Southern New Hampshire University

CS255 Business Requirements

Dante A. Trisciuzzi

CS 255: System Analysis and Design

Prof. Kim-Marie Foss

January 31st, 2023

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, is looking to upgrade their workflow and provide a better educational experience for driving students. Driverpass is a small driver's ed company that has a brick-and-mortar business, but they are looking to expand their services. The client is looking to reduce the number of people who fail DMV driving exams by making education more accessible to new drivers. They need us to develop an inclusive online learning management system for these purposes.
-

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 wants a fully client manageable learning management system. As hinted above, they are looking to bridge the gap in their business model by offering different levels of driver training ranging from online lectures to one-on-one in-person driving assessments. The system should be able to accommodate customizable packages for students to select from, as well as course material presentation and exam capabilities.

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?

- A user should be able to register and manage their account.
- An administrator should be able to manage users and course packages.
- An instructor should be able to create and manage course offerings, as well as interact with their students.
- A student should be able to register for classes and pay for them online.
- A student should be able to access their course materials and exams, as well as schedule in-person appointments online.
- All users should be able to access troubleshooting information and contacts.
- The UI/UX of the system should be easy to understand and use.

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 should consist of a backend and a front end, the backend will run on a server of cloud service, and the front end will run in the customer's browser.
- The backend should be designed with scalability and performance in mind; a compiled language should be employed to inherently increase performance. C# would be my first choice since it is a great balance of run-time performance and development speed. Other options might include Go, Java or even Javascript running on something like Bun or NodeJS.
- Updates to the system should come in the form of feature additions, bug fixes, and vulnerability patches. Rolling releases would serve a product like this well since there are not expected to be any external projects dependent on the codebase.

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?

- Platform independence is always a factor in choosing development tools. All of my recommendations are cross-platform.
- The backend should use a NoSQL type of database. These include in-memory databases (Like LiteDB for C#), which are also cross-platform. A LiteDB or similar database system supports encryption and backup capabilities that make them highly reliable. Additionally, they have near-native speed when compared to something like even a performant SQL database (Like MariaDB).

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?

- During development we will employ the use of inheritance, such that different types of users (students, instructors, administrators, etc) may have different traits.
- Each user will have a unique id within the system, stored in the in-memory database solution. This allows us to distinguish users, and has the added benefit of being able to use hashing for a constant-time lookup algorithm.
- Every action that any user or system component makes should be logged; a severity level should be associated with these events (log, info, warning, error, critical). System administrators should have the ability to access logs based on a filtering system, and should be able to set notification levels independently for different system components.

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?

- The user management system should be based on inheritance. Users of different types will have similar registration processes, but only root IT administrators should be able to elevate a user to administrator. Likewise, these administrators should be able to manage instructor and student accounts.
- The sysadmin staff should have access to the server backend to troubleshoot issues with the backend environment.

- Since the IT staff may want to create other types of users in the future, they should be able to manage user permissions and templates. This would allow them to add users with arbitrarily configurable permission stacks.

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?

- User login should have integration for 2 factor authentication.
- Login information should be salted and hashed before heading to the server. All data transmissions should use a secure connection and employ proper encryption techniques.
- Brute forcing attacks can be mitigated by controlling the amount of user login attempts via a lock-out system. Additionally, since passwords will be hashed, no clear text passwords will ever be stored on the server.
- Users should be able to use their associated email or phone number (or other 2FA methods) to reset their password. Additionally they should be able to submit a request directly to IT staff via email or phone, where identity verification should take place before a password reset link is generated.

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 rely on a secure connection for user interaction.
- The system shall not allow users to access data outside the scope of their permissions.
- The system shall not allow spam and traffic.
- The system shall employ a DDoS mitigation service.
- The system shall allow students to access all their education information in a profile.
- The system shall generate reports on student progress for instructors to review.
- The system shall allow instructors to configure their course material and exams.
- The system shall not trust the client to validate information.

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.)?

- The user interface should be accessible from most devices that have a web browser.
- Layouts for common aspect ratios should be integrated and applied automatically.
- A mobile specific layout that is optimized for touch-screen access should also be included in the design.
- Since so many people use cell phones or tablets as their primary browsing device, full functionality should be implemented with touch support. This should include a warning for exams to ensure the device has plenty of battery life or is connected to a charger.

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?

