Advanced Programming

Season 2024-III

Report Advanced Final Project #1

Juan Nicolas Diaz Salamanca 20232020059

Mathew Zahav Rodriguez Clavijo 20232020050

Based on the principles of drive software used for commercial, personal, and enterprise purposes, one drive from Microsoft and Google Drive from Google, we find that in the daily use of a drive software a client expects to:

User stories:

- The client expects access to his storage wherever required
- The client expects to upload, see, and organize every file he uses in one place
- The client expects to keep his files hidden from other clients

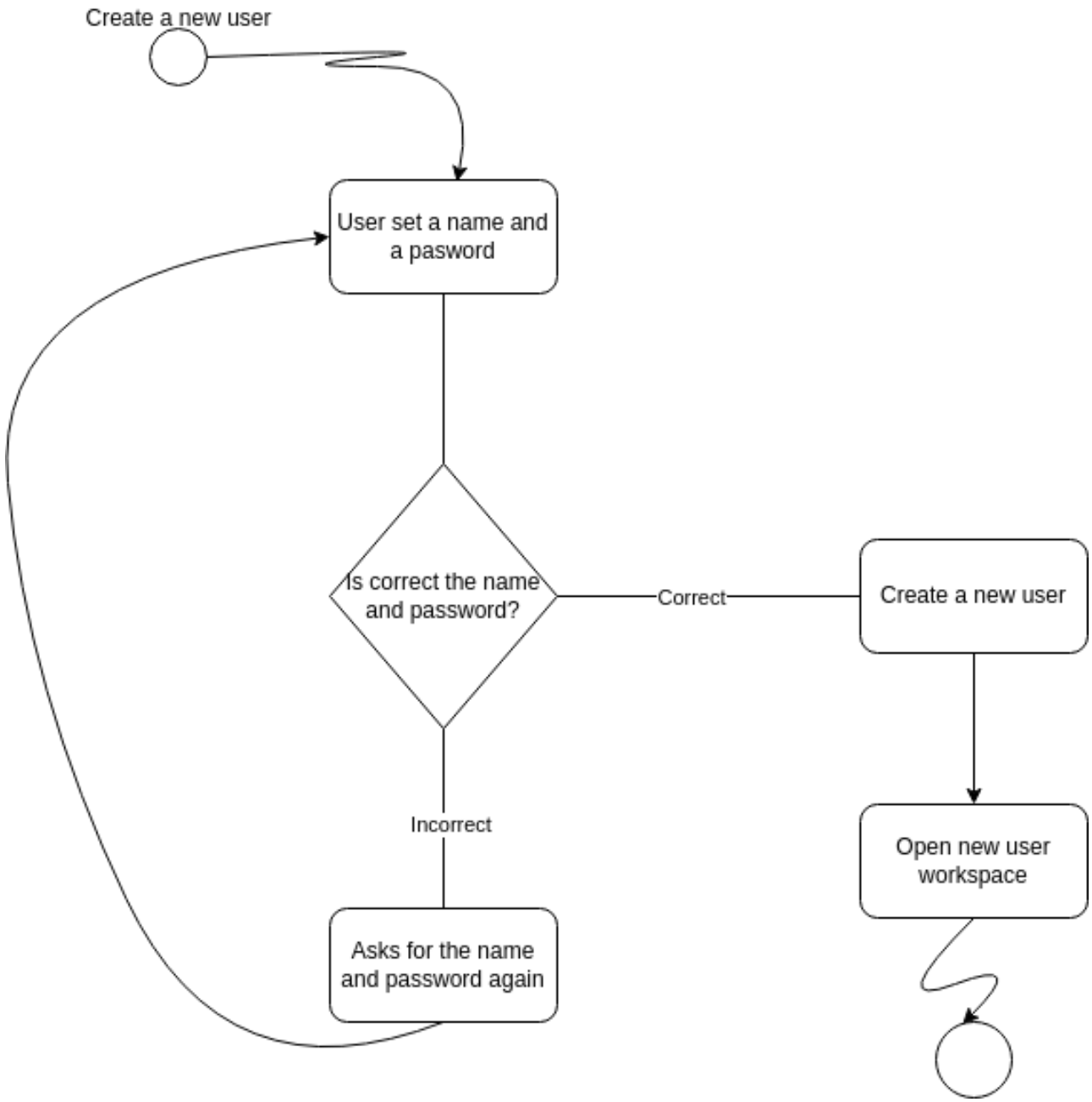
In line with user stories, the following functional requirements are proposed:

- The client needs to register with a name and password when using the application for the first time to access its functionalities.
- The client needs to access his files by name and password when is already registered and close his account when required
- Different clients need to access their accounts and swap accounts when required
- The client needs to visualize his files ordered in folders in his account
- The client needs to upload the desired file to his storage
- The client needs to download from his account the file of his choice to the space he requires
- The client needs to move files and folders in his storage as he required
- The client needs to create and name folders in his storage as he required
- The client needs to rename files and folders as he required

