

# Project Outline

## EN3350 - Software Design Competition

Last Modified: Aug 31, 2025

### TABLE OF CONTENTS

1. Overview	2
1.1. Background	2
1.2. Your Challenge	2
1.3. Required Features (Phase-wise)	2
1.4. What You'll Be Judged On	3
1.5. Resources Provided	3
1.6. Final Deliverables	3
2. Tech Stack	3
2.1. Technologies	3
2.2. Development Guidelines	4
3. Phase 1 – Transformer and Baseline Image Management	4
3.1. Overview	4
3.2. Scope	4
FR1.1: Admin Interface for Transformer Management	4
FR1.2: Thermal Image Upload and Tagging	4
FR1.3: Categorization by Environmental Conditions	4
Additional Technical Requirements:	5
3.3. Resources Provided	5
3.4. Deliverables	5
3.5. Evaluation Rubric	5
4. Phase 2 – Automated Anomaly Detection	6
4.1. Overview	6
4.2. Scope	6
FR2.1: AI-Based Anomaly Detection Engine	6
FR2.2: Side-by-Side Image Comparison View	6
FR2.3: Automatic Anomaly Marking	6
Additional Technical Requirements:	6
4.3. Resources Provided	7
4.4. Deliverables	7

4.5. Evaluation Criteria	7
5. Phase 3 – Interactive Annotation & Feedback	8
5.1. Overview	8
5.2. Scope	8
FR3.1: Interactive Annotation Tools	8
FR3.2: Metadata and Annotation Persistence	8
FR3.3: Feedback Integration for Model Improvement	8
Additional Technical Requirements:	9
5.3. Resources Provided	9
5.4. Deliverables	9
5.5. Evaluation Criteria	10
6. Phase 4 – Maintenance Record Sheet Generation	10
6.1. Overview	10
6.2. Scope	10
FR4.1: Generate Maintenance Record Form	10
FR4.2: Editable Engineer Input Fields	10
FR4.3: Save and Retrieve Completed Records	11
Additional Technical Requirements:	11
6.3. Resources Provided	11
6.4. Deliverables	11
6.5. Evaluation Criteria	12

# 1. Overview

## 1.1. Background

Power utilities perform routine thermal inspections of distribution transformers to detect anomalies like overheating, insulation degradation, and load imbalances. Currently, these inspections rely on manual comparison of thermal images, making them time-consuming, subjective, and error-prone. There is a strong need to automate and digitize this workflow while ensuring traceability, efficiency, and adaptability.

## 1.2. Your Challenge

Design and develop a complete software solution that allows users to:

1. Record and manage transformers and their associated image data
2. Automatically detect temperature anomalies in new images using computer vision
3. Allow users to manually validate or correct detected anomalies
4. Generate a maintenance record sheet with marked anomalies and editable fields

This project will be executed in **four phases**, and your solution must support the flow of data and insights across all of them.

## 1.3. Required Features (Phase-wise)

Phase 1 – Transformer and Baseline Image Management

- FR1.1: Build an admin interface to add transformer records (ID, location, capacity).
- FR1.2: Enable uploading of thermal images (baseline and maintenance) tagged to transformers.
- FR1.3: Categorize baseline images by environmental conditions (sunny, cloudy, rainy).

Phase 2 – Automated Anomaly Detection

- FR2.1: Implement an AI-based comparison engine to detect anomalies in new images.
- FR2.2: Display side-by-side comparison of new and baseline images.
- FR2.3: Automatically highlight potential hotspots and deviations based on thresholds.

Phase 3 – Interactive Annotation & Feedback

- FR3.1: Allow engineers to manually adjust detected anomalies or annotate new ones.
- FR3.2: Save annotated images with metadata (user, timestamp).
- FR3.3: Feed annotations back into the model pipeline for improved accuracy (retraining or correction tagging).

Phase 4 – Maintenance Record Sheet Generation

- FR4.1: Generate a transformer-specific digital maintenance form with the thermal diagram pre-marked.

- FR4.2: Allow manual input of other required fields.
- FR4.3: Save completed records in the system under the relevant transformer.

## 1.4. What You'll Be Judged On

This is not just about completing the system—it's also a competition. You'll be assessed on:

Area	Criteria
Functionality	Completion of all required features
Scalability	Modular and well-structured architecture
Efficiency	Speed of image uploads, model inference, and overall responsiveness
ML Integration	Accuracy and robustness of anomaly detection (fine-tuning or integration)
Creativity	Innovative annotation tools or workflow improvements
Quality	Test coverage and coding best practices

## 1.5. Resources Provided

- Sample dataset of thermal images and baseline records
- Sample diagrams and formats of handwritten maintenance sheets.

