**Momentum: A Content Creator’s All in One Productivity Cross-Platform App**

Marqice A. Rector

Keiser University

CEN4090: Software Engineering Capstone

Professor Kruayatidee, Somard

August 28,2025

# 

# **Project Overview**

Content creators face unique challenges in organizing ideas, planning content, and managing production across multiple platforms. Most existing productivity tools fail to address these specialized needs, resulting in fragmented workflows and lost efficiency.

This capstone project proposes a cross-platform productivity and planning application tailored specifically for content creators. The app enables users to capture ideas, schedule content, manage production workflows, and track key analytics—all within a unified and intuitive interface.

Developed using React Native and Firebase (React Native, n.d.; Firebase, n.d.), the solution is designed for both mobile and web use, supporting creators wherever they work. The aim is to streamline the content creation process and empower creators to consistently move from concept to publication with greater organization and less stress.

**User Stories**

This application is designed to address the core needs of content creators who aim to manage their ideas, track productivity, and gain actionable insights across platforms. The following user stories reflect the primary goals and motivations of the intended users:

* As a content creator, I want to easily log and organize my ideas so that I never lose track of valuable content concepts.
* As a user, I want to set tasks and deadlines on a calendar so that I can stay accountable and plan my content creation schedule efficiently.
* As a creator managing multiple platforms, I want to manually log analytics for each of my videos or posts so I can visualize growth and performance trends over time.
* As a user, I want to quickly view upcoming tasks and major ideas on my dashboard so that I can focus on what’s most important each day.
* As a user, I want to edit or delete my ideas, tasks, or analytics entries if my plans or content strategies change.
* As a user, I want the ability to adjust my profile information, application theme, and notification preferences, so the app fits my workflow and style.
* As a user, I want secure login and logout features, ensuring my data remains private and only accessible by me.

These user stories form the foundation of the application’s design and feature set, ensuring that the software remains focused on solving the real challenges faced by modern content creators.

# **Problem Statement**

Content creators often manage multiple projects, platforms, and deadlines simultaneously, requiring careful coordination and organization. However, most available productivity tools are designed for general task management and lack features that address the specific workflows and requirements of content creation. As a result, creators are forced to rely on a combination of generic apps, spreadsheets, and manual reminders, leading to inefficiency, missed opportunities, and increased risk of burnout.

There is a clear need for a unified solution that caters to the distinct planning, tracking, and organizational demands of modern content creators.

# **Project Scope**

This project will deliver a cross-platform productivity and planning application specifically designed for content creators. The core features will include:

* Idea Bank: Capture, categorize, and manage content ideas.
* Content Calendar: Visually schedule upcoming posts across multiple platforms.
* Workflow Tracker: Use customizable checklists to monitor each stage of content production, from brainstorming to publication.
* Manual Analytics Logging: Record and review basic performance metrics for published content.
* Reminders and Notifications: Receive alerts for deadlines and important tasks.
* User Authentication: Secure sign-up and login functionality.
* Responsive Design: Seamless experience across mobile devices and web browsers.

The app will be developed using React Native (with Expo for web compatibility) and Firebase for backend services.

**Out of Scope**

This project will not include automated integration with external analytics APIs (such as YouTube or Instagram), advanced media editing capabilities, or real-time team collaboration beyond individual user accounts.

The goal is to deliver a fully functional minimum viable product (MVP) within an eight-week development timeline.

# **Objectives**

The primary objective is to equip content creators with a unified tool that increases productivity, enhances organization, and reduces friction in the content creation process.

* Develop a cross-platform application that streamlines content planning and workflow management for content creators.
* Enable users to efficiently capture and organize content ideas in a structured and accessible format.
* Provide a visual scheduling tool that allows creators to plan, track, and manage upcoming content across multiple platforms.
* Implement customizable production workflows that guide users through each step of content creation.
* Offer manual analytics tracking to help creators monitor performance and growth.
* Deliver a responsive, user-friendly interface that works seamlessly on both mobile devices and web browsers.
* Ensure secure user authentication and data management using modern technologies and best practices.

# **Requirements**

## **Functional Requirements**

The functional requirements define the core features and behaviors that the application must support to meet the needs of content creators. These requirements specify the essential actions and interactions that users should be able to perform within the app:

User Authentication

* The system shall allow users to register and log in securely.
* The system shall support password reset functionality.

Idea Bank

* Users shall be able to add, edit, and delete content ideas.
* Users shall be able to categorize and tag ideas.

Content Calendar

* Users shall be able to schedule upcoming posts on a visual calendar.
* The calendar shall display scheduled content and deadlines.

Workflow Tracker

* Users shall have customizable checklists for each content piece.
* Users shall be able to update the status of content as it progresses through production stages (e.g., idea, script, record, edit, publish).

Manual Analytics Logging

* Users shall be able to manually enter performance metrics (e.g., views, likes, comments) for each published piece of content.
* The system shall display basic charts or summaries of analytics data.

Reminders and Notifications

* Users shall receive reminders for upcoming deadlines and tasks.
* Users shall be able to customize notification settings.

Responsive **Design**

* The application shall function seamlessly on both mobile and web platforms.

**Non-Functional Requirements**

The non-functional requirements describe the overall qualities and constraints of the application, ensuring a reliable and high-quality user experience. These requirements address aspects such as performance, security, usability, compatibility, and maintainability:

Performance:

* The application shall load all main screens within two seconds on standard hardware and network conditions.

Security:

* All user data shall be stored securely and transmitted over encrypted channels.
* The app shall comply with basic data privacy standards.

Reliability:

* The system shall maintain at least 99% uptime during the project demonstration period.

Usability:

* The interface shall be intuitive, with minimal learning curve for new users.
* Help and support information shall be accessible within the app.

