

## CS 255 Business Requirements Document

### System Components and Design

#### Purpose

This project is for our client DriverPass, which is a company that wishes to “take advantage of a void in the market when it comes to training students for the driving test at their local department of motor vehicles” (Southern New Hampshire University, n. d.). Their purpose is to help students pass their driving tests by offering services like training, practice tests, and access to online courses with content and material (Southern New Hampshire University, n. d.). The purpose of the project is to help DriverPass to create a system which allows them to achieve this goal.

#### System Background

- The problem which they would like to fix is the fact that “so many people fail their driving tests at the DMV” (Southern New Hampshire University, n. d.).
- The client asked us to help build a system that enables driving students to access a series of services including driving training, practice tests, online courses, and the ability to schedule in-person training sessions with driving instructors and driving tests at the Department of Motor Vehicles.
- The system will provide an intuitive interface to students where they can access all of these services online.
- The system will consist of a website or application where students can access these services via the internet.

#### Objectives and Goals

The system will provide the user with the ability to access the following services (Southern New Hampshire University, n. d.) online:

- Schedule in-person training sessions with driving instructors
- Access practice tests designed to prepare students for their driving tests
- Access online courses designed to prepare students for their driving tests
- Schedule their actual driving tests at the department of motor vehicles
- Choose their package based on their desired features and the price difference between packages

### Requirements

#### Nonfunctional Requirements

According to the specifications stated by DriverPass owner Liam, “the system needs to run off the web” (Southern New Hampshire University, n. d.). In other words, the system must be able to run in a web-

based environment. Liam would also like to be able to enable or disable packages offered to DriverPass customers, or make packages available or unavailable as needed. Liam would also like to implement an intuitive interface for students to easily access DriverPass services and navigate the web page. Ian, the DriverPass IT Officer, stated that DriverPass has “different employees at the company with different rights and roles. For example, for me I need to have full access over all accounts so I can reset them if someone forgets their password, or if we let go of someone and I need to be able to block their access” (Southern New Hampshire University, n. d.). In other words, access control will be a necessary security requirement for this system. We assume that DriverPass would also like for the system to include other security features such as user authentication and the encryption of sensitive user data such as personal information and credit card and bank account numbers. We also assume that DriverPass would like for the system to be able to accommodate as many users as needed without wasting unnecessary resources during times when less users are accessing the system. According to these needs, these are the non-functional requirements for the DriverPass system:

## Performance Requirements

- Accessibility
- Scalability
- Efficiency
- Usability
- Accuracy
- Adaptability
- Security

## Platform Constraints

- Considering the non-functional requirement of accessibility, this web-based system must be accessible through users’ browsers (TechTarget, n. d.). While virtually all browsers are able to access web applications, the application should be tested within a variety of commonly used browsers such as Chrome, Firefox, Safari, Edge, and Opera.
- Considering the non-functional requirement of Scalability, many cloud platforms such as AWS have an incredibly powerful scalability tool called “load balancing,” which allows web applications to spread workloads across a series of virtual servers as needed. This way, DriverPass can easily scale up or down in order to control expenses while maintaining constant and reliable functionality for their users. Linux is an inexpensive and highly adaptable web server operating system option, which is probably why “roughly 80% of all servers use some variation of Linux” (PhoenixNAP, 2022). For this reason, the DriverPass system should run on a virtual AWS cloud Linux server. In order to address the requirement of scalability in terms of the ability to access and query user information, MongoDB is our database of choice. “MongoDB Atlas makes scaling as easy as setting the right configuration” (MongoDB, n. d.).
- Considering the non-functional requirement of efficiency, the system should be set up in a way which will optimize the resources used by the system. The variety of cloud tools available to AWS clients which can make the system more efficient in this way makes AWS the best cloud platform on which to run the DriverPass web application. In order to further enhance the efficiency of the system in terms of the ability for developers to make modifications to the system, the AWS cloud-based Linux servers hosting the application should use the Vim text

editor. Vim is a back-end tool known for giving developers efficient methods for “complex tasks with just a few keystrokes” (Kedzierski, 2023).

