

CS 255 DriverPass Interview Transcript

This transcript contains two different sections. The first is from the interview between representatives from your consulting company, Sam and Jennifer, and representatives from your client, DriverPass. The second part of the transcript is a conversation between Sam and Jennifer about the schedule for the DriverPass project.

Take notes as you read and focus on capturing details about what the client, DriverPass, wants from their system. The following questions may help guide your thinking:

- What does the client want the system to do?
- Who will be the different users of the system? What will each user need to be able to do?
- What are the different requirements for the system?
- How can you organize the schedule in a meaningful way?

Remember to refer back to your notes and this interview as you complete the work for Projects One and Two. Check to make sure that your notes and designs capture what the client needs.

DriverPass Interview

You are sitting in a meeting room and discussing a new project with four other people. Two are from your new client, DriverPass, and two are from your consulting company.

Liam, the owner of DriverPass, is explaining his vision for the new system he wants built. Liam is hoping to take advantage of a void in the market when it comes to training students for the driving test at their local department of motor vehicles (DMV). With Liam is his information technology (IT) officer, **Ian**.

On the other side of the table sits **Sam**, the former system analyst for your consulting company, and **Jennifer**, your project lead.

Sam: Welcome, Liam and Ian. Liam, can you start by telling us a little about DriverPass? What is your vision for this company?

Liam: I noticed that there is a need for better driver training. So many people fail their driving tests at the DMV. I'm starting this company to provide this type of training for my customers. I want them to be able to take online classes and practice tests. My company will also provide them with on-the-road training if they wish. I need you guys to help me build a system that will handle all of this.

Sam: Sounds great. What do you need from this system?

Liam: I want the system to help me access my data from anywhere, online as well offline.

Sam: OK. One thought. You can't modify or update the data unless you're online; otherwise, we might end up with duplicate data on different servers. This might cause data redundancy.

Liam: Sure, sure—I knew that. That's what I meant. I want to access data online from any computer or mobile device. I need to be able to download the reports and some information that I can work on at home, using Excel, for example.

Jennifer: Absolutely. What about security?

Ian: We have different employees at the company with different rights and roles. For example, for me I need to have full access over all accounts so I can reset them if someone forgets their password, or if we let go of someone and I need to be able to block their access.

Sam: What about tracking? Do you wish to know when a user makes a change to a record in the system?

Liam: Absolutely! For tracking, I need to make sure I know who made a reservation, who canceled it, who modified it last. All this must be clear in case something goes wrong. I want to be able to print an activity report and figure out who is responsible.

Sam: OK. Can you tell me a little more about these reservations? What are the reservations for?

Liam: Our customers need to be able to make reservations for driving lessons. Each lesson is two hours long, and the customer should be able to tell us the day and time when they want to take that lesson. They should be able to make this reservation online using their account. Or they could call or visit our office to schedule an appointment with our secretary.

We also need to be able to identify the driver the customer is scheduled to go out with, since we have many drivers and many cars. We have to be able to track which user is matched up with a certain driver, time, and car.

Jennifer: OK, great. We can include those aspects. We have talked about it a little bit already, but can you tell us about the different users of your system?

Liam: Sure. You have me, the big boss. You have my IT officer, Ian, who will be responsible for maintaining the system, modifying it, et cetera. You also have my secretary who answers the phone and makes appointments. Oh, and I want the user to be able to make appointments, cancel, and modify appointments online if they wish.

Sam: OK, great. How do the driving appointments work?

Liam: Sure. I have 10 cars. Each car has a driver. The customer can pick one of three packages:

- **Package One:** Six hours in a car with a trainer
- **Package Two:** Eight hours in a car with a trainer and an in-person lesson where we explain the DMV rules and policies
- **Package Three:** Twelve hours in a car with a trainer, an in-person lesson where we explain the DMV rules and policies—*plus* access to our online class with all the content and material. The online class also includes practice tests.

Each driving session is two hours long. So, for example, in Package One the six hours would be spread over three separate sessions. In the future, I would like to be able to customize these packages—remove some of them, add new ones—so the system has to be flexible.

