To implement the requirements for the password manager console application, you can follow this plan:

**Design the File Structure:**

Decide on a file format to store the encrypted passwords, such as JSON or CSV.

Define the structure of each password entry, including the required and optional fields.

**Implement Encryption/Decryption:**

Design and implement an encryption algorithm that uses the main password to encrypt and decrypt the file contents.

Ensure that the encryption method is original and provides a reasonable level of security.

Test the encryption and decryption functions to ensure they work correctly.

Handle File Selection and Initialization:

Provide the user with the option to choose a source file or enter the absolute path to the file.

Implement the logic to check if the file exists and initialize it if it's a new file.

Prompt the user to enter the main password for the source file encryption.

Add Password:

Implement the "Add password" command to allow the user to add a new password to the file.

Prompt the user to enter the required information, such as name, password, and category.

Provide an option to generate a secure password based on the user's preferences.

Check if the entered password has been used before and provide a security rating.

Edit Password:

Implement the "Edit password" command to allow the user to modify the data in an existing password entry.

Prompt the user to select a password entry to edit.

Provide options to modify the name, password, category, website, and login fields.

Update the corresponding entry in the encrypted file with the new data.

Delete Password(s):

Implement the "Delete password(s)" command to allow the user to delete one or more password entries.

Prompt the user to select the passwords to delete, either by name or using a search command.

Display a confirmation prompt before deleting the selected passwords.

Remove the selected entries from the encrypted file.

Add Category:

Implement the "Add category" command to allow the user to add a new category for password organization.

Prompt the user to enter the name of the new category.

Add the new category to the encrypted file.

Delete Category:

Implement the "Delete category" command to allow the user to delete a category and all passwords assigned to it.

Prompt the user to select a category to delete.

Display a confirmation prompt before deleting the category and associated passwords.

Remove the category and its associated passwords from the encrypted file.

Search Passwords:

Implement the "Search passwords" command to allow the user to search for passwords based on specific parameters.

Prompt the user to enter the search criteria, such as name, category, website, or login.

Display the passwords that match the search criteria.

Sort Passwords:

Implement the "Sort passwords" command to allow the user to sort the passwords based on different parameters.

Prompt the user to select the sorting parameters, such as name, category, or creation date.

Display the sorted list of passwords.

CLI Interface:

Design and implement a command-line interface (CLI) that displays prompts, menus, and user inputs.

Handle user commands and execute the corresponding functions based on the command.

Testing and Error Handling:

Test each functionality of the password manager thoroughly, considering different scenarios and edge cases.

Implement proper error handling to handle invalid inputs and unexpected situations.

Ensure the program behaves correctly and provides meaningful error messages to the user.

Remember to iterate and refine your implementation as necessary. Good luck with your project