PluriNotes

In this project, it is about designing and developing the application PluriNotes, intended to edit and manage a set of notes (memos) that can correspond to text or images. For example, a note may be the report of a meeting, notes taken during a lesson, or tasks to be performed.

1. Description of main functions:

In this document, the specifications of the operation of the requested application are presented. These specifications voluntarily leave you with design choices, functional, conceptual and technological. Whatever the choices and adaptations you make, you will be careful to expose them and justify them in the report made with the project. It may lack specifications. In this case, make a choice by making it clear in your report.

1.1 Different types of notes

Each note is characterized by an identifier, a title, a creation date and a date of last modification. The identifier is a kind of label that makes it possible to refer to this note in another note. The title is a kind of summary of the note that allows you to quickly understand the content. Among the notes, one distinguishes the articles, the tasks, the images, the audio or video recordings:

- Articles are notes with text.
- Images, audio recordings, and video recordings are characterized by a description and an image file.
- The tasks refer to a certain action (described as text) to be performed. A task may have some priority (optional). You may also want a task to be completed before a certain (optional) expiry date. A job has status "pending", "in progress" or "completed". A task is initially pending.

1.2 Management of different versions of notes

Each time a note is edited, a new version of the note is created and the old version is saved. Part of the application interface allows you to navigate through the different versions of a given note and restore an earlier version as the current version of the note. Apart from its identifier which can never be edited, all the elements of a note are modifiable.

1.3 Relationships between notes

In order to enrich the semantics of a set of notes, one can create relations (in the mathematical sense) between notes.

Part of the application interface allows you to create, enrich, edit or delete a relationship. A relationship is characterized by a title, a description, and a set of pairs of notes. The application allows to add or remove a couple. Each couple can be characterized by a label. This label can be edited.

By default, a relationship is oriented. Thus a pair (x, y) of a relation signifies that there is a relation from the note x to the note y. However, the user may decide that a relationship is un-oriented. This will mean that a pair (x, y) of this relation holds for a relation from x to y, as well as from y to x.

A relationship is only defined between the latest versions of the notes.

The application must allow to visualize a relation (in a form that you will choose). The application must also make it possible to visualize for a selected note, the tree of the ancestors and the tree of the descendants of the note in all the existing relations simultaneously.

There is a special oriented relationship "preexisting" called Reference that cannot be deleted. This relation makes it possible to materialize the references that a note can make to other notes thanks to a special syntax. Thus, as soon as in any specific field of a note x that can be assimilated to a written text (the title, description, text, action ...), we find the text \ref {idy}, where idy is the identifier of a note y, it means that the pair (x, y) is added to the relation Reference.

1.4 Deleting a note

The user can delete a note and all of its versions if it wishes. The removal of a note leads to the elimination of all couples involving the superseded note in all existing relationships. This does not apply to the reference relationship. If a note is "referenced" by at least one other note, it cannot be deleted (as this implies the modification of the texts using this reference). Instead of being deleted, the note is archived. This means that it is still viewable (including its earlier versions) and can always be restored and become active again. However, as long as a note is archived, it is no longer editable. Couples of existing relationships to or from an archived note are not eliminated. When, after various actions by the user, the last reference (from an active or archived note) to an archived note x has been eliminated, the application proposes to the user to permanently eliminate the note x. If after various actions of the user, the only references which exist to a set A of archived notes have their origin since one of the notes of this set A, then the application proposes to eliminate the notes of the set A. When the deletion of a note is requested and this deletion is possible (that is not requiring archiving), it is placed in a basket awaiting a possible final restoration. It is therefore also considered as a kind of archived note but in suspension. The user can request the emptying of the basket causing the final elimination of this note. The emptying of the recycle bin is always proposed when the application is released unless it has been decided thanks to a configuration of the application that this dump was done automatically.

1.5 Interface Elements

- In its main view, the left part of the application has a part dedicated to the display of all the active notes, a part dedicated to the ergonomic display (taking into account the priorities and the due dates) of the tasks, and another (more discreet) part to all archived notes.

- In its main view, the central part of the application allows you to view a particular note.

- In its main view, the right part of the application whose view will be optional (retractable), will be dedicated to the display of the ascending tree and the tree of the descendants of the note in the set of existing relationships simultaneously. The goal is to visualize the trees from a note representing his ancestors and his descendants in a synthetic way. An ascending or descending note may appear multiple times in the trees to handle existing circuits in a relationship. These trees allow a simple click to move to another note.

- A secondary view of the application will be dedicated to the management and visualization of each of the relationships.

1.6 Saving the context

When the application starts, the state of the application, the settings present during the last run are retrieved.

1.7 Cancel and Restore Features

The application has the functions "cancel" and "restore". These functions can be called by a menu or by the Ctrl-Z and Ctrl-Y shortcuts.