Usability Document

CS 345-346

Problem 2 : Classnote Taking Application

Group 18

Name	Roll Number
Param Aryan Singh	180101055
Parth Dhananjay Bakare	180101056
Pooja Gajendra Bhagat	180101057
Tejas Prashant Khairnar	180101081

Contents

1 Eight Golden Rules of Shneiderman

1.1 Strive for consistency

The application's consistency, both **internal** and **external**, would be ensured by adhering to fixed conventions in every place.

- For example, **constructive actions** like adding elements and creating Notes would be displayed on the screen in **green buttons**
- Whereas **destructive actions** like deletions would be displayed in **red buttons**.
- Other **manipulative actions** like searching and editing would be displayed in **blue buttons**.
- Apart from button colours, icons for Notes and Notebooks would resemble **standard file** and **folder symbols**, quitting and minimizing window buttons would be present in the **same corner** of the screen for all windows and so on and so forth.

All these design choices are in line with the current designs we see in the real world, thus establishing both **internal and external consistency** in the application.

1.2 Design for universal usability

- Our contextual enquiry revealed that users want different types of content tools while making notes, depending on their **expertise in making notes** and using technology.
- Our application caters to this variety of needs, by offering all sorts of tools for making notes along with **step-by-step instructions** on how to use them, making it viable for all types of users - **novices, intermediates and experts**.
- **Hovering over the button** would give a detailed description of the button's function and usage.
- **Keyboard shortcuts** for simple actions like cut, copy, paste, undo etc would be present, thus allowing highly proficient users to perform these very quickly.

1.3 Offer informative feedback

Almost every process in the application would offer informative feedback.

- Right from the beginning, the logging in event would display a **message confirming** a successful or an unsuccessful login attempt.
- Time consuming and computation-intensive processes such as searching keywords would involve **progress bars**, indicating to the user the percentage of Notes that have already been searched while looking for the keyword.

Just like these processes, other processes would provide informative feedback to the user as well, thus communicating with the user at regular intervals.

1.4 Design dialogues to yield closure

The entire procedure of making a Note is divided into subtasks in the application.

- A **register/login** prompt would appear as soon as the application is launched, followed by a display of existing Notebooks and Notes. This would allow the user to browse previous Notes if he/she wants to revise or edit them.
- After choosing to **open** an existing Note or creating a new Note, the user would make notes and save and close the Note once his/her work is over.
- While **saving**, the user would be asked the location where he/she wants to store the note.

This entire process would be handled by different modules, guiding the user at each step, making the experience very seamless.

1.5 Offer error prevention and simple error handling

Note making experience in the platform would be largely void of errors, owing to the neat interface.

- The most error-prone events would be **logging in** and **failed searches**. Both these cases would be handled by displaying simple messages on the screen, such as **"Wrong Password"**, **"No matches found"** etc which would be easily understood by all users.
- Moreover, options to correct these errors would be present, for instance **"Forgot Password"**, which would be implemented using the Update Details function in the Manage Users module. Also, proper instructions for error handling will be provided to the user through pop-up messages.

1.6 Permit easy reversal of actions

- Actions like **redo and undo** would be implemented in the application interface, allowing easy reversal of actions.
- Alongside these features, functionalities to **delete objects** - Notebooks, Notes and elements - would assist in easy reversal of actions as well.
- **Easy navigation** through the notebooks will also be ensured using **back and forward** buttons for navigation.

1.7 Keep users in control

Making notes using the application would be largely graphical and visual.

- Features like adding images, shapes and tables would be **very interactive** and would make the user feel that he/she is in control.
- Organising Notes inside Notebooks would be done using **drag and drop**, and the entire structuring of Notebooks would be as per the user's convenience, thus adding onto the previous point.

1.8 Reduce short-term memory load

While making notes, at no point would the user be required to remember a large number of chunks of information **(7+2) Rule**.

- The user would only need to **recall the actual content** he/she wants to write down in the notes, which cannot be done away with in any case.
- All the buttons in the interface would be accompanied by **appropriate texts and/or suitable icons**, so that the user would not need to memorise which button would do what action.
- In the case that these auxiliary texts/icons would not suffice to tell what the button does, **hovering over the button** would give a detailed description of the button's function and usage.

Note: All the points mentioned above will be taken care of when designing the prototype (interface) and more points will be added as well.