

# Product Requirements Document

*Enhancing Engagement for Young Users (18-24) on the Aperture App*

## 1. PROBLEM ALIGNMENT

### 1.1 The Problem

Young users (18–24) are disengaging from Aperture due to outdated filter functionality and poor usability. This decline threatens the app's relevance as competitors offer more intuitive experiences.

**Key Insight:** Frustration with filter discoverability, usability, and navigation is driving users to alternative platforms.

---

## 2. AUDIENCE

**Young Users (18–24):** Tech-savvy, active on social media, seeking dynamic, intuitive tools.

**Frequent Social Media Users:** Look for easy-to-use, visually appealing photo editing tools.

**Casual Photo Editors:** Prioritize simplicity and frustration-free editing.

---

## 3. GOALS AND SUCCESS METRICS

3.1. Product Goals	3.2. Success Metrics
Improve ease of use for filter-related tasks. Recapture engagement among the 18–24 demographic. Enhance user retention and increase session duration by reducing frustrations with saving edits	<b>Task Success Rate:</b> 70% of users save edits successfully without errors within 90 days. <b>Adoption Rate:</b> 20% adoption of Save functionality within the first 90 days. <b>Satisfaction:</b> Average rating of 4.0/5 for the Save functionality. <b>Retention &amp; Engagement:</b> 15% growth in return visits from 18–24 users in 3 months. <b>Save Button Usage:</b> >80% engagement with the Save button. <b>Save Prompt Interaction:</b> >80% interaction with the confirmation prompt.

---

## 4. SOLUTION ALIGNMENT

### 4.1. Key Features

- **Save Button:** Easily accessible to save edited photos.
- **Save Confirmation Prompt:** Appears when users attempt to exit without saving.
  - **Yes:** Saves and exits.
  - **No:** Discards changes and exits.

---

## 5. KEY FLOWS AND PROTOTYPES

Flow 1: Save workflow	Flow 2: Exit without Saving Workflow
-----------------------	--------------------------------------

<p>User applies a filter → Taps "Save" → Prompt: "Your changes have been saved."</p> <p>User exits or continues editing - <a href="#">User flow 1 and 2</a></p>	<p>The user applies a filter → Attempts to exit without saving → Prompt: "Do you wish to save changes?"</p> <ul style="list-style-type: none"> <li>• <b>Yes:</b> Saves and exits.</li> <li>• <b>No:</b> Discards and exits</li> </ul>
---	---

## 6. ASSUMPTIONS AND DEPENDENCIES

6.1. Assumptions	6.2 Dependencies
<ul style="list-style-type: none"> <li>• Users will adopt the Save functionality to solve core pain points.</li> <li>• The prompt will effectively reduce exits without saving.</li> <li>• Users trust the Save functionality, fostering repeat usage.</li> <li>• Users are familiar with similar features, easing adoption</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Engineering bandwidth for UI revamp.</i></li> <li>• <i>Availability of analytics data to train recommendations.</i></li> </ul>

## 7. LAUNCH PLAN

7.1. Key Milestones	7.3 Post-Launch Plan	7.3 Task Duration
<ul style="list-style-type: none"> <li>• <b>MVP Beta Launch:</b> (January 1- 2025) Select group for initial feedback. (3 weeks).</li> <li>• <b>Full Launch:</b> 1-week post-beta with refinements.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Gather User Feedback:</b> Refine features based on real-world usage.</li> <li>• <b>Monitor Metrics:</b> Track task success, feature engagement, and satisfaction to assess impact.</li> </ul>	<ul style="list-style-type: none"> <li>• Finalize Features &amp; Design - 1 week</li> <li>• Development(backend and frontend) - 2-3 weeks</li> <li>• QA testing - 1 week</li> <li>• Beta launch - 1 week</li> <li>• Full launch &amp; post-launch updates - ongoing</li> </ul>

## 8. RISKS AND MITIGATIONS

Risk	Mitigations
Confusing prompt language	Iterate language based on user testing.
Low feature adoption	Educate users with tooltips/onboarding.
Technical issues with Save	Conduct robust QA and implement error tracking.
Low Interaction with Save Prompt	Increase the visibility of the prompt through animations or additional confirmation messaging
Delays in MVP Development	Prioritize critical features and use agile sprints to stay on schedule.

