

Dokumentation der SWE Photography DB

Benutzerhandbuch

Beim Start der Applikation sieht der Benutzer auf der unteren Seite des Fensters bereits eine Auflistung an Bildern und darüber das ausgewählte Bild in erhöhter Auflösung. Auf der rechten Seite befinden sich zwei Tabs mit je den EXIF und IPTC Informationen des ausgewählten Bildes. Durch die Tabs kann frei gewechselt werden und im IPTC Tab befindet sich ein Knopf mit der Aufschrift „Save IPTC“, der die in den Textfeldern darüber befindlichen Informationen des Bildes speichert.

Um zur Verwaltung der Fotografen zu gelangen, wählt man den Menüpunkt „Manage Photographers“ unter „Edit“ an und es öffnet sich ein neues Fenster. Dort sieht man auf der rechten Seite eine Auflistung an Knöpfen mit den Namen der Fotografen als Beschriftung. Durch Klicken der Knöpfe wird der ausgewählte Fotograf selektiert, dessen Informationen links im Fenster zu sehen sind. Wie auch die IPTC Informationen können jene in den Textfeldern stehenden Daten beim Drücken des Knopfes „Save Info“ gespeichert werden. Alle Speicherungen synchronisieren direkt mit der Datenbank.

Zurück im Hauptfenster gibt es noch zwei weitere Menüpunkte, „Show Image Folder“ unter „File“ sowie „Visit GitHub“ unter „Help“.

Lösungsbeschreibung

Die Applikation wurde in Java geschrieben mit JavaFX als GUI-API und Gradle für IntelliJ Idea als Build-Tool. Für die Datenbank habe ich JDBC mit SQLite verwendet und zum Loggen TinyLog.

Ich bin das Problem beziehungsweise die Aufgabenstellung eher spielerisch angegangen, ich habe also weniger versucht, schon im Vorhinein eine vollständige Beschreibung der Prozesse zu verfassen, sondern mich erst einmal mit der Technologie auseinandergesetzt. Ich hätte mich sonst schwergetan, einen umfassenden Entwicklungsplan ganz am Anfang zu erstellen.

Worauf ich stolz bin

Ich bin stolz darauf, endlich eine GUI Anwendung programmiert zu haben, die Wahl der Sprache ist dabei nebensächlich. Ebenso habe ich zum ersten Mal „gescheit“ eine Datenbankbindung programmiert, da ich nicht nur mit bestehenden Tools wie SQL Workbench gearbeitet habe. SQLite und JDBC haben sich als eine ausgezeichnete Wahl für Projekte dieses Umfangs erwiesen.

Was ich das nächste Mal anders machen würde

Jetzt, wo ich viel mehr von GUIs beziehungsweise Patterns wie MVVM verstehe, werde ich nicht mehr „auf Gut Glück“ drauf los programmieren, sondern einen (hoffentlich) vollständigen Design/Entwicklungsplan entwerfen. Außerdem kam es mir während der Entwicklung so vor, als ob GUI Alternativen für C# doch ein wenig leichter zu verwenden sind.