# CS 255 Business Requirements Document

**Tammy Hartline**

## 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 develop an online system for DriverPass, a driving school company, to help their students prepare for driving tests.
* The system should allow students to take practice exams, schedule on-the-road driving lessons, and view their progress.

### 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 an online system where students can take practice exams and schedule driving lessons with instructors.
* The current problem is students are not well-prepared for driving tests and there is no centralized scheduling system for lessons.
* The system will need user accounts, a scheduling component, a learning/practice exam component, and a reporting component.

### 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?*

* When completed, the system should allow students to log in, access learning materials, take practice exams, schedule driving lessons, and track their progress.
* Measurable goals include improving exam scores over time, easy scheduling of lessons, and high student satisfaction ratings.

## 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 will be web-based and needs to support up to 1000 concurrent users with page load times under 5 seconds.
* The system should be updated with new learning content and exam questions on a quarterly basis.

#### 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 run on any modern web browser on Windows, Mac, iOS and Android.
* The back end will require a relational database like MySQL to store user accounts, scheduling info, and exam results.

#### 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?*

* Users will be differentiated based on unique usernames. Usernames will be case-insensitive.
* The admin should be alerted if a user fails an exam 3 times in a row.

#### 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?*

* An admin interface will allow adding/removing users and modifying user roles without code changes.
* The system will adapt automatically to web browser updates.
* IT admin will need full access to manage system configuration.

#### 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 a username and password. Passwords must be 8+ characters.
* All data exchanged between client and server will use SSL/TLS encryption.
* After 5 failed login attempts the account will be locked for 15 minutes.
* For forgotten passwords, users can request a password reset link via email.

### 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.”*

* Allow users to log in with username and password
* Allow users to reset forgotten passwords
* Display a dashboard showing the user's practice exam scores and upcoming scheduled lessons
* Allow users to select and take practice exams
* Display practice exam results and explanations of right/wrong answers
* Allow users to view a calendar of available lesson time slots
* Allow users to schedule driving lessons
* Send users email confirmations of scheduled lessons
* Allow admins to add/remove users and assign roles
* Allow admins to add/update learning materials and practice exam questions

### 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 needs to be user-friendly for beginner computer users while still being visually appealing.
* There will be 3 main types of users: students, driving instructors, and admins.
* Students need to be able to log in, view learning materials, take exams, see scores, and schedule lessons.
* Instructors need to be able to view their teaching schedule and access student contact info.
* Admins need full access to manage all aspects of the system.
* The interface will be browser-based and mobile-friendly.

### 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?*

* We assume all users will have access to a modern web browser on a computer or mobile device.
* We assume DriverPass will provide the learning content and exam questions to populate the system.
* We assume DriverPass has the necessary vehicles and instructors to accommodate the scheduled lessons.

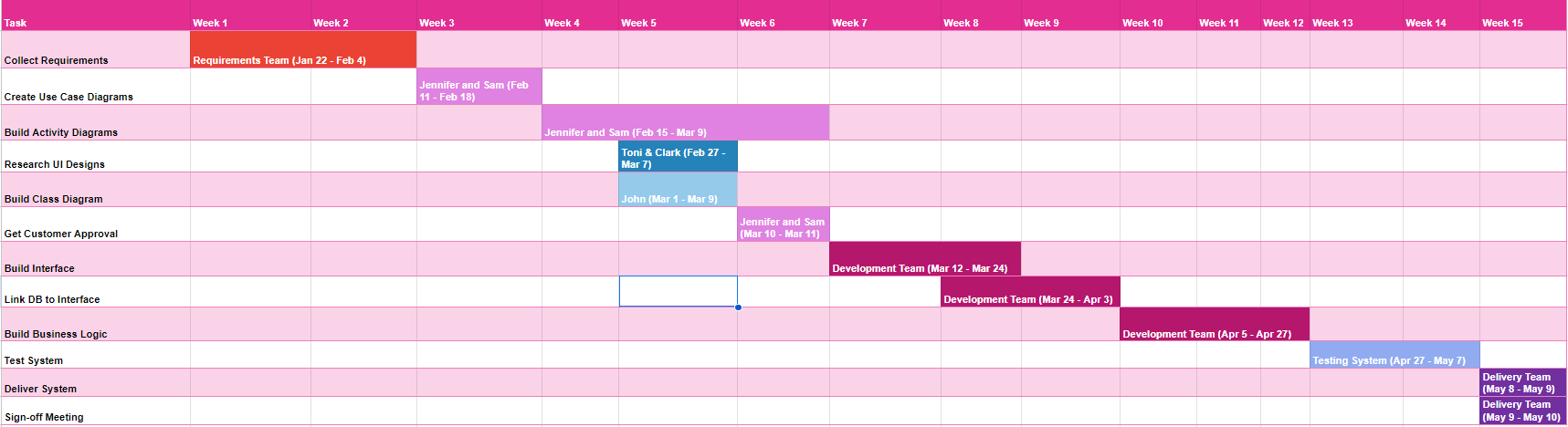
### Limitations

*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 is limited to only supporting scheduling for the number of instructors/vehicles DriverPass has available.
* Development time is limited to 6 months with a fixed budget, so not all desired features may be in the initial release.
* The system's performance may be affected by the user's internet connection speed and the number of concurrent users.
* The system is dependent on third-party services for email delivery and SSL certificates.

### 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.*



**Text Version of Gantt Chart:**

The Gantt chart for the DriverPass project includes the following main tasks, durations, and date ranges:

1. Requirements Gathering and Analysis - 2 weeks (January 1-14, 2024)
2. System Design - 4 weeks (January 15 - February 11, 2024)
   * Architecture Design - 2 weeks (January 15-28, 2024)
   * Database Design - 1 week (January 29 - February 4, 2024)
   * UI Design - 1 week (February 5-11, 2024)
3. Development - 12 weeks (February 12 - May 5, 2024)
   * User Management Module - 2 weeks (February 12-25, 2024)
   * Learning Content Module - 4 weeks (February 26 - March 24, 2024)
   * Practice Exam Module - 3 weeks (March 25 - April 14, 2024)
   * Scheduling Module - 3 weeks (April 15 - May 5, 2024)
4. Testing - 4 weeks (May 6 - June 2, 2024)
   * Unit Testing - 2 weeks (May 6-19, 2024)
   * Integration Testing - 1 week (May 20-26, 2024)
   * User Acceptance Testing - 1 week (May 27 - June 2, 2024)
5. Deployment - 1 week (June 3-9, 2024)
6. Post-Launch Support - 2 weeks (June 10-23, 2024)

The total project duration is estimated at 25 weeks (approximately 6 months), starting from January 1, 2024, and ending on June 23, 2024.