# 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 client DriverPass wants to provide students with access to online practice exams and on-the-road training to better prepare them for driving tests. The general functionality of this application will include:

• The application will be an online accessible application that can allow users to sign up for 3 different tiers of driving test packages.

• The application will allow multi-role user support with differing functionality, i.e. Admin users can reset passwords, block access, run reports on user activity. Regular users (Customers) can sign-up for differing driving test packages and can schedule, modify or cancel driving test training appointments. Driver users (Employees) will be able to accept, modify or cancel pending driving test appointments.

### 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?*

The client DriverPass, wants their system to bring the education of automobile driving out of the classroom and into the convenience of modern young aspiring drivers (a void in the market). The “problem” they are wanting to fix is the technicality of scheduling drivers (teachers) to their customers (students) and at the same time applying a flexible environment that will allow the customers to gain a convenient driving lesson experience.

### 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 delivered, this application will be able to:

* Customers (students) should be able to sign up for any tier of driving packages (of 3), after choosing their driving package tier and submitting their payment, the user will need to input their account information (address, name, email, phone, photo) (users can also mention any special needs that might need to be accommodated – maybe to be expanded upon later).

(Only after a customer (student) has their account information complete, can they be enabled to schedule driving lessons).

* Customers (students) once signed up for a driving package tier, should be able to sign-up for driving lessons, they will then be assigned a driving instructor.

## 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 is primarily intended to be a web-based system and hosted in the cloud. Ian and Liam specifically mentioned they need: the application to be accessible from any computer or mobile device, Users should be allowed to interact with the system (online) and finally avoiding dealing with technical aspects like backups or security – this is why cloud hosting is also convenient.

While from our interview no specific performance metrics are declared so, we can infer: Speed is important for task scheduling, report access and loading & handling of customer data. The system should also ideally have real-time responsive operations such as: booking/modifying reservations (all reservation actions would be in real-time), accessing/ updating customer profiles and viewing test progress or activities.

The system should also support regular updates, in reference to: Law updates (state/local applies), service-wide application policies, security updates, feature deployments. A modular system architecture should be applied to allow updates to be deployed with minimum disruptions to end users.

#### 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?*

Since the system is web-based and will be cloud-hosted, it will be platform independent for end users, which means the DriverPass application should work on any modern operating system (Windows, Mac, Linux, iOS, Android), with any modern browser (Chrome, Edge, Firefox). The system will require several back-end components:

1. Database
   1. I would recommend a relational database to be used for managing data. Possible choices could include MySQL, PostgreSQL, MS SQL Server (or Amazon RDS)
      1. Student data (name, address, phone, credit card information)
      2. Instructor data (name, address, phone, driver information)
      3. Reservations (Linked to students & instructors)
      4. Packages (Tier types, availability)
      5. User roles and rights
      6. Test progress data
      7. Driver notes
      8. Appointment logs
2. Auth system / Security layer
   1. Needed to manage user roles and permissions, access levels of the application (Administrator, ITUser, Secretary, Customer)
   2. Oauth implementation, JWT Tokens, session-based authentication required
3. Cloud Storage & Hosting
   1. Since we would be utilizing the cloud (minimal technical maintenance), AWS, Azure, etc. is necessary
      1. Automated Backups
      2. Security
      3. Scalability
4. Optional Tools
   1. Excel Export Functionality (for offline reports Liam wants to review).
   2. Notification system for DMV updates (e.g., via email, SMS, or in-app).

#### 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 will be able to distinguish users based on their role:

* Administrator (Liam) – full access
* ITUser (Ian) – full control over accounts (reset passwords, blocks users)
* Secretary – can create, delete, update appointments and customer data
* Customer – can register, purchase (unlock) different package tiers, books or cancels appointments, can take online tests

Each user will have a username and password. Once the user logs in, the system will utilize the user’s role to show/hide application features accordingly. The inputs will not be case sensitive (usernames/ emails, other user inputs), and for security purposes passwords will be case sensitive. The application should inform the admin of a problem on:

* Security alerts (failed logon attempts, unauthorized access)
* System failures (server down / database down)
* Any Client-UI exceptions (Show default error page, logout user, report issue)
* Any integration errors (Failed DMV issue, failed to receive data from Instructor, Student)

Any system errors that incur should be emailed to Administrators, IT users.

#### 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?*

Yes, this should be possible without modifying any code. The system should include a user management UI through the administrator mode. Administrators should be able to: Modify user data (role, contact info, etc.), Deactivate users, Add new users (new ITUser, Secretary), Be able to reset user passwords. Main functionality will be handled in separate API to there will be no need to make code updates. The system should be built using a modern approach (modular, cloud-based architecture), which will enable congruent platform updates, cloud hosting will be required (AWS or Azure configuration will be needed), they will also need access to the CI/CD Pipeline which will allow the developers to push new updates to the application (Azure Pipelines, git branching strategy, Jenkins, etc.). With these methodologies implemented, the system will have no problem adopting platform updates. The IT Admin (Ian) will need some level of administrative access (not full), Ian will specifically require:

* View, add, remove or reset user accounts
* View system logs and view report data
* Be able to utilize passwords resets for accounts
* Disable/Enable features (by tier)
* Be able to manage system settings, 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?*

For a user to be able to login, an email will be required to sign up initially. After the initial registration, a user will use their username and password to login to the system. For added security a reCAPTCHA will be added as a layer of authentication. By utilizing HTTPS, we ensure that all data exchanged between the client and server will be encrypted. By also utilizing password hashing, we guarantee to store only hashed and salted passwords in the database. Continuing, as mentioned earlier, by implementing JWT (JSON Web Tokens) we can ensure that the cookie session will be secure. As far as Cross-site scripting is concerned, enabling input sanitization on all POST methods will be way to go forward (client and server).

If a “brute force” hacking attempt was successful on an account, we could implement these defenses as deemed necessary:

1. After failing anticipated login attempts, perform automatic lock of account (20min timeout)
2. Display generic error message “don’t reveal too much to attacker”
3. Error emails are sent out to administrators
4. reCAPTCHA helps to prevent automated botting attacks

If a user forgets their password, the DriverPass system will implement a “Forgot my password” feature to allow a unique, time-limited reset link to be sent to the user.

### 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.”*

User Access & User Authentication –

* The system shall distinguish users between their roles
* The system shall enforce access rights based on their role
* The system shall allow users to reset their password
* The system shall lock users out temporarily after multiple fail logins

Customer Scheduling & Making Reservations –

* The system shall allow students to make, cancel or modify driving lesson reservations online
* The system shall allow secretaries to update or cancel driving lesson reservations
* The system shall allow instructors to accept, update or cancel driving lesson reservations
* The system shall be able to display all scheduled reservations for the students and instructors
* The system shall be able to prevent scheduling conflicts and enforce scheduling policies

Online practice test modules & features –

* The system shall display online practice tests for eligible students
* The system shall record time taken, score and result for each student

Reports –

* The system shall allow admins to export reports (e.g., student data, reservations, activity logs)
* The system shall provide print-friendly versions of activity reports, test summaries

Security –

* The system shall log all changes made to students, reservations, packages
* The system shall allow administrators to view activity reports, reservation management reports
* The system shall notify the administrator of any errors or suspicious login activity

User Management –

* The system shall allow IT users to add, update or deactivate user accounts
* The system shall store and display all student & instructor information
* The system shall allow secretaries to update student information
* The system shall allow secretaries to update instructor information

Notifications –

* The system shall notify students of upcoming lessons (via email or text)
* The system shall notify instructors of upcoming lessons (via email or text)
* The system shall notify admins of compliance-related updates (from the DMV)
* The system shall allow instructors and students to communicate with each other (via form)

### 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 DriverPass application will implement an intuitive and clean user interface design. The application will need a responsive layout (role-based layouts) and should follow current accessibility standards (ensuring usability for users with disabilities). The DriverPass application will also provide real-time validation and feedback (form input errors, scheduling conflicts). The differing functionality per users:

* Student – Customer
  + Can register for an account
  + Can login to the DriverPass system
  + Can make a driving lesson reservation
  + Can take an online practice test
  + Can change or update a driving lesson reservation
  + Can view lesson schedules, Instructor comments
  + Can submit contact forms to DriverPass company
* Instructor
  + Can Login to DriverPass system
  + Can accept, cancel or modify driving reservation
  + Can view student progress, student notes
  + Can input lesson notes, student feedback
* Secretary
  + Can Login to DriverPass system
  + Can Manage Student Profile Information
  + Can update reservations
  + Can assign students to instructors
* IT User
  + Can Login to DriverPass system (admin privileges)
  + Can deactivate student, instructor accounts
  + Can add or modify user roles
  + Can view system log data
  + Can manage access rights, role-based permissions
* Administrator
  + Can Login to DriverPass system (admin privileges)
  + Can enable or disable driving package tiers
  + Can view system log data
  + Can view financial report data
  + Can view student, instructor activity report data

### 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 believe thus far in our design implementation we have a great foundation of the DriverPass application, but we are missing some key areas that would be needed before full rollout, such as:

1. Payment processing implementation
   1. Since the DriverPass system is collecting credit card details, we should ensure that the following components are ironed out:
      1. Integration of payment gateway (Stripe, Paypal)
      2. PCI compliance with handling sensitive information
2. Legal compliance, privacy regulations
   1. Since the DriverPass design is assuming handling of sensitive information, we must ensure that the following compliances are met:
      1. CCPA (California)
3. Data backup & recovery (to be addressed: backwards compatible)
   1. Since the DriverPass application will be utilizing the cloud for rollouts, it’s important to understand how automatic backup processing will be incurred, AWS setup or local backup methodologies could be applied.
   2. Since the DriverPass application will be built modularly it could be assumed that if any rollbacks were to take place the backwards compatibility with any APIs should be ensured.

I believe that there are assumptions we are making in design, we are assuming that all students and instructors have internet access & devices (of course). We are also assuming that our mobile design will be conforming enough for modern web browsers (no additional mobile app is needed). Another assumption could also be that the DMV integration will be straight forward (working with the government could always go slow and entail numerous compliances).

### 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?*

Some limitations I can see coming from the DriverPass system design would include:

* Offline functionality is limited (The DriverPass system only allows viewing of downloaded reports, any data changes must be made online)
* Limited user customization (Say if a user wanted to upgrade their package tier they would likely need to require an elevated user with permission to update their package)
  + A later feature could be added to allow students to “upgrade” their package tier
* No advanced scheduling (No support for recurring appointments)
* No native mobile application (mobile access would be through a browser only, no dedicated application would exist)

Some other limitations I can see when it comes to time constraints could be that, based on the current timeline, the project is expected to be complete in just a couple of months, some additional time might be needed for testing (feedback + next iteration) and also with the amount of advanced features that could be added, there couldn’t be enough time in the world! Continuing, when it comes to budget limitations, I think that the current support & staff won’t have enough capital for an enterprise level infrastructure, with this being said, third-party dependencies could be very real (since there’s not enough time to develop any independent technology in house). If too many third-party dependencies incur then there could also be additional costs or delays that come with that. Lastly, when it comes to the technology limitations, I don’t think there’s any immediate concern, if the foundation is strong and we set ourselves up for success, DriverPass will never hit the technological ceiling!

### 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.*

A screenshot of a graph

AI-generated content may be incorrect.