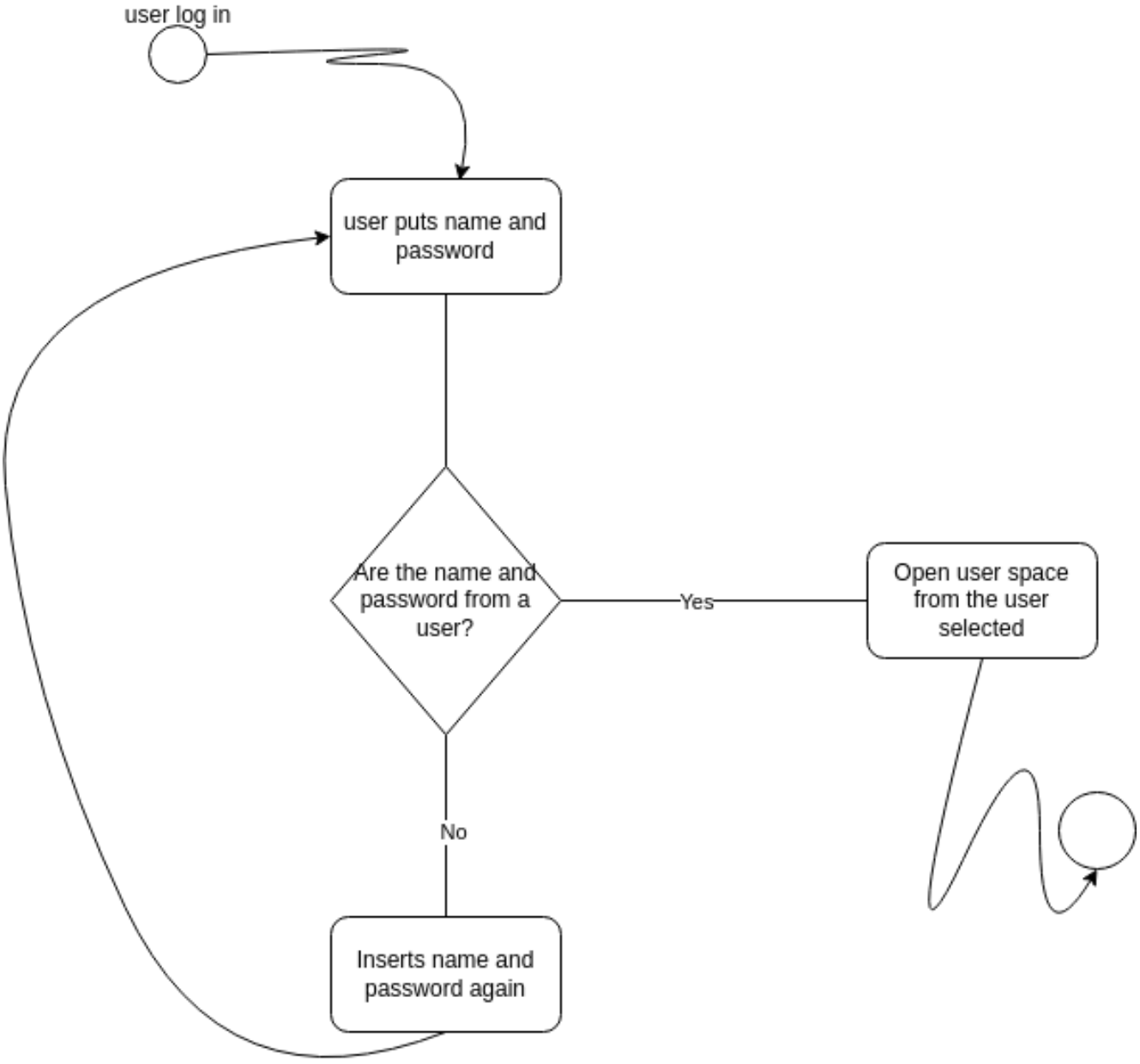
Activity Diagrams

These functional requirements are represented in the next activities diagrams:

