

"Lab 5: GUI I: Layouts and Widgets"

Build the LMS main window with nested layouts and a table

Lab 5 of 9

Branch: week8-gui-layouts

About 2 to 3 hours

Guide: LMS Guide, Step 5

Objective

Build the **look** of the LMS: a main window with a heading, a data table, an add-item form, and buttons, using BorderLayout, GridLayout and FlowLayout. Buttons will not work yet. That is next week.

Before you start

- Lab 4 merged. Lecture 8 notes read.
- Create a package `lms.ui`.
- Sketch the window on paper **first**. Mark which layout manager each region uses (`docs/ui-sketch.jpg`).

Git: start and submit

```
git checkout main
git pull origin main
git checkout -b week8-gui-layouts
```

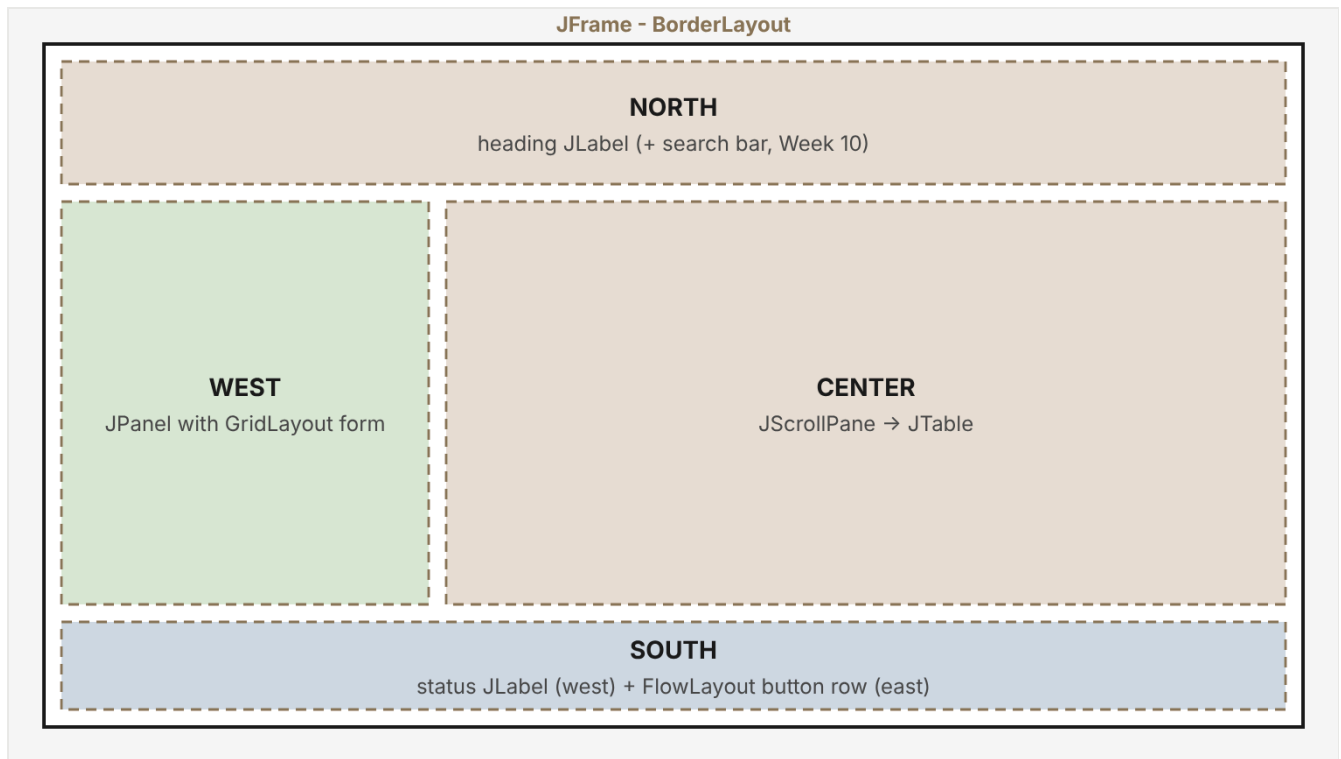
Commit small and often (see the *Git Support Guide*). When you finish:

```
git push -u origin week8-gui-layouts
```

Open a Pull Request titled **Week 8: GUI I (Swing Layouts & Widgets)**, fill in the template, and request a review from your lecturer.

