# CS 255 Business Requirements Document Template

Sean Chase

## 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 client is DriverPass – the system should be able to receive purchases of various driving lesson packages. It should also have a web interface with online tests. It should allow employees to access the data offline and it should track changes made by each employee. Edits to the data should be made online only.

### 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?*

* Problem – too many kids fail driving tests.
* Database – customer information, driver availability, scheduled sessions
* Frontend – interface for employees to enter information and make reservations when customers call, web interface for customers to access and make reservations online, or to access online tests

### 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?*

•Access data offline, edit data online

•Track edits to data and manage multiple user roles and permissions

•Allow employees to make reservations for customers or customers to make reservations online

•Handle multiple package types

•Allow customers to reset their own passwords

•Receive notification when DMV makes a rules change

•Web interface for customers to take online tests and view scheduled lessons, as well as make new reservationsRequirements

### 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?*

•System should be web-based

•Website should be responsive and modern web app

•Load times of no more than 2 seconds for majority of users

•Update system monthly or as needed with no more than 2 hours of downtime per updatePlatform •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?*

•Web frontend should run on most desktop and mobile browsers – IE, Chrome, Firefox, Safari, equivalent mobile browsers

•User Interface should automatically adapt for mobile browsers

•Backend requires database to store user information and system logs

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?*

* Web frontend uses session cookies to distinguish between users.
* Passwords are case-sensitive
* Daily aggregate report of all error reports; immediate notification of critical errors

#### 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?*

* Web app will need to remain up to date with breaking changes to browsers
* User changes will be done in the backend without changing code
* IT admin needs database access and access to server running web app

#### 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?*

* Password required for login
* Optional 2FA via SMS
* Accounts locked after 5 bad login attempts
* Forgotten password or locked account triggers email to user with temporary password

### 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 validate user password when logging in.
* The system shall send a verification SMS to the user’s registered phone number when 2FA is enabled.
* The system shall lock user accounts after five consecutive failed logins.
* The system shall send email a temporary password if an account locks or has a forgotten password.
* The system shall update user information on the backend in response to user or admin commands.
* The system shall track available appointment times and user appointments.
* The system shall schedule user appointments in response to user or admin commands.
* The system shall notify admins when DMV rules change.

### 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 User Interface will be web-based and adapt to desktop and mobile environments
* Users will access the frontend through desktop or mobile browsers.
* Customers will be provided with access to their own account including purchase and order history, and allow customers to purchase packages.
* Admin users will be provided with access to the entire schedule and include the ability to access customer user accounts to make appointments for them.

### 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?*

•I assume most customer users have access to the web and a modern web browser.

•I assume customer users have email.

•I assume DMV rules changes can be tracked automatically via an API or other interface.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?*

* 15 weeks to complete project
* Inability to anticipate changes to web browsers
* Inability to control DMVs rules and policies
* Must design frontend to be broadly compatible with all major web browsersGantt 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]

|  | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Collect Requirements | Jan. 22- Feb.4 | Task 2 | Task 2 |  |  |  |  |  |  |  |  |  |
| Create Use Case Designs |  | Feb. 11- Feb. 18 | Task 2 | Task 3 |  |  |  |  |  |  |  |  |
| Build Activity Diagrams |  |  | Feb. 15- Mar. 9 | Task 2 |  |  |  |  |  |  |  |  |
| Research User Interface 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 Database to Interface |  |  |  |  |  |  | Mar. 24- Apr. 3 |  |  |  |  |  |
| Build Business Logic |  |  |  |  |  |  |  | Apr. 5- Apr. 27 |  |  |  |  |
| Test System |  |  |  |  |  |  |  |  | Apr. 27- May 7 |  |  |  |
| De;over Suste, |  |  |  |  |  |  |  |  |  | May 8- May 9 |  |  |
| Sign off Meeting |  |  |  |  |  |  |  |  |  |  | May 9- May 10 |  |