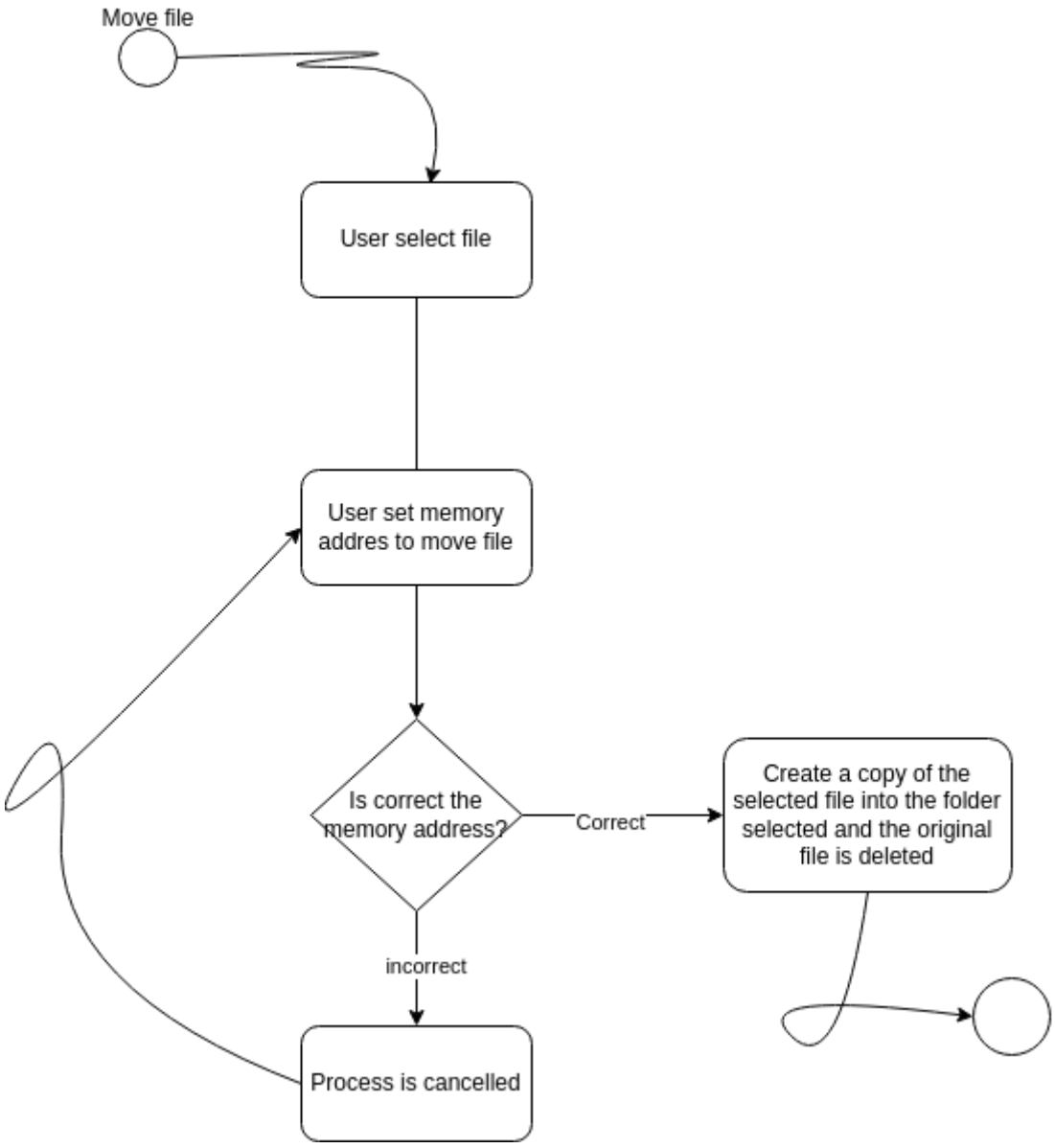
- The client needs to register with a name and password when using the application for the first time to access its functionalities.

