

# Todo List User Manual.

Name: Kopach Artem

Neptun-code: JQBOLI

## Description for users:

**All fields must be field before adding, another you will get message about wrong input!**

All of this information about buttons you can see also in program, just click to Help button

Main information about buttons:

- Add: add new task with all possibilities
- Save: save to taskList.bin file all your task in List
- Load: load your task from .bin file
- Delete List: delete task List (,but your tasks in file save)
- Delete: delete selected task
- Update: with this button you can update your done, priority or date
- Clear: clear your display

**COMPARE WORK AUTOMATIC** on buttons add, update, save, load.

First condition, done/ not done, after if first condition same for both of tasks deadline date (the newest will be up), then if second same priority, last is estimate hours.

## Description for developers:

Create the application using Swing GUI library (**topic: SWING**). Our application will allow to create new tasks, assign them a title and choose a project for that task to belong to, text with JScrollPane for JTextArea. Besides, we will provide the realisation another features such removing, marking tasks as done (class: JCheckBox), save/load to the file (**topic: Fájlbá írás, fájlból olvasás valamilyen fajta standard I/O**) and get the priority for tasks (**topic: JComboBox**), also we can put estimate hours for every task (with JTextField && Duration), besides can mark the deadline time for all tasks too (with JTextField && rather different parse to Date). Also, used layouts, main layout for main panel is BorderLayout, in another panels FlowLayout and BoxLayout with vertical. Used JOptionPane a lot for say user about wrong input or CONFIRM click (Delete List (, because it is so important button, which dele all List of Tasks and you will lose this if do not save before)).

```
JFrame frame = new JFrame("TODOList");
```

```
    frame.setSize(950, 500);
```

```
(initial size)
```

Also, developers can find Javadoc comments in a code!

For all of this we provide a class Task which has rather good constructor:

```
public Task(String name, String text, Boolean done,
String priority, Date date, Duration estimate){
    this.name = name;
    this.text = text;
    this.done = done;
    this.priority = priority;
    this.date = date;
    this.estimate = estimate;
}
```

After that this class will a lot of use for serialization and compare. Task class a lot of help with this, and get for us simpler and readable access.

For compare we assume TaskComparator class, which will get for us

@Override

public int compare(Task a, Task b) { ... } for this compare() I wrote my realisation,

which includes some interesting moments for example used compare for date, and the most interesting moment is priority, because we used ArrayList<>.sort() which get (a,b) and work from top i must implement such not simple realisation and check manual this and automatic with Junit test \*\*see test

public void TestTaskComparator() . There I checked TaskComparator class.

And final I did Model class which will help to solve problem with division of „roles”.

In this class I collect methods which use in ActionListener in buttons.

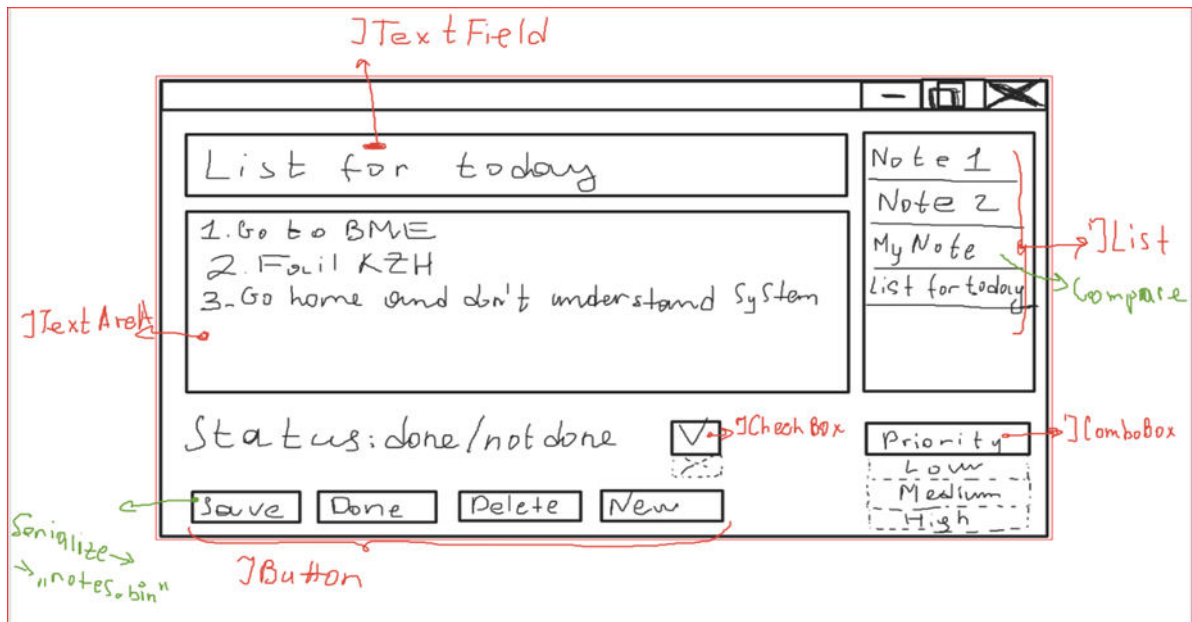
Methods a lot of work with „Model” of program. With HashMap and ArrayList<Task>, which sort(). For this we provide the important part from Main class and properly good divide this. This solution help to debug a lot of errors and do our code more-more readable. Such a good method is the last one:

```
public ArrayList<Task> update(String oldName, String newName, String text, Boolean done,
String priority, Date date, Duration estimate) {
    delete(oldName);
    add(newName, text, done, priority, date, estimate);
    return getTasks();
}
```

