

Notes Manager

Universal notes binder

Sebastian Smoliński
[version 1.0.0]

13.03.2019

Contents	
Introduction	2
Features	3
UI visualisation	6
Details of the implementation of the functionality	8
Add category	9
Delete category	10
Edit category	11
Add note	12
Delete note	13
Edit note	14
Prototype system design	15
User manual	17
Versions	18

Introduction

Project assume create application for managing notes that can be grouped in named and colored categories. Application should be very simple in action and easy in implementation. For more information about application's capabilities (see **Features** chapter).

For create application we're going to use Java 11 technology and Java's standard libraries (OpenJDK). For creating UI we're going to use Swing library. For creating Database and communication with it, we're going to use SQLite3 and JDBC.

Application will be on GPLv3 license, it'll be free and open software. For version control we're going to use GitLab (with private repository) and GitHub (final version, public repository).

Features

For better understanding application's needs a use-case diagram was drawn up. This diagram is completed — it contains all features from user perspective.

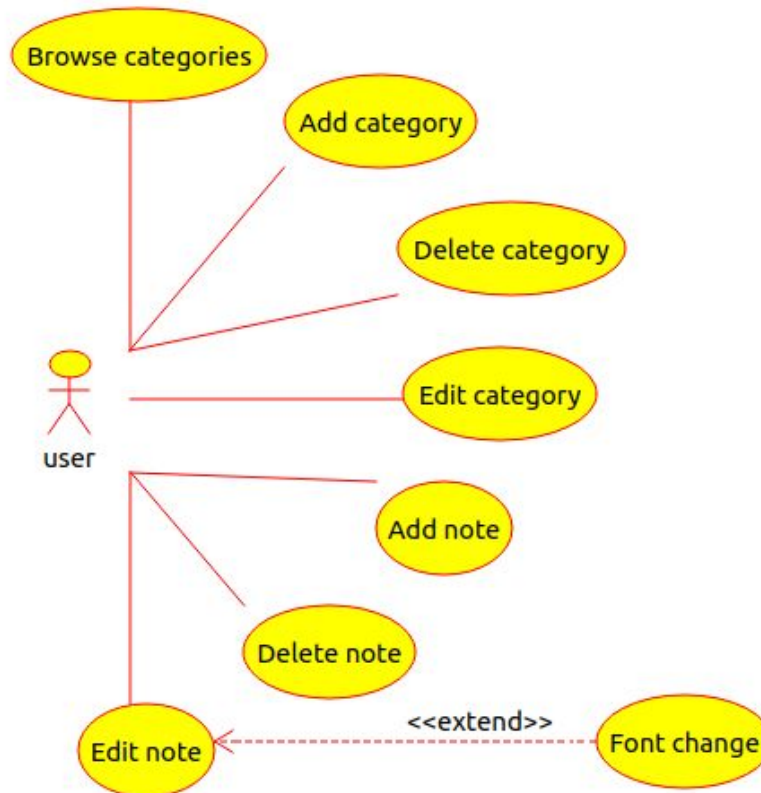


Fig. 1: System use-case diagram

For every use case was created unified description for better understand every possible feature in our system.

Case name: add note

Standard process:

1. User initializes action by click "Add note" button in the category view.
2. Window with simple form for note name and color appears.
3. User confirms the entered data.
4. The view is refreshed.

Alternative process:

1. User initializes action by click "Add note" button in the category view.
 2. Window with simple form for note name and color appears.
 3. User clicks cancel button and window disappears.
-
1. User initializes action by click "Add note" button in the category view.
 2. Window with simple form for note name and color appears.
 3. User enters not unique note name.
 4. In the window text field is marked in red and above it text "That name is already exist." appears.

Time dependencies:

1. Typical execution time: ~1 min.

2. Max execution time: undefined.

Result: New note in specific category.

Case name: delete note

Standard process:

1. User selects "Delete" option from popup menu after clicking right mouse button at specified note.
2. Window with confirmation request appears.
3. User confirms delete request.
4. The view is refreshed.

Alternative process:

1. User selects "Delete" option from popup menu after clicking right mouse button at specified note.
2. Window with confirmation request appears.
3. User withdraws operation by clicking cancel button.

Time dependencies:

1. Typical execution time: <1 min.
2. Max execution time: undefined.

Result: Deleted note from the system.

Case name: edit note

Standard process:

1. User selects "Edit" option from popup menu after clicking right mouse button at specified note or edit button inside note view.
2. Edit note view appears.
3. User edits note informations.
4. User confirms changes by clicking confirm button.