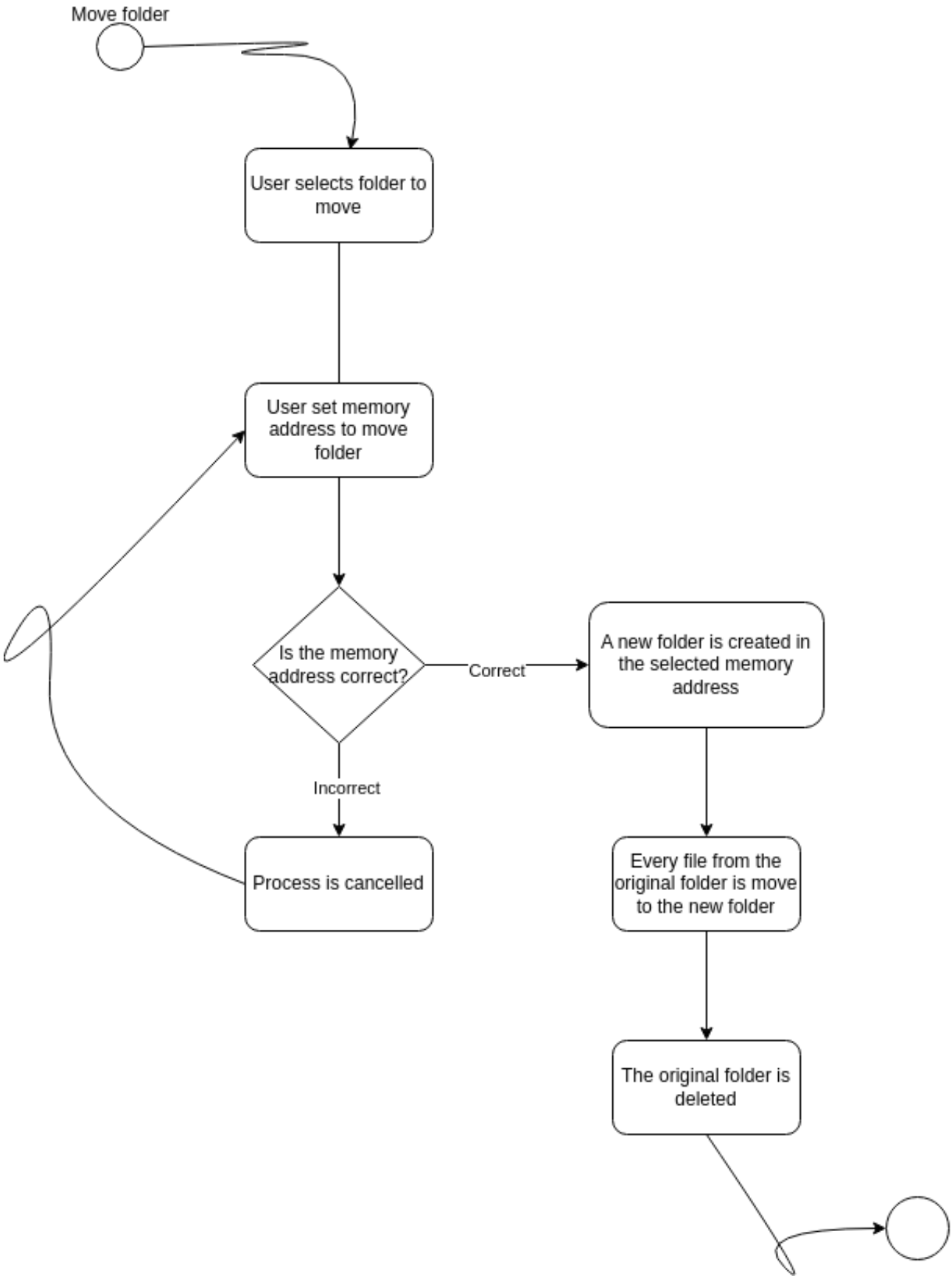


- The client needs to access his files by name and password when is already registered and close his account when required

