# 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?*

* Project purpose: “to take advantage of a void in the market. . .training students for the driving test at their local DMV”
* Client: DriverPass, a company which trains driving test candidates. Liam is the owner and Ian is the IT officer.
* To do: for the system to handle the needs of customers “to be able to take online classes and practice tests. . .also. . .on-the-road training if they wish.”

### 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?*

* System to do: help DriverPass organize, schedule and track their business via an online application
* Problem to fix: 65% of applicants fail driving license exam because they only studied previous tests and not other sources
* Components: recording customer information, allowing scheduling of driver appointments, offering online classes and tests, accepting online payment

### 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?*

* Allow Liam to access his data online, download reports for offline work
* User permissions to allow varied levels of access:
  + levels/user types:
    - boss (Liam)
      * reporting
      * disable packages (modify packages in future iterations)
      * access blocking
    - IT officer (Ian)
      * maintenance
      * modifying system settings
      * password reset
    - Secretary
      * answers phone
      * creates new student profiles
      * makes appointments
    - Drivers
      * DMV notifications
      * student (client) profiles
      * Session notes
    - Student
      * make appointments
      * cancel/modify appointments
      * make online payments
      * DMV rules/policies lesson (package #2 and #3)
      * Online class content and material, practice tests (package #3 only)
* user action tracking
  + creation, modification, deletion of data:
    - reservations, cancellations, modifications to appointments
  + printable activity report (boss)
* phone call or online scheduling of appointments:
  + First name, last name, address, phone number, state, credit card number, expiration, security code, pickup location of customer, drop-off location of customer (same as pickup)
* online reservation scheduling (client or secretary for call-ins)
  + track client reservations by driver, time, and car
    - 10 cars, 10 drivers
* Training packages:
  + One: 3 lessons, 2 hours each (6 hours)
  + Two: same as One, plus one more lesson, plus DMV rules/policies lesson
  + Three: same as Two, plus two more lessons, online access to class content and material, practice tests
* Compliance: updates about DMV policies
  + product connected with the DMV
  + system notifies when new updates from DMV come in
* DMV Interface:
  + web/cloud (built-in security, no on-site storage of info)
  + triggers notifications when new info is added by DMV
* Look/style (see drawing)
  + Profile page
    - Logo
    - Tests completed/not completed
    - Driver info
    - Driver notes
      * Lesson time
      * start hour
      * end hour
      * driver comments
  + Contact Us page
  + New Student input form (secretary)
  + Other pages (later)
* Tasks:
  + Collect Requirements
    - interviews, observation, document analysis, JAD, prototyping
  + Functional Modeling
    - use case, use case diagrams, activity diagrams
    - research user interface designs
  + Structural Modeling
    - CRC cards, class diagrams, object diagrams
  + Behavioral Modeling
    - interaction diagrams, UML models, behavioral state machine
  + Get Liam’s approval
  + Create user interface
  + Design database and file structure
  + Map classes between interface and database
  + Build business logic
  + Test system
  + Document system
  + Deliver system
  + Sign-off meeting

## 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 accessible from anywhere via internet hosting and web browser access.
* The website should fully load within three seconds.
* The system will update in real time to provide accurate reflection of reservations calendar, payment, and profile updates.
* It must also feed new DMV notifications to admin users’ account pages in real-time.

#### 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 must run on web browsers. Web browsers run on Windows, Linux, and MacOS, as well as iOS and Android.
* The system must include horizontally- and vertically-scalable cloud-based database, security, and hosting services.

#### 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?*

* System users will require users log in using a registered username.
* The system must tag all user actions with a unique user identification to distinguish between users.
* The system must perform validation of text fields to prevent entry errors.
* The system must use check boxes and multiple-choice options wherever possible.
* The system must notify admin when a user is locked out of the system after three failed login attempts.

#### 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 will allow user profile updates of user name, address, and payment information by .
* The system will disallow user updates to username in order to preserve the distinction between users and preserve a consistent record of user actions.
* The system will adapt to platform updates by implementing them in off-hours (overnight), after being tested on a duplicate of the production system.
* IT admin requires access to reset and block any user accounts
* IT admin requires access to disable registration for particular packages.

