

# MemoryRevive – Project Overview & Functional Specification (MVP)

## 1. What is MemoryRevive?

**MemoryRevive** is a mobile and web application focused on **restoring, colorizing, and lightly animating old photographs using AI**, with a strong emphasis on **emotional value, realism, and respect for personal and family memories**.

The goal of the app is not to apply visual effects or filters, but to **revive historical and family photos in a believable and human way**, preserving facial identity, age, and the original context of the image.

The primary target audience is **people aged 55–70**, so the product must be:

- Extremely simple to use
  - Emotionally driven
  - Technically robust but visually calm and reassuring
- 

## 2. Product Philosophy

- **Realism over spectacle**  
No cartoonish, exaggerated, or “AI-looking” results.
  - **Facial identity preservation is critical**  
The person in the photo must still look like themselves.
  - **Emotional respect**  
Every output should feel respectful and authentic.
  - **Simple UX, powerful backend**  
Users should never deal with technical complexity.
  - **Future scalability**  
MVP is only the first step; the architecture must allow growth.
- 

## 3. MVP Scope – Core Functionalities

### 3.1 Photo Restoration & Colorization

**(1 single core functionality in the MVP)**

This is the primary function of MemoryRevive at launch.

## What it does:

Given an old photograph (black & white, faded, damaged, or low quality), the system:

- Removes common defects:
  - Scratches
  - Grain
  - Stains
  - Mild blur
- Enhances sharpness carefully (no over-processing)
- Reconstructs damaged areas coherently
- Applies **realistic colorization**, respecting:
  - Natural skin tones
  - Clothing appropriate to the era
  - Original lighting
  - Paper and texture characteristics

## Expected result:

- A restored, colorized image
  - High visual quality
  - No “AI artifact” look
  - Ready for download or animation
- 

## 3.2 Photo Animation

(1 animation engine with 2 modes in the MVP)

Animation in MemoryRevive is **subtle, emotional, and controlled**.

It is not full video generation, but **micro-movements that bring the photo to life**.

---

### Animation Mode 1: Subtle Animation

Includes:

- Natural blinking
- Very slight head movement
- Micro facial expressions
- Calm, neutral presence

### Rules:

- No facial deformation

- No exaggeration
  - Maximum identity consistency
- 

### **Animation Mode 2: Light Expressive Animation**

Includes:

- Very soft smile (optional)
- Livelier gaze
- Gentle expressive gestures (e.g. light greeting)

#### **Important:**

Not all photos should smile. Expression must remain subtle and respectful.

---

## **4. Home Screen Clarification (CRITICAL)**

The home screen visually displays **many options** such as:

### **“Restore & Color”**

- Couple
- Portrait Plus
- Family
- Family Plus
- Sharpness
- Contrast
- Fix Blacks
- Cinematic
- Fix Color
- Sepia Tones

### **“Bring Memories to Life”**

- Subtle (Portrait)
- Greeting (Portrait)
- Couple
- Greeting Couple
- Group
- Greeting Group
- Kiss
- Passion Kiss

**⚠ VERY IMPORTANT:**

These are **NOT** separate technical features in the MVP.

---

## 5. How the Developer Should Interpret the Home

### Restoration & Colorization

- All restoration cards call **the same restoration + colorization engine**
- Cards exist for:
  - Emotional clarity
  - User identification (“this looks like my photo”)
  - Future preset expansion

In the MVP:

- No separate models are required
- No different pipelines are required
- All cards can trigger the same backend process

Example (simplified):

Engine: `restore_color_v1`

Input:

- Image
- Optional context label (portrait / couple / family)

Output:

- Restored & colorized image

The context label can:

- Be ignored in MVP
  - Or stored for future logic
- 

### Animation

- All animation cards map to **one animation engine**
- Only **two real modes exist in MVP**:
  - subtle
  - expressive

Example (simplified):

Engine: `animate_photo_v1`

Mode:

- subtle
- expressive

Input:

- Image
- Mode

Output:

- Short animated clip
- 

## 6. What the Developer SHOULD NOT Do (MVP)

- ✗ Do not create multiple restoration engines
- ✗ Do not build separate animation pipelines per card
- ✗ Do not over-engineer logic based on UI naming
- ✗ Do not treat home cards as independent features

This complexity is intentionally postponed to later phases.

---

## 7. What the Developer SHOULD Do

- ✓ Build a strong, reusable core
  - ✓ Keep the logic centralized
  - ✓ Treat home options as UI/UX layers
  - ✓ Prepare the system for future presets and expansions
- 

## 8. User Flow (Simplified)

1. User uploads a photo
  2. User selects an action:
    - Restore & Color
    - Animate (Subtle or Expressive)
  3. AI processing
  4. Before / After comparison
  5. Result:
    - Download image or animation
    - Save to user gallery
- 

## 9. Quality & Technical Priorities

- **Facial consistency is top priority**
  - No beautification or rejuvenation
  - No identity alteration
  - Automatic retry or fallback if result quality is poor
  - Outputs must feel human, not synthetic
- 

## 10. Scalability & Future Expansion (Not MVP)

The architecture should allow future features such as:

- Additional animation styles
  - Historical video generation
  - AI voice narration
  - Family albums & timelines
  - Face recognition for grouping photos
  - Automatic historical context detection
- 

## 11. Final Summary for the Developer

- MVP includes:
    - 1 restoration & colorization engine
    - 1 animation engine with 2 modes
  - Home screen shows examples and categories, not real features
  - UX is emotional and simple
  - Backend must be clean, scalable, and respectful
  - The product is about **memories, not effects**
- 

If you want, next I can:

- Turn this into a **technical task breakdown**
- Create a **developer checklist**
- Or prepare a **system architecture diagram (text-based)**