Criterion A: Planning

The Problem:

My client is John Hammond, the owner of Jurassic Park. To keep the park secure, Mr. Hammond uses a manual locking system that requires extra labor and is inefficient. He wants to use a computerized system that would neglect the use of extraneous workers and make the process more efficient. Mr. Hammond planned to install automatic doors that would lock with a computer. Therefore, Mr. Hammond needed software that would mimic his park and control the doors of the park. He specified the details of the park in an interview conducted while designing the product. (See Appendix A)

During the initial consultation with Mr. Hammond, he asked me if I could make software that can meet those requirements and I said I would be willing to help. I proposed an initial prototype on paper and asked him if this is what this was looking for. (See Appendix B) Mr. Hammond agreed, however, suggested that the program should also have a login and register system, where employees can be registered. He also added that the only the employees should be able to access the system. So, I proposed another design for the login system as well in the initial consultation. (See Appendix B) The prototypes I suggested had to be graphical user interfaces to make the software easy for the employees to use.

Rationale for Proposed Solution:

After the initial consultation with Mr. Hammond, I realized that the best approach would be to implement this software using Java. Java is an object-oriented programming language that has the suited qualities for Mr. Hammond’s security system. Java has several features like polymorphism, inheritance of classes that are related to each other. Java also implements encapsulation which keeps the data secure from the user, so the program cannot be changed. In case Mr. Hammond does not come to me for maintenance, Java is widely known, which would make it easier for Mr. Hammond to find other people for maintenance. Java is also supported by a large range of platforms if there is a valid installation of the Java Virtual Machine. I also plan on using the Java Swing GUI because it is already built into java, which would allow Mr. Hammond to be satisfied with just one product. He would not have to use other software to implement a graphical user interface. I have also decided to use some sort of database (google sheets) to store the users and their passwords, so the users are persistent/non-volatile. The software would be an application that would be ready to use for the employees when installed on their computers by me.

Criteria for Success

Developed a Rubric to Establish a Range of Success

0-2 (Poor)	**2-4 (Decent)**	**4-6 (Excellent)**
Application stores the username and password locally and does not initialize a database	Application stores the username and password in a live database that updates along	Application stores the username and password in a live database that updates along as we register more users. The user can also reset their password
Application Checks the local arraylist for the user	Application check the live database and ensures all of their credentials match	Application checks the live database and ensures all of their credentials match. It also updates the database when the password is changed
Application does not allow the user to reset their password	Application allows the user to create a new user instead of changing the password in the old one	Application allows the user to change their password while keeping the badge and username the same. Updates in the database
Application allows new registration	New username and password get registered inside the live database	New username and password get registered inside the live database and they are assigned a badge number that is unique to the employee
Application has a command line interface	Application has a menu based interface that is easy to use	Application contains a graphical user interface that

		makes the user experience very easy
Application does not lock doors, without an accidental lock/unlock method	Application locks the doors however it cannot unlock the doors, without fail lock unlock method	Application is able to successfullty lock and unlock doors; there is also a fail safe method to lock and unlock the door so that cannot be accidentally unlocked
Application does not have an error message system; the user does not know what it going on if the program does not work	Application has an error system that tells the user what is going on in most cases	Application has an error system that tells the user what is going on in all cases, the user always knows what is going on

Criterion A Word Count: 450