- Considering the non-functional requirement of usability, the DriverPass web application should include an intuitive and easy to use. We will model this interface after the sketch given to us by Liam.
- Considering the non-functional requirement of accuracy, the system must have the ability to distinguish between users and to inform admins of any problems as soon as they are identified.
- Considering the non-functional requirement of adaptability, the ability to enable or disable packages offered to users should be made available to Liam as per his request. Contrary to the assertion that the system “will still need a developer or a system analyst to add or remove modules” (Southern New Hampshire University, n. d.), and that the system “can’t be built in a way that a nondeveloper can do easily” (Southern New Hampshire University, n. d.), a separate interface can be made which will only be accessible by administrators where Liam or any of his employees can easily modify packages available to their customers.
- Considering the non-functional requirement of security, the system must include user authentication methods and access control so that different users have different privileges and are able to access different resources within the system.

### **Accuracy and Precision**

- The system will distinguish between different users using the common method of user authentication where each user is assigned (or chooses) a unique username and password. When a user is authenticated by entering their password, their username can be queried by the system from the database so that the system has access to the user’s information which can include their name, email address, billing information and their active package if they have purchased one.
- Passwords should be case sensitive because the presence of two possible characters for each letter (uppercase and lowercase) adds to the possible characters used in a password and therefore increases the difficulty of cracking a user’s password. Case-sensitive usernames, on the other hand, could cause confusion for users. For example if usernames “john mcadams,” “John Mcadams,” and “John McAdams” can all exist simultaneously, then it can be more difficult for a user to remember their username. Email addresses are also not case-sensitive, so if the email address is used as a username then username case sensitivity could confuse the system.
- The system should inform admins of a problem whenever an error occurs within the system, or whenever data queried from a database is different than what is expected. Users should also be able to report a problem if necessary and those reports should be sent directly to admins or those on the administrative side of the system.

### **Adaptability**

- As per Liam’s request to be able to enable and disable packages offered to customers, Liam should be able to make these changes without the need for intervention from developers. As previously mentioned, contrary to the assertion that the system “will still need a developer or a system analyst to add or remove modules” (Southern New Hampshire University, n. d.), and that the system “can’t be built in a way that a nondeveloper can do easily” (Southern New

Hampshire University, n. d.), a separate interface can be made which will only be accessible by administrators where Liam or any of his employees can easily modify packages available to their customers without having to interact with the code.

- In order for the system to adapt to platform updates, it would be helpful for developers to try to orient the systems usage of the platform as much as possible to functionalities which are unlikely to be discontinued or changed drastically in the near future. Companies providing platform components also often announce changes to be made, or if they will cease providing support for a specific component of the platform. Developers should stay informed regarding these announcements to make sure that they begin adaptation to any system updates as early as possible.
- The IT admin will need as much access to the system as possible without giving them the ability to involuntarily do damage to the system. The specifics depend on the level of technical expertise of the admin. For example, an admin who is not a seasoned programmer should probably not have the ability to change the actual code which makes up the system. Admins should have the ability to make changes to user accounts as needed in case a user becomes unable to access their account, or if users violate codes of conduct or rules set forth by DriverPass. As an extreme example, if a user acts abusively toward a driving instructor or constantly misses appointments then the admin should have the ability to either lock or shut down their account.

## Security

- A user must specify their username (or email) and enter their correct password before being allowed access to the system. Two-factor authentication is another method which would further secure a user's account and prevent unauthorized access. Since some users may find two-factor authentication to be tedious and unnecessary, any two-factor authentication included in the login process should be optional.
- In order to secure the connection and data exchange between the client and the server, the application will use end-to-end RSA encryption. This way any malicious actors who are able to intercept information going from the client to the server or from the server to the client will be unable to decipher the information. This is extremely important since sensitive user information will need to be communicated to the system such as credit card and/or banking information, as well as personal information.
- If the user forgets their password, then they will have the option to have the system send an email to the email which they used to create their account containing a link which leads to a page prompting them for their new password. If the user is unable to access that email account, then the user must prove their identity in order for their password to be reset by an admin. Methods for identity verification must be limited to state-issued identification cards or documents in order to prevent forgery leading to social engineering attacks. An example of such a social engineering attack could involve a malicious actor contacting DriverPass claiming to be a user who had been unable to access their DriverPass account or the email account used to create their DriverPass account.

