**Project One: Business Requirements Document**

Ryan Rieth

Prof. McHann — CS 255 — SNHU

Abstract

[System Components and Design 3](#_Toc115289206)

[Purpose 3](#_Toc115289207)

[System Background 3](#_Toc115289208)

[Objectives and Goals 3](#_Toc115289209)

[Requirements 5](#_Toc115289210)

[Functional Requirements (*more may be necessary deeper into product development*) 5](#_Toc115289211)

[Nonfunctional Requirements (*more may be necessary deeper into product development*) 5](#_Toc115289212)

[Performance Requirements (*more may be necessary deeper into product development*) 6](#_Toc115289213)

[Assumptions 6](#_Toc115289214)

[Limitations 6](#_Toc115289215)

[User Interface 6](#_Toc115289216)

[Gantt Chart 8](#_Toc115289217)

## 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 Liam, the owner of *DriverPass*, a company intentioned to properly prepare students for their driving tests as their local DMV via online educational modules, practice tests, and requested on-the-road training.
* The purpose of the project is to create a server-based application service that accomplishes the mission of *DriverPass* while also meeting Liam’s goal of he and any other admins being able to access the service’s information from “anywhere”.

### 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?*

* Liam wants *DriverPass* to buttress students’ current efforts to learn how to drive and secure a driver’s license from their local DMV. Liam states that too many students are failing their driver’s tests and wants to fill in a void regarding properly preparing future drivers.
* Liam wants *DriverPass* to be cloud-based so that backing up and storing data has minimal interaction and headaches. Beyond this, *DriverPass* should be able to be accessed from online — including web browsers — for every student after they have signed up. The sign-up process is old-fashioned: a phone call with a secretary to input personal information. From here, the signee should be able to choose between three learning packages that Liam can “delete” if they are full; schedule/delete the aspects of these packages per their leisure; and keep track of vehicles/trainers that are reserved and not reserved. Liam also wants the system to record changes made such as reservations confirmed, cancelled, and modified, as well as itemize this information for printing.
* Servers and cloud-based systems management are required for this system. A secretary is also required, as is a systems analyst specialist for future modular changes to the system that were mentioned but not requested in the *DriverPass* interview transcript. A web-based version of the application as well as a mobile-based version must be compatible and must communicate with the online servers and cloud-based storage. At least three different interfaces must be available: the admin version, the trainer version, and the student version; each interface will allow or not allow access to aspects of the system (respectively, most access to least access).

### 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?*

STUDENT SIDE

* Create student account.
* Store student’s personal information.
* Delete account.
* View and choose packages.
* Schedule sessions allowed in package and available via telephone or online application.
* Un-schedule sessions.
* Delete/update chosen package(s).
* Showcase vital information regarding session start time, length of time, instructor, etc.

CLIENT SIDE

* Record changes made to account, including but not limited to updating personal information; scheduling, deleting, or modifying sessions; etc.
* Print recorded information for analysis.
* Remove package(s) from viewable website due to number of students.
* Allow access to information on local devices.
* Allow access to private information for administrators.

DEVELOPER SIDE

* Build system for modularity to future proof accessible/offered elements.
* Communicate with in-house servers.
* Communicate with third-party servers such as DMV to keep up to date.
* Store information in cloud-based network.
* Privatize/encode information for security’s sake.

## Requirements

### Functional Requirements (*more may be necessary deeper into product development*)

* *The system shall…* allow downloadable content for administrators from a computer or mobile device.
* *The system shall…* support roles with differing levels of functionality and information access.
* *The system shall…* store and manage lesson materials in electronic and in print forms.
* *The system shall…* validate student credentials when logging in.
* *The system shall…* allow changes to user information.
* *The system shall…* allow an administrator to view student records and information.
* *The system shall…* store student driving data such as lessons and driving scores.
* *The system shall…* display unique identifiers for each lesson/package.
* *The system shall…* connect driving lessons to driving instructors.
* *The system shall…* connect lesson material to driving lessons.
* *The system shall…* provide instructor the ability to score student’s driving lessons.
* *The system shall…* prevent registering for a lesson package if the package is deemed full.
* *The system shall…* allow students to self-register in driving lessons/packages.
* *The system shall…* send confirmation notifications to students upon registering.
* *The system shall…* track student’s progress through lessons.
* *The system shall…* verify a student is not double enrolled in a lesson.
* *The system shall…* verify a student is not enrolled in a lesson that has already been taken.