Alternative process:

1. User selects "Edit" option from popup menu after clicking right mouse button at specified note or edit button inside note view.
2. Edit note view appears.
3. User withdraws operation by clicking cancel button.

Time dependencies:

1. Typical execution time: 3-5 min.
2. Max execution time: undefined.

Result: Note with new content.

Case name: font change

Standard process:

1. User chooses font family in combo box.
2. User chooses font size in combo box.
3. New font is visible in text areas and fields.

Alternative process: None.

Time dependencies:

1. Typical execution time: <1 min.
2. Max execution time: <1 min.

Result: New font in note and edit view.

Case name: browse categories

Standard process:

1. User selects willd category by clicking it.

2. Categories sorted alphabetically appears on the view as colored tiles.

Alternative process: None.

Time dependencies:

1. Typical execution time: instant.
2. Max execution time: instant.

Result: Screen with every category in the system.

Case name: add category

Standard process:

1. User clicks "Add new" button in the dashboard.
2. Window with simple form for category name and color appears.
3. User enters name and choose color.
4. User confirms form by clicking confirm button.
5. The view is refreshed.

Alternative process:

1. User clicks "Add new" button in the dashboard.
 2. Window with simple form for category name and color appears.
 3. User withdraws operation by clicking cancel button.
-
1. User clicks "Add new" button in the dashboard.
 2. Window with simple form for category name and color appears.
 3. User enters not unique category name.
 4. In the window text field is marked in red and above it text "That name is already exist." appears.

Time dependencies:

1. Typical execution time: ~1 min.
2. Max execution time: undefined.

Result: New category in the system.

Case name: delete category

Standard process:

1. User selects "Delete" option from popup menu after clicking right mouse button at specified category.
2. Window with confirmation request appears.
3. User confirms delete request.
4. The view is refreshed.

Alternative process:

1. User selects "Delete" option from popup menu after clicking right mouse button at specified category.
2. Window with confirmation request appears.
3. User withdraw operation by clicking cancel button.

Time dependencies:

1. Typical execution time: <1 min.
2. Max execution time: undefined.

Result: Deleted category from the system.

Case name: edit category

Standard process:

1. User selects "Edit" option from popup menu after clicking right mouse button at specified category.
2. Window with confirmation request appears.

3. User confirms new name and color information.
4. User confirms edited data.

Alternative process:

1. User selects "Edit" option from popup menu after clicking right mouse button at specified category.
2. Window with confirmation request appears
3. User withdraw operation by clicking cancel button.

Time dependencies:

1. Typical execution time: ~1 min.
2. Max execution time: undefined.

Result: Category with edited name and/or color.

