Test cases

Each test case includes a registration of a new account as the first step. A fillable Excel sheet for the test cases can be found in the GitHub repository.

1. **Note addition and search function**
    - User adds a note, and then searches for it

2. **Note deletion and search function**
    - User adds a note
    - User deletes the note
    - User searches for the deleted note

3. **Account deletion**
    - User deletes the account
    - User tries to log in with the deleted credentials

4. **Note editing and saving**
    - User adds a note, and then edits it
    - User logs out of the account and logs back in
    - User examines the saved changes to the note

5. **Note editing and search function**
    - User adds a note, and then searches for it using the original title
    - User edits the note's title
    - User searches for the note using the original title, and then searches for it using the new title

6. **Language change and saving**
    - User edits the account's language choice
    - User logs out of the account and logs back in
    - User examines the application's language and the selected language for the account

7. **Notification functionality**
    - User adds a note with a notification date and time
    - User logs out of the account and logs back in after the notification's time has passed
    - User examines that the application has notified the account

8. **Notification gets removed once note is edited**
    - User adds a note with a notification date and time
    - User waits until the notification's time has passed
    - User opens the note and confirms an edit
    - User examines that the indication of the notification has disappeared

9. **Notification time is saved and visible upon editing**
    - User adds a note with a notification date and time
    - User edits the notification's time
    - User examines that the correct notification time is visible on the note, and that upon opening it, the correct notification time is selected

10. **Image changes are saved**
    - User adds a profile picture
    - User adds a note with an image attached
    - User changes the profile picture as well as the image on the note
    - User logs out of the account and logs back in
    - User examines the updated and saved images

Table showing the outcomes of each test case per team member:

Test case No.	Maria Aalto	Suphitcha Pitkänen	Ville Schulz
1.	pass	pass	pass
2.	pass	pass	pass
3.	pass	pass	pass
4.	pass	pass	pass
5.	pass	pass	pass
6.	pass	pass	pass
7.	fail	fail	fail
8.	pass	pass	pass
9.	fail	fail	fail
10.	pass	pass	pass

Evaluation of the results

We all discovered the same bugs during the 7th and 9th test case. The 7th test failed because the indication of the notification was not visible when it should've been. The 9th failed because the notification time appeared to be the previous time even after changing it.

We determined that the errors found during testing were moderately severe, and fixed them:

Before (error in test case 7.)

```
dateSelector.setValue(currentNote.getDate().toLocalDate());
hourSpinner.getValueFactory().setValue(currentNote.getDate().getHour());
minuteSpinner.getValueFactory().setValue(currentNote.getDate().getMinute());
```

After

```
dateSelector.setValue(currentNote.getNotificationTime().toLocalDate());
hourSpinner.getValueFactory().setValue(currentNote.getNotificationTime().getHour());
minuteSpinner.getValueFactory().setValue(currentNote.getNotificationTime().getMinute());
```

before (error in test case 9.)

```
private void startNotificationChecker() {  1 usage   Ville +1
    scheduler.scheduleAtFixedRate(() -> {
        LocalDateTime now = LocalDateTime.now().truncatedTo(ChronoUnit.MINUTES);
        if (notificationDate != null && now.equals(notificationDate.truncatedTo(ChronoUnit.MINUTES))) {
            Platform.runLater(() -> notificationShown.set(true));
        }
    }, initialDelay: 0, period: 1, TimeUnit.MINUTES);
}
```

after

```
private void startNotificationChecker() {  1 usage   Ville +1 *
    scheduler.scheduleAtFixedRate(() -> {
        LocalDateTime now = LocalDateTime.now().truncatedTo(ChronoUnit.MINUTES);
        if (notificationDate != null && !notificationShown.get() && !now.isBefore(notificationDate.truncatedTo(ChronoUnit.MINUTES))) {
            Platform.runLater(() -> notificationShown.set(true));
        }
    }, initialDelay: 0, period: 1, TimeUnit.MINUTES);
}
```