### Nonfunctional Requirements (*more may be necessary deeper into product development*)

* *The system shall…* have unique lesson numbers
* *The system shall…* relate various functionalities or lessons to respectively tiered packages.
* *The system shall…* exist with a public cloud-based environment.
* *The system shall…* notify any system-affecting changes or downtimes ahead of time.
* *The system shall…* continue operating as normal during updates or support.
* *The system shall…* provide reports upon request.
* *The system shall…* exhibit material in various compatible file types such as Word and PDF, .mov and .mp4, .jpg and .png, etc.
* *The system shall…* support closed captioning.
* *The system shall…* connect to help service database with information and a contact page.

### Performance Requirements (*more may be necessary deeper into product development*)

* *The system shall…* support access via computer devices.
* *The system shall…* support access via web browsers.
* *The system shall…* support access via mobile devices.
* *The system shall…* be able to retrieve current and past student information within a few seconds’ time.
* *The system shall…* be able to expand or limit capacity of instructors as necessary.
* *The system shall…* pull from an update database every hour.

### 

### Assumptions

* Users provide accurate information.
* Users set up accounts properly.
* Users’ machines have at least minimal recommended specs.
* Users have internet access either by WiFi or LTE connectivity.
* Users consistently partake in lessons.

### Limitations

* Preferred 16-week development lifespan.
* Small development team.
* Client with nuanced aspirations.
* Compatibility across platforms not guaranteed.
* Old-fashioned secretary position necessitated for most student account creation.
* Client wants most maintenance of website to be handled behind-the-scenes.
* Design is module-based sanctioned into a three-tiered package system.

#### Platform Constraints

* *The system shall…* run on Windows and Mac OSX.
* *The system shall…* have a backend support database for students.
* *The system shall…* have a cloud-based information database for students and instructors.

#### Accuracy and Precision

* Users will possess their own unique profiles authenticated by user ID and password.
* Input for users will be case sensitive.
* Admin should have access to system records and information at all times from home-base and remote locations.
* System should notify the admin of any problem as instantaneously as possible.

#### Adaptability

* System must support modularity so that admin can adjust offered packages or offer more, newer packages at any time.
* Lessons must be able to be adjusted, removed, or added at any time.
* User must be able to make changes to their account either online or over the phone.
* System should be able to run during minor patch upgrades or updates.
* The IT administrator needs access to all various roles and their respective functionalities as well as the cloud-based information system and help database system.

#### Security

* User ID and password are required for the user to log in.
* Two-Step verification must also be met to log in.
* Client-to-server information must be secured via SSL or TLS security measures.
* Account information should be encoded and encrypted upon a brute force hacking attempt.
* User must be able to easily reset their password via a usual *Forgot Password?* manner or by calling in to the secretary.

### User Interface

* The interface needs a modular design that very quickly displays accessible information to users who have permissions to access the information.
* The different users are Administrator, Instructor, and Student.
* Administrator must be able to access all information from the other two user types as well as make changes to the system’s functionalities and offerings as well as make detailed reports of the systems operation and changes made to the system either by instructors or students.
* Instructors must be able to access similar features as Students but have more ability to make changes to lessons they cover, or information contained with lessons. They must also be able to link or unlink themselves to lessons. Instructors must be able to see all their students’ progress.
* Students must be able to register for lessons or sign up for packages that are available. They must also be able to see their progress and history as they navigate through each lesson. They must be able to create their own account either online or by phone.
* The user must be able to interact with the interface in various capacities ranging from point-and-click (computers and web browsers) to drag-and-touch (mobile devices). Each device must be able to adjust the layout to properly showcase the information in an easy-to-read but equally informative manner. Beyond this, navigating the system must be very similar across all different types of devices even if the physical interaction is different (such as clicking or touching).

## Gantt Chart