#### 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 must be registered with the system by the creation of their user account by IT admin before logging in.
* Users will be authenticated by their username and password match when logging in.
* User’s passwords must include at least eight characters, upper and lowercase letters, numbers, and/or symbols, and not include words found in the dictionary (Swoopnow.com, 2020).
* The system will help users reset their password by sending a password reset email to their registered address.
* The system website pages will use encrypted HTTPS protocols with Transport Layer Security (TLS) (kaspersky.com, 2020) to secure the data connection between the browser and the website.
* To protect against brute force hacking attempts, users are allowed three login attempts before they’re blocked.
* User passwords expire after 60 days.

### 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.”*

* When users log in, the system shall verify user login credentials.
* When a DMV notice is received, the system shall notify admin users.
* When report data is requested from the system, it shall be downloaded from the database to the user’s computer.
* When a user is blocked by admin, the system shall render their username inactive.
* When a password is reset, the system shall render the username active if they have not been blocked.
* The system shall record every action in an audit log.
* The system shall maintain a connection to the DMV notification feed.
* The system shall notify admin when the DMV connection is offline.
* The system shall notify admin when the connection to the cloud database is offline.
* The system shall notify admin when a security issue has been detected.
* The system shall not allow user appointments without a payment confirmation.
* The system shall not allow user access to online classes payment is confirmed.
* The system shall define user access by authorization levels.
* The system shall utilize token-based authentication for all client-server communication.
* The system shall provide student profile and calendar access to driver users when an appointment is logged with the driver.
* The system shall enable user profile creation for IT, Admin, Secretary, and Student authorization levels.
* The system shall enable payment functionality for Secretary and Student authorization levels.
* The system shall enable reporting functionality for Admin, Secretary , and IT authorization levels.
* The system shall enable maintenance functionality for the IT authorization level.
* The system shall enable package editing functionality for the Admin authorization level.
* The system shall enable appointment creation and editing functionality for the IT, Secretary, Driver, and Student authorization levels.
* The system shall enable test access for Admin, IT, Secretary Driver, and Student authorization levels.

### 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 must present the login page, followed by a one-page view of functions according to user authorization level.
* The user interface must present the most-frequently used functionality in an easy-to-see manner, requiring only one click to enter the workflow. These features include viewing driver schedules, creating appointments, accepting payment, creating driver notes, and navigating to course materials and tests.
* The interface may use drop-down menus to hide less-frequently used functionality to be accessed in three clicks or less. These features include reporting, updating student profiles, adding student and driver photos, contacting DriverPass, and performing maintenance functions.
* The driver schedules page will show an overview of all driver schedules on one page, with a detailed view available for each driver by clicking on their name.
* The appointments creation and edit page will show current student appointments for two weeks starting with the current week, and including the driver name, start and end times, and pickup/dropoff location.
* The payment page will present a credit/debit card payment widget from a trusted third-party service.
* Driver notes will be available from the driving session page (default view for drivers), and allow screen keyboard input into a secure text field.
* Student course material will present the course heading, descriptions of modules, and links to course material within a window in the student view page.
* Updateable portions of student profiles, appointments, photos, and driver notes will feature an edit symbol in the bottom-right corner of the corresponding section of the page.
* A list of available reports will exist within the Admin and IT user views to run and schedule reports for download.
* Only those functionalities which are permissible for users to access will be visible to those users’ view pages (outlined below).
* Admin users will be using the user interface to review online test progress and to download reports.
* IT users will use the interface to enter the maintenance menu, as well as require access to all system functions for troubleshooting purposes.
* Secretary users will use the interface to create new student profiles, create and update appointments, and process payments.
* Driver users will use the interface to review appointment calendars, add notes during student driving practice sessions, and add their driver photo.
* Student users will use the interface to update their profile information, photo, review online course material and take online tests, and schedule driving appointments.
* Users will interact with the interface from a computer using a mouse, or from a touch screen device using touch.
* Loading web page elements should be accomplished via client-downloaded browser cookies due to frequent use of the page.
* Form fields will be remembered and automatically reloaded if an entry error is detected.
* Form fields will include default values to describe the necessary content to be added.
* Website database resources will be sourced from a cloud platform.
* The website shall use multi-compatible browser controls like the Python webbrowser module (Python.org, 2020) for forms, multimedia, and test-taking features.
* The user interface will use consistent color schemes, fonts, and capitalization.
* The user interface will use informative feedback for completion of actions and tasks, as well as for clicking on buttons and links.
* The user interface will include the ability to reverse, cancel, or abort an action before it is submitted, completed, or otherwise put in process (e.g., updates, payments, test submission, etc.)
* The user interface must adapt for display on either horizontal or vertical screen orientations.

