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?*

* Provide new drivers with a resource to practice and prepare for driving tests through online tests and on-the-road training.
* The clients are Liam and Ian of DriverPass. DriverPass is their new company whose goal is to fill a void in the market when it comes to providing student drivers with resources and opportunities to ensure they get their license through an easy-to-use, online-provided course.

### 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?*

* Manage user’s training.
* Match users to instructors.
* Keep track of user, instructor, and car groups.
* Give users access to:
  + Register
  + Training materials.
  + Tools for making and managing appointments.
  + Progress tracking
* Give admins access to revoke permissions, manage users and employees, disable access, manage package availability.

### 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?*

* Be accessible for online and offline use including practice tests, and course materials.
* Handle scheduling between users and instructors.
* Make, modify, cancel appointments.
* Secure data.
* Authenticate users.
* Update with changes to DMV rules, policies, and sample questions.

## 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?*

* Web-based:
  + No server maintenance.
  + Automated backups.
  + Security.
  + Reliable.
* Support for different browsers, aspect ratios, rendering engines, etc.
* Update to any change from users and admins promptly.
* Regularly query the DMV for any changes.

#### 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?*

* Cloud-based application
* Unix-based OS – Linux or other Linux Distros like Ubuntu, Alpine, or Debian
  + No licensing costs.
  + Minimal hardware needs
  + Customizable and flexible
  + Compatible with many cloud platforms and tools.
  + Scalable
  + Portable – change cloud service provider, reproduce exact OS state for development.
* Cloud host providers, Microsoft Azure, AWS, …
* An SQL database like MySQL for managing data (CRUD) and linking data.
  + Form relationships between data to limit user permissions.
  + Modify those relationships as needed.
  + Prevent redundancy and duplication.
  + Is a relational db so data can be exported to CSV then imported to Excel.
* Possible Redis use for reducing webpage load times through caching and to send notifications, queueing actions.
* PCI compliant third-party company to handle transactions and credit card information.
  + Less risk, less to keep track of, no audits. Third-party companies specialize in this type of security.

#### 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?*

* Full name associated with email – use original case.
  + Maintain user preferences for their name.
* Unique emails – use original case.
  + Store original case and lowercase form.
    - Send emails to original case.
  + Case sensitivity is rare but can still happen. Allows those rare cases to sign up an receive emails.
  + Use canonicalized form of the user entry for login.
* Passwords – use original case.
  + Case sensitive – must meet requirements (e.g., character count, symbol, number, capital letter, etc.) for increased security.
* Notify admin with time detected if:
  + Server is down for any reason.
  + Server is running slower than anticipated.
  + Server is unreachable.
  + Mass amounts of data is downloaded, created, or deleted.
  + Corrupted data detected.
  + DMV makes any change to policies and requirements.

#### 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?*

* Changes to users can be made with the CLI or a GUI client/web UI.
* Using Linux, the IT admins can control when and how the platform updates.
  + Can opt to not update.
  + Can clone server to update and test changes and find conflicts/bugs.
  + Inform ahead of time for server maintenance, deploy updates and changes.
    - Retain server and db backups to roll back to if errors occur.
* IT admins will need proper credentials to access:
  + Cloud host for web app.
  + Cloud host for database (if different).
  + Web app source code and files.
  + Database and its information.

#### 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?*

* Login requirements:
  + Users must provide full name and a unique email and password for sign up.
    - Names are not used for login, possibly used for password recovery.
    - Emails are not case sensitive for login.
      * [jOhnDoe@example.com](mailto:jOhnDoe@example.com) is converted to [johndoe@example.com](mailto:johndoe@example.com) for authentication.
      * If [JohnDoe@example.com](mailto:JohnDoe@example.com) was used for signing up and verified, used that original case for email communication.
        + Satisfies rare case where email is case sensitive.
    - Passwords are case sensitive for login.
      * Never stored in plain text. Option to reveal password entry.
      * Store in db after “salting” the password’s hash.
      * Admins can only change or reset per verified user’s instruction via random temp password or link/code in email.
  + Emails will only have one associated password.
    - Authentication compares login attempt email and password to db by searching for email and converting password to salted hash to compare to salted hash password stored.
* Secure connection:
  + Use SSL/TLS protocols to encrypt exchanges between client and server.
* Brute force handling/prevention:
  + Lock user account from login attempt after many failed logins attempt from the same IP address in a period.
  + Form of 2FA for new device login – text, email, third party 2FA host.
  + Use CAPTCHAs
* Forgot password:
  + Send email with reset link/code.
  + Send text with reset link/code.
  + Customer service cannot reset password – difficult to verify identity.

### 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 have the user verify their password for account creation.
* The system shall never store passwords in plain text.
* The system shall validate user credentials when logging in.
* The system shall update the user’s account with any changes upon login or refresh.
* The system shall keep track of user, instructor, and car groups.
* The system shall use SSL/TLS certificates for any web traffic.
* The system shall have different user permissions.
* The system shall verify the user lives within a certain jurisdiction (relevant DMV information and instructor availability).
* The system shall back up automatically.
* The system shall have implementations to handle exploits.

### 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.)?*

* Interface needs:
  + Online test progress.
  + Driver notes.
  + General student information.
  + Accommodations for the student.
  + Photo of the student and instructor.
  + Student contact information form.
  + Company’s/Instructor’s contact information.
* Interface users:
  + Student
    - Access progress, notes, learning resources, necessary instructor information, meet up and drop off location, update personal information, relevant vehicle information, schedule.
  + Instructor
    - Access student’s progress, driver notes, learning resources, necessary student information, meet up and drop off location, update personal information, relevant vehicle information, schedule, current students.
  + Secretary
    - Access students’ relevant information, update students’ personal information, student schedules.
  + Admins, IT Admins
    - Access system functionality, modify page content and site design, modify site availability, modify other user’s permissions and roles.
* Interaction:
  + Provide cross platform support to make changes and access information on any device with a verified login.

### 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?*

* Not specified:
  + Does the DMV have an API for accessing their data?
    - If not, how will it be accessed? Manually? Scraping with a script?
  + How to obtain SSL/TLS certificate.
  + How will admins access necessary tools for management?
* Assumptions for students/clients of service:
  + Users have an internet connection and it’s fast enough.
  + They have an email address.
  + Students will trust the service enough to provide sensitive information.
  + They trust that the instructor and company itself is legit and safe.
  + The browser they use is supported.
* Assumptions for company employees:
  + They will get training for proper use and functions.
  + They understand how to use certain a certain OS, database, and cloud host effectively and safely.
  + They understand limitations for the web app and dependencies.

### 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?*

* System design limits:
  + Not enough backups – 3-2-1 backup strategy isn’t used.
    - 3 copies of the data – original and 2 copies.
    - 2 data storage types – e.g. NAS via HDD and external storage via SSD.
    - 1 off-site copy – different location from other storage methods.
  + Cloud based web apps have limited/no control over hardware and firmware.
  + Must have internet connection to manage web host.
  + Assuming it’s a web application, will run in browsers:
    - Unable to work offline.
    - Limited to browser caching capacity.
    - No push notifications for reminders, updates, and messages.
* External limits:
  + Limited to unknown budget needed hosting and database integration.
  + Timeline is uncertain due to possible changes, new features, and approval from customer.
  + Limited to cloud hosting
  + Must be low maintenance, minimal intervention needed for proper operation.
  + Must rely on third parties for hosting, hardware, backups, credit card processing, and APIs and that they’re available.
    - More points of failure out of the company’s control.
    - More dependencies to learn and manage.

### 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 graph with a graph and numbers

Description automatically generated with medium confidence