# CS 255 System Design Document Template

## UML Diagrams

### Diagram Description automatically generatedUML Use Case Diagram

Graphical user interface

Description automatically generated**UML Activity Diagrams**

Graphical user interface

Description automatically generated

**UML Sequence Diagram**

### Diagram Description automatically generated with low confidence

### Diagram, schematic Description automatically generatedUML Class Diagram

## Technical Requirements

Looking at the UML Sequence Diagram, we can see that DriverPass needs a front-end interface that interacts with users, as well as a backend that contains a server, application, and database (“What is the difference between front-end and back-end development?”, 2019). People working on the front-end need to know whatever front-end language has been decided on, for instance, JavaScript, CSS, or HTML (“What is the difference between front-end and back-end development?”, 2019). The front-end team will need computers that can run whatever language they are using. If the front-end team decides to use JavaScript with the Visual Studio Code IDE, then they need at least a 1.6 GHz processor, 1 GB of RAM, and an up-to-date OS, Windows, or Linux operating system (“Requirements for Visual Studio Code”, n.d.).

For the back end, the team once again needs to know the language they are working in such as Python, Java, or Ruby on Rails (Kapoor, 2021). If Python is chosen it is a rather versatile language so it will run on most modern computers, this gives the team freedom to work on whatever computer/OS they deem fit (Aaberge, 2020). They will also need an IDE, one example for Python is PyCharm by JetBrains (*PyCharm*, n.d.). DriverPass wants the system to run on the cloud, this means it should be a cloud app. It also means it can run on any system such as Windows, Unix, Apple, and Android (Marcak, 2021). One good thing about cloud apps is that they work regardless of device or screen size (Marcak, 2021). If the system is to be run on the cloud then that means it will be serverless. An application framework provided by companies such as Amazon Web Services allows a web or cloud application to be built and run (Mahajan, 2019). Amazon Web Services is good for companies just getting into websites and applications because they provide all the infrastructure. This means the company does not need to buy infrastructure, and they also only pay for what they use. This prevents a company from building too large initially and wasting a lot of money.

Resources

Aaberge, M. A. (2020, September 11). *Everything you need to dive into Python programming*. Towards Data Science. https://towardsdatascience.com/everything-you-need-to-get-started-with-python-programming-4a37a46e427b#:~:text=To%20start%20programming%2C%20you%20need,to%20choosing%20an%20operating%20system.

JetBrains. (n.d.). *PyCharm*. https://www.jetbrains.com/pycharm/

Kapoor, A. (2021, June 28). *5 Most scalable backend development languages/ frameworks in 2021-22*. Medium. https://medium.com/backenders-club/5-most-scalable-backend-development-languages-frameworks-in-2021-22-f7f96101c19b

Mahajan, K. (2019, September 7). *Exploring how AWS serverless web apps work*. Srijan. https://www.srijan.net/resources/blog/exploring-how-serverless-web-apps-work#:~:text=Serverless%20refers%20to%20an%20application,are%20ephemeral%20and%20event%2Dtriggered.

Marcak, M. (2021, March 31). *Cross-platform apps vs native apps vs progressive web apps*. BusinessofApps. https://www.businessofapps.com/insights/cross-platform-apps-vs-native-apps-vs-progressive-web-apps/

Visual Studio Code. (n.d.). *Requirements for Visual Studio Code*. https://code.visualstudio.com/docs/supporting/requirements#:~:text=VS%20Code%20is%20lightweight%20and,1%20GB%20of%20RAM

*What is the difference between front-end and back-end development?* (2019, February 27). Concepta. https://www.conceptatech.com/blog/difference-front-end-back-end-development#:~:text=The%20term%20%E2%80%9Cfront%2Dend%E2%80%9D,deliver%20information%20to%20the%20user.