## Functional Requirements

- The system shall validate user credentials when logging in.
- The system shall allow users to schedule in-person lessons explaining DMV rules and policies.
- The system shall allow users to schedule in-car lessons with driving instructors.
- The system shall allow users to access online driving courses.
- The system shall allow users to access practice driving tests.
- The system shall allow users to schedule their driving test at their local DMV.
- The system shall allow admins to add, remove, or edit the DriverPass packages being offered to users.
- The system shall provide a user interface for users to interact with the system.

### **User Interface**

- The interface should include a homepage which is visually appealing and which provides an intuitive way for users to navigate the web application including an 'About' page explaining DriverPass' goals and the features of their web application, a 'Packages' page offering different packages to users, a 'Contact' page for users to contact the company itself. The interface should also include separate pages for each of the services provided by DriverPass to their customers including a page for scheduling driving lessons with instructors, a page for scheduling one's driving test at the DMV, a page where users can take practice tests, and a page where students can access the online training courses offered by DriverPass.
- According to Liam's requirements stated in the driverPass interview, the different users for the interface will include customers, developers, and administrators. Different types of users should also be specified in order for users with more expensive and inclusive packages to have the ability to access tools which are unavailable to users who decide to purchase less expensive packages.
- Users should be able to access online training courses, schedule in-person training sessions with driving instructors, access practice tests, and schedule their driving test appointment at the DMV. Users should also be able to choose their package based on their desired features and the price difference between packages.
- Since this will be a web application, the user should be able to interact with the interface through their web browser (TechTarget, n. d.) of choice.

### **Assumptions**

- We assume that DriverPass has a specific budget and would like for us to develop the system as cost-effectively as possible while still meeting the requirements of the client.
- We assume that DriverPass would like for the system to be able to accommodate as many users as needed without wasting unnecessary resources during times when less users are accessing the system.
- We assume that DriverPass would like for the system to include other security features such as user authentication and the encryption of sensitive user data such as personal information and credit card and bank account numbers.
- We assume that most users will interact with the system through their desktop, but that some users will interact with the system through mobile devices.

## Limitations

- Since the DriverPass web application will run within a web browser, the system will not be as easy to use through mobile devices due to the limited browser screen space available on such devices.
- According to the timeline discussed during the schedule planning meeting between Sam and Jenniffer, the development process should take 16 weeks. After reporting this timeframe to the client, this length of time should not be lengthened too much so that the client's planning done in consideration of this timeframe will not be heavily affected by any extensive delays.
- A specific budget was not discussed in the DriverPass interview, but the development team should take care to allocate time and resources efficiently in order to reduce the cost to the client as much as possible in order to increase the chances of client satisfaction and repeat business.

## Gantt Chart



**Sources:**

Kedzierski, T. (2023, October 30). *Top 25 Back-End Web Development Tools*.

<https://www.netguru.com/blog/top-back-end-web-developement-tools>

MongoDB. (n. d.). Database Scaling. <https://www.mongodb.com/basics/scaling>

PhoenixNAP. (2022, March 10). *Server Operating Systems: Server OS Types & How to Choose*.

<https://phoenixnap.com/kb/server-operating-system>

Southern New Hampshire University. (n. d.). *CS 255 DriverPass Interview Transcript*.

[https://learn.snhu.edu/content/enforced/1546013-CS-255-J9874-OL-TRAD-UG.24EW4/course\\_documents/CS%20255%20DriverPass%20Interview%20Transcript.pdf?ou=1546013](https://learn.snhu.edu/content/enforced/1546013-CS-255-J9874-OL-TRAD-UG.24EW4/course_documents/CS%20255%20DriverPass%20Interview%20Transcript.pdf?ou=1546013)

Southern New Hampshire University. (n.d.). *CS 255 Sample Business Requirements Document for an ATM*.

[https://learn.snhu.edu/content/enforced/1546013-CS-255-J9874-OL-TRAD-UG.24EW4/course\\_documents/CS%20255%20Sample%20Business%20Requirements%20Document%20for%20an%20ATM.pdf?ou=1546013](https://learn.snhu.edu/content/enforced/1546013-CS-255-J9874-OL-TRAD-UG.24EW4/course_documents/CS%20255%20Sample%20Business%20Requirements%20Document%20for%20an%20ATM.pdf?ou=1546013)

TechTarget. (n. d.). Web Application (Web App).

<https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app>