1. Objectives

- Refine cropping functionality to ensure accurate coordinate mapping and alignment with displayed image data.
- Implement a 1:1 image display mode to show images at their original size without compression.
- Push the project to GitHub, document the functionality and current issues, and notify the team lead.
- Conclude internship responsibilities and finalize pending tasks.

---

2. Activities

- **Refining Cropping Functionality:**
  - **Problem:**
    - Cropping used inconsistent coordinate mapping, leading to misaligned and unexpected outputs.
  - **Solutions:**
    - Calculated scaling factors (scaleX, scaleY) based on window and image dimensions for accurate mapping.
    - Updated cropToSelection to use viewToImageCoordinates from GraphicsView for precise transformation.
    - Clamped coordinates with std::clamp to ensure cropping remained within image boundaries.
    - Added debugging prints to validate coordinates during cropping.
    - Simplified logic by directly using RectItem coordinates mapped to image space.
  - **Outcome:**
    - Cropping now aligns with the displayed image in the window view, though slight offsets from floating-point rounding may persist.

- **Implementing 1:1 Image Display Mode:**
    - **Problem:**
        - Images were compressed to fit the screen, losing original size integrity.
    - **Solutions:**
        - Updated TextureItem::updateGeometry to map image pixels directly to screen pixels.
        - Ensured identity transforms in GraphicsView::updateViewMatrix for 1:1 mapping.
        - Removed scaling logic, applying only translations to center the image.
        - Corrected errors by:
            - Including <algorithm> for std::clamp.
            - Fixing typos and missing function definitions.
            - Adjusting matrix logic in updateDisplayImage for pixel-perfect centering.
        - Debugged matrix transformations and explored alternative scaling and viewport adjustments.
    - **Outcome:**
        - Improved 1:1 display logic, though persistent compression in edge cases suggests hidden scaling operations requiring further refinement.
- **Project Finalization and Documentation:**
    - Pushed the project to GitHub, ensuring all functionalities, libraries (ImGui, Boost.Signals2, SDL), and known issues were documented in README.md.
    - Highlighted faulty functionalities, such as edge cases in cropping and scaling, to notify users.
- **Team Lead Update:**
    - Notified the team lead about the GitHub push and current project issues.

- Shared progress on final fixes and the status of unresolved edge cases.

---

3. Achievements

- Refined cropping functionality, ensuring alignment with the displayed image and maintaining user-selected regions.
- Improved 1:1 image display mode to minimize compression and support pixel-perfect rendering.
- Documented project functionalities and issues comprehensively in GitHub README.md.
- Successfully concluded internship responsibilities with a final status update to the team lead.

---

4. Problems & Solutions

1. **Problem:** Cropping functionality produced misaligned results.
   - **Solution:** Integrated scaling factors and view transformation utilities to map coordinates accurately to image space.
2. **Problem:** 1:1 image display mode still exhibited compression in edge cases.
   - **Solution:** Debugged matrix transformations and removed scaling logic, though further refinement is needed to address hidden scaling operations.
3. **Problem:** Debugging output visibility was limited in Visual Studio.
   - **Solution:** Added additional debugging outputs but noted the need for a more robust debugging strategy.
4. **Problem:** Documentation required updates to reflect project status and issues.
   - **Solution:** Wrote a detailed README.md explaining project functions, used libraries, and known issues.