Southern New Hampshire University

CS 255 Business Requirements Document

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?

- Fill void in market
- Train students for driving test
- Partner with local dmv
- Better driver training
- Reduce failing driving test

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?

- Access data from anywhere
- Accessible online and offline
- Only update data while online
- Accessible Both on computers and mobile devices
- Ability to Export data
- Data reports compatible with excel
- System security with principle of least privilege in mind
- Tracking and auditing system changes
- Make/modify reservations
- Match student with driver, time and a car
- Enable/disable package offerings
- Create/update/verify profiles with personal data for each customer
- Import live data from third party dmv
- Serverless architecture

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?

- Ability to reserve/update/cancel appointments
- Offer student 3 packages
- Track progress
- Allow feedback from drivers about customers
- Customize profile with picture and personal data

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?

- The system needs real-time connection to the DMV database, so an API would be developed
- Updates should be made when there is an update to requirements from DMV database
- Cloud based support for backup and security

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 integration to Amazon S3.
- Running on Windows for reliability and 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?

- Users are allowed 10 tries before being locked out
- Usernames and passwords are case sensitive
- Usernames and passwords are validated via API from cloud-based storage In real-time
- Admin should be notified of any locked-out accounts
- Regular probes and system checks should be done

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?

- Rule of least privilege should be used for IT admin access or otherwise
- The access can be added and revoked via API to our AMAZON S3 databse
- Updates are made by microservices if needing to implement future capabilities, otherwise just an update to the existing database by modifying offerings and tests/diagrams.

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?

- We can offer a one-time password reset link sent directly to the users registered email
- Two-Factor authentication required on all accounts
- Username and password are required to log-in, as well as a one-time text message for validation
- The security of the exchange is taken care of by Amazon S3 cloud provider.
- A "brute-force" attack should be prevented by being proactive with security, but in an instance where a user is attempting multiple passwords, that account will be temporarily banned via IP after 10 failed attempts within a 1 second interval. Admin can unlock accounts once verification of false-positive.

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 load within 2 seconds
- The system should provide real-time status of any tests/assessment completions and results.
- The system should show the photo of the driver
- The system should show the driver notes
- The system should show the name, email, address and other personal information of the user
- The system should be able to schedule/cancel and update any appointments
- The system should be able to assign coaches/drivers to users and provide pertinent details
- The system should be able to provide a breakdown of different products/packages as well as disable any full or discontinued products/packages
- The system should prompt a security code via text after valid username/password for user to login
- The system should be able to receive updates from DMV database in real-time.

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.)?

- Users are admin, user, driver, parent
- There is a login page followed by a 2-factor page
- The main screen shows current test progress and results
- Driver notes provided to user
- Driver and student photo
- All personal details such as email and address
- A schedule page link that takes us to a new page detailing the packages offerings with a tab of any completed or current appointments to be updates/cancelled.
- A link to update personal details
- A link to pay and safely checkout

- The program should be optimized for mobile use

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?

- The user has a mobile device
- The user has internet connectivity
- The user has a credit/debit card
- The user has texting service
- The user is over the age of 13
- The user has a parent/guardian
- The user speaks english

Limitations

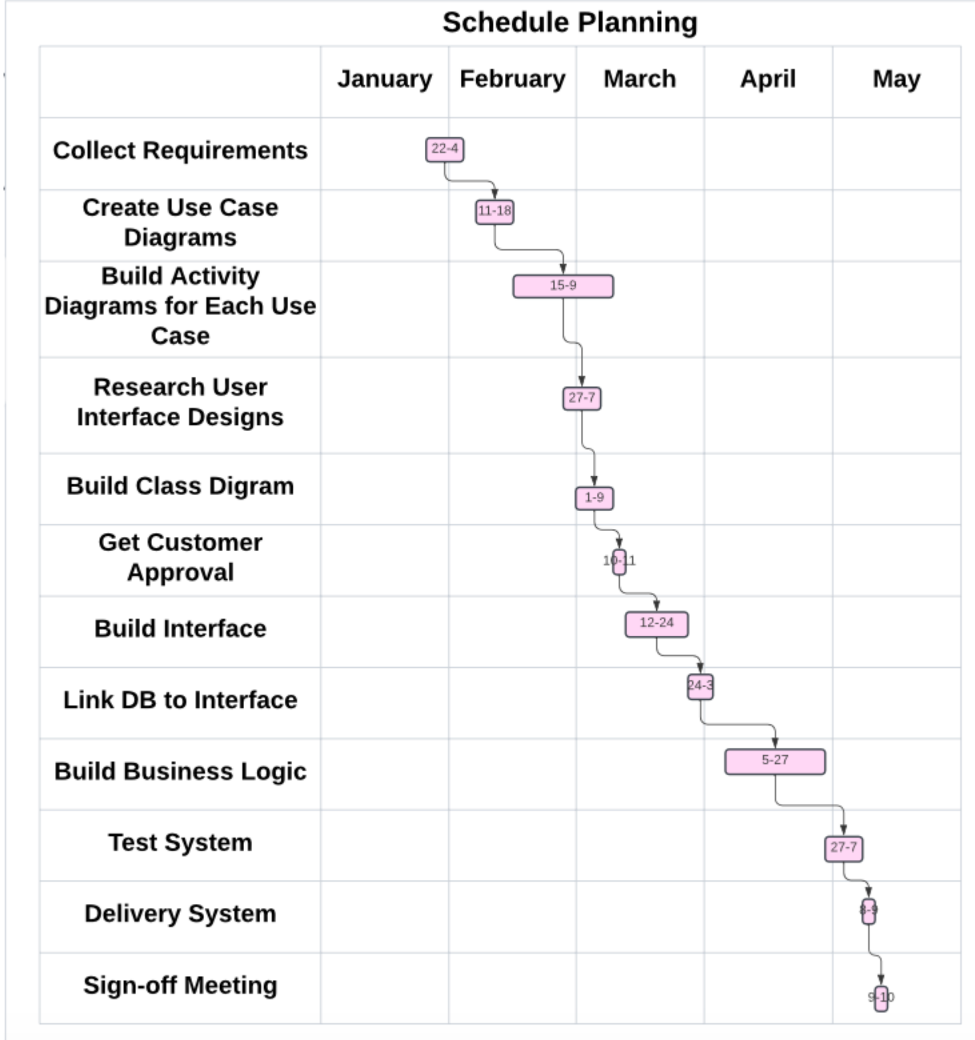
Any system you build will naturally have limitations. What limitations do you see in your system design? What limitations do you have as far as resources, time, budget, or technology?

- Cloud based support reduces overhead but requires implementation team
- Relying on mobile offerings might limit the immediate offerings given the screen constraints
- No offline support might prevent younger users from fully utilizing services
- No cash support might be problematic for a small portion of users

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.

Project Timeline for January to May



Schedule Planning

	January	February	March	April	May
Collect Requirements	22-4				
Create Use Case Diagrams		11-18			
Build Activity Diagrams for Each Use Case		15-9			
Research User Interface Designs			27-7		
Build Class Digram			1-9		
Get Customer Approval			10-11		
Build Interface			12-24		
Link DB to Interface			24-3		
Build Business Logic				5-27	
Test System					27-7
Delivery System					8-9
Sign-off Meeting					9-10