# DriverPass Business Requirements

## 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” would like to have a system designed around their current business model. DriverPass offers driver tests and practice to students hoping to get their driver’s license from the DMV. The system that we design will allow them to expand their business to allow users access to online reservations, scheduling, and payments. The system will also allow DriverPass employees access to user data and schedules, and will handle scheduling of drivers to users.

#### 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 has noticed a potential niche in the market, where drivers hoping to take their driver skills test at the DMV are ill-prepared and do not pass. Offering better practice tests and driver training, DriverPass hopes to capitalize on these user’s needs.
* A system to support this would require the ability to store user information in accounts, and allow those users to schedule driver tests or driving practice. The system will let users track their progress, and make payments online for the services they purchase. The system will also allow DriverPass to track its own employees and the cars they drive for training, as well as users and their data. The entire system will be cloud-based, and accessible from any type of device with an internet connection.

#### 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?*

| **Objective** | **Task** |
| --- | --- |
| House a database of users | Create a function that allows users of the system to input personal information into a secure account that can be accessed via log-in.  Create a secure database that is housed in the cloud. |
| Intuitive User Interface | Get customer approval for overall design before user interface can be built  Create a UI that shows the client-user information about their account, including personal information and progress tracking, notes from their driver, scheduled services, and their photo  Create an admin UI that shows the admin-user options for scheduling, car selection, user payments processed, and other user client-user information |
| Automatic car/schedule matching for DriverPass employees | Create a function that organizes employees and cars that are currently at DriverPass.  Create a function that matches available employees with available cars and schedules them to train drivers on the user requested days. Allow users to override the automatic selection through modification function. |
| Roles for DriverPass employees | Create a parameter that distinguishes employee accounts from general user accounts.  Create a parameter for each type of employee that will be using the system and the permissions that go with that type. |
| Automatic system updates from DMV | Create a function that regularly checks for DMV updates via the DMV website/database.  Create a notification service that updates administrator accounts of DMV updates. |
| Cloud access for all users | Design the system to be used in the cloud.  Create a function to compile specific data sets to be downloaded to a csv spreadsheet file.  Create a function to upload modified data to the cloud and update cloud data with most recent modifications. |
| User profile interactions for scheduling, pay, etc. | Create a function that allows users to access their schedule of services purchased, make changes when necessary, cancel, or pay for services.  Users will be able to purchase any of three different packages offered.  Create a function that displays user progress on a “dashboard” type of user interface.  Create a service that connects to a map repository online for setting pick-up and drop-off locations when a user purchases a service. |
| Link DB to interface | Build UI per research gathered by Toni and Clark, prepare UI to be linked to DB |
| Test System | Testing team will use static and dynamic tests to ensure the system performs correctly (10 days) |
| Deliver a finished product | System testing with the test-team must be finished before the product can be delivered |

# Requirements: Nonfunctional Requirements

#### 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 DriverPass system will need to run on both PC’s/Mac’s, as well as smart phones. For this to work, a web app should be developed that runs in a typical PC web browser such as Firefox or Chrome, while a smartphone app should be developed to handle mobile access.
* The system will need to run fast enough to support video streaming, as that is one component of the DriverPass business model. User’s bandwidth will ultimately determine the quality of the stream, but the system should be able to handle multiple users streaming driver’s education videos at once.
* The system will need to be updated at least as often as the web browsers that run it are updated, as well as when the mobile operating systems of the respective smartphones are updated. That is, if Google Chrome pushes an update to its users, then the DriverPass web app should be tested on the new update, necessary fixes made, followed by a system update to support the browser update.

#### 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?*

* If the DriverPass system is developed for web browsers using HTML as the delivery language, then any web browser-supporting operating system will be supported by default. The mobile application will need to be supported by Android and iOS at least. While other niche mobile operating systems exist, marketing should focus on having those users use the web application until a dedicated application can be developed (provided market research supports this).
* In order to store user information, both for clients and administrators (drivers included), a database will need to be developed and linked to the DriverPass interface. Depending on the developers preferences, a backend development tool will be necessary as well, such as ASP.NET Core or Node.Js. For the server to handle database calls, SQL is a standard tool.