Sam: Well, you will still need a developer or a system analyst to add or remove modules; it can't be built in a way that a nondeveloper can do easily. We can talk about adding or removing modules for a future release, but it does not need to be in the system we build now.

Liam: Sure, sure. But I at least want to be able to disable a package if I don't want any more customers to register for it.

Sam: OK. Anything else?

Liam: Yes. The way the registration should happen is that we get a phone call, and then the customer gives us their information. This information would include their first name, last name, address, phone number, state, and their credit card number, expiration date, and security code.

It should also include the pickup location from where the customer wants to be picked up. It should also ask them for a drop-off location, which should be the same as the pickup location.

Ian: As mentioned before, we also want the customer to be able to schedule appointments over the internet. If the customer forgets their password, we need them to be able to automatically reset it.

Jennifer: What about compliance? How will you keep up-to-date with changes the DMV might make?

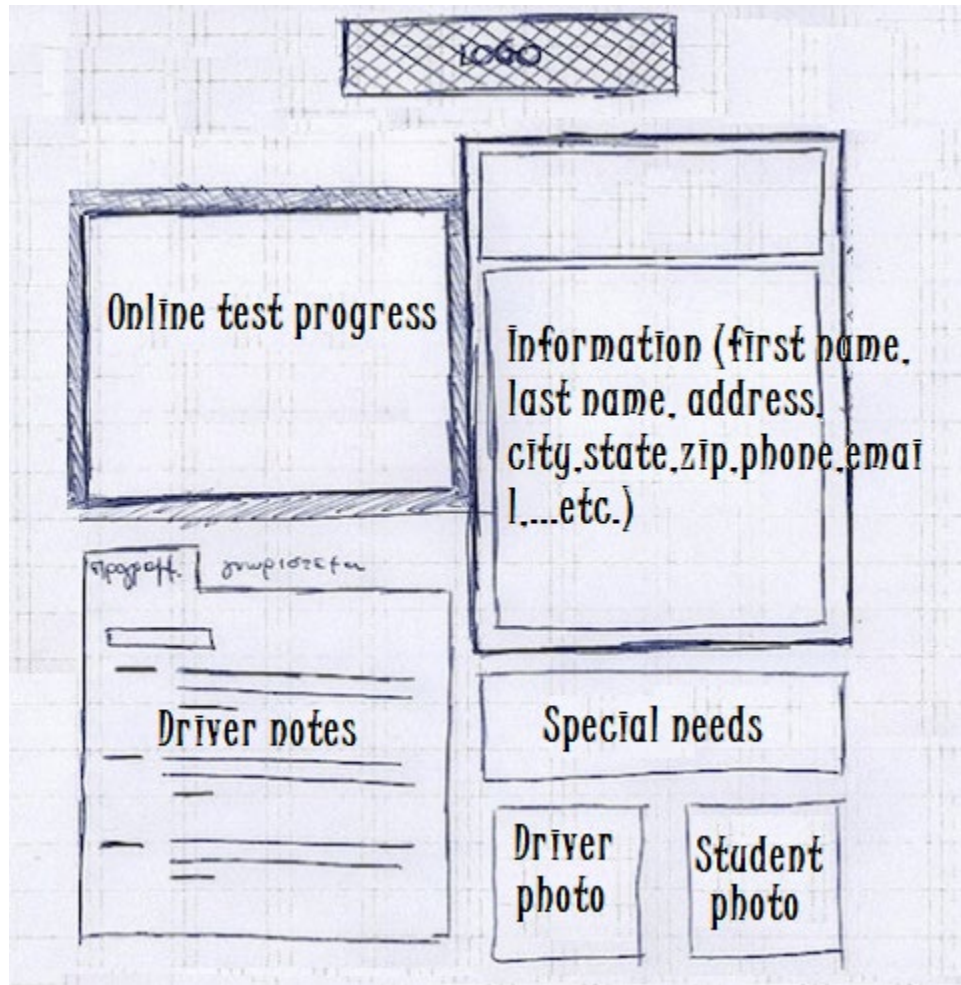
Liam: Oh, good question! We definitely want to make sure that the tests and practice we are providing are current with what the DMV requires. So we need to be able to be connected to the DMV so that they can update us with new rules, policies, or sample questions. We should get a notification whenever they have an update.

