Feed App: Sprint Development Log

Introduction

When I began developing Feed, my goal was to create an application that simplified the way people discover and interact with digital content. I wanted to replace the endless, passive scrolling found in most modern feeds with something intentional and focused, an experience guided by categories, purpose, and clarity. Over the course of three days, in roughly sixteen non-consecutive hours, I designed and developed this app from the ground up, using AI tools to help translate design concepts into functioning code. Each sprint represented a step forward in shaping both the product and my own understanding of how AI could extend my design process.

My role spanned from UX and interface design to front-end development. I began with sketches, wireframes, and acceptance criteria that defined each feature's purpose and constraints. Using these documents as a guide, I built, tested, and refined Feed across multiple iterations. This journal reflects that journey, an honest record of progress, setbacks, fixes, and moments when the design evolved through interaction with both code and AI.

Sprint 0 – Foundations

Sprint 0 established the foundation of the project. I built the first working prototype using React Native within Expo, setting up a basic tab navigation system and creating essential screens such as Login, Create Account, Choose Interests, Feed, Search, Profile, and Settings. My initial focus was on structure and flow rather than visual polish. I implemented feed cards that displayed image previews, short summaries, and source links, with long-press interactions that allowed users to Save, Send, or Delete items.

At this stage, I wanted a working skeleton, a minimum viable application that could render, navigate, and respond. Although basic, it was a critical milestone. It allowed me to test my initial wireframes in code and confirm that the UX logic held up in practice. The early persistence model and simple theming were placeholders, but they gave me a sense of how the product would eventually feel.

Key Updates:

- Created base React Native structure with Expo and bottom tab navigation.
- Developed initial screens: Login, Create Account, Choose Interests, Feed, Search, Profile, and Settings.
- Implemented feed cards with image previews, actions, and timed updates.
- Introduced long-press menu for Save, Send, and Delete actions.
- Added placeholder theming and dark UI tokens for testing.

Sprint 1 – Refinement and Visual Identity

The first real improvements began in Sprint 1. I integrated Montserrat as the global typeface, giving the app a clean, cohesive visual tone that matched my design system. Category search

was redefined to display topic categories instead of direct article results, aligning with Feed's vision of structured discovery.

This sprint also addressed usability concerns. Buttons were standardized with consistent radii and spacing, and the "Build My Feed" call-to-action became full-width for clarity. I refined long-press actions to remain visible until dismissed, creating a smoother interaction pattern. AI tools helped identify layout inconsistencies and optimize how components rendered in different viewports. This sprint solidified the app's identity, transitioning it from a functional prototype to a visually coherent interface.

Key Updates:

- Implemented global Montserrat font for system-wide typography.
- Updated search tab to show categories instead of articles.
- Added consistent 4px corner radius across all buttons.
- Improved keyboard behavior by ensuring fields scroll above input focus.
- Enhanced long-press menu persistence for smoother interactions.

Sprint 2 – Personalization and Structure

Sprint 2 marked a turning point where personalization began to take shape. I restructured the action menu into a horizontal layout, improved saved-state visibility by adding star icons to category chips, and implemented pull-to-refresh with a dynamic yellow spinner.

One major addition was the Settings screen, which introduced multiple customization options such as theme selection, account management, and content blocking. The AI assistant helped me scaffold these new components quickly, but I spent time refining the details, ensuring that each theme change persisted correctly and that the experience remained responsive. This sprint highlighted how a clear design system paired with AI-assisted code generation could significantly reduce build time while keeping user experience at the forefront.

Key Updates:

- Added horizontal action menu for feed cards.
- Implemented saved-state indicators with star icons.
- Introduced pull-to-refresh with yellow spinner animation.
- Built comprehensive Settings screen with themes, content blocking, and notifications.
- Integrated per-user account management fields and responsive layout tuning.

Sprint 3 – Multilingual Functionality and Data Persistence

Sprint 3 introduced language support, a step that required thoughtful design consideration. I added English, Spanish, French, and Dutch translations, and created a rotating greeting on the feed that cycled every few seconds. I also implemented full per-account persistence using SecureStore, allowing user data such as interests, saved items, and preferences to remain intact between sessions.

This sprint tested my patience technically. The initial data persistence caused unexpected resets and conflicts between local and mock data states. AI-assisted debugging helped

pinpoint a few mismanaged hooks, and I rewrote parts of the logic to ensure stable storage. The end result was a more resilient app and the foundation for user personalization across devices.

Key Updates:

- Added multilingual greetings with timed rotation on the home feed.
- Implemented SecureStore-based account persistence for user data.
- Improved onboarding flow to collect and save profile details.
- Refined interest selection interactions for better usability.
- Debugged storage conflicts and stabilized data handling logic.

Sprint 4 – Expanded Languages and Accessibility

In Sprint 4, I deepened localization by expanding supported languages to include Hebrew, Japanese, Runic, Dutch, and Finnish. I also added the current date under the rotating greeting to enhance context and user familiarity. This sprint was driven by inclusivity and subtle polish.

However, adding new text variations exposed contrast issues in dark mode. I corrected these using accessibility guidelines and my acceptance criteria for legibility. These refinements improved both form and function, ensuring that Feed maintained accessibility regardless of language or theme.

Key Updates:

- Expanded greeting language set to seven total.
- Displayed live date under greeting header.
- Enhanced contrast and color visibility in dark mode.
- Simplified “Choose Interests” screen by removing hero image.
- Improved keyboard navigation and scroll behavior for multi-language input.

Sprint 5 – Stability and Cleanup

This sprint focused primarily on stability. I created a sticky “Build My Feed” button that maintained visibility above the bottom navigation and keyboard, improving consistency during onboarding. I resolved a persistent keyboard overlay issue that caused input fields to be partially hidden, and I corrected several hook ordering problems and stray token declarations in the codebase.

This was the first sprint where AI truly felt like a coding partner. By feeding it snippets of malfunctioning logic and acceptance notes, it suggested corrections and linting improvements that saved significant time. This sprint was all about reliability, making sure Feed behaved predictably in real-world conditions.

Key Updates:

- Implemented sticky “Build My Feed” call-to-action above navigation.
- Fixed keyboard overlay and input focus issues.
- Cleaned redundant state declarations and stray variables.
- Corrected translation map runtime error in i18n logic.
- Improved code consistency and overall app stability.

Sprint 6 – Subscription Plans and Catalog

In Sprint 6, I built out the subscription flow and expanded the onboarding experience. Users could now select from tiered plans (Basic, General, or Unlimited) that simulated a payment process. I also created a large, categorized interest catalog with sections like Sports, Science, Travel, and Finance to give users a meaningful starting point for content curation.

Implementing this required restructuring the account creation flow into two stages: user info and plan selection. I refined transitions between these steps to ensure logical progression and minimal friction. The acceptance criteria guided these refinements, ensuring that tier limits and transitions adhered to design expectations. The result was a more realistic and user-driven onboarding process.

Key Updates:

- Added subscription plan flow with tiered options and simulated payments.
- Built categorized interest catalog with dynamic filters.
- Redesigned onboarding into multi-step process.
- Added password visibility toggle for signup forms.
- Improved layout spacing for back and create buttons.

Sprint 7 – Brand Presence and Usability Enhancements

This sprint was all about polish and brand visibility. I added the official app icon and integrated it across the system. I refined the Settings screen to include active payment methods, plan upgrades, and wellness messages displayed under the feed footer.

This version felt closer to the Feed I envisioned from the start: calm, human, and supportive. The rotating “wellness nudge” messages were small, but they contributed to the tone of intention I wanted users to feel. I also reworked the sticky button’s layering to ensure it behaved correctly regardless of navigation depth. This sprint represented the shift from functional development to thoughtful design curation.

Key Updates:

- Integrated official app icon across UI and navigation.
- Added payment method and plan details to Settings.
- Introduced wellness microcopy under feed footer.
- Improved z-index layering for sticky elements.
- Enhanced delete account button visibility and contrast.

Sprint 8 – Theming and Upgrade UX

Sprint 8 deepened personalization. I redesigned the theme system with friendly color names like Daffodil Yellow, Gentian Blue, and Lavender Pink, and moved plan upgrades to a dedicated

screen for better clarity. The theming dropdown gained swatches, spacing adjustments, and descriptive help text for ease of use.

One of my goals was to make Feed feel human and expressive without overcomplicating it. I revisited several component layouts to clean spacing and improve readability. These refinements, while subtle, reinforced the visual rhythm and personality of the interface. Each improvement was a direct reflection of earlier design documentation and accessibility notes.

Key Updates:

- Reworked theme selection with color swatches and friendly names.
- Added dedicated plan upgrade screen with descriptive copy.
- Improved layout of dropdowns and confirmation CTAs.
- Increased spacing and hierarchy for interest selection sections.
- Updated neutral theme to new Jasmine White tone.

Sprint 9 – Profile Enhancements and Saved Content

Here, I focused on organization and recall. Saved items were grouped by time—Today, Yesterday, Last Week, and so on. Tapping a date opened a carousel viewer, creating a seamless way to revisit past content. The layout was redesigned for horizontal scrolling, making it more intuitive for mobile use.

This sprint tested design-to-code consistency more than any other. Ensuring the carousel logic and grouping aligned with my design expectations required multiple iterations. AI-assisted testing flagged several rendering conflicts that I fine-tuned manually. By the end, Feed's profile section had evolved from a static grid into an elegant, living archive.

Key Updates:

- Grouped saved items by date with labeled headers.
- Added modal carousel viewer for saved content.
- Designed single-row horizontal scrolling by category.
- Fixed sticky CTA behavior consistency with keyboard interactions.
- Enhanced layout spacing and target sizing for touch input.

Sprint 10 – Finalization and Release

The final sprint tied everything together. I refined the content blocking system so that it worked dynamically within the search feature, ensuring blocked terms properly filtered results without breaking logic elsewhere. I updated copy across the interface, improved contrast, and finalized the onboarding sequence with a confirmation message reading "Your feed is ready."

This sprint focused on cohesiveness and small details: localized text corrections, better category labels, and logical counter feedback for plan limits. It was here that the project reached the standard I envisioned at the start, an intentional, accessible, and complete product experience. Seeing Feed come to life from sketches to a stable version 1.0 was both rewarding and affirming.

Key Updates:

- Finalized content blocking across search and feed filtering.
- Unified app copy and interface tone for consistency.

- Enhanced accessibility contrast and text clarity.
- Added plan confirmation and active plan indicators.
- Completed onboarding flow with success confirmation screen.

Reflection

Across ten sprints, Feed grew from a concept into a functioning, human-centered application that demonstrates how AI can accelerate design and development without replacing creativity. I used AI to fill technical gaps, automate repetitive build tasks, and validate components against acceptance criteria. Each sprint was a dialogue between design and technology, and between intuition and precision.

Looking back, I learned that structure and vision are what make AI effective. With clear documentation, intentional design systems, and accessibility-driven standards, the AI tools became extensions of my process. The result was more than a working product, it was a study in how modern design can harness intelligence and iteration to create meaningful, efficient, and beautiful experiences.