## 1.6. Final Deliverables

- Working web-based system with all four phases integrated
- Test coverage report
- Deployment instructions and README
- GitHub repository with source code

# 2. Tech Stack

## 2.1. Technologies

- Maintenance recorded keeper
  - Front-end: React
    - Utilize React for the front end. It provides a responsive and interactive user interface for the questionnaire application.
  - Back-end: Java with Spring framework
    - Java, with the Spring framework, offers a scalable back-end solution.

- Spring facilitates the development of RESTful APIs for communication between the front-end and back-end components of the questionnaire application.
  - Database interactions, user authentication, and API endpoints can be efficiently managed using Java with Spring.
- Target platform: Web browser
  - The tool will target web browsers as the primary platform.

## 2.2. Development Guidelines

- Development: Use a GitHub public repository to maintain the source code.
- Follow software engineering best practices and design principles in the development work.

# 3. Phase 1 – Transformer and Baseline Image Management

## 3.1. Overview

In this phase, your team will lay the foundation of the system by implementing essential functionalities to manage transformer records and their associated baseline thermal images. This includes setting up the core data models, admin interfaces for data entry, and image uploading mechanisms. The primary goal is to create a structured and searchable repository of transformers and their thermal imaging data under various environmental conditions.

## 3.2. Scope

You are required to implement the following functional requirements for Phase 1:

### FR1.1: Admin Interface for Transformer Management

- Add new transformer records
- View and edit existing transformer records
- Delete transformer entries if necessary

### FR1.2: Thermal Image Upload and Tagging

- Allow uploading of thermal images to specific transformer entries
- Images must be tagged as either:
  - **Baseline:** Used as the reference image for future comparisons
  - **Maintenance:** New images used for periodic inspections
- Each image must be associated with metadata such as:
  - Upload date/time
  - Image type (Baseline / Maintenance)
  - Uploader (admin user ID or name)

### FR1.3: Categorization by Environmental Conditions

- While uploading a baseline image, the user must tag it with the observed environmental condition:
  - Sunny
  - Cloudy
  - Rainy
- Environmental conditions should be selectable via a dropdown during image upload.

#### Additional Technical Requirements:

- Image storage should support efficient retrieval and viewing
- Transformer and image metadata should be stored in a relational database
- Admin interface should be accessible via a web browser
- Follow a modular architecture that supports easy extension in future phases

### 3.3. Resources Provided

The following resources will be provided to assist with implementation:

- **User Interface Designs:** Figma/PNG/HTML mockups for all required Phase 1 screens (admin panel, image upload, etc.): [Link](#)

### 3.4. Deliverables

- A working web-based system with the following:
  - Admin dashboard for managing transformers
  - Image upload interface with appropriate tagging options
  - Database schema and seed data for initial testing
- Clean and readable source code hosted in a **GitHub public repository**
- A short **demo video (2–3 minutes)** showing the working features of Phase 1
- A **README.md** file with:
  - Setup instructions
  - List of implemented features
  - Any known limitations or issues
- Test data (minimum of 5 transformers with baseline images) included in the repository
- All material must be added in a single zip file with a logical subfolder structure and uploaded to Moodle.

### 3.5. Evaluation Rubric

Criterion	Weight
Completeness of all required features (FR1.1–FR1.3)	35%
Clarity and intuitiveness of the admin interface	15%
Clean structure, modularity, and adherence to best practices	15%
Proper tagging, storage, and retrieval of images and metadata	15%

Clear README and usage instructions	10%
Any additional thoughtful features	10% (bonus)*

\*Bonus marks can push the total above 100 for exceptionally creative or technically impressive additions, but the core score will be capped at 100 for the final grade.

## 4. Phase 2 – Automated Anomaly Detection

### 4.1. Overview

In Phase 2, you will build and integrate an AI-based engine to automatically detect thermal anomalies in transformer images. This is a critical capability in digitizing and scaling thermal inspection workflows. Your task is to compare new (maintenance) thermal images with previously stored baseline images, identify deviations such as hotspots, and present them to the user in a clear and intuitive interface. The system must support comparison logic, anomaly detection based on pre-defined thresholds, and a visual display that helps users understand the insights derived by the AI model.