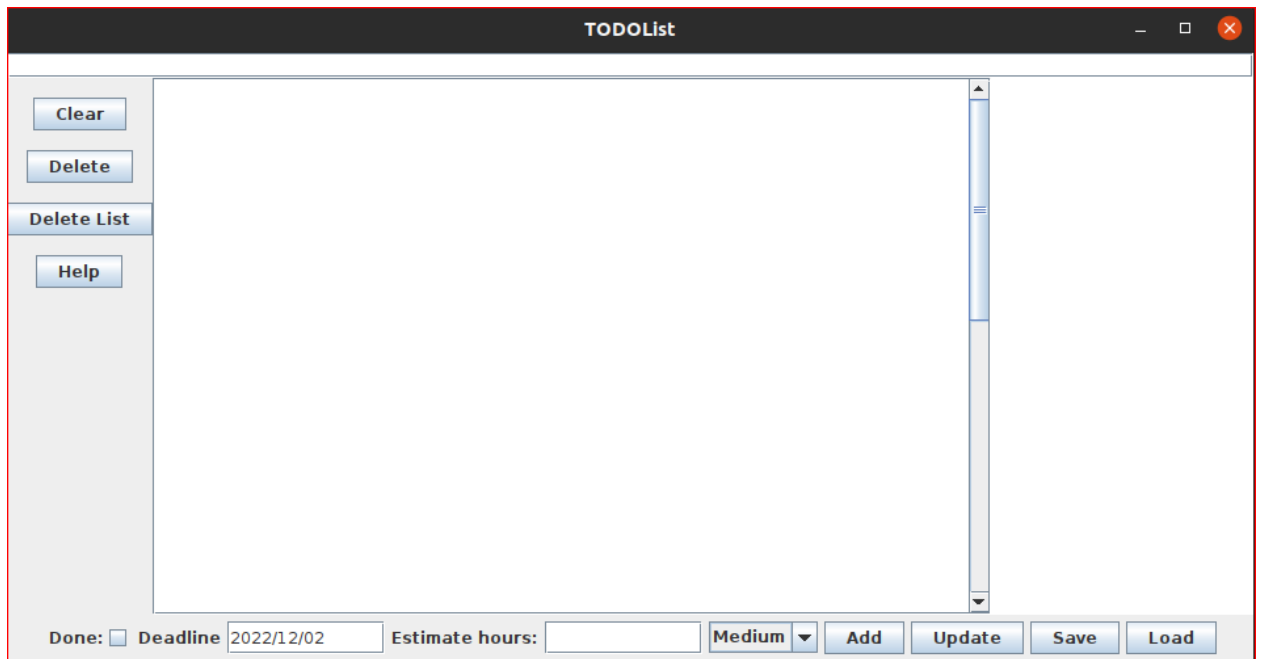
Here we used all methods, which was before (-\_-). This is really simple and fast one of the hardest button in our program.

Our application was planed like this:

The mockup shows a window with a title bar. Inside, there's a text field at the top containing "List for today". Below it is a text area with a list: "1. Go to BME", "2. Fail KZH", and "3. Go home and don't understand System". To the right of the text area is a list box containing "Note 1", "Note 2", "My Note", and "List for today". Below the text area is a label "Status: done/not done" next to a checked checkbox. At the bottom are four buttons: "Save", "Done", "Delete", and "New". To the right of the buttons is a priority dropdown menu with options "Low", "Medium", and "High".



**Now it is look like:**



You can see new Buttons: Help, Delete List, Clear, Update! Read upper description for user to more information!

### **Testing part:**

I did class TODOListTest in which I wrote 12 different tests for all methods from Model class, check Task class, check Exceptions and see also TaskComparator too.

### **Class diagram by using diagram plug-in:**

