4156 Revised Project Proposal

Group WLTT
Junyue Wu (jw3674) Bowen Tan (bt2484)
Xuheng Li (xl2784) Shulan Tang (st3174)

1 Abstract

Overall, we will build our application by constructing the MVC model using SQLite for data storage and NW.js to create views and build control logic. To achieve the functionalities of our note, We will employ some public libraries as well as APIs. Specifically, html-pdf to generate pdf preview while taking note and Marked.js that serves as markdown translator. We will use Maps JavaScript API that can track and mark the note with the user location. By providing this, the user can easily recall the content that is associated with this location. After finishing the implementation, we will use nw.js to wrap up our application and turn it into the cross-platform software support Windows, Linux as well as MacOS.

2 Project Description

The project that our team is going to purpose is a note-taking app with markdown support. Nowadays, as more and more people tend to use technology products to take notes to avoid the loss of notes, the need for a user-friendly note-taking app is high. Markdown is a simple yet efficient way to add formatting to make notes easier to read. Here, our group will provide users with a pc-based software that satisfy users' need. What makes a good note-taking app? First, we want to provide users with a simple way to edit and organize notes. Markdown is a good choice for an efficient yet not too overwhelming text formatting. Therefore, our application will support markdown syntax for users to generate the well-organized note. Further, during the editing, users should be able to preview to notes they taken with parsed markdown. Our application will also provide users an interface to export their notes as PDF files so that they can share and print notes. Second, the notes can be saved and retrieved easily every time the user opens the software. In our project, we plan to use a relational database to store notes created before. Notes can be stored and retrieved locally without interacting with a remote server by using SQLite. To support note retrievemcent in other devices, notes are also stored in the backend MySQL database and can be located with the associated user.
We will prioritize main functionalities for our application and develop it in an iterative

approach. For the first iteration, we will focus on simple markdown support and local note storage and retrievemcent.

3 User Stories

3.1

As a note taker, I want to add common markdown formatting to my notes so that I can organize it to be more readable. My conditions of satisfaction are:

- I can create numbered and bulleted lists
- I can separate Heading and Body

3.2

As a note taker, I want to see my pdf preview and script side by side so that I can preview the note directly. My condition of satisfaction is:

- I can see two split views in one window. One shows the plain text I typed, the other one shows the parsed and rendered note according to my markdown input

3.3

As a note taker, I want to preview the note in real-time so that I can better understand whether the note is correctly taken. My conditions of satisfaction are:

- The text shows in the original format should be synchronized with the parsed version
- When I scroll the preview window, the original note will scroll as well
- When I add/delete text, the preview window will be updated within a few seconds

3.4

As a note taker, I want to view the note and the markdown script simultaneously so that when I write a long note and make a change in the middle, I can see the effect in real-time. My condition of satisfaction is:

- When I scroll the original note, the parsed note will change correspondingly

3.5

As a note taker, I want to save my notes in the application so that I can read it afterward. My condition of satisfaction is

- All the contents and formatting of my note should be preserved.

3.6

As a note taker, I want to save my notes with a title and see its date so that I can easily recall the content of my notes. My condition of satisfaction are

- I can change the title of my note
- The metadata of my notes, such as the title, time of creation/editing should be preserved and I can browse and review my notes by the title.

4 Acceptance Test Plan

4.1 Data Storage

4.1.1 Store notes

- Input: User types in an input text. Then click the Save button.
- Expected outcome:
  - A new record of note of the unsaved note should be created in the database correctly.
  - The note?s creation date and last modified time are correctly stored in the database as well.
  - The state of note in the GUI should be changed from unsaved to saved so that the user knows this note has been saved.
  - If the note is not saved successfully, the user should see a notification of the failure.

4.1.2 Update notes

- Input: User modifies the existing note. Then store the note.
- Expected outcome:
  - The stored note file is modified. When reopened, the user?s modification is preserved.

* The metadata for size and last modified time changes accordingly.
* The state of note in the GUI should be changed from unsaved to saved so that the user knows this note has been saved.
* If the note is not saved successfully, the user should see a notification of the failure.

4.1.3 Delete notes

- Input: User clicks the delete button for the existing note.
- Expected outcome:
  * The note record should disappear immediately in GUI.
  * When the user reopens the app, the deleted note should not show up again.
  * If the note is not deleted successfully, the user should see a notification of the failure.

4.1.4 Retrieve notes

- Input: User opens the app.
- Expected outcome:
  * All previously stored notes should be correctly retrieved.
  * All the notes metadata including note title, create date and modified date should be correctly retrieved.

4.2 Testing objective: Note formatting

- Input: Valid markdown gramma for Bold: bold text, Italic _italic_, different level of titles:

$$\#, \#\#, \#\#\#, \#\#\#\#, \#\#\#\#\#, \#\#\#\#\#\#$$

, numbered and bulleted lists

$$, -, 1., 2., \ldots$$

- Expected outcome:
  * The corresponding formatted text.

- Input: Invalid gramma such as ###### title to specify a title, bold for bold
- Expected outcome:
  * Unparsed plain text of ###### title or bold

4.3 Testing objective: Simultaneous scrolling

- Input: The user inputs some notes more than one page and scrolls the text editing window up and down
- Expected outcome:
  - The first line of the editing view and the preview view should be in the same position.
  - The process of auto scrolling of the preview should be within 1 second.
- Input: The user scrolls the preview window up and down
- Expected outcome:
  - The first line of the preview view and the editing view should be in the same position.
  - The process of auto scrolling of the editing window should be within 1 second.