Criterion A: Planning

Defining the Problem	My client is the owner of a vehicle parking company. His business works as an intermediate between people who are willing to rent their parking space in exchange for cash and people who need to access parking spaces around the city. Therefore, my client is thinking of a program where he will be able to easily and effectively keep track of every parking space in his business.
Solution	In order to solve my client's problem, I had an initial conversation with him (Appendix 1), where he suggested that he needed a program where he could perform three main tasks. Firstly, he wants to be able to add each new parking space and store them in an organized manner. Secondly, my client needs a function where he will be able to easily rent out parking places. Finally, my client will need a function that allows him to easily view all the available parking spaces in his files. As a result, the solution will be based on a program that contains a main menu and allows my client to access any of the three functions he mentioned.
The Rationale For The Proposed Solution	The reasons why I choose to use Java for this program: 1. OOP. With this feature, I can create classes and methods which will allow me to separate the three main functions of the program (adding spots, renting out spots, and viewing spots) and as a result have an organized product. As well with OOP, I will be able to easily add new features to the program and leave room for extensions or further improvements. 2. Java has powerful development tools; IntelliJ Idea is free for students; plus includes code completion and a debugger to help solve abnormalities, allowing the user to have a better and more professional experience. 3. Java has great community support. Stackoverflow (Stack Overflow, 2020), for example, is great at solving queries when you are having issues in a certain part of code. If there is a request from my client that I cannot code, expert coders on Stackoverflow could be of great help. 4. Java has versatile features. ArrayList can be used to create a selection of accessible data that can be stored and managed, allowing my client to easily keep track of every parking place. Loops could be implemented for our menu to be repeated as long as the user requires. I can also import open source libraries such as scanners in order to get the input of a user. 5. Compared to other programming languages like C++, Java has a better packaging technique, so that the libraries are much easier to find, download, and debug; making it easier to use. In terms of speed, Java is generally faster and more

	efficient than Python. Java's efficiency largely comes from its Just-In-Time (JIT) compiler and support for concurrency. JIT It improves the performance of Java programs by compiling bytecode into native machine code "just in time" to run. Python is an interpreted language and it determines the type of data at run time which makes it slower comparatively. (SnapLogic 2020)
Stating the Success Criteria	Based on my client's needs during our first conversation (appendix 1), I've designed the following success criteria, to ensure that I built an accurate, coherent, and successful product. 1. The client will be able to access three different main options. One for adding spaces, another for renting out spaces, and another one for viewing the spaces in his files. 2. The program has a menu, which allows the users to choose the options and exit whenever he wants. 3. When adding a parking space, the program will ask for the city of the space and the available days for rental (weekday or weekends) to store each address in its specific place. 4. Every time a parking space is added, the Arraylist storing the addresses will increase and the new address will be stored. On the other hand, whenever a parking space is rented, the address will disappear from the specific Arraylist, in order for my client to keep track and know exactly which spaces are currently available and dismiss those which have been rented out. 5. When renting out a parking space, the program will ask for the city where the client needs to park and the day (weekday or weekend) to show the available spaces considering those specific conditions (city and day). 6. When renting out a parking space, the program will ask the number of days, in order to calculate the prices which should be charged to the customer. As suggested by the client, (Appendix 1) 7. The program will run in an infinite loop 'While(true)' and therefore will allow the users to perform multiple tasks without ending or closing the program. Whenever the user wants to exit, an option will be provided and the program will break/terminate. 8. The code meets the coding complexity requirements (appendix 2), to ensure a sufficient level of detail, functionality, and complexity, for the client to have a quality end product that meets his requirements and solves his problem. 9. The code uses imported libraries from java that can enrich the user experience. This might include Array List, File Output Stream, and Scanners. 10. The code uses java serialization, in order to store each Arrallist containing the addresses on organized files, keep

	them updated and use them every time the program starts. This will allow my client to close the program at the end of every workday, and the next day when he starts the program he will have all of the addresses updated and organized thanks to the files which are keeping track of the data. Serialization will also help my user keep its data safe in case the program crashes, the data will still be kept safe and updated within the files created.