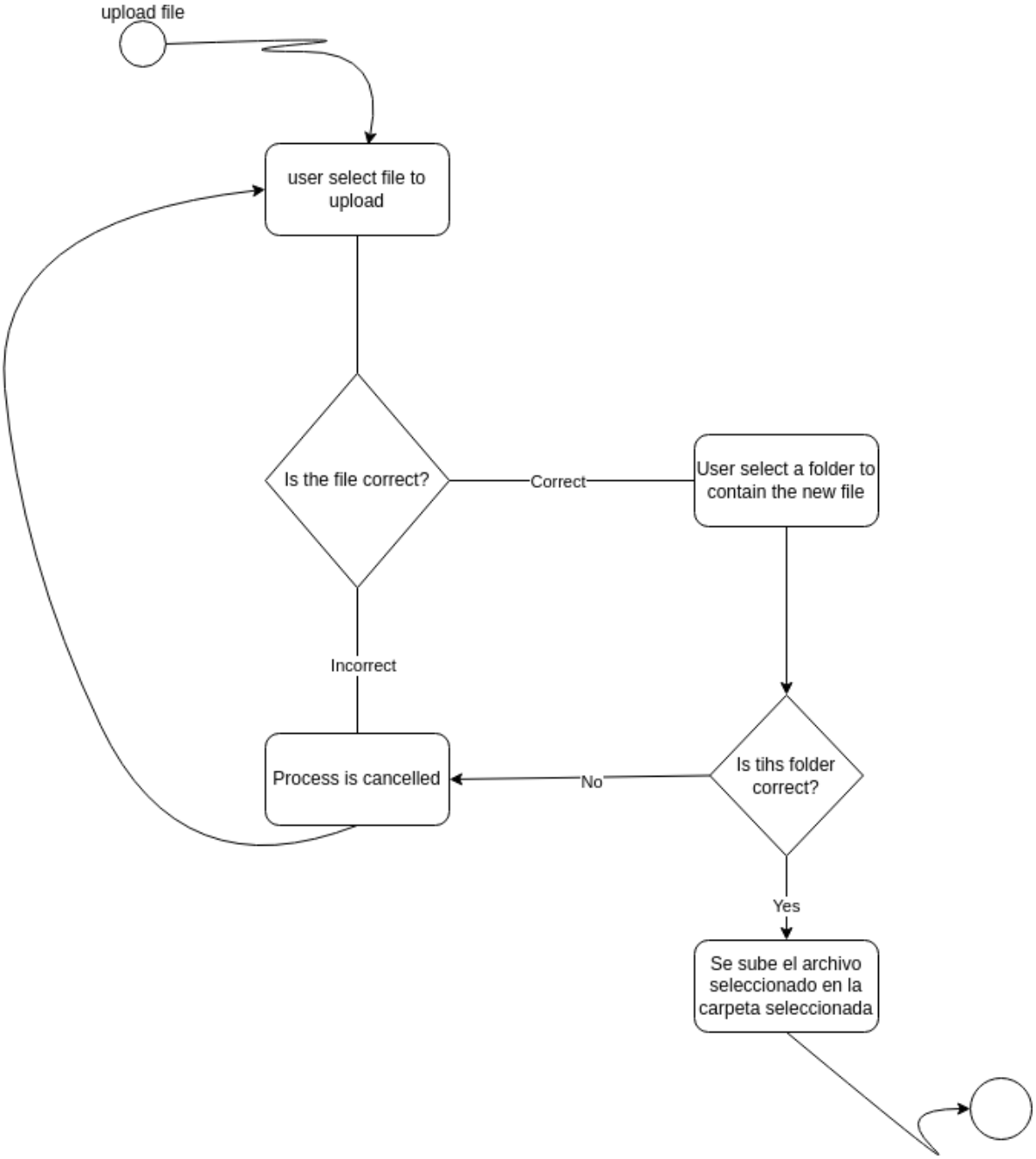


- The client needs to visualize his files ordered in folders in his account

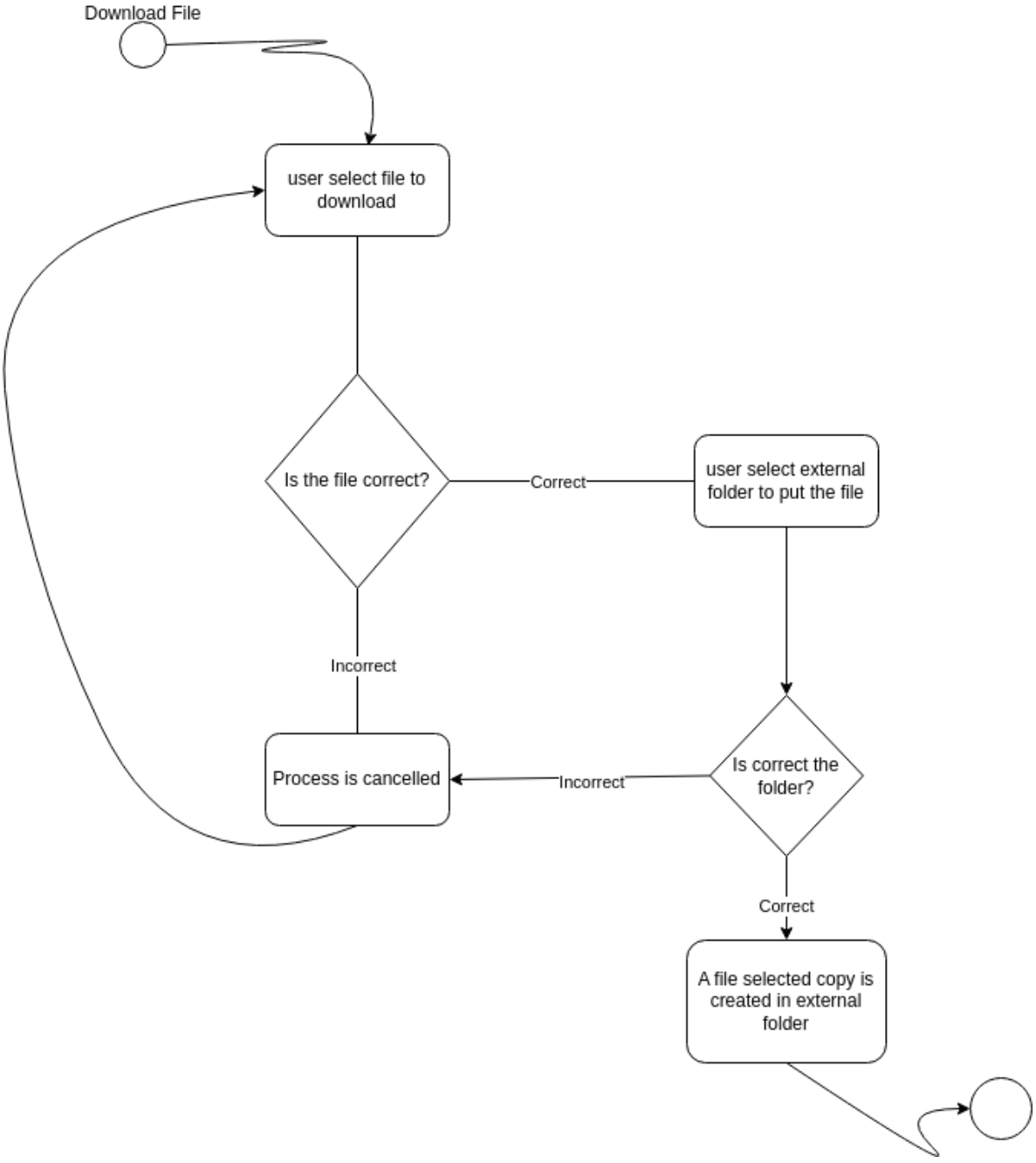



Move folder
User selects folder to move
User set memory address to move folder
Is the memory address correct?
Correct
A new folder is created in the selected memory address
Incorrect
Process is cancelled
Every file from the original folder is move to the new folder
The original folder is deleted

