Decision Logs Report

Inter IIT Tech Meet 14.0

"Re-imagining Photoshop - The AI Editor of 2030"

Team 94

1 Executive Summary

This report documents the major decisions taken during the development of our solution for the Adobe Problem Statement at Inter IIT Tech Meet 14.0. Rather than presenting standalone decisions, this document explains why each decision emerged, how constraints shaped our choices, and what impact each decision had on the final direction of the product. The project evolved under considerable time pressure and compute limitations. These constraints guided many of the pivots and tradeoffs described in the following sections.

2 Approach to Logging Decisions

Decision logs were compiled from chronological meeting notes, team updates, architecture documentation, and retrospective analysis. The resulting report is narrative in style so that the reasoning behind each decision becomes clear, and so that the reader understands how technical, organizational, and design choices converged into a coherent product.

3 Chronological Development of Key Decisions

3.1 Meeting 1, 11 November

The earliest meeting focused on establishing discipline and clarity in team operations. The team recognized that the workload would be substantial and that disorganization would become a bottleneck. As a result, documentation was made compulsory, and team members committed to recording progress, decisions, and micro deadlines. Work was divided into balanced subteams so that design, application development, and AI could evolve simultaneously. Tools such as Perplexity and Notion were introduced to structure research and planning, and subscriptions were considered to improve workflow efficiency.

3.2 Meeting 2, 11 November (later)

Early technical experiments revealed model instability, dependency conflicts, and storage issues. This led to the request for Google Colab Pro. The team initially planned to create a single workspace called Lab Space, which contained standalone models such as inpainting and deblurring. This plan later evolved as user experience considerations deepened.

3.3 Meeting 3, 13 November

Integration concerns were raised, and the team discussed embedding Python scripts directly to reduce backend latency. A single script build system for the APK was proposed to ease testing. Daily progress updates were reinforced to maintain visibility across subteams.

3.4 Meeting 4, 14 November

Advanced subscriptions such as Claude Max and Colab Pro Plus were procured to support development. The design team introduced a strong minimalistic direction informed by modern UI trends and cognitive load considerations. Long work sessions were planned to accelerate progress, and empathy in collaboration was emphasized to maintain morale during intensive work periods.

3.5 Meeting 5, 17 November

A deadline was established for moving from research into implementation. This marked the transition from exploratory ideas to concrete execution.

3.6 Meeting 6, 18 November

A monorepo was created to centralize development. Legal and documentation requirements were drafted, and the team explored HPC access. The lack of HPC resources later informed model selection and integration choices.

3.7 Meeting 7, 19 November

This meeting served as a simple progress review.

3.8 Meeting 8, 20 November

The team learned that the Adobe Problem Statement had the earliest submission deadline. This accelerated convergence toward a smaller but stronger set of final features. A workathon schedule was introduced to accommodate the shortened timeline.

3.9 Meeting 9, 29 November

The team clarified expectations for the APK, repository structure, and demo video. This ensured that post exam work would continue smoothly.

3.10 Post November 30

After exams, the team continued through intensive workathons. No new conceptual decisions were introduced during this phase.

4 System Level Design and Product Decisions

4.1 The Transition to Dual Workspaces

User experience requirements and editing patterns motivated the transition from the original single workspace design to a dual workspace system. Flow Space was created for guided, beginner friendly pipelines, while Lab Space preserved flexibility for power users. This separation reduced cognitive load and improved clarity. It also allowed pipeline workflows to reuse underlying model components without overwhelming the user.

4.2 AI Workflow Finalization

Two core workflows were selected. Background replacement offered high creative value, while auto enhancement provided low light correction, denoising, deblurring, and facial restoration. These workflows aligned with common user needs and could be implemented reliably within compute constraints. More experimental ideas were discarded due to instability or lack of usable pretrained weights.

4.3 Multimodal Input

The team added support for prompts, dictation, and button interactions. This improved accessibility and prepared the app for conversational editing patterns that are expected to become more common by 2030.

4.4 Layer System and Foreground Background Separation

Layer based editing improved control and reversibility. Automatic foreground background separation was crucial for background replacement, and history states provided traceability and transparency in editing.

5 Constraints and Their Impact

5.1 Compute Constraints

The Problem Statement's vision of a lightweight editor influenced many technical choices. Heavy models were moved to backend Docker containers to avoid increasing the app's size. Hybrid inference reduced device side load but introduced latency. These constraints made diffusion based methods impractical and guided the selection of stable, efficient models.

5.2 Time Constraints

End semester examinations sharply reduced development time. This required the team to narrow the feature list and focus on high reliability workflows that could be completed on schedule.

5.3 Lack of HPC Access

Without HPC support, the team could not explore research heavy models extensively. This limitation encouraged the use of models with available implementations and mature codebases.

6 Subteam Decision Narratives

6.1 Design Team

The design team adopted a minimalistic, future proof style. Navigation paths were optimized to ensure that any screen was reachable within four to five clicks. Both light and dark modes were included for user comfort. The team avoided visual clutter and focused on clarity and accessibility.

6.2 App Development Team

The app development team adopted a RESTful API first approach, React Native for cross platform functionality, and Docker based AI services to ensure consistent behavior across environments. A monorepo simplified coordination between frontend and backend.

6.3 AI Team

The AI team selected PCT Net for its reliable implementation. U2Net and U2Net Human were chosen for segmentation due to their clean mattes. SAM provided superior

segmentation quality compared to MobileSAM, and LaMa was selected for object removal. Quantization resulted in degraded output, so full precision models were retained.

7 Pivots and Dropped Ideas

Several ideas were dropped when they became infeasible or inconsistent with the product's goals. These include natural relighting, diffusion based editing, and lightweight segmentation methods that produced poor results. The final system reflects a balance of feasibility, quality, and user experience.

8 Trade Offs

The dual workspace increased design workload but improved clarity. Docker containers increased inference time but reduced app size and dependency issues. Full precision models consumed more space but preserved quality. Hybrid inference reduced device load but introduced latency.

9 Final Outcomes

Flow Space and Lab Space together serve both beginners and professionals. Multimodal inputs increase accessibility. Backend model processing ensures that even mid range devices can use advanced tools. The resulting editor is practical today and aligned with creative expectations for 2030.