

# PhotoDitE - User's Guide

Andrei Grad

February 8, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Features . . . . .	2
<b>2</b>	<b>Design</b>	<b>2</b>
2.1	Tehnologies Used . . . . .	2
2.2	Architecture . . . . .	3
<b>3</b>	<b>Implementation</b>	<b>4</b>
3.1	Importing the Application in Eclipse . . . . .	4
3.2	JUnit Auto Tests . . . . .	5
3.3	Utility Classes for Extensibility . . . . .	5
<b>4</b>	<b>How to Use</b>	<b>6</b>
4.1	Launching the Application . . . . .	6
4.2	Editing an Image . . . . .	6
4.3	Saving and Exporting . . . . .	7
<b>5</b>	<b>Future Enhancements</b>	<b>8</b>

# 1 Introduction

PhotoDitE is a Java-based photo editing tool that provides users with basic and advanced image adjustment features. Supports popular image formats such as PNG, JPG, and JPEG. The application is designed with a user-friendly interface developed using Java Swing. The goal is to provide an intuitive and efficient experience for users who need quick and simple image modifications.

## 1.1 Features

The application provides the following features:

- Image import in different format types (PNG, JPG, JPEG);
- Save and export modified images in different formats with customizable settings;
- Image modifications such as:
  - Rotation;
  - Zoom (in and out with smooth scaling);
  - Flip (horizontal and vertical orientation);
  - Apply various filters (grayscale, sepia, negative);
  - Adjust brightness (increase/decrease intensity levels);
  - Adjust contrast (modify color differentiation and depth);
- Undo functionality to revert recent changes;
- User-friendly graphical interface with real-time preview of changes;
- Efficient memory management for handling large images;

# 2 Design

## 2.1 Tehnologies Used

- **Backend:** Java;
- **GUI Framework/Frontend:** Java Swing;
- **Image Processing:** Java AWT and BufferedImage;
- **File Handling:** Java I/O for reading and writing image files;
- **Database:** MySQL Server;

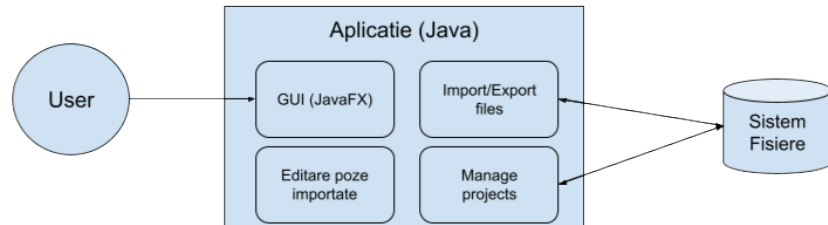


Figure 1: Application Architecture

## 2.2 Architecture

The application is structured into two main frames:

- **StartUpFrame** - Handles the initial setup and user interface, allowing users to load images and navigate to editing features.
- **MainFrame** - Manages image editing and processing functionalities, including applying filters, transformations, and adjustments.

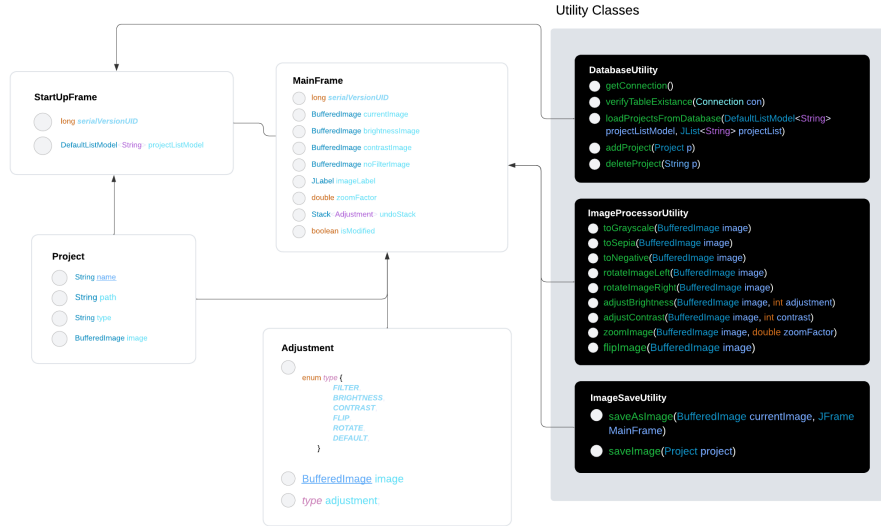


Figure 2: Application Architecture

### 3 Implementation

The application is implemented using Java and Java Swing for the graphical user interface. The core image processing functionalities rely on Java's built-in image manipulation libraries and additional third-party libraries for advanced processing.

#### 3.1 Importing the Application in Eclipse

To use the Photo Filter Application in Eclipse, follow these steps:

1. **Install Git and Eclipse:** Ensure that you have Git installed on your system. You can download it from <https://github.com/AF1DC2/PhotoDitE>. Also, make sure Eclipse IDE is installed and set up for Java development.
2. **Clone the Repository:** Open a terminal or command prompt and execute the following command to clone the project repository from GitHub:

```
git clone https://github.com/AF1DC2/PhotoDitE.git
```

3. **Open Eclipse and Import the Project:**

- Open Eclipse and go to File > Import.

- Select **Git > Projects from Git** and click **Next**.
  - Choose **Clone URI** and paste the URL of the GitHub repository.
  - Follow the instructions to copy the project to a local directory.
  - Select **Import existing Eclipse projects** and click **Next**.
  - Browse to the cloned directory, select the project, and click **Finish**.
4. **Ensure Dependencies are Configured:** Check that all required libraries and dependencies are correctly set in **Build Path** under project properties.

### 3.2 JUnit Auto Tests

The application includes automated tests using JUnit to ensure stability and reliability. The tests cover core functionalities like transformations and filter applications. To run the tests:

1. Ensure that JUnit is included in the project dependencies. If not, add it by:
  - Right-click the project in Eclipse and go to **Build Path > Configure Build Path**.
  - Under the **Libraries** tab, click **Add Library** and select **JUnit**.
  - Choose the latest version and click **Finish**.
2. Open the `src/test/java` directory in the project.
3. Locate test class.
4. Right-click a test file or the entire `test` package and select **Run As > JUnit Test**.

### 3.3 Utility Classes for Extensibility

The application is designed with modular utility classes to facilitate easy extension and additional functionalities. Some key utility classes include:

- **ImageProcessorUtility:** Provides common image processing functions such as resizing, color adjustments and filters.
- **DatabaseUtility:** Handles the connection to the database and the loading of existing projects.
- **ImageSaveUtility:** Manages file export operations, ensuring smooth handling of different image formats or saving adjustments in place.

Developers can extend the application's capabilities by creating new utility classes or enhancing existing ones. By following the structured architecture, new features can be integrated seamlessly while maintaining code clarity and maintainability.

## 4 How to Use

The Photo Filter Application provides an intuitive and user-friendly interface designed for easy image editing. Below are the steps to navigate and utilize the application effectively:

### 4.1 Launching the Application

1. Open PhotoDitE from your system.
2. Upon launch, you will see the **Startup Window**, where you can:
  - View all your existing projects.
  - Create a new project by selecting **New Project**.
3. Once a project is selected or created, you will be taken to the **Main Window**, where image editing functionalities are available.

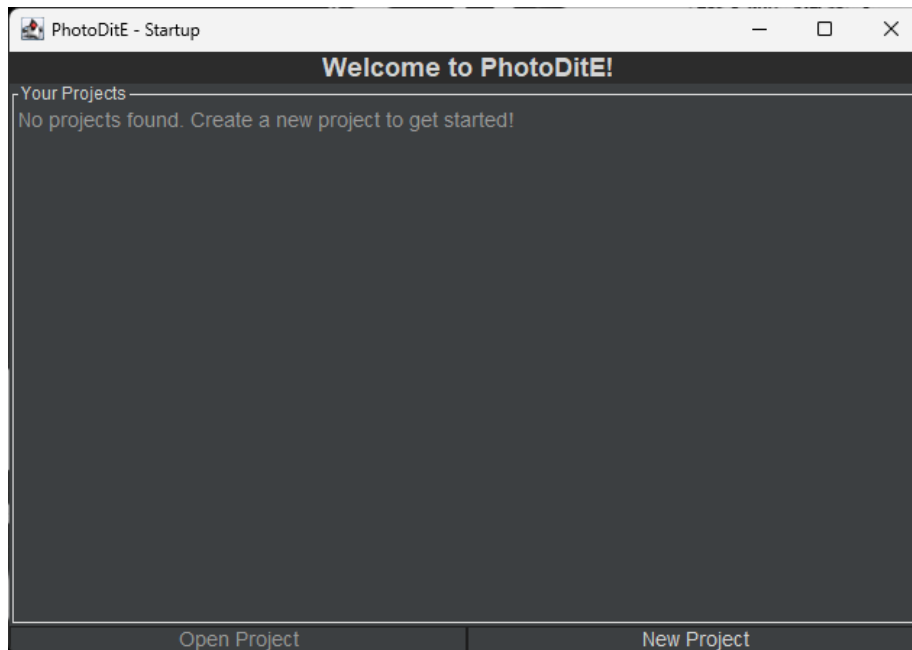


Figure 3: StartUp Window

### 4.2 Editing an Image

1. **Brightness and Contrast:** Adjust intensity levels using the provided sliders.

2. **Flip and Rotate:** Rotate the image clockwise/counterclockwise or flip it horizontally/vertically.
3. **Filters:** Apply various filters such as grayscale, sepia, and negative to modify the image appearance.
4. View the real-time preview of your changes before saving.

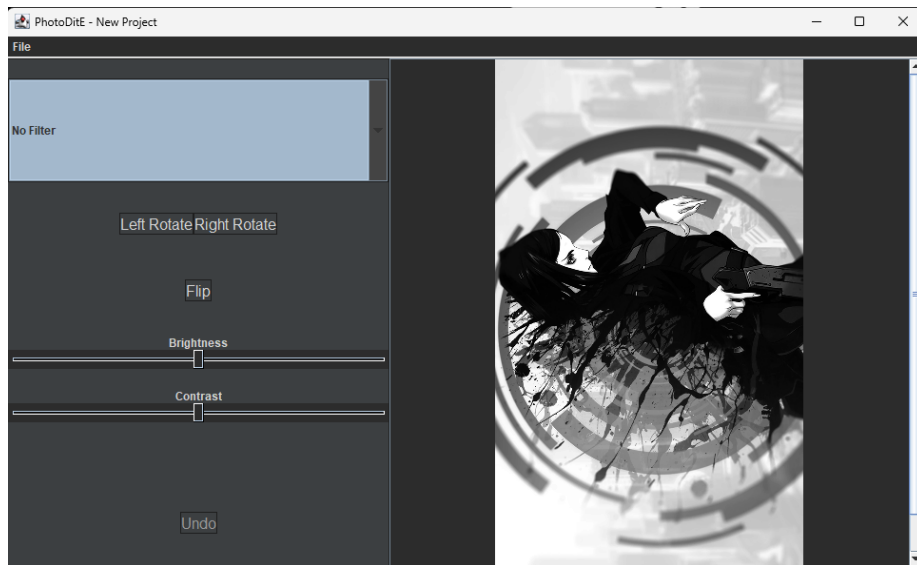


Figure 4: Main Window

### 4.3 Saving and Exporting

1. Once editing is complete, save the modified image by selecting **File > Save**.
2. To export the image in a different format, go to **File > Save As**, choose the desired format (PNG, JPG, JPEG), and configure settings as needed.

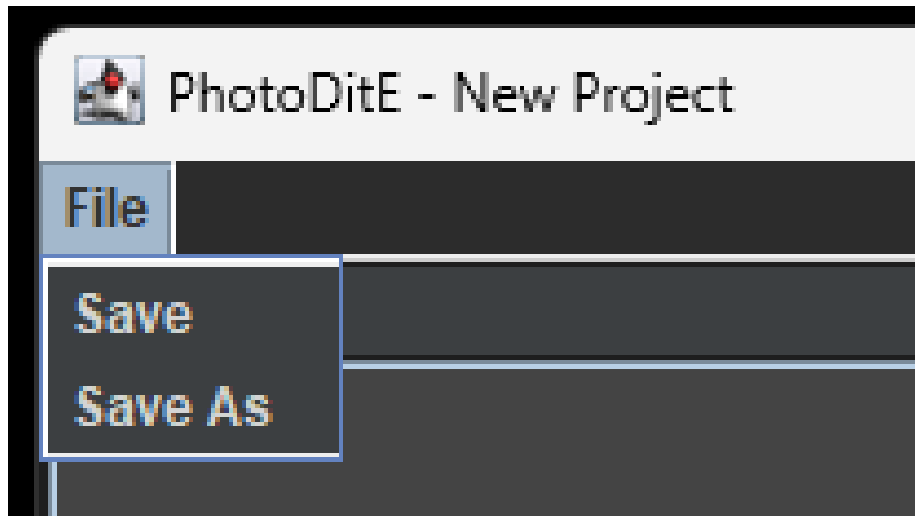


Figure 5: Saving options

## 5 Future Enhancements

The application can be further improved by adding:

- Advanced AI-based filters for automatic enhancement.
- Batch processing for editing multiple images simultaneously.
- Cloud integration for storing and retrieving images.
- Support for additional image formats (e.g., TIFF, BMP, GIF).