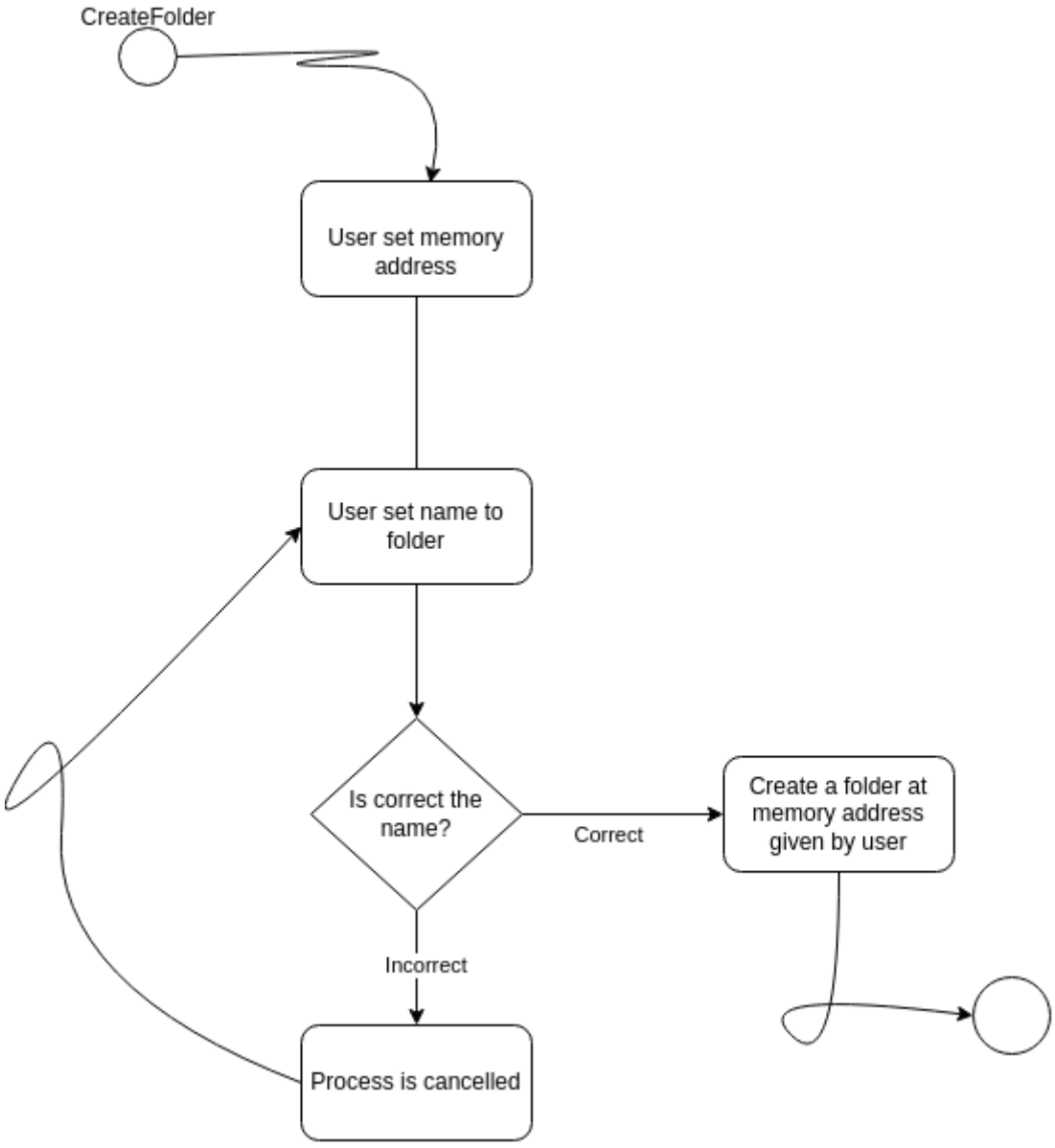
- The client needs to upload the desired file to his storage