UI visualisation

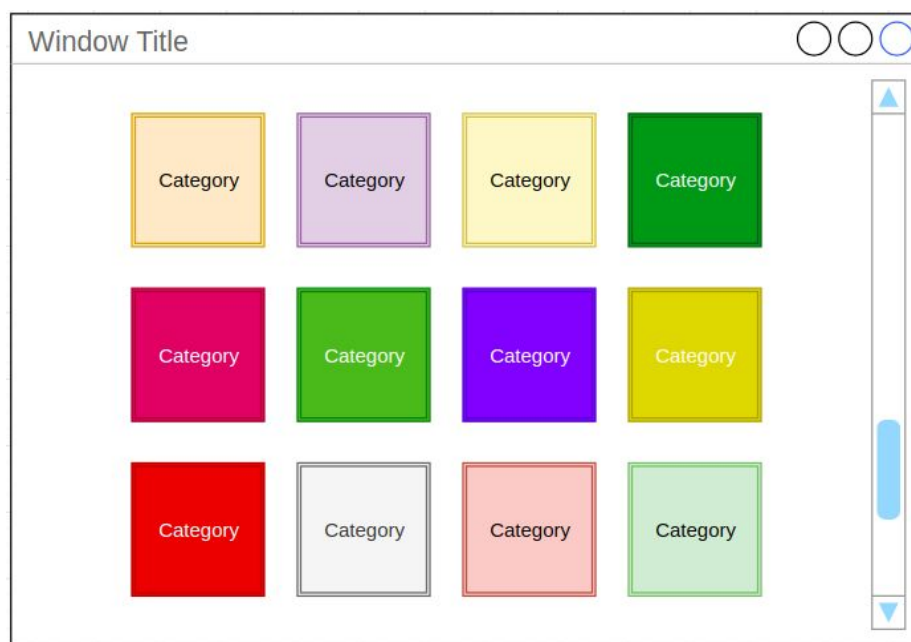


Fig. 2: Visualisation of the main application view

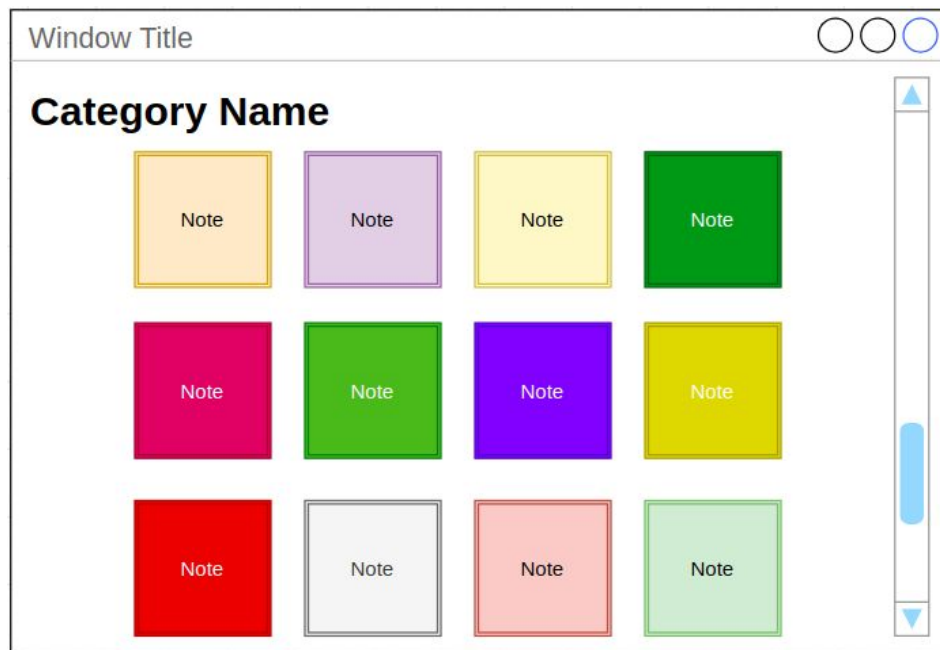


Fig. 3: Visualisation of the category view



Fig. 4: Visualisation of the note view



Fig. 5: Visualisation of the note's edit mode

Details of the implementation of the functionality

Whole application code will be separated into three layers. Just like in design pattern Model View Controller.

- Model – entities, physical representation of the object.
- View – UI elements and way of refreshing UI.
- Controller – intermediate layer. Has methods for creating model, getting parameters, rendering view, handling user actions etc.

Add category

Operation starts when user clicks “Add new” in dashboard view. It ends when window with simple data form disappears. If operation ends with success, then in dashboard user can see new category.

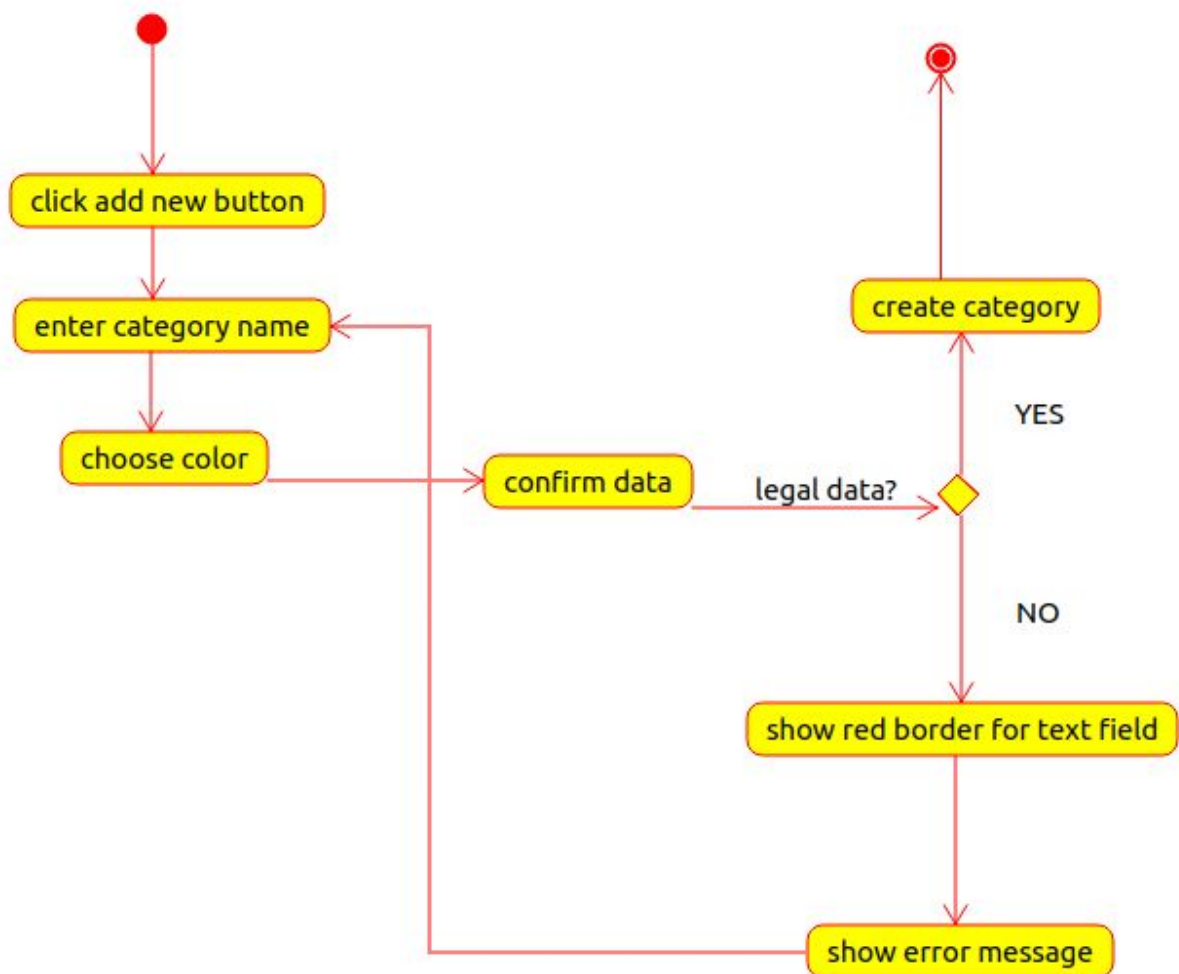


Fig. 6: Activity diagram for adding category

Delete category

Operation starts when user clicks "Delete" from popup menu which can be shown by click right mouse button on specified category in dashboards. It ends when warning dialog disappears. After success specified category disappears from the system.

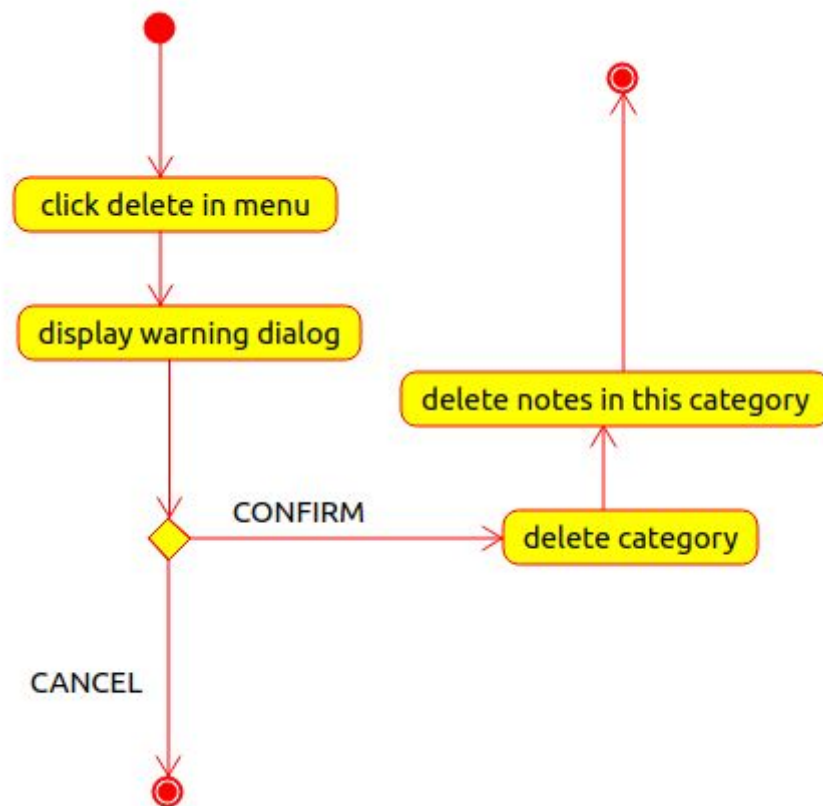


Fig. 7: Activity diagram for deleting category

Deleting one category means that every note written in that category should be also deleted!

Edit category

Operation starts when user clicks “Edit” from popup menu which can be shown by click right mouse button on specified category in dashboards. It ends when warning dialog disappears. After success specified category is updated with new information.

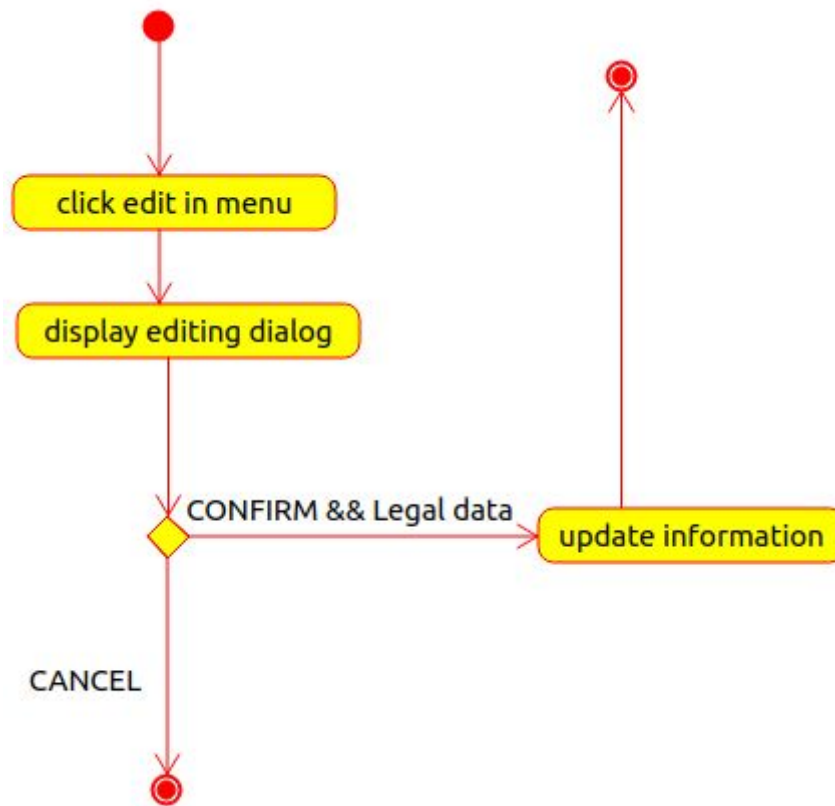


Fig. 8: Activity diagram for editing category

Add note

Operation consist in adding new note in specified category by fill simple form in dialog window. If operation ends with success in specified category new note will be visible.

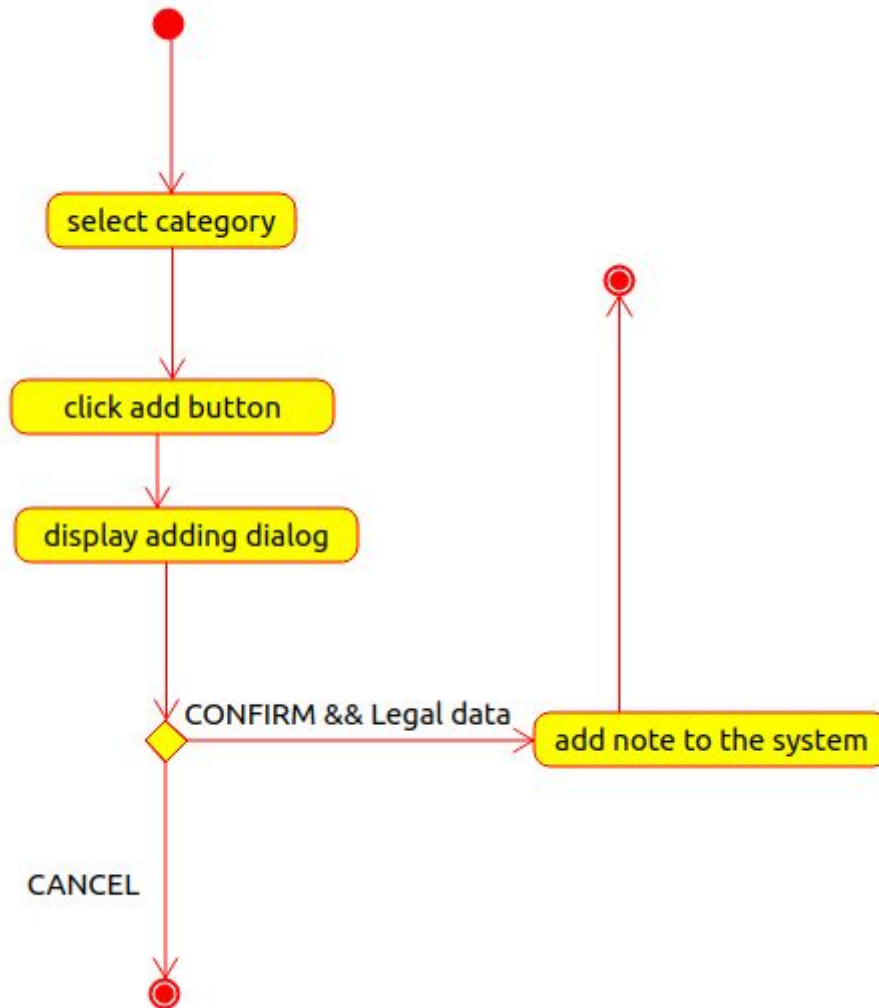


Fig. 9: Activity diagram for adding note

Just like in adding category case – system should have simple filter and decide which note has unique name in that category (because in the system we can create same note but in two different categories). To identify note system should has two field key — note name and category name.

Delete note

Deleting note process is very similar to deleting category. It's pretty simple – user selects “delete” option in popup menu and clicks confirm in warning dialog.

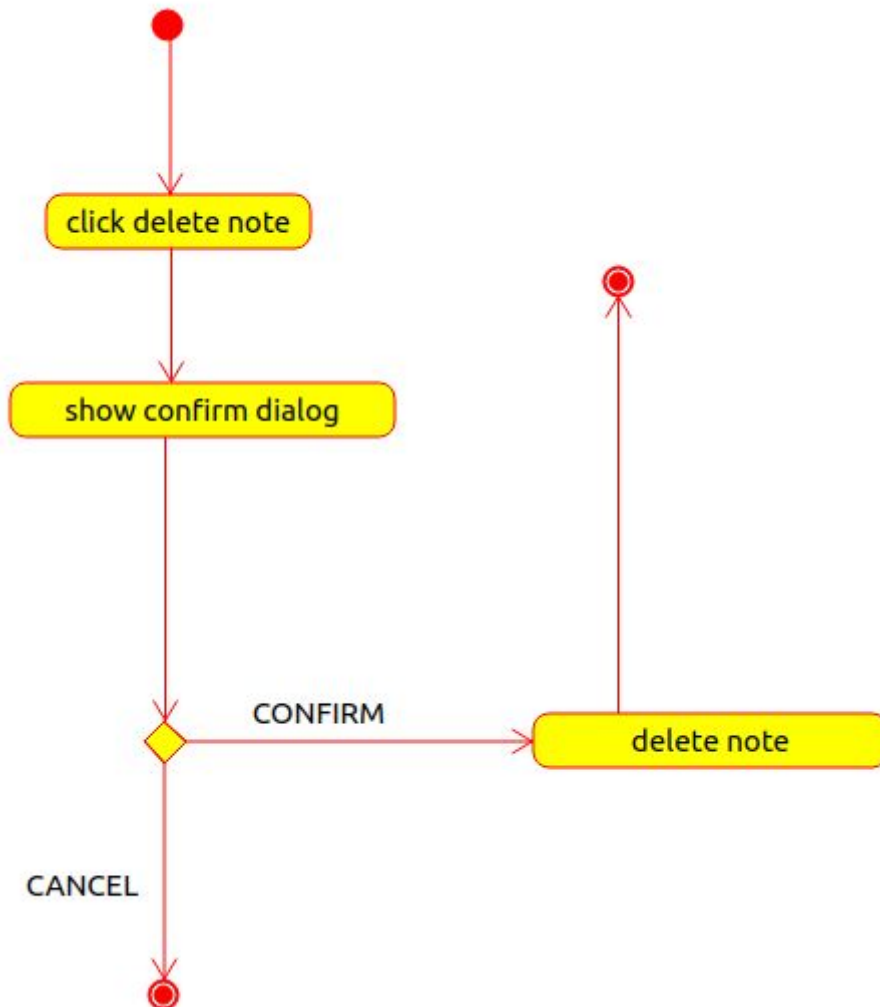


Fig. 10: Activity diagram for deleting note

Edit note

Editing note is more complicated than editing categories, because system needs create whole new view for it. Editing view is quite similar to normal note view but instead inactive text field there's field that is able to edit content.

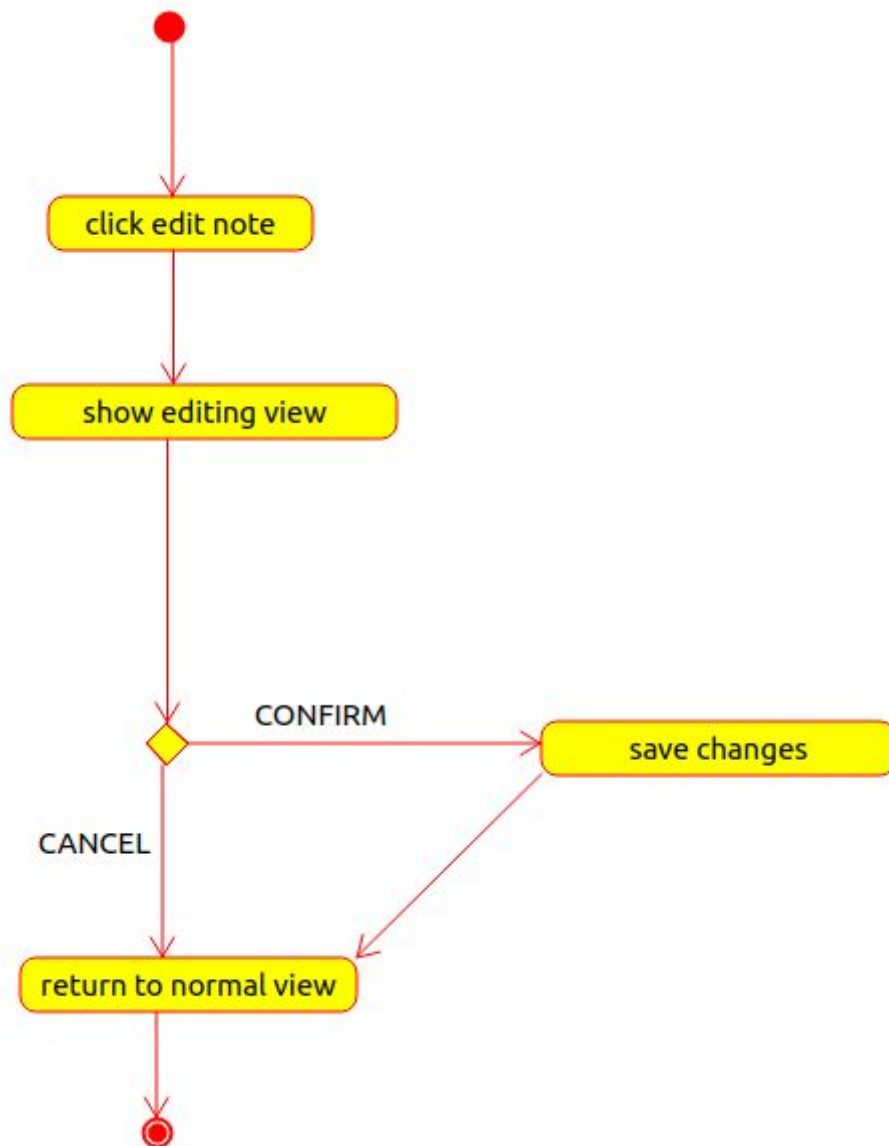


Fig. 11: Activity diagram for editing note

Prototype system design

First part of the class diagram — Dashboard module as a initial point of the application. Dashboard is responsible for displaying and manipulating the list of categories as graphical tiles. Window class is built as Singleton.

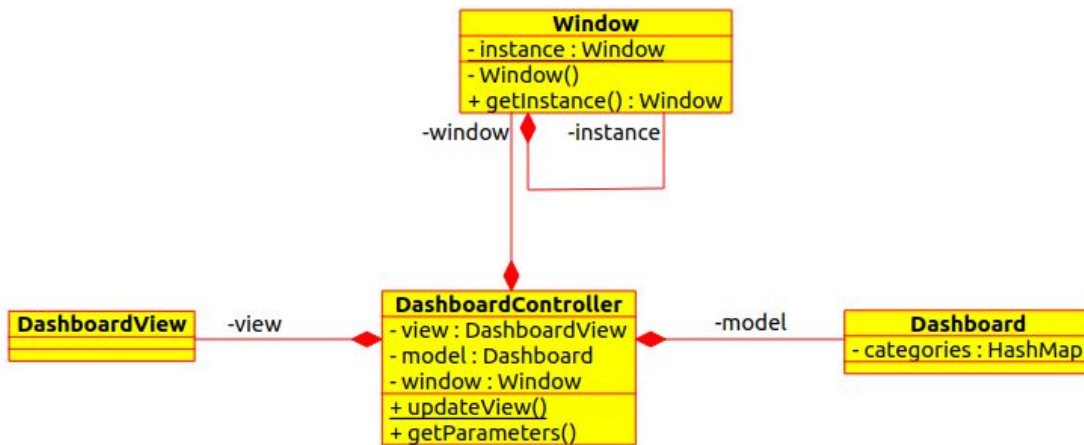


Fig. 12: First part of the diagram

Second part of the diagram is MVC construction. Parts of the Dashboard module is extended and there is Database here.

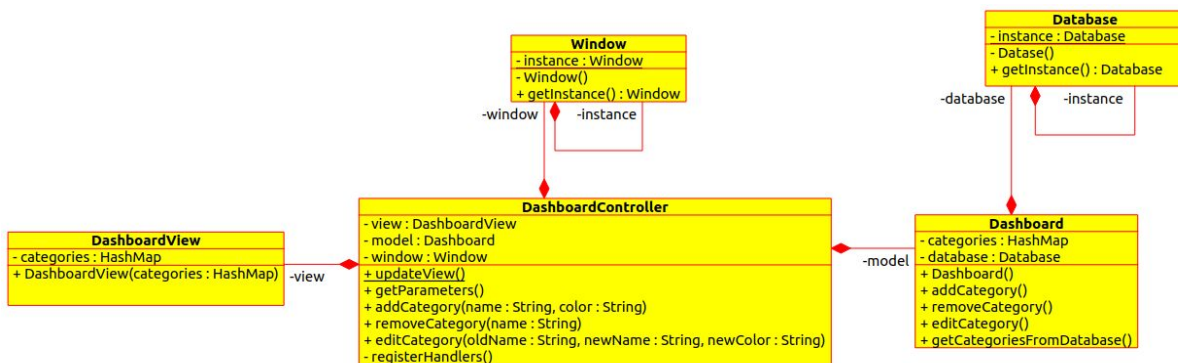


Fig. 13: Second part of the diagram

The last figure represents bottom part of the diagram. There are two modules — for Category and Note.

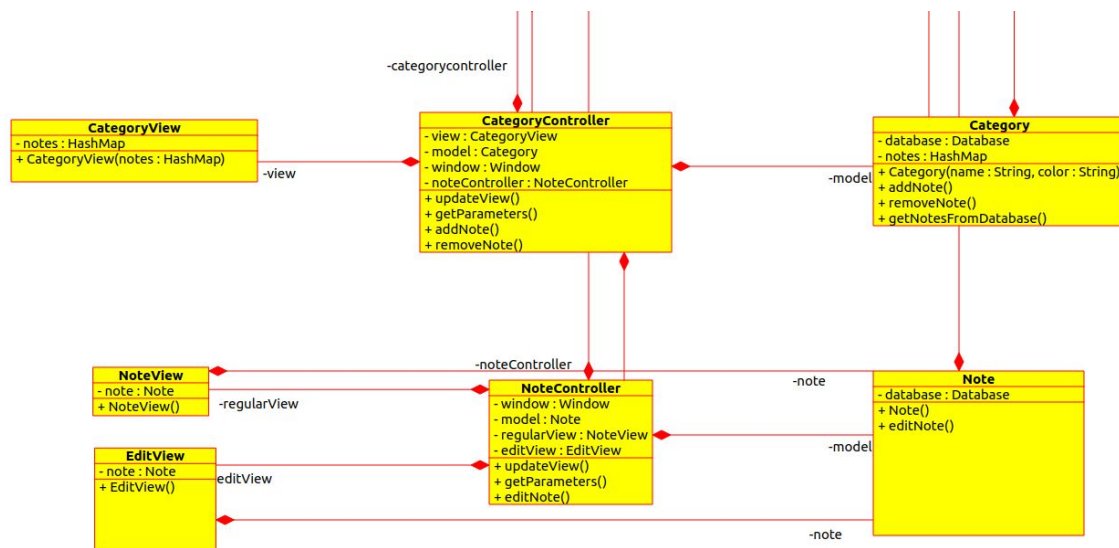


Fig. 14: Last part of the diagram

Diagram doesn't include all methods and fields – it's only for visualisation. Diagram is for show how classes should be connected.

User manual

Use mouse to navigate system (e.g. click on the tile in Dashboard to see specified category with their notes). If you want to back to the previous screen you have to click right button in free space (don't click on text fields or buttons).

You can manage your categories and notes in categories. Also you can specify font in each note.

Before you use this application you should install fonts in your system: Lato, Cantarell, Cousine and Ubuntu – these are default font on Ubuntu system, but not sure if you have it on your Windows or Mac machine.

Versions

Version number	Features	Branches	Date
0.0.0	GUI prototype	version2.0	17.03.2019
0.0.1	Main application window and the database	0.0.1	18.03.2019
0.0.2	Dashboard module without action handling	0.0.2	28.03.2019
0.0.3	Adding, editing and deleting categories	0.0.3	29.03.2019
0.0.4	Category and Note module	0.0.4	31.03.2019
1.0.0	First release	1.0.0	02.04.2019