- I am assuming that devices used to access the application have modern web browsing capabilities, such as script support and HTTPS support.
- Browsers without secure protocol support will not be supported.
- Browsers without scripting support will not be supported.
- I assume the user is not visually impaired to the point that they cannot interact with text on the screen. (They may not be eligible to drive either?)

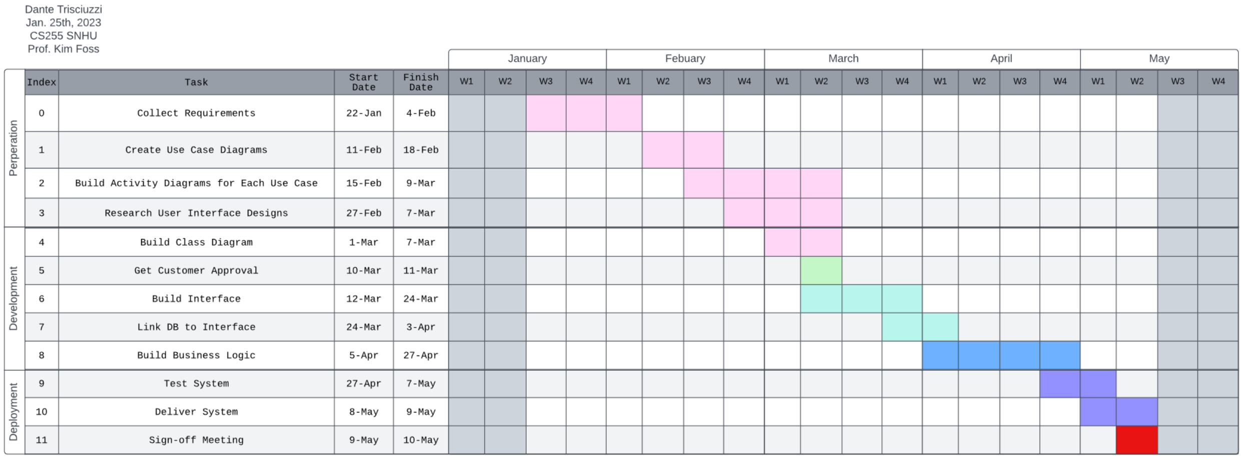
Limitations

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 will have only a certain set of course material presentation methods that are designed during development. Expansion of these systems would require further development and patching down the road.
- The client has a budget, so development planning needs to account for features that are necessary before adding ancillary features.
- The time frame is somewhat tight with only a few months for the project, this means the development team has less headroom to work with in terms of unforeseen challenges.
- Depending on the customer’s prioritization, some aspects of the product may rely on certain external resources or libraries.

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.



Dante Trisciuzzi
Jan. 25th, 2023
CS255 SNHU
Prof. Kim Foss

					January				Febuary				March				April				May			
	Index	Task	Start Date	Finish Date	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Perperation	0	Collect Requirements	22-Jan	4-Feb			■	■	■															
	1	Create Use Case Diagrams	11-Feb	18-Feb						■	■													
	2	Build Activity Diagrams for Each Use Case	15-Feb	9-Mar							■	■	■	■										
	3	Research User Interface Designs	27-Feb	7-Mar								■	■	■										
Development	4	Build Class Diagram	1-Mar	7-Mar									■	■										
	5	Get Customer Approval	10-Mar	11-Mar										■										
	6	Build Interface	12-Mar	24-Mar										■	■	■								
	7	Link DB to Interface	24-Mar	3-Apr												■	■							
	8	Build Business Logic	5-Apr	27-Apr													■	■	■	■				
Deployment	9	Test System	27-Apr	7-May																■	■			
	10	Deliver System	8-May	9-May																	■	■		
	11	Sign-off Meeting	9-May	10-May																		■		

Resources

SNHU. (2023). *DriverPass Interview Transcript*. SNHU.edu. Retrieved January 31, 2023, from https://learn.snhu.edu/content/enforced/1233071-CS-255-T3290-OL-TRAD-UG.23EW3/course_documents/CS%20255%20DriverPass%20Interview%20Transcript.pdf?_&d2lSessionVal=XQirjHL4vjK56MJrrvVMrmlMT&ou=1233071

Valacich, J., George, J., & Hoffer, J. (2020). *Modern Systems Analysis and Design* (9th ed.). Pearson.