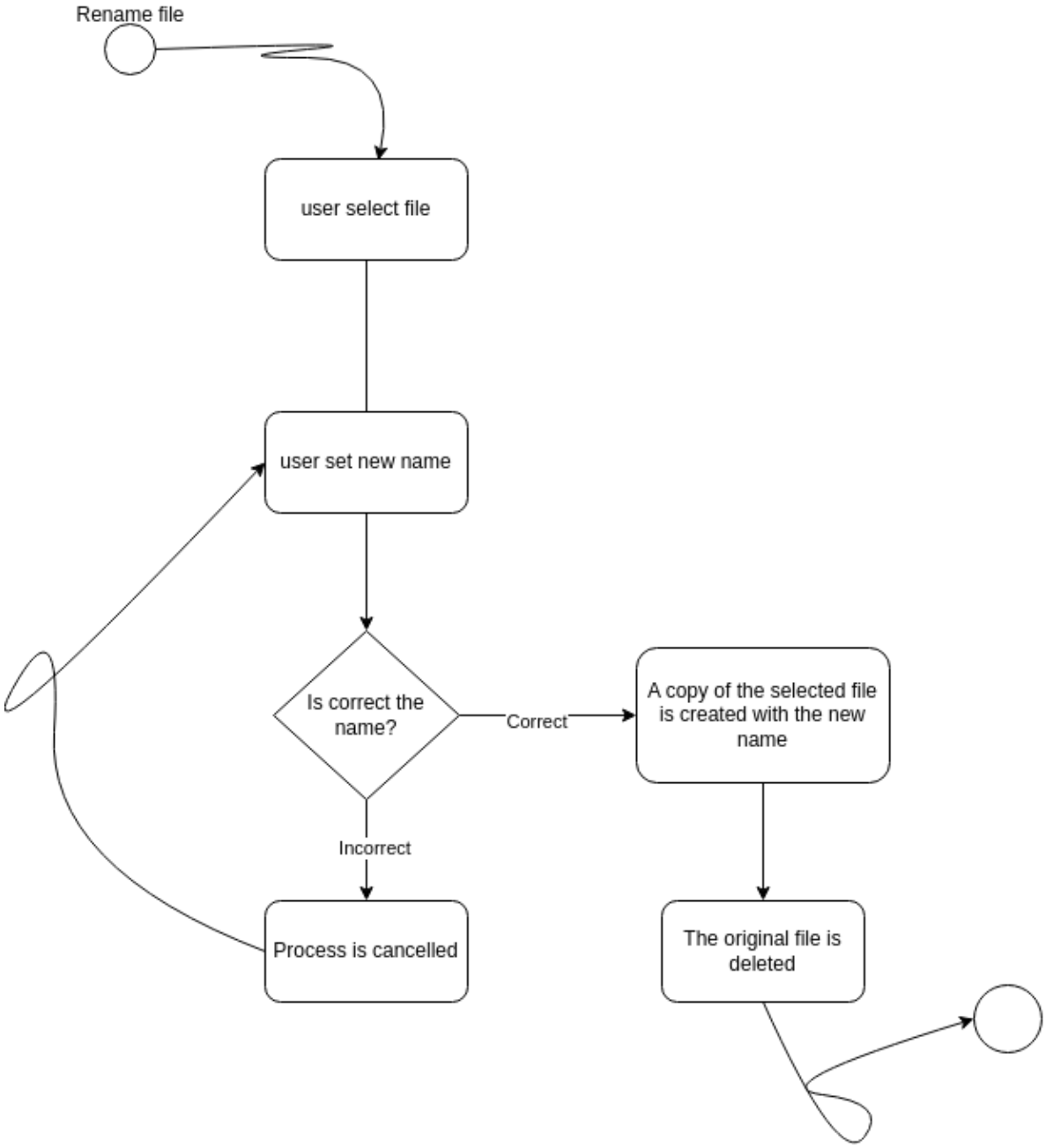
- The client needs to download from his account the file of his choice to the space he requires

