

License Manager

Jack Doyle & Jose Tovar

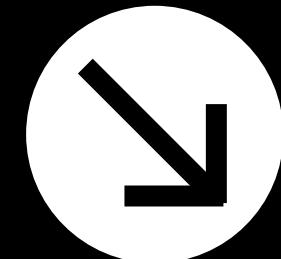


Jack

Core Development

Jack's background is in building iOS apps and full stack web tools, so he took the lead on most of the system engineering. Jack came into this, not knowing too much Java.

He built the core backend logic, the database layer, and the main functional interface that makes the license manager actually work day to day. His experience with app design and system flow shaped how the whole project runs.

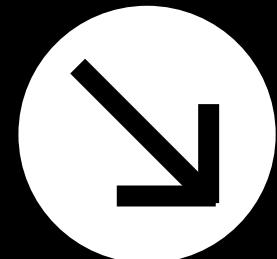


Jose

Interface Setup & Entry Points

Jose's strength is Java, which made him well-suited for setting up the project foundation.

He built the login system and the initial GUI structure that the rest of the interface was built on. His familiarity with Java helped establish the starting point the system grew from. He also built a few other modules like Dashboard, AddNewLicense, and ReportsandAnalystics. Without Jose's knowledge of Java, this project wouldn't have gone far.



What did we build?

We wanted to build something actually useful. Picture a company with around three hundred employees, each needing access to multiple paid tools, Excel, Word, Zoom, Adobe, and more. That means hundreds of licenses the company is paying for. You'd want to know those licenses are actually being used, not wasted.

That's where Jack & Jose come in.



DEMO TIME



Success

Success came early because we did our best to work efficiently.

Using GitHub to coordinate the project felt like a real workflow win. Sharing files, tracking changes, and keeping everything organized made the whole build process feel more professional.

Impediments

Our biggest obstacle was scheduling.

Finding meeting times was nearly impossible, but we still kept progress steady and never let it stall the project.

Failure

Not everything worked on the first try.

Our original idea was to use a PowerShell script to scan the machine, but that approach didn't pan out, so we shifted to a different method. We also started with MySQL but ran into account and payment issues, which pushed us to switch to PostgreSQL instead.

Success

Our biggest win was seeing the system run end to end.

Once the program could scan a machine, detect active software, and update the database live, we knew the full pipeline worked exactly as intended. It nearly brought tears to my eyes.

Jose

Better use of Github for version control of the project. We created a repository but did not use it as intended. This made it difficult to build a project with one GUI template.

Jose

Success is not linear. We had a vision in the beginning and we were able to execute the plan we proposed. I learned that it's okay to ask for help. We also should have used a schedule plan better

Jack

Looking back, I think I could've pushed this further. The project worked and came out solid, but compared to teams building things like AI bots or full event platforms, ours felt a bit contained. If I had more time, I would've expanded the scope and made it more ambitious.

Jack

I'd also streamline deployment. Right now, an IT person would need to manually install this on each machine. If I were to redo it, I'd build an easier distribution method so companies could roll it out automatically across their devices.

What would we do differently?

Future Growth

Our tool works, but it can grow far beyond its current scope. The next stage is turning this from a class project into a scalable product.

Something a real company could deploy across hundreds of machines, monitor licenses in real time, and use to make smarter budgeting decisions.

Auto Deployment

We want to package the scanner so it installs itself across an entire company network, no manual setup needed.

Advanced Analytics

Future versions could track trends, flag unused licenses automatically, and help companies cut costs with data-driven or even AI recommendations.



Remember we're graduating.

You need computer geniuses, and we need jobs.

Jack

+1 (978) 727-4865

jack@jgcks.com

Jose

+1 (401) 536-2842

josedtovar22@gmail.com

Doyle:

Tovar:



Any Questions?

Ask us anything, go on, I double dog dare you.

Project Infrastructure

- Desktop application built in Java Swing.
- DatabaseHelper manages JDBC connection.
- PostgreSQL database 3 main tables
 - users, products, licenses.
- Excel files (via Apache POI)
 - Import and export data.



