Southern New Hampshire University

CS 255 Business Requirements Document

Brooks Maerder

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 give easy access to driving lessons, online classes, and practice tests to users looking to take their driving test at the DMV.
- They want their system to give users the ability to sign up for driving lessons, online classes, and practice tests via web interface.
- They want role based permissions, in particular, they want to give the IT Officer full access over accounts for retrieval of lost data(e.g. usernames and passwords), and for closing inactive accounts.

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 is attempting to make an impact on the high rate of failure on drivers tests among users by giving easy access to lessons and tests.
- They need their system to be able to register users either via phone call, in house visit, or online registration.
- They need users to have the ability to purchase one of three package plans, the bottom tier giving 6 hours of driving time with a driver/car, the middle tier 8 hours and an in person lesson on DMV policy and regulation, and the top tier with 12 hours, the in person lesson, and access to online courses and practice tests.
- They need the system to be able to track which users modify, cancel, and make reservations, at what time and date, and with which driver.
    - They also mentioned there are 10 drivers, so the system must not allow more than 10 appointments to be scheduled at the same date and time.
- They want their interface to display user info, test progress, previous lessons + notes from them, any special needs, and photos of user + driver.
    - They also want a page for contacting management/support team, and a way for the support/management to contact a user.

- They want the interface to be managed on the cloud, the ability to access anywhere(mobile phone, home desktop, etc..)
- Lastly, they need the system to be compliant with DMV poilcy and regulations, getting notified of any updates on their end to update the system accordingly.

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?

- The system needs to be able to create or register users, which can be measured by creating a sample user.
- The system must allow users to purchase a package(all 3 packages must be tested), and schedule a reservation via web interface, which can be tested with the sample user once registration is tested successful.
    - The system must enforce a maximum of 10 concurrent reservations at any given time slot
- The system must give access to the owner to access user data and delete accounts, and track reservation details such as userID/username, date and time, and driver assigned, which can be tested by attempting these procedures on a sample user.
    - Must also ensure that all of this functionality remains congruent across all platforms (mobile phone, home desktop, etc…)
- Using one of the sample users, sample drives must be entered, as well as take a sample test or two, to ensure the main interface functions properly and displayed test progress and prior lessons + notes.
    - The contact page must be tested using a sample user and check with the company to ensure they got the sample email and vice versa, have the company contact a sample user.
- DMV compliance can be tested by issuing a sample update of policy and regulations passed ot the system as if it came from the DMV to ensure the system receives these updates and the system can be updated in compliance.
    - This test update MUST be revoked before release to ensure compliance with DMV regulation and policy

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?

- System should be web based, and cloud hosted, accessible via both desktop/laptop computer and mobile devices
- System should support many concurrent users without any performance issues
- System should allow users to export reports and data for use off site or offline.
- System should stay up to date with current DMV regulations immediately upon release.

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?

- System should support use on modern web browsers on Windows, Linux, macOS, IOS, and Android systems.
- System should be supported by relational database to store customer, reservation, training, and testing data.
- System should be hosted on a secure cloud environment to handle backups, updates, and scaling.

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 should enforce role based access to distinguish between admins, drivers, and customers.
- System should log all activity including tests, drives, purchases, etc… with timestamps and userIDs.
- System should inform admins of failed logins, any scheduling errors that may occur, and any data mismatches/errors in database.
- Input fields should be format and/or case sensitive when appropriate(I.e. usernames, passwords, card info).

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?

- System should allow IT admin to create, update, and delete user accounts and appointments.
- System should allow enabling or disabling of packages available to customers without redevelopment.
- System should be flexible to allow DMV regulation changes and updates

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?

- System should require secure and valid login credentials for all users.
- System should user secure connections for all data transfers and transactions.
- Card data should be stored securely in compliance with PCI standards.
- System should lock user accounts after numerous failed attempts.
- System should allow users to reset forgotten passwords through email verification.

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."

- The system shall allow customers to register and manage their personal accounts online.
- The system shall allow customers to book, modify, and cancel driving lessons online.
- The system shall allow administrators to book, modify, and cancel driving appointments online on behalf of the customer.
- The system shall match each customer appointment with a specific/unique driver, car, date, and time.
- The system shall allow purchase of three packages(6 hours, 8 hours, 12 hours) with content specific to each package.
- The system shall allow administrators to disable packages so that no new customers may enroll if disabled.
- The system shall provide online learning lessons/materials and DMV practice tests.
- The system shall track customer individual learning progress including test scores and completion dates.
- The system shall allow drivers to record lesson start and finish times, and make comments on the portal.
- The system shall record and display registration details, contact information, pickup/drop off location, and payment details.
- The system shall log all activity including test scores/times, driving lesson history(scheduling/cancellations) and purchases.
- The system shall connect to DMV systems to receive updates on rules, regulations, and policies, as well as notify administrators of these updates.
- System shall provide a contacting platform for drivers and customers to communicate when necessary.

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.)?

- Customers must be able to register, login, purchase packages, schedule lessons, and take practice tests easily through the interface. The interface must clearly display individual test progress, lesson history, and driver notes for customers.
- Drivers must be able to easily log lesson times, contact their customers if necessary, and make comments on each lesson.
- Admins must be able to create, delete accounts, or modify accounts easily as well as disable/enable packages, and modify/create/cancel driving appointments through the web interface.
- The system must be web based, user friendly, and easily accessible for both users on desktop/laptop computers through web browser and mobile users through an app(iOS/Android).

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?

- Users will have secure connection to the internet for online services.

- Customers will have access to modern web browsers and have basic computing skills for navigating system components.
- The DMV will provide timely updates to DriverPass in line with any policy/regulation changes.
- Secure cloud hosting services will provide timely backups and maintenance on infrastructure.

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 system can not be modified by non tech staff beyond disabling/enabling packages. Any further changes will require support from development team.
- Offline access will be limited to exporting reports on system data.
- System performance is reliant upon user connection and availability/performance of cloud hosting services.
- Budget and timeline may limit scope of features upon initial release, further features will require updates upon the initial system release.

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.

Task	Duration(Days)	Start Date	Finish Date	January	Feburary	March	April	May
Collect Requirements	14	Jan 22	Feb 4					
Create Use Case Diagrams	7	Feb 11	Feb 18					
Build Activity Diagrams	22	Feb 15	Mar 9					
Research UI Designs	8	Feb 27	Mar 7					
Build Class Diagrams	8	Mar 1	Mar 9					
Get Customer Approval	1	Mar 10	Mar 11					
Build Interface	12	Mar 12	Mar 24					
Link DB to Interface	10	Mar 24	Apr 3					
Build Business Logic	22	Apr 5	Apr 27					
Test System	10	Apr 27	May 7					
Deliver System	1	May 8	May 9					
Sign Off Meeting	1	May 9	May 10					