### 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?*

* The layout of reports is assumed to include a single domain and variable (x and y axis).
* It is assumed that users will have internet access whenever they need to use the website.
* It is assumed that student users have an email address to which a password reset email will be sent.
* Development of the DriverPass system will remain within budgetary limitations.
* It is assumed that development staff will not call out sick.

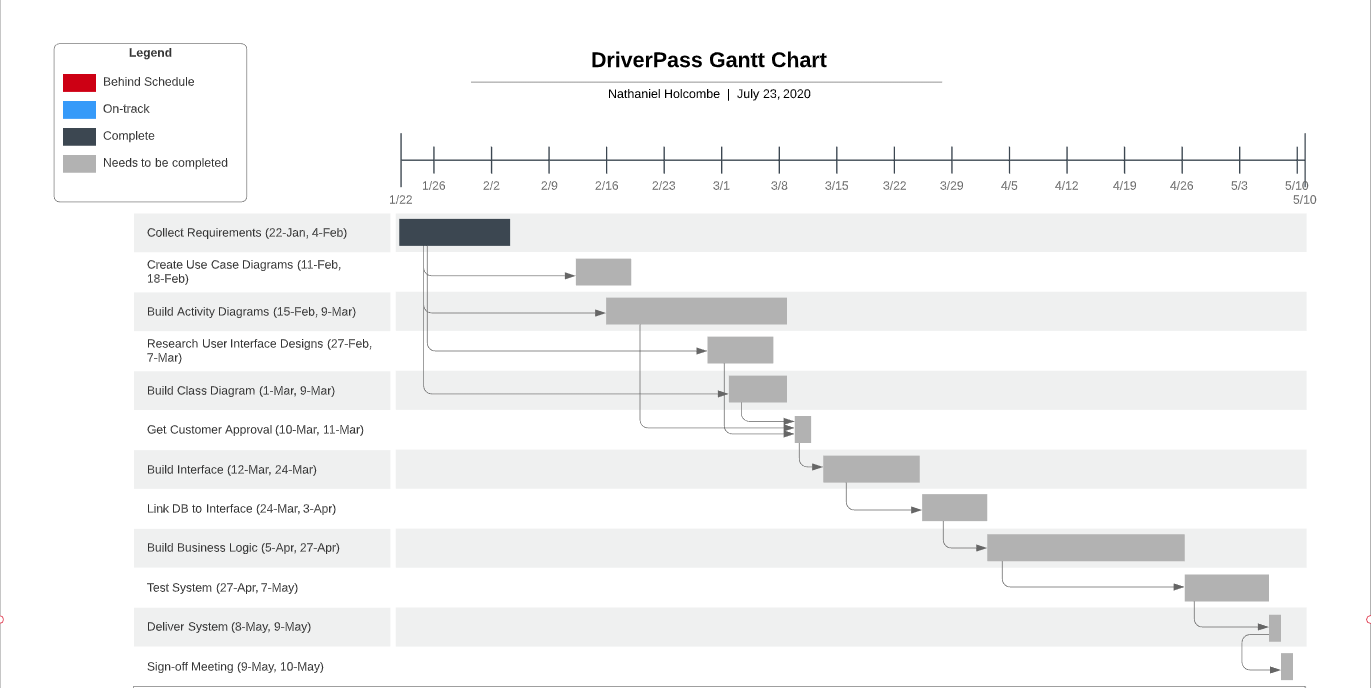
### Limitations

*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 available time to be ready to launch the system is four months.
* The system requires an internet connection to be accessible to users.
* The development team includes only five people: Jennifer, Sam, Toni, Clark, and John.
* The scalability of the cloud resources must remain within limitations for profitability.

### 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.*



References:

Swoop. (2020, April 20). Authentication API: A Company’s Guide for Passwordless Login. Swoopnow.com. Retrieved from <https://swoopnow.com/authentication-api-guide/>

Kaspersky. (2020). Brute Force Attack: What you need to know to keep your passwords safe. Kaspersky.com. Retrieved from <https://www.kaspersky.com/resource-center/definitions/brute-force-attack>

Python Software Foundation. (2020). Convenient Web-browser Controller. docs.python.org. Retrieved from <https://docs.python.org/3/library/webbrowser.html>