the admin in case of errors like failed bookings or missing progress updates.

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 should be able to add/remove/modify user accounts through the dashboard without touching the code.
- The system should remain compatible with browser and mobile OS updates.
- IT admin will require full access to user management, database logs, and security configurations.

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 log in with secure credentials (username/email and password).
- Data transfers will be secured using HTTPS/SSL encryption.
- Lock accounts for 15 minutes after 5 failed login attempts to prevent brute-force attacks.
- Provide password reset via email verification 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 allow students to create and manage their accounts.
- The system shall validate user credentials when logging in.
- The system shall allow students to schedule and reschedule driving lessons.
- The system shall allow instructors to manage their calendar and lesson availability.
- The system shall send automated reminders for scheduled lessons and test eligibility.
- The system shall provide real-time updates to student progress.
- The system shall allow administrators to manage users and view system usage.

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.)?

- User types:
- Students: book lessons, view progress, receive reminders
- Instructors: manage availability, view assigned students
- Admins: manage users, oversee system operations
- The interface must be intuitive and mobile-friendly.
- Users will access the system via modern web browsers on desktop and mobile devices.

- The system will use clear visual cues (e.g., progress bars, color-coded status) to guide users.

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 reliable internet access.
- Users can receive email and/or SMS for notifications.
- The DMV test schedule integration will be a future upgrade, not part of initial release.
- Students will actively use the platform after registration.

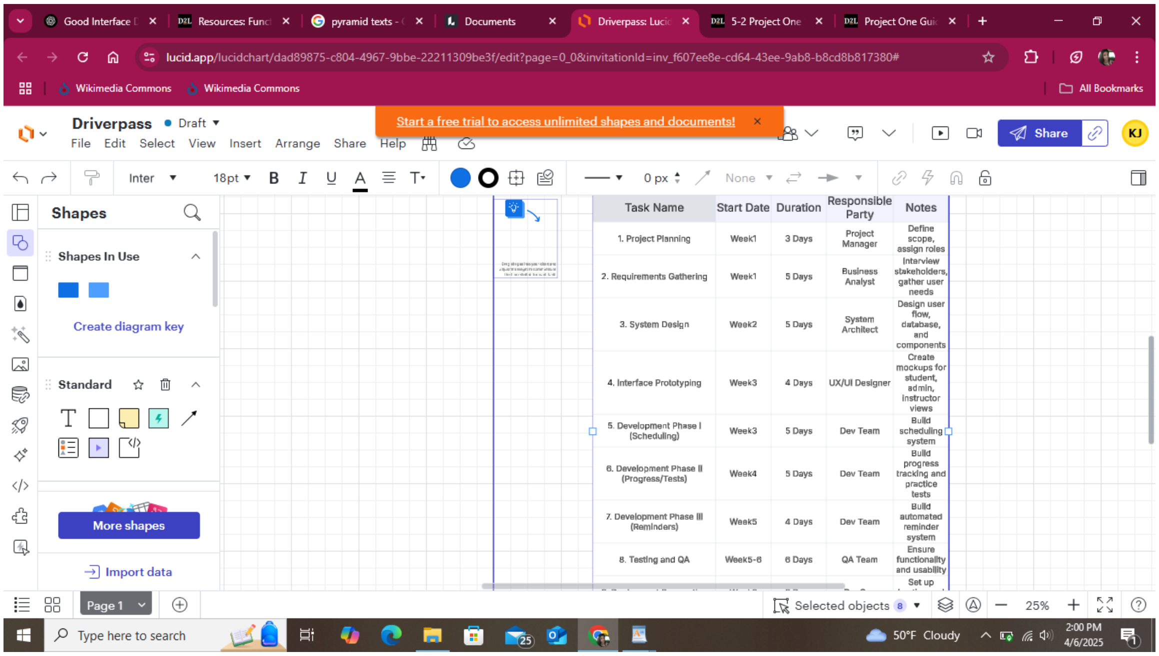
Limitations

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?

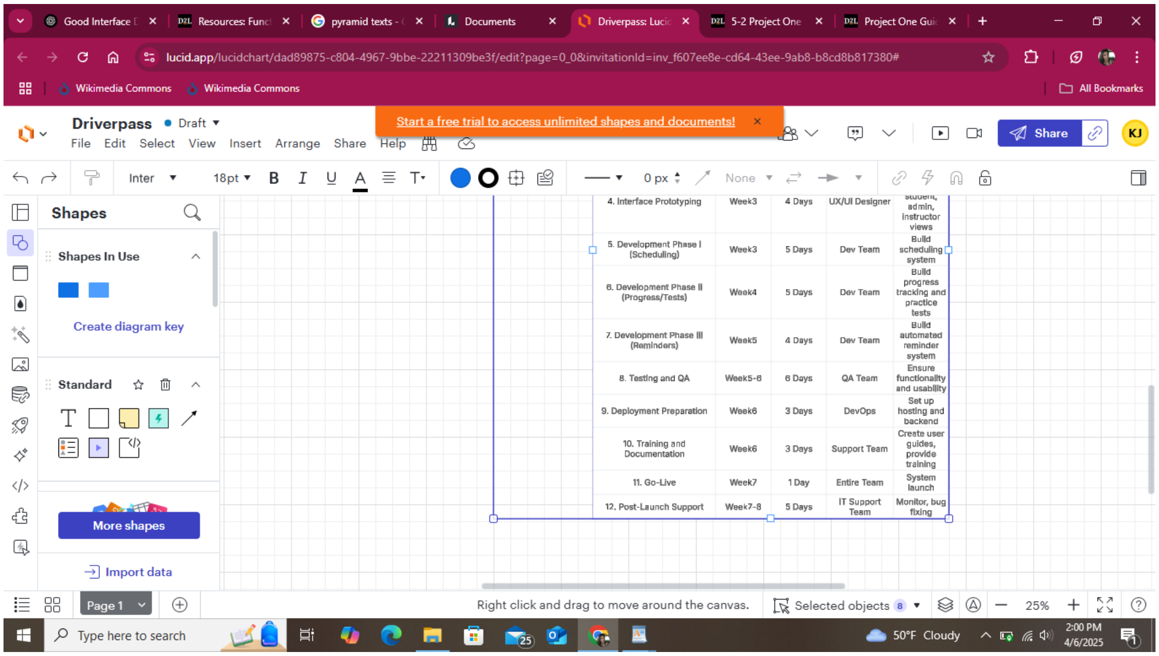
- The system will only be available in English during the first phase.
- Limited budget may restrict the inclusion of real-time DMV data integration.
- Timeline may limit full mobile app development—initial release will focus on web-based access.
- Testing and support resources are limited to internal staff initially.

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.



Task Name	Start Date	Duration	Responsible Party	Notes
1. Project Planning	Week1	3 Days	Project Manager	Define scope, assign roles
2. Requirements Gathering	Week1	5 Days	Business Analyst	Interview stakeholders, gather user needs
3. System Design	Week2	5 Days	System Architect	Design user flow, database, and components
4. Interface Prototyping	Week3	4 Days	UX/UI Designer	Create mockups for student, admin, instructor views
5. Development Phase I (Scheduling)	Week3	5 Days	Dev Team	Build scheduling system
6. Development Phase II (Progress/Tests)	Week4	5 Days	Dev Team	Build progress tracking and practice tests
7. Development Phase III (Reminders)	Week5	4 Days	Dev Team	Build automated reminder system
8. Testing and QA	Week5-6	6 Days	QA Team	Ensure functionality and usability
				Set up



4. Interface Prototyping	Week3	4 Days	UX/UI Designer	student, admin, instructor views
5. Development Phase I (Scheduling)	Week3	5 Days	Dev Team	Build scheduling system
6. Development Phase II (Progress/Tests)	Week4	5 Days	Dev Team	Build progress tracking and practice tests
7. Development Phase III (Reminders)	Week5	4 Days	Dev Team	Build automated reminder system
8. Testing and QA	Week5-6	6 Days	QA Team	Ensure functionality and usability
9. Deployment Preparation	Week6	3 Days	DevOps	Set up hosting and backend
10. Training and Documentation	Week6	3 Days	Support Team	Create user guides, provide training
11. Go-Live	Week7	1 Day	Entire Team	System launch
12. Post-Launch Support	Week7-8	5 Days	IT Support Team	Monitor, bug fixing

Right click and drag to move around the canvas.