Design first (10 minutes)

Your layout plan should match this structure (`docs/ui-sketch`):



Question to answer in your PR: *which container uses which layout manager, and why that one?*

Tasks

Task 1: MainFrame extends JFrame

```
Constructor MainFrame(Library library); title Library Management System;
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); size about 820 × 460;
setLocationRelativeTo(null).
```

Task 2: the table (CENTER)

Columns `ID`, `Title`, `Type`, `Detail`, `Status`. Use a `DefaultTableModel` inside a `JTable` inside a `JScrollPane`. Write `refreshTable()` that clears the model and adds one row per item (use `instanceof` to build the `Detail` text: author for books, 120 min for DVDs, Issue 42 for magazines).

Task 3: the form (WEST)

A `JPanel` with `new GridLayout(0, 2, 6, 8)` holding: `Type:` + `JComboBox<String>` (Book, DVD, Magazine), `ID:`, `Title:`, `Detail:` text fields, and an `Add Item` button. Give it a titled border. Put it in **another** `JPanel` with `BorderLayout` and add the form to its `NORTH` so it keeps its natural height.

Task 4: heading (NORTH) and buttons (SOUTH)

A bold, centred `JLabel` heading. A `FlowLayout(FlowLayout.RIGHT)` panel with `Borrow` and `Return`; a status `JLabel` (" Ready") on the left of the same `SOUTH` area.

Task 5: launch

Write `lms.GuiMain` that builds a `Library` with a few items and members, then:

```
SwingUtilities.invokeLater(() -> new MainFrame(library).setVisible(true));
```

Why `invokeLater`? Put your one-sentence answer in the PR.

Task 6: screenshot

Take a screenshot of the running window and save it as `docs/week8-window.png`.

Expected output

Your window should look like this (colours depend on your operating system):

ID	Title	Type	Detail	Status
B001	Clean Code	Book	Robert C. Martin	Available
B002	Refactoring	Book	Martin Fowler	Available
D001	Introduction to ...	DVD	120 min	Available
G001	Software Weekly	Magazine	Issue 42	Available

Checkpoints

Tick each before moving on. If a box stays empty, ask for help *now*; later steps build on it.

- ☐ The window opens centred and closes the program when you close it
- ☐ The table shows your items with all five columns and a header row
- ☐ The form stays compact at the top-left, not stretched to the bottom
- ☐ Buttons are at the bottom right
- ☐ Resize the window: the table grows, the form does not
- ☐ `docs/week8-window.png` committed

Experiments: break it on purpose

Learning happens when you predict, run, and compare. Write your **prediction** first, then paste the real compiler message or output into your PR description.

1. Add the form panel straight to `BorderLayout.WEST` (without the wrapper). Take a screenshot and explain what happened.
2. Remove `JScrollPane`. What disappears?
3. Change the form's `GridLayout(0, 2, ...)` to `GridLayout(2, 0, ...)`. Predict first.
4. Remove `setVisible(true)`. What do you see, and what does the program do?

Stretch tasks (optional, no marks lost if skipped)

- Make the heading use a bigger font and a coloured border.
- Add a `JMenuBar` with *File* → *Exit*.
- Disable table editing by overriding `isCellEditable` to return `false`.
- Try `table.setAutoCreateRowSorter(true)` and click a header.

Common pitfalls

Symptom	Fix
Blank window	<code>add(...)</code> was called on the wrong container, or components were added after <code>setVisible</code> without <code>revalidate()</code>
Window is a tiny sliver	Missing <code>setSize</code> (or <code>pack()</code>)
<code>NullPointerException</code> in the constructor	You used a field before creating it; create widgets in the field declaration
Nothing happens when I click	Correct: events are next week

What to submit

- ☐ PR **Week 8: GUI I (Swing Layouts & Widgets)** from branch `week8-gui-layouts`
- ☐ `docs/week8-window.png` and your sketch
- ☐ PR description: which layout where and why; the `invokeLater` answer; what experiments 1 to 4 showed

Marking guide

Your PR is marked on four criteria. Each is scored 4 (excellent) to 1 (needs work) and weighted as shown.

Criterion (weight)	4	3	2	1
Works (40%)	All tasks run, expected output matches	Minor differences	Partly works	Does not compile or run
OOP design (30%)	Concepts of this week used correctly and deliberately	Mostly correct	Concepts present but misused	Concepts missing
Code quality (15%)	Clear names, small methods, no dead code, comments explain <i>why</i>	Mostly clean	Messy in places	Hard to read
Git and PR (15%)	Exact branch/PR name, small clear commits, full PR description, UML in docs/	One slip	Several slips	One giant commit or direct to <code>main</code>

Check Before you open the PR, read your own diff on GitHub as if you were the reviewer. Most deductions are things you would have spotted yourself.