Sam: OK, got it. What about the interface?

Ian: The system needs to run off the web, preferably over the cloud. We do not want to deal with backup and security; we need that to be taken care of. We need our focus to be on running the business with minimal technical problems.

Sam: OK. What about look? Do you have any specific thoughts about the look of the interface, or do you wish us to come up with our own?

Liam: Oh, no—I do have a picture of what I want. Here, check this sketch out:



Liam: As you can see, the online test progress should show the tests the customer took. It should show what's in progress and the ones that the customer completed. So, it would say something like test name, time taken, score, and status. The status could be not taken, in progress, failed, or passed.

In the driver notes, I need to show any comments the driver left as well as the times for the lessons. So, it should show me something like this table:

Lesson Time	Start Hour	End Hour	Driver Comments

Sam: OK, so this is one of many pages we need to build, right?

Liam: Absolutely. There should be an input form where the student (or secretary) fills in the student information, such as first name, last name, address, et cetera. There also should be a page for contacting us, and a way to contact the student.

Sam: Great. Any idea of what you might want to do in the future? Like future features?

Liam: No, not yet. Let's focus on this; then maybe we can introduce new features to it.

Jennifer: Wonderful. I think we have enough here to get started. Sam and I will begin working on the system design and communicate this to our team. We will check in with you at different points to make sure that our design is meeting your needs. Thank you both for coming!

Schedule Planning

A short time after the initial meeting with the client, DriverPass, Sam and Jennifer sit down in a room. They have a conversation about how to approach the project and the different scheduling needs of the team.

Jennifer: OK, Sam. How are we going to approach this?

Sam: Well, I was thinking of laying out the tasks for this project. Then maybe we can arrange them according to dependency or importance. What do you think?

Jennifer: Perfect. So we started this project by sitting with the client January 22nd and today is February 4th. That took 14 days for requirement collection.

Sam: Correct. So next, I think we should start building the use case diagrams and activity diagrams. If we start February 11th (since this week is a holiday), that should take us about eight business days.

Jennifer: That's right. Meanwhile, we can at the same time have Toni and Clark research user interface designs. That should take them around nine days. Their task is not related to the use case diagram, so there is no dependency there.

Sam: Absolutely. Similarly, we can have John work on the class diagram. He needs nine days, and I know he's returning from his vacation March 1st. So he can start then.

Jennifer: Wonderful. We should be done by March 10th. Let's set that day and March 11th to meet with the customer to discuss the work with them.

Sam: Absolutely. Next, once the customer approves our work, and assuming he doesn't ask for modifications, we can start working on the interface March 12th. It will take us 12 days to complete. After that, we can build the database tables and link it to the interface. That should take nine days.

Jennifer: Correct. Once that's done, we need to add the business logic (security, role, right) layer, which will take 22 days.

Sam: Perfect. The system delivery will take a couple of days; then the sign-off meeting will take a couple of days as well.

Jennifer: Great. Here is the table I built based on what we discussed. I hope it helps.

Task	Start Date	End Date
Collect Requirements	22-Jan	4-Feb
Create Use Case Diagrams	11-Feb	18-Feb
Build Activity Diagrams for Each Use Case	15-Feb	9-Mar
Research User Interface Designs	27-Feb	7-Mar
Build Class Diagram	1-Mar	9-Mar
Get Customer Approval	10-Mar	11-Mar
Build Interface	12-Mar	24-Mar
Link DB to Interface	24-Mar	3-Apr
Build Business Logic	5-Apr	27-Apr
Test System	27-Apr	7-May
Deliver System	8-May	9-May
Sign-off Meeting	9-May	10-May

Sam: Thanks. That works just fine.

Jennifer: Great! You can use this basic idea to create a GANTT chart for our documentation.