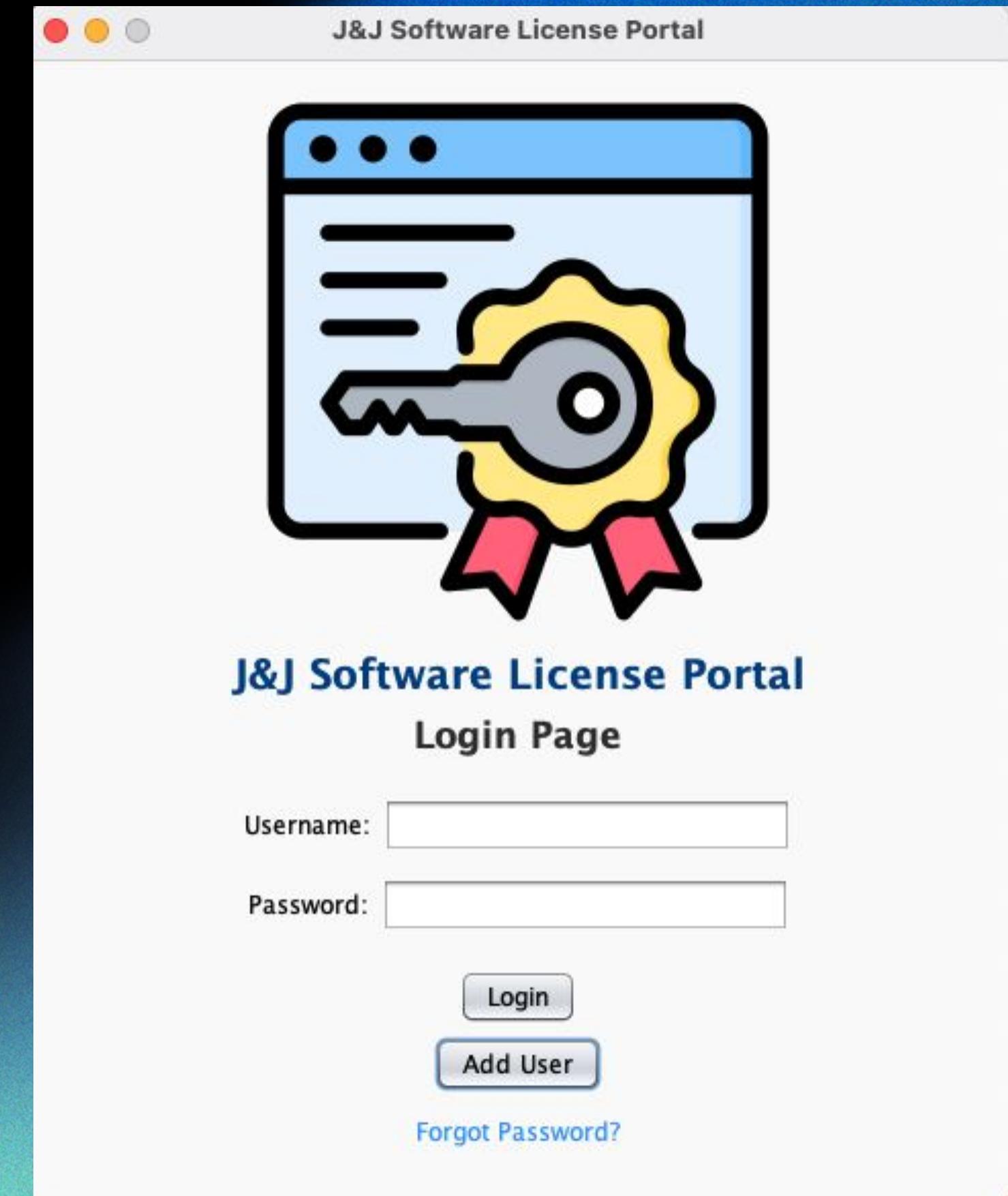
LoginPage.java

Purpose: First screen of the app; authenticates the user.

What it does:

- Takes username + password
- Calls DatabaseHelper.getuser(username) to lookup the user in users table.
- Adds user
- On success, opens Dashboard.

All access to the system starts at the LoginPage, which uses the users table to enforce basic authentication.



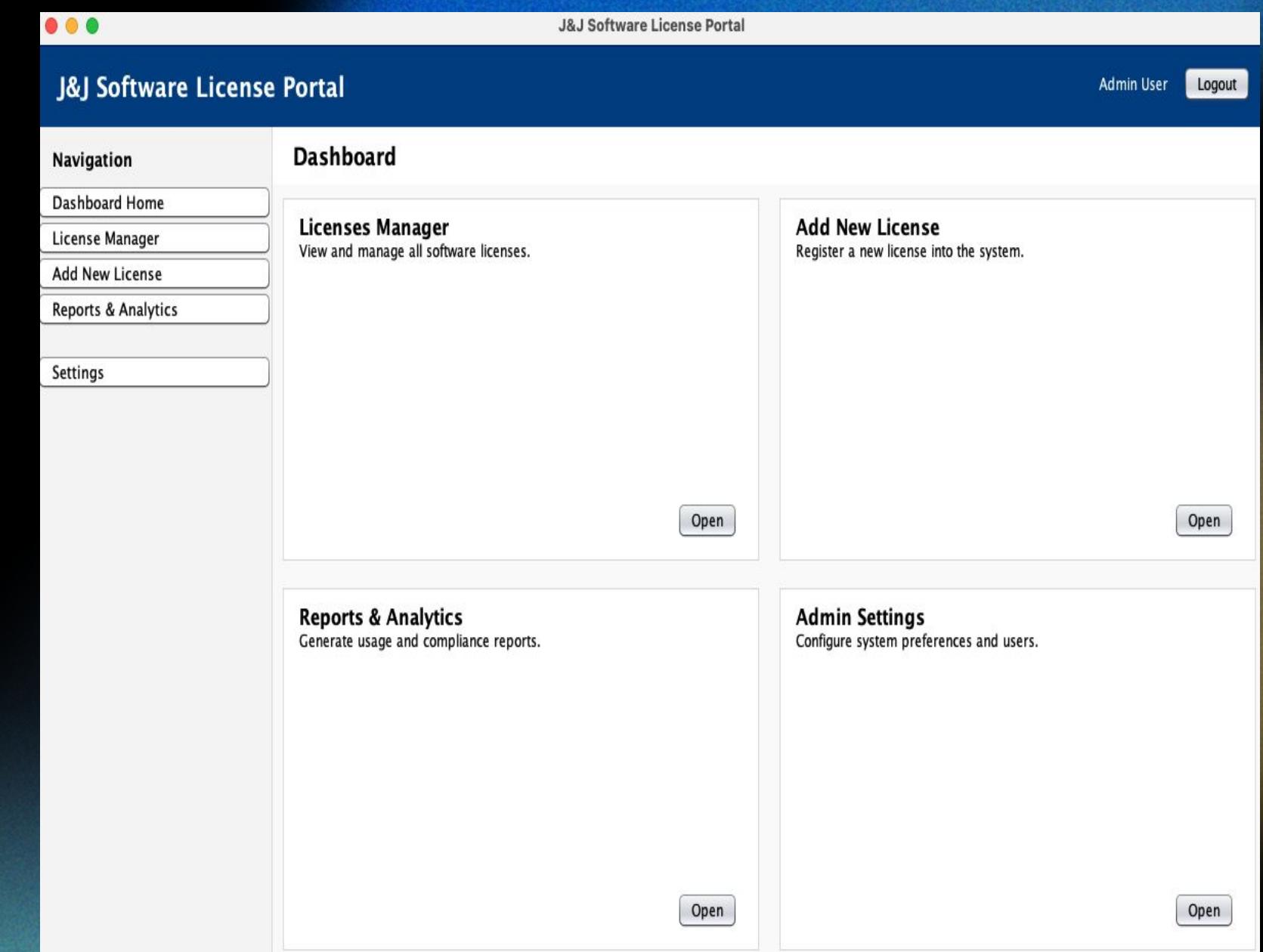
Dashboard.java

Purpose: Main menu after login

What it does:

- Opens License Manager (LicenseManagerGUI)
- Add New license (AddNewLicensePage)
- View Reports & Analytics (ReportsAnalyticsPage)

The Dashboard is the hub of the application. From here admins can manage licenses, add new ones, and open analytics page.

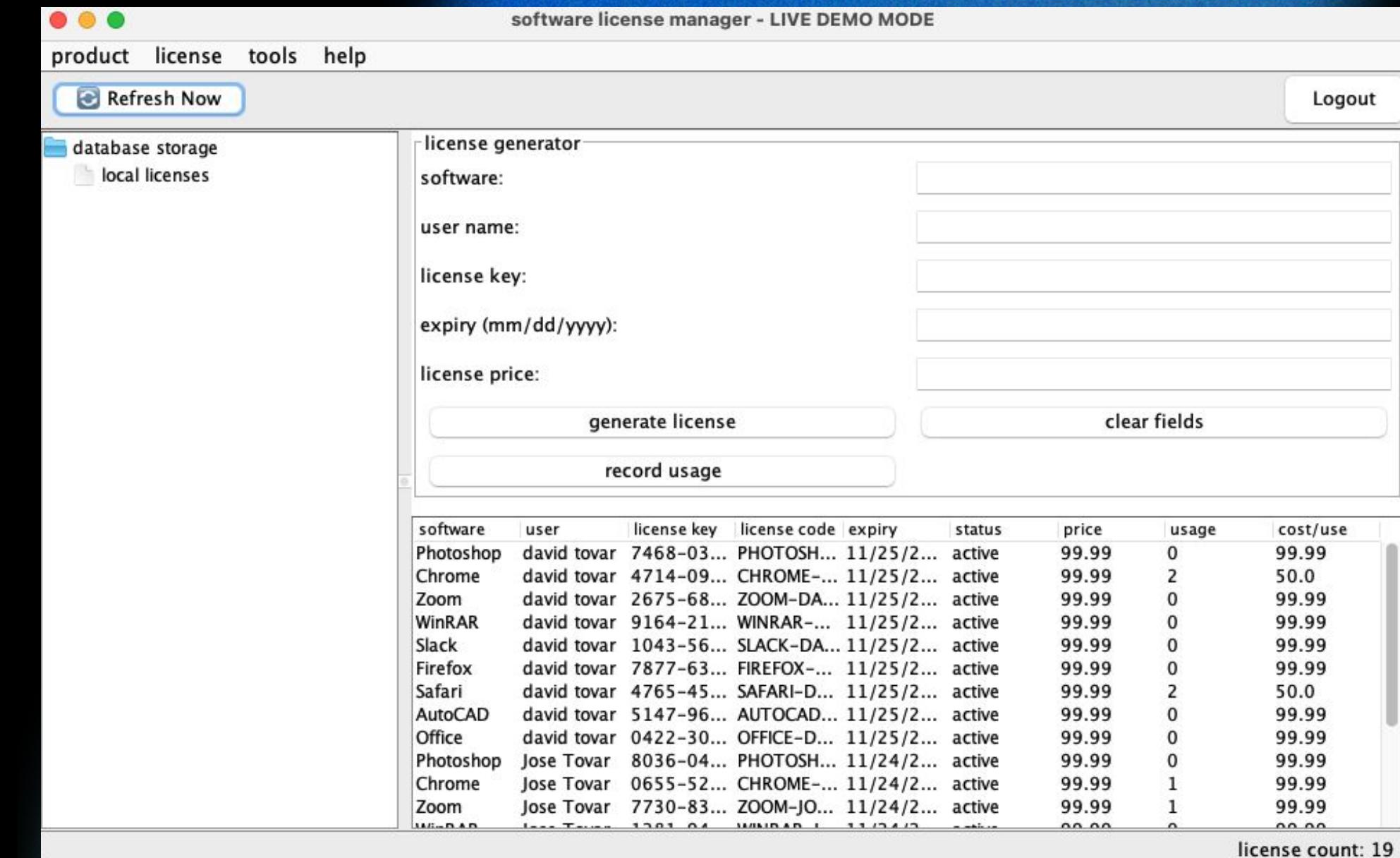


LicenseManagerGui.java

Purpose: Main License Inventory

What it does:

- Calls DatabaseHelper to load data from licenses table
- Shows licenses in a Jtable from Monitor Agent
- Generates software licenses for Users



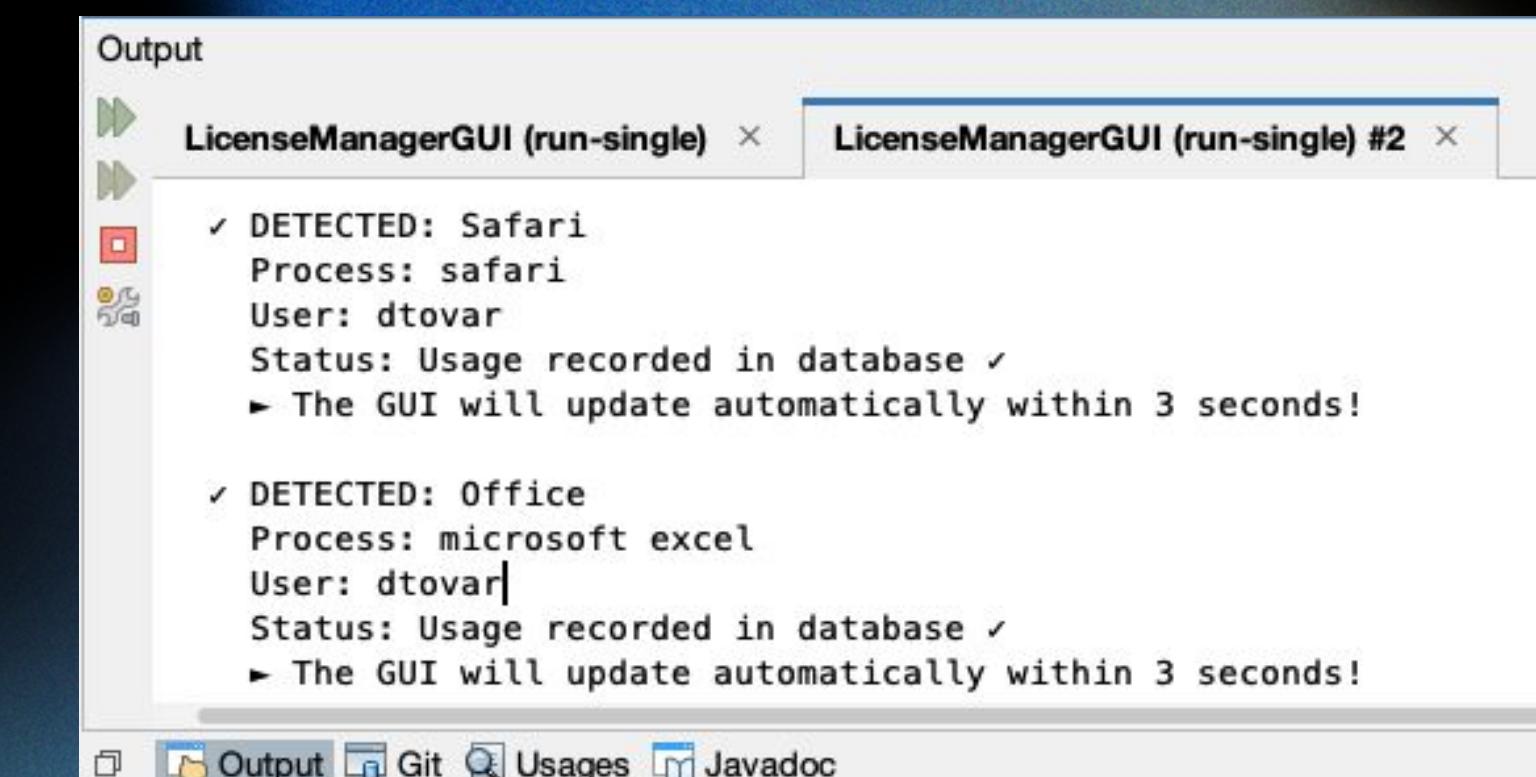
MonitorAgent.java

Purpose: Background usage tracking

What it does:

- Runs on a workstation and watches for a specific application is running in the system.
- Logs each usage session to a table in the database

The monitorAgent tracks user usage based on user and software.