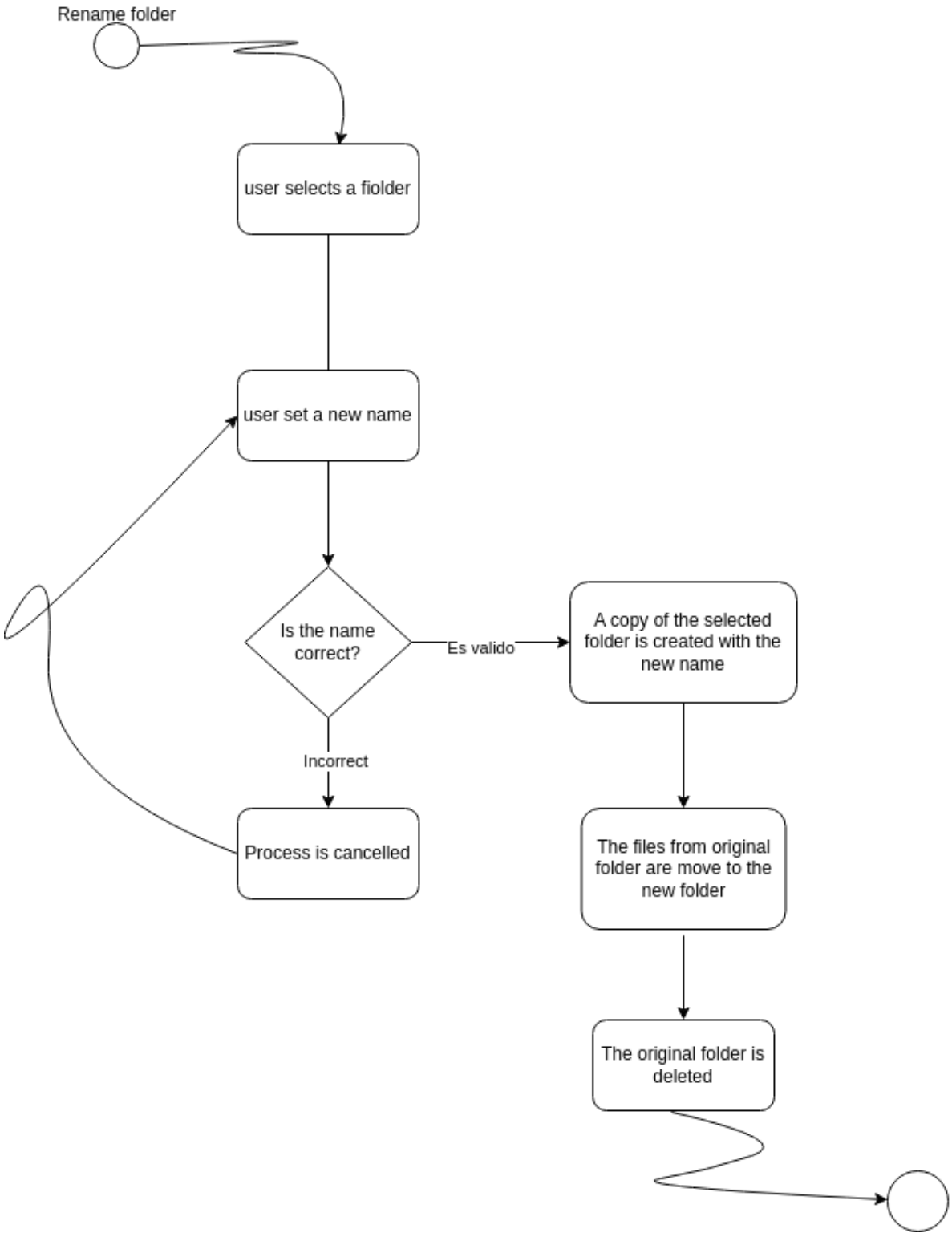


- The client needs to create and name folders in his storage as he required



- The client needs to rename files and folders as he required





And non-functional requirements:

- The application must be able to work with multiple user account storage
- The application must keep hiding the files and user information from other accounts

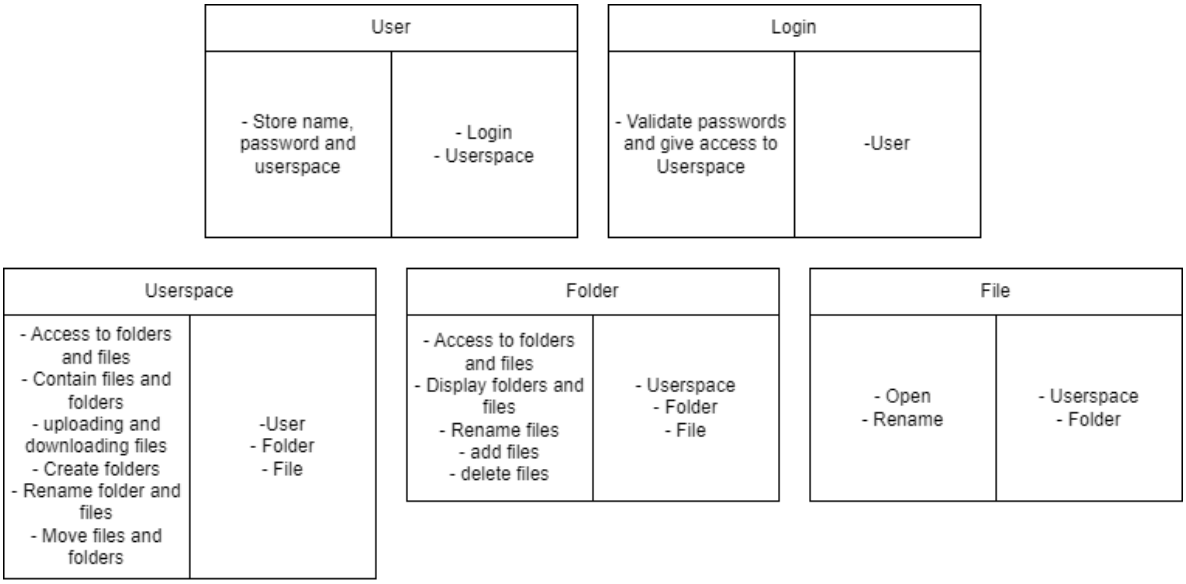
- The application must be able to do user requests in good performance

Based on these requirements and activities diagrams, the following entities are abstracted as necessary for the job flow of a drive:

- User: Entity that represents the client and stores his name, password, and connects the user with his account
- Login: Entity that represents the control to validate which user accesses a storage
- User space: Entity that represents the root folder of a specific user and gives access to the user to his files
- Folder: Entity that represents a space to store files or other folders
- File: Entity that represents a file that can be renamed and open

CRC Class diagram

These entities are represented in the following CRC Class Diagrams:



Object Oriented Analysis

The user class generates passive objects since its responsibilities of giving access to the user's storage are delegated to the user space class and the responsibility of verifying the user's name and password is delegated to the login class, from composition.

The user space class generates active objects in charge of giving access to the user's folders and files, it is composed of the folder and file class, the instances of these classes are added using a list and, in its methods, it accesses the methods of these classes to create, add, delete, and rename. Also, the folder class is composed of the files class since it contains a list of files.

In conclusion, the object-oriented analysis of the drive applications is based on composite over other object-oriented principles to create objects in order to keep responsibilities

separate as the CRC class diagram proposes. The next class diagram shows this relation between entities: