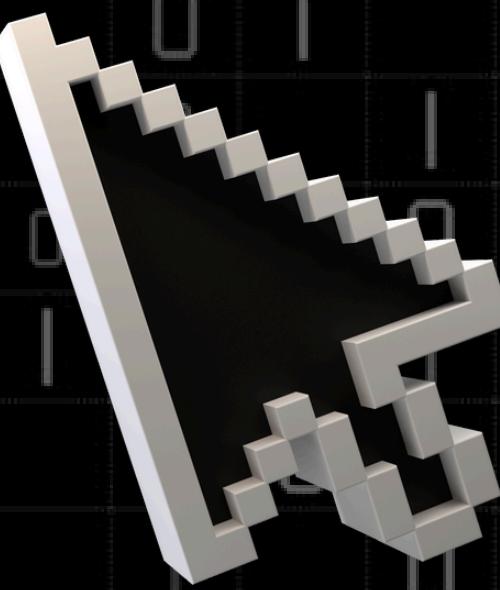


CS102 Smart Attendance System

BY: GROUP 8

JIN RAE, JEREME, NICHOLAS, JUSTIN, MAY, ASHTON



Tech Stack/ Libraries for our Project

Front End GUI

JavaFx

JavaSwing

JavaAwt



Face Recognition

OpenCV:

- Basic computer vision and image processing

Arcface:

- DNN model for embedding generation and recognition

Backend

Supabase postgresql



Tech Stack for our Project

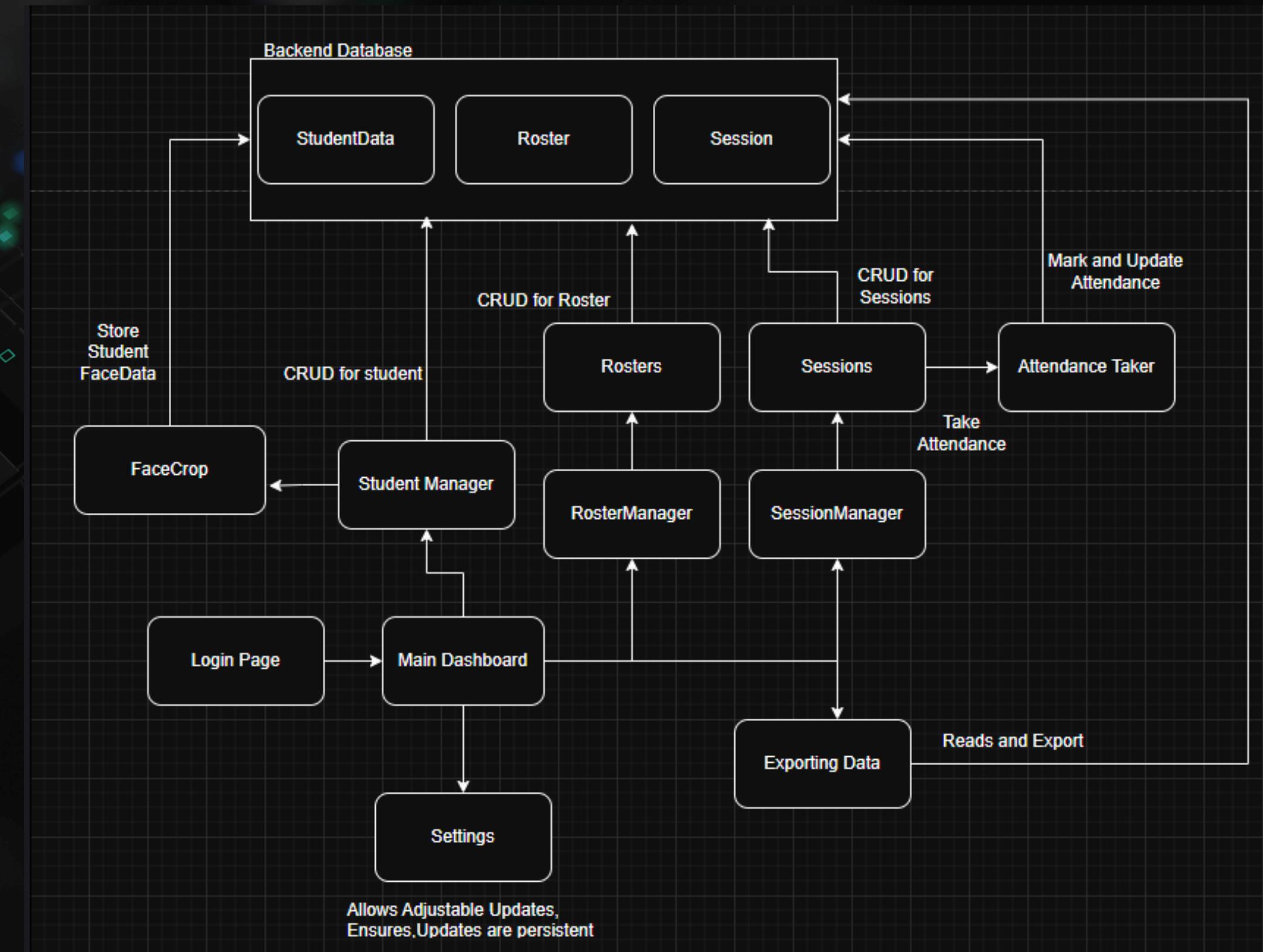
Exporting Data

IJFileChooser (File Explorer GUI)
ITextPDF (PDF)
Apache (Excel)
FileWriter (CSV)

Configuration & Logging

Java Logger

Team's System Architecture



Team's OOP Design

Modular Package



App

Application Entry Point,
Intializing Configs and Logging.

Config

Configuration and Logging of
settings
Listeners for change in
configuration
Updating and storing settings in
a phisical file.

Entity

Entities like Student, Roster,
Session

GUI

Main Pages of the app, GUI styler,
Component parts, Interfaces of
the app

Model

Data Objects for processing, Face
Capturing, Face Recognition

Report

Converting data into Different
types of Files such as CSV, Excel,
PDF.

Team's OOP Design

Modular Package

Repository

Database Connection and functions

CRUD functionality for Database,
Data access from Database

Util

Shared helper functions, library
loaders, Image Preprocessing

Service

Core business Logic, Camera
Features, Processors, Face
Embeddings, Face Detection

Core App Functions

Student
Enrollment &
management

Attendance
Session
Management

Face Detection and
Recognition

Marking
Attendance

Reporting
and Export

GUI

Configuration &
Logging

Features will be shown in the demo!

Additional Features

Database

Multi Face
Recognition

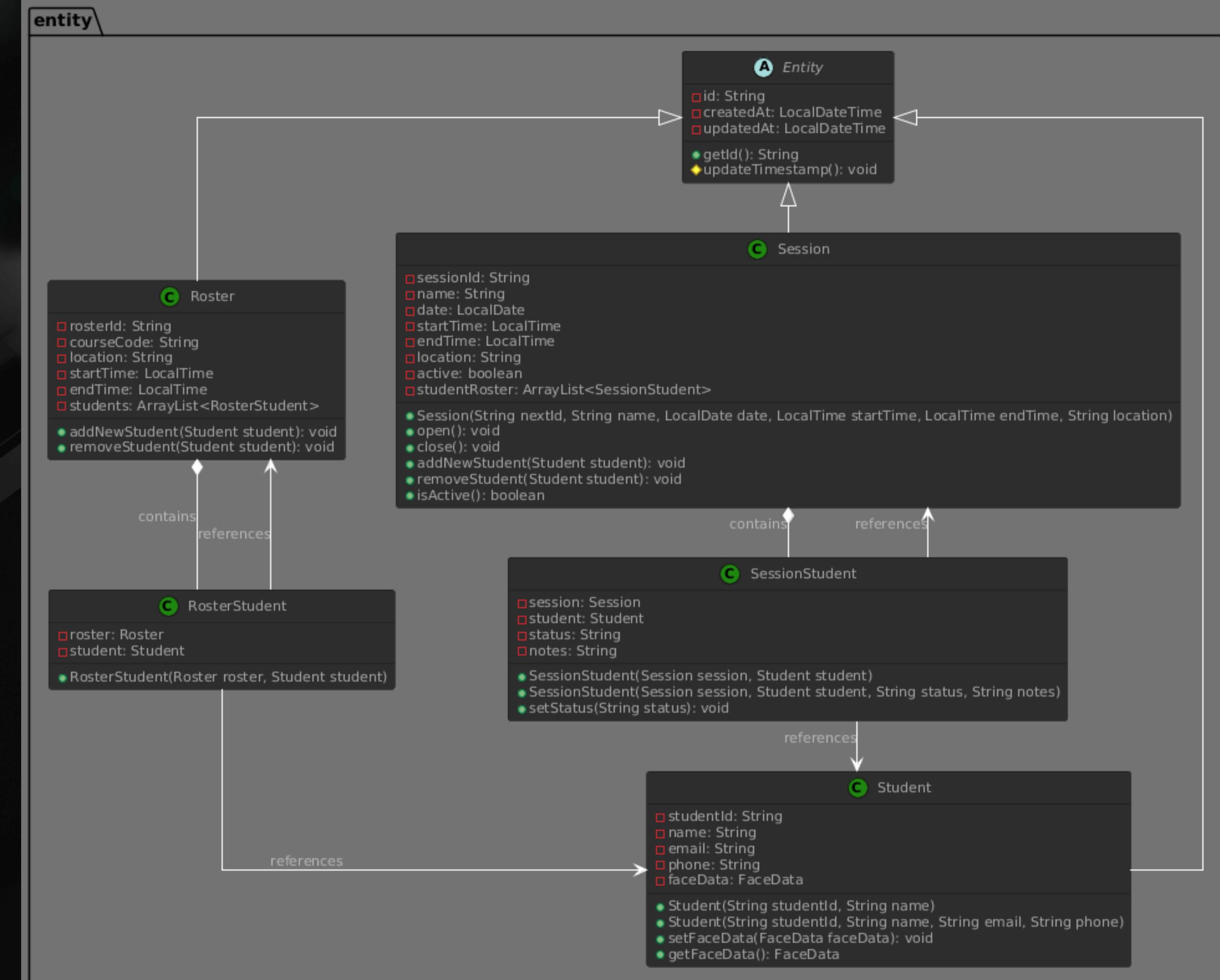
Performance
Optimizing

Image Quality
Validation

Features will be shown in the demo!

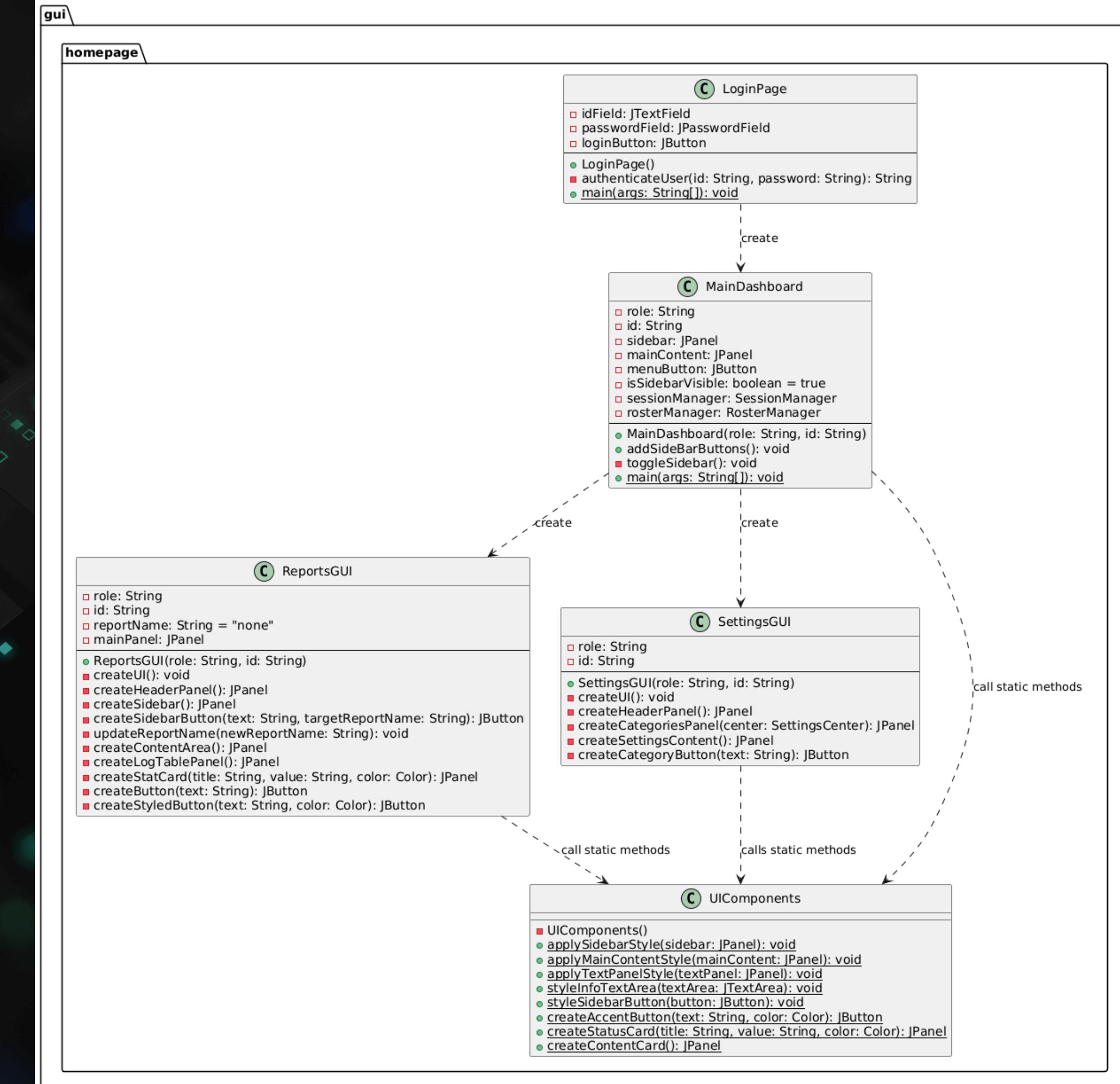
Entity

- Entity
 - Defines key properties and methods
 - Inheritance and Polymorphism
- Roster, Session, Student
 - Encapsulates their properties
 - Abstracts method logic for GUI / service calls
- RosterStudent, SessionStudent:
 - Single, focused responsibility to model association between entities
 - Allows for navigation between entities



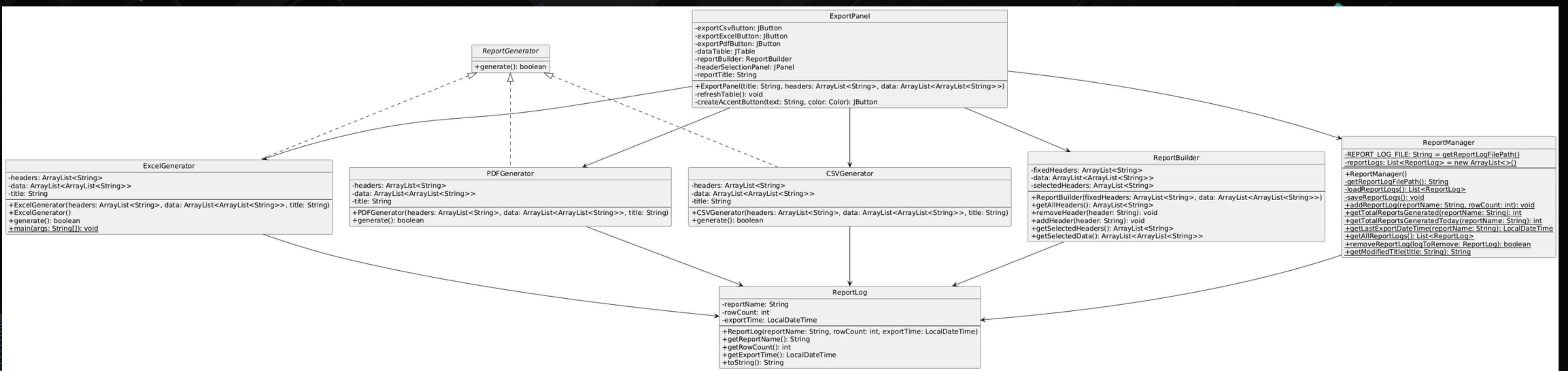
GUI

- Login & MainDashboard
 - **Central interface** of the system with quick access to key features
 - **Inheritance** and **Polymorphism**
- UIComponents
 - **Utility class** with static methods
 - **Composition** of UI styling
- ReportsGUI & SettingsGUI
 - **Encapsulate** the properties
 - **Polymorphism**



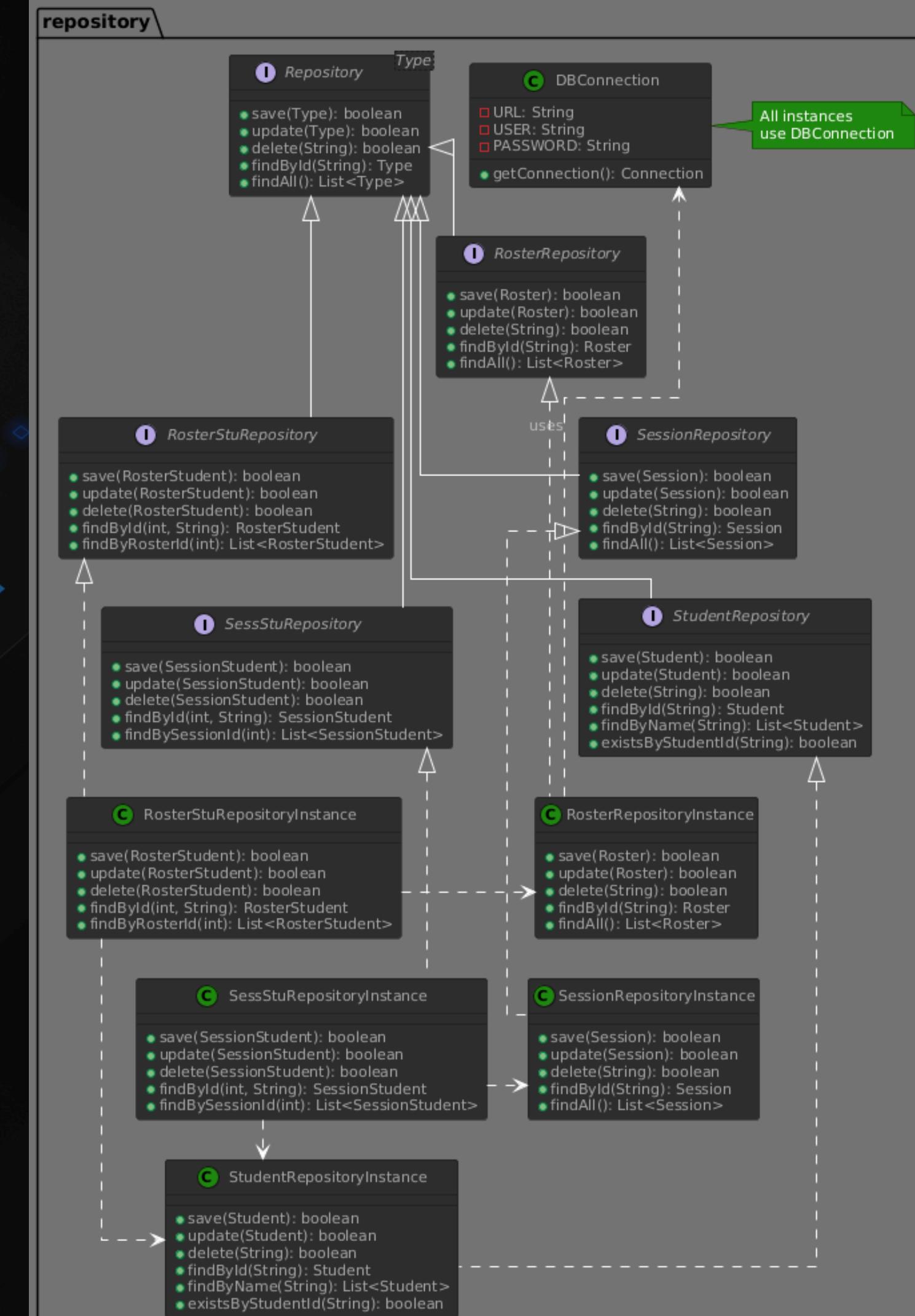
Report & Export

- ExportGenerator
 - Interface to ExcelGenerator, PDFGenerator, CSVGenerator
 - Inheritance and Polymorphism
- ReportBuilder
 - Builder class that filters fields for user to export
- Export Panel
 - Depends on ReportManager (handles logs tracking) & ReportBuilder (builds report fields)



Repository Package

- Repository interface:
 - Defines all DB CRUD operations
 - Framework for Abstraction
- Secondary repository interfaces define operations for specific classes
 - Polymorphism and Inheritance
- RepositoryInstances:
 - Handle connection to the DB
 - Abstracts database SQL calls for GUI/services to use easily



Core Recognition Features

- **Multi-Face Recognition:** Group recognition, processing all faces detected in a single frame and recognising individually
- **Multi-Threading:** Camera UI and image processing + recognition use different threads to ensure UI responsiveness
- **Recognition Smoothing:** Tracks previous frames to ensure consistent recognition decisions.
- **Adaptive Frame Skipping:** Skips frames based on the number of profiles in the dataset, to ensure smooth performance
- **Quality Validation:** Shared util class that validates all frames for brightness, contrast, blur and size to filter bad frames.

Why arcface?

- Most accurate recognition from the libraries we evaluated
 - Excellent discrimination; confidence scores between profiles are significant!
 - Able to recognise multiple poses, orientations
 - Performed well even under different lighting, accessories
 - Could effectively detect faces of different sizes
 - Easy implementation with Java OpenCV

Before arcface...

Haar Cascades

- Insufficient discrimination for our targeted recognition
- Fails to detect non-frontal faces
- Highly sensitive to lighting

OpenFace

- Embeddings generated were too similar
 - Recognition decisions were inconsistent
- Face detection still relied on Haar Cascades

dlib

- Failed to implement into Java
 - C++ library with no direct port to Java
 - Tried but failed to compile and convert to Java

Recognition Pipeline

Class	Responsibility
LiveRecognitionService	Coordinates and calls other services
RecognitionGeometry	Pads face rect to ensure full face captured
ImageProcessor	Scores and validates frame image quality
LiveRecognitionPreprocessor	Processes frames for Arcface embeddings
FaceEmbeddingGenerator	Generates Arcface embeddings for the current frame
RecognitionHistory	Refers to rolling window of embeddings for smoothing
Recognition Scorer	Compares frame to profile data to assign a score
RecognitionDatasetRepository	Loads face profile data for recognition
RecognitionDecisionEngine	Evaluates the score to recognise/reject profile

LiveRecognitionService.analyzeFace()

LiveRecognitionService

RecognitionSession

RecognitionGeometry

ImageProcessor

LiveRecognitionPreprocessor

FaceEmbeddingGenerator

RecognitionHistory

RecognitionScorer

RecognitionDatasetRepository

RecognitionDecisionEngine

Single Responsibility Principle

Each class only has one responsibility in the pipeline

Code Reuse

Embedding / util classes reused for capture and recognition to ensure embedding pipeline matches

Persistence

Fixed Data!

Database

Database for persistent data storage

Configuration Files

Configuration settings are stored in a single file, allows independent user-side configurations



DEMO!

1. Login
2. Capturing Face
3. Creating Rosters
4. Create Sessions
5. Auto Mark Attendance
6. Exporting
7. Editing Configurations

Thank You!

Glossary

Mainly for UML diagrams