### 4.2. Scope

You are required to implement the following functional requirements for Phase 2:

#### FR2.1: AI-Based Anomaly Detection Engine

- Design or integrate a computer vision model (classical or DL-based) to:
  - Compare new maintenance images with baseline images of the same transformer
  - Detect thermal anomalies such as temperature spikes, asymmetries, or changes in hotspot locations.
- Define a thresholding mechanism (fixed or adaptive) to flag anomalies

#### FR2.2: Side-by-Side Image Comparison View

- Display new (maintenance) and baseline images side by side
- Include basic controls such as zoom, move (click and drag), and reset
- Highlight anomaly regions visually (e.g., bounding boxes, overlays, heatmaps)

#### FR2.3: Automatic Anomaly Marking

- When anomalies are detected:
  - Automatically annotate the image with color-coded overlays or markers
  - Display metadata such as pixel coordinates, anomaly size, and severity score
- Include a confidence score or flag where the model is unsure

#### Additional Technical Requirements:

- Ensure model inference time is optimized for responsive performance

- Support modular integration so that the anomaly detection engine can evolve over time
- Record all detection outputs and metadata for later retrieval or feedback (for Phase 3)

### 4.3. Resources Provided

The following resources will be provided to assist with Phase 2 implementation:

- **Sample Image Pairs:** Baseline and maintenance thermal image pairs for selected transformers. [Link](#)
- **Extended UI Designs:** Layout for side-by-side comparison with interactive overlays.
- **Ruleset and Threshold Guidelines:** Suggested anomaly detection rules (e.g., temperature deviation > 10%) and thresholds (where applicable). [Link](#)

### 4.4. Deliverables

- A working web-based system with:
  - Functional AI-based comparison engine
  - Frontend interface showing side-by-side image comparisons with highlighted anomalies
- Clean and modular code hosted in the **GitHub public repository**
- A **demo video (max 10 minutes)** clearly explaining:
  - The detection logic, the user interface and example outputs
  - More info about the video:
    - You must keep your video on throughout the demo.
    - The video must be done by another member of the group who did not do the first milestone video.
- A **README.md** file that includes:
  - An overview of your detection approach
  - Setup and run instructions
  - Dependencies and known limitations
- Logs or data outputs showing detection metadata for at least **5 image pairs**
- All material must be added in a single zip file with a logical subfolder structure and uploaded to Moodle.

### 4.5. Evaluation Criteria

Criterion	Weight (out of 100)
Functionality	30 marks
Accuracy & Robustness	40 marks
Code Quality and Modularity	20 marks
Documentation	10 marks

\*Teams are free to use classical computer vision techniques or integrate a deep learning model, as long as detection outputs are meaningful and clearly visualized.



## 5. Phase 3 – Interactive Annotation & Feedback

### 5.1. Overview

In Phase 2, you developed a system that automatically detects anomalies in transformer thermal images. In Phase 3, you will extend this functionality by enabling **human-in-the-loop feedback** through interactive annotation tools. This allows engineers or admin users to **validate, correct, or reject** detected anomalies and **add their own annotations** where necessary. This phase is essential to enhance the system's accuracy and usability in real-world operations, as manual oversight plays a key role in critical maintenance workflows. Additionally, user corrections and annotations must be stored for potential model retraining or analysis in later phases.

### 5.2. Scope

You are required to implement the following **functional requirements**:

#### FR3.1: Interactive Annotation Tools

- On the anomaly detection view (from Phase 2), enable users to:
  - **Adjust** existing anomaly markers (resize, reposition)
  - **Delete** incorrectly detected anomalies
  - **Add** new anomaly markers by drawing bounding boxes or polygonal regions
- All annotations should include:
  - Annotation type (e.g., added / edited / deleted)
  - Optional comments or notes
  - Timestamp and user ID

#### FR3.2: Metadata and Annotation Persistence

- When a user interacts with the anomaly detection view, all changes must be:
  - Captured and saved in the backend
  - Stored along with metadata (user ID, timestamp, image ID, transformer ID, action taken)
  - Shown in the UI
- Existing annotations should be automatically reloaded when the same image is revisited.

#### FR3.3: Feedback Integration for Model Improvement

- Maintain a **feedback log** that includes:
  - Original AI-generated detections
  - Final user-modified annotations
- This log will serve as training or validation data for improving the detection model
- Use the user-modified annotations to improve the accuracy of the AI model
- The feedback log must be exportable in JSON or CSV format with:
  - Image ID

- Model-predicted anomalies
- Final accepted annotations
- Annotator metadata
- Saving previous versions of the annotated image in the user interface and switching between them is not within the scope of this project.

#### Additional Technical Requirements:

- Annotation tools must be intuitive and user-friendly (use standard UI components where possible)
- All annotation actions must be logged without requiring the user to manually "save"
- Store image-to-annotation relationships in a structured and queryable format (relational DB or NoSQL)

### 5.3. Resources Provided

The following will be provided to assist in Phase 3 implementation:

- **Extended UI Designs:** Annotated mockups showing the expected interaction flow (edit/delete/add annotations) - refer to the Phase 3 explanation video
- **Demo Data:** Sample image sets with expected ground-truth annotations for validation. [Link](#)

### 5.4. Deliverables

Your Phase 3 submission must include:

- A fully working interface to interact with anomalies detected in Phase 2
  - Users should be able to view, modify, delete, and add annotations
- Backend support for:
  - Annotation storage and retrieval
  - User metadata logging
  - Export of annotation logs
- A **README.md** file with:
  - Description of the annotation system
  - Backend structure used to persist annotations
  - Known bugs or limitations
- A **demo video (max 10 minutes)** showcasing:
  - How annotations are added/modified
  - How metadata is captured and logged
  - How annotation logs can be exported
  - More info about the video:
    - You must keep your video on throughout the demo.
    - The demo video must be presented by a group member who has not presented in either of the previous milestone videos.
- Final submission must be a **single ZIP file**, structured clearly and uploaded to Moodle

### 5.5. Evaluation Criteria

Criterion	Weight (out of 100)
Annotation Functionality	40 marks
Backend Integration	25 marks
Usability & UI Design	15 marks
Data Logging & Export	10 marks
Documentation	10 marks

\*This phase simulates real-world use cases. Usability and data traceability are as important as functionality.

## 6. Phase 4 – Maintenance Record Sheet Generation

### 6.1. Overview

In the final phase of this project, your team will implement the functionality to generate transformer-specific digital maintenance records based on thermal inspection results. These maintenance records must include anomaly annotations (auto-detected and/or user-validated from previous phases), editable fields for additional engineer input, and a method to store and retrieve finalized reports.

The goal is to ensure that every inspected transformer has a structured, traceable, and digitally accessible maintenance record, reducing the reliance on handwritten documentation and streamlining future audits or interventions.

### 6.2. Scope

You are required to implement the following **functional requirements**:

#### FR4.1: Generate Maintenance Record Form

- For each transformer with a new maintenance image and corresponding analysis:
  - Generate a maintenance record form that includes:
    - Transformer metadata (ID, location, capacity)
    - Timestamp of inspection
    - Thumbnail or embedded thermal image with anomaly markers (from Phase 3)
    - List of detected/annotated anomalies with metadata (type, location, details)

#### FR4.2: Editable Engineer Input Fields

- Allow authorized users (e.g., maintenance engineers) to:
  - Add notes, comments, and corrective actions
- Input fields such as:
  - Inspector name

- Status of transformer (OK / Needs Maintenance / Urgent Attention)
- Electrical readings (voltage, current, etc.)
- Recommended action
- Additional remarks
- Support text inputs, dropdowns, and date pickers as appropriate
- Editable fields should be clearly separated from system-generated content

#### FR4.3: Save and Retrieve Completed Records

- Once filled, allow saving the maintenance record to the database
- Each record should be:
  - Associated with a specific transformer and inspection timestamp
  - Stored in a format that allows easy retrieval, filtering, and future exports
- Implement a simple record history viewer:
  - View all past maintenance records for a given transformer

#### Additional Technical Requirements:

- Ensure the UI is clean and printable (PDF-ready design)
- Backend must support versioning or timestamping of saved forms for traceability

### 6.3. Resources Provided

The following resources will be provided to support implementation:

- **Form Layout Designs:** Wireframes/UI designs showing the expected structure and layout of the digital maintenance sheet

### 6.4. Deliverables

Your Phase 4 submission must include:

- Functional web interface to:
  - Generate a pre-filled digital maintenance form from a processed image
  - Input additional engineer notes and status
  - Save and retrieve records linked to a specific transformer
- Backend implementation to:
  - Store and query completed records
  - Support retrieval of historical records per transformer
- A **README.md** file with:
  - Description of how the form generation and saving mechanism works
  - Database schema for record storage
  - Setup and usage instructions
- A **demo video (max 10 minutes)** showcasing:
  - How a form is generated after image analysis

- How engineers fill and save the form
- How historical records are viewed
- More info about the video:
  - You must keep your video on throughout the demo.
- Final submission must be a **single ZIP file**, structured clearly, and uploaded to Moodle

## 6.5. Evaluation Criteria

Criterion	Weight (out of 100)
Maintenance Form Generation	30 marks
Editable Fields & UI	20 marks
Backend Integration	20 marks
Record Retrieval & Export	20 marks
Documentation	10 marks

Note: Focus on creating a robust, traceable digital record that accurately reflects the inspection outcome. Think about what a field engineer or utility company would need to use this in practice.