The screenshot shows an IDE's Output window with two tabs: "LicenseManagerGUI (run-single)" and "LicenseManagerGUI (run-single) #2". The "LicenseManagerGUI (run-single) #2" tab is active, displaying the following log entries:

```
✓ DETECTED: Safari
Process: safari
User: dtovar
Status: Usage recorded in database ✓
► The GUI will update automatically within 3 seconds!

✓ DETECTED: Office
Process: microsoft excel
User: dtovar|
Status: Usage recorded in database ✓
► The GUI will update automatically within 3 seconds!
```

At the bottom of the window, there are tabs for Output, Git, Usages, and Javadoc, with "Output" being the selected tab.

AddNewLicensePage.java

Purpose: Create and manage individual license records

What it does:

- Admin can record a new license
- Buttons:
 - Add
 - Update / Delete / Clear
- Excel file
 - used to Jtable to the right. Stores metadata.

This page is where the admin enters a new license. The form fields map directly to columns in the license table. The page supports Excel format data.

J&J Software License Portal - Add New License

Software	Vendor	Type
AlphaOffice Pro	AlphaSoft Inc.	Subscription
GlideVPN Secure	Glide Networks	Subscription
RenderFX Studio	PixelForge Labs	Perpetual
TrackPoint Analytics	AnalytIQ Systems	Subscription
CloudDrive Sync	NebulaWare	User-based
SecureShield AV	ShieldTech	Subscription
StudioPlay Media	Orion Digital	Perpetual
FlowChart Designer	Graphite Systems	Perpetual
VectorLine CAD	VectorLab	Subscription
Nimbus Project Manager	NimbusSoft	Subscription
EncryptPlus Vault	SecureLogic	User-based
StreamCore Server Tools	StreamCore Inc.	Device-based
AtlasDB Client	AtlasData	Subscription
ZenWriter Pro	CalmTech	Perpetual
PulseScan Monitor	Pulse Innovations	Subscription
BlueGate Firewall Client	BlueGate Security	User-based
OptiDesk Tools	OptiWare	Subscription
LumaScreen Designer	LumaStudio	Perpetual
SyncLine Scheduler	SyncLine Systems	Subscription
MetaCheck Compliance	MetaCheck Corp.	Subscription
Adobe Pro	Adobe	Subscription
wireshark	cisco	Subscription

J&J Software License Portal

Admin User Close

Software Name:

Vendor:

License Key:

License Type:

Seats Purchased:

Purchase Date:

Expiration / Renewal:

Cost (USD):

Notes:

Add License Update Delete Clear Reload from Excel

ReportsandAnalytics.java

Purpose: Create and manage individual license records

What it does:

- Reads data from licenses table via Excel
- Gives admin report of upcoming expirations
 - 30, 60, 90, 180 days
- Yearly Cost of licenses
- Cost per Vendor

The screenshot shows a web-based software interface titled "J&J Software License Portal". At the top right, there are "Admin User" and "Logout" buttons. On the left, a vertical navigation menu includes "Dashboard Home", "License Manager", "Add New License", "Reports & Analytics" (which is currently selected), and "Settings". The main content area is titled "Reports & Analytics" and features a sub-header "Show expirations within: 180 days" with an "Apply" button. Below this are three tabs: "Upcoming Expirations" (selected), "Cost per Vendor", and "Total Spend by Year". A table displays license information with columns: Software, Vendor, Expiration, and Cost. The data is as follows:

Software	Vendor	Expiration	Cost
AtlasDB Client	AtlasData	01-09-2026	2300
MetaCheck Compliance	MetaCheck Corp.	01-25-2026	2600
wireshark	cisco	03-24-2026	105
StreamCore Server Tools	StreamCore Inc.	05-05-2026	6700