You will be provided with a Microsoft excel .csv file. This file contains a large volume of data. You must develop an executable that respects the permissions and rules that the file has been created with. There are several possible rules that each cell or section can have, and they will be detailed both below and within the sheet itself.

This program should be able to read and write to the .csv file, as well as support multiple levels of user access (Guest, User, Superuser, and Administrator). Security information, terms, and definitions are detailed below in the Task Instructions.

**Context**

One of the fundamental, core concepts of Security through Design is the separation of functionality into different roles. This assessment follows that separation. Many programs opt to make different accounts into a ‘flow - down’ structure, where;

* •  Users have all the permissions of guests,
* •  ‘Superusers’ have all the permissions of Users,
* •  and administrators have full access.

**Task Instructions**

Submission for this assignment should be in the form of an executable application that can be run on the university machines. The language used to develop it is up to you (discuss with the lecturer prior) but it must have an existing API to read and output to Microsoft Excel .csv files.

**The User types are as follows:**

**Guest**: This user is not ‘logged in’ so to speak, and should only have a minimal level of access. At every available protected instance, the guest should have requested credentials (such as a username or password dialog that would allow a User to sign in)

**User**: This user is ‘logged in’, and can be thought of as a ‘client login’. This user should have access to their own data, but not access to any other clients’ data.

**Superuser**: This user is ‘logged in’, and can be thought of as a staff member. This user can create new users, add and view data in their accounts, and can also view company-specific information. They can also view information on their own account, but not other Superusers. They cannot create Superuser accounts.

**Administrator**:

This account level cannot view user information, or create user accounts. However, the Administrator can view Superuser information, as well as create Superuser accounts. The Administrator (like the Superuser) can also view the Company information.

**The security types are:**

**Public**: This information is available to all users. This includes file names, sheet names, and other miscellaneous data. This also includes access to the tool itself. It should also be public to create a user account (which then elevates the guest to a user.)

**Client / User**: This information is visible to both the specific user and Superusers, but not guest or administrators. This may include client details - such as names, addresses, and other private information. Client information should be visible to the specified User and Superusers only, and not to other Clients.

**Company**: This information is visible to users, Superusers, and Administrators. This information is only relevant to the company. This may include employment information, or company relevant policies, procedures, and plans.

Submission

Should include one-to-two page document outlining the various specifications OR a design document that details justification for the appropriate architecture.

**Your application will be marked on the following criteria: Commenting and Programming standards - (10%)**

- The quality of your documentation within the code, as well as your usage of your technical skills. A well-performing example in this area would be a well- commented, dynamic program with no visible lag or slowdown, no errors, and very few to no minor issues. The program is also well commented and laid out, with clear and concise information provided.

**Adherence to SBD principles - (30%)**

- This area focuses on how closely you follow, develop, and display the principles of Security by Design, both for the hypothetical user, and within the implementation itself. A well-performing example in this area would be a secure program that does not allow users to contradict the secure principles. The systems in place cannot be used to intentionally access protected data.

**Technical Implementation - (30%)**

- This area focuses on the technical depth of the submitted application. A good performer in this area has many relevant features implemented. These features are approaching the level of a professional-level application. The features also respect SBD principles, and cannot be easily exploited in order to retrieve protected data.

**Documentation - (30%)**

- The accompanying documentation for the application. A good submission for this documentation not only explains the purpose of the application, but also lays out and justifies decisions made during development. The documentation includes a number of cited, well-researched sources that add to the overall justification of design decisions.

**Referencing**

References should be included in the accompanying document that you have created. There is no minimum amount of references however, remember that the primary component of this assessment is **Justification**, which makes up a large amount of the overall documentation section.

It is essential that you use the appropriate APA style for citing and referencing research.