#### 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 have a unique username that is used to access the site. The usernames will be housed in a database. When a new user attempts to create an account using a username that already exists, an error will prevent account creation and the user will be informed of the error. Usernames and passwords will be case sensitive, so user “Bob123” will be different from user “bob123”.
* Any time an error is logged within the system, that error should be forwarded to admin, as well as added to a log. Some items to inform admin about are processes consuming more resources than expected, CPU resources being improperly used, memory issues due to cache size, zombie processes, load average issues, and any issues with disk read/write operation speeds.

#### 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?*

* Since users will be objects created within the system and stored in the database, the ability to add/remove/modify users without changing system code will apply. If implemented properly, the interface will ensure the entire system does not need to be refactored in order to update one class.
* IT will need administrator access to maintain the system properly. They will have access to usernames and some user information, with the ability to update some user information, but they cannot see the information that they will be changing. For example, a user may wish to update payment methods; IT support will be able to take new payment information and inject it into the user’s current account, but they will not be able to access the previous payment information, nor will they be able to see the new information once it has been submitted to the system.

#### 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 require a unique user ID as well as a “strong” password (standard character length 10+, capitals and lowercase, numbers, and symbols)
* Data transfers will take place via HTTPS over TLS to ensure data is secure
* Users will be given multiple attempts to access their account in the event of a forgotten password, but after a certain number of attempts their account will be locked and they will be required to contact IT for a password reset (either directly or via “forgot password” link).
* The above policy will thwart a brute force attack from outside the system via user log-in.

# Requirements: Functional Requirements

#### System Functions

*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 credentials when logging in
* The system shall support two-factor authentication for all users
* The system shall keep track of user’s progress in testing
* The system shall store user information in private accounts (driver notes, special needs, driver photo, address information, payment information)
* The system shall allow users to schedule driver tests and driving practice through one of three pre-loaded packages
* The system shall allow users to make payments online
* The system shall track employees of the business
* The system shall track when cars that employees drive are scheduled to be used
* The system shall track user data and progress

#### 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 system will be accessible from any type of internet connected device that allows user interaction (phone, tablet, PC). Users of the application will be customers, employees, and administrators.
* Customer-users will be able to log-in via a credentials input, check their account status and progress, check their account balance, and schedule driver tests and practice. Customer’s will also be able to update profile and account information, and update billing information
* Employee users will be able to check their schedule, and check the schedule of cars they are to drive. Employees will be able to access their own employee-profile and make updates to it, including their availability
* Admin users will have access to most user and employee information, and will have the ability to reset passwords and unlock accounts. Admin users will be able to see the schedule of employees, make adjustments, and add / remove employees and other users from the system. Admin users will get updates about driver-test related items from the DMV. Admin will also be able to schedule users’ appointments for driver tests and practice.
* The interface will be touch or click based, with keyboard support for typing in names and other pertinent information. The interface will be simple 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?*

* It is assumed that all users will access the application via a modern device, i.e. smartphone or PC, and through an updated web browser.
* The coding language used to write the application is not known yet
* The color scheme or aesthetic of the final site is not known yet, as well as what settings will be available to users to enhance their experience on their preferred device (color-blind friendly, slow connection friendly, high-contrast view, etc)
* Where the driver-test videos will be hosted has not been established yet; if they are hosted on the DriverPass server then that will take up bandwidth and space, while hosting on a free-use third-party site will reduce bandwidth but may incur extra costs or create other issues (hosting on YouTube, for instance, will have ads play that may not be in line with DriverPass’ business model).

### 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 will arise from the possible time frame versus team size. If the development team is relatively small, then putting together a full system in a short amount of time could cause some features to be cut or minimized.
* Depending on the database host and their allotted bandwidth, the user interface would need to be adjusted to account for lower bandwidths with smaller pictures or less animations.
* The budget has not been established for this project yet. If the project starts to run over budget, a meeting will have to be held with the DriverPass CEO to renegotiate requirements or funding.

### 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.*