Compatibility:

* The application shall support the latest versions of major browsers and mobile operating systems (iOS and Android).

Maintainability:

* The codebase shall follow standard development practices and be documented for future updates.

**System Design**

The content creator productivity app is designed as a cross-platform solution, leveraging a modern client-server architecture to ensure a responsive, reliable, and scalable experience for users on both mobile and web platforms. The system is divided into the following main components:

* Client Application:Developed with React Native and Expo for web compatibility (React Native, n.d.; Expo, n.d.), the client application delivers a unified codebase for both mobile devices and web browsers. The front end is responsible for user interface rendering, input handling, local data validation, and communication with backend services.
* Backend Services: Firebase serves as the primary backend solution (Firebase, n.d.), providing user authentication, real-time data storage, and cloud-based notification support. All sensitive operations, including authentication and data management, are handled securely through Firebase’s managed services.
* Data Management:User data—including content ideas, calendar events, workflow progress, and analytics entries—is securely stored in Firebase. Data access is managed through well-defined security rules to ensure privacy and integrity.
* Notifications:Reminders and deadline alerts are managed via Firebase Cloud Messaging and/or Expo Notifications, enabling timely and platform-appropriate push notifications to users on both mobile and web.
* **Security:** All data transmission between the client and backend is secured using HTTPS. Authentication tokens are managed securely, and access is restricted to authorized users.

The overall system design prioritizes modularity, scalability, and ease of future maintenance. The architecture supports the seamless addition of new features and allows for efficient updates and troubleshooting.

# **High-Level Architecture**

The application is built upon a modular client-server architecture that ensures robust communication and secure data management across platforms. The primary components and their interactions are as follows:

## **Client Application**

A cross-platform React Native app (using Expo) deployed to both mobile devices and web browsers. The client interfaces directly with backend services for data storage, user authentication, and notifications.

**Backend Services**  
Firebase provides core backend functionality, including:

* Authentication: Manages user sign-up, login, and session handling.
* Firebase Database: Stores all user data (content ideas, schedules, workflows, analytics).
* Cloud Messaging: Delivers reminders and deadline notifications to user devices.

## **Data Flow**

* Users interact with the app to submit, update, and retrieve information.
* The client communicates securely with Firebase APIs to store and access user data.
* Notification triggers (e.g., reminders, deadlines) are set within the app and managed through Firebase Cloud Messaging or Expo Notifications.

**Security**

All communications between the client and Firebase services occur over encrypted connections. Access to data is governed by Firebase security rules to ensure that only authorized users can access their own information.

# **Technology Stack**

| **Layer** | **Technology** | **Purpose/Notes** |
| --- | --- | --- |
| Frontend | React Native | Cross-platform UI for mobile and web (via Expo) |
| Frontend | Expo | Streamlines mobile/web builds, device features, and testing |
| Backend | Firebase | Provides backend-as-a-service, authentication, database |
| Database | Firestore | Cloud NoSQL database for storing user/content data |
| Notifications | Expo Notifications / Firebase Cloud Messaging | Push notifications/reminders |
| Deployment | Expo Go / Expo Build | App deployment to iOS, Android, and web |

**Component/Modular Design**

The Momentum Productivity App is structured around a modular architecture, where each core feature is encapsulated within its own component or screen. This design approach supports scalability, ease of maintenance, and clear separation of concerns. The following describes the major components and their primary responsibilities within the application. References to the corresponding wireframes and diagrams are provided for further clarity.

**1. Dashboard**

* Purpose: Serves as the central hub for users, providing a summary view of upcoming events, major ideas, and quick-access statistics relevant to content creation.
* Features:
  + Displays the next 3–5 upcoming tasks or events (with edit/delete options for rapid management).
  + Highlights major or “flagged” ideas for immediate attention.
  + Presents quick stats (number of tasks due, total ideas, productivity streaks).
  + Includes navigation shortcuts to other sections (e.g., Calendar, Idea Bank).
* Diagram Reference**:** See Dashboard Wireframe and User Flow Diagram.

**2. Idea Bank**

* Purpose: Allows users to log, organize, edit, and delete content ideas in a dedicated space, ensuring no creative concept is lost.
* Features:
  + Add, edit, and delete ideas with title and description.
  + Display ideas in a scrollable list, sorted by creation or update date.
  + (Optional) Star or flag key ideas for focus.
* Diagram Reference: See Idea Bank Wireframe and Use Case Diagram.

**3. Calendar**

* Purpose: Enables users to schedule, review, and manage tasks/events tied to specific dates.
* Features:
  + Visual calendar interface for selecting and viewing dates.
  + List of tasks/events for the selected date, with options to add, edit, or delete.
  + Integration with Dashboard to surface urgent/upcoming events.
* Diagram Reference: See Calendar Wireframe and User Flow Diagram.

**4. Analytics**

* Purpose: Provides manual tracking and visualization of content performance analytics across multiple platforms.
* Features:
  + Users log analytics data for content (views, likes, comments, etc.) per platform.
  + Data is visualized via interactive graphs and charts, with filtering by platform.
  + List of logged analytics entries, filterable by platform and date.
* Diagram Reference: See Analytics Wireframe and ER Diagram.

**5. Settings**

* Purpose: Centralizes all user preferences and account management.
* Features:
  + View and edit profile information.
  + Toggle application theme (light/dark/system).
  + Mute or unmute notifications.
  + Log in, log out, and view app version.
* Diagram Reference: See Settings Wireframe.

**6. Navigation**

* Purpose: Facilitates smooth movement between major app sections.
* Features:
  + Tab navigation (Dashboard, Idea Bank, Calendar, Analytics, Settings).
  + Bottom navigation bar, with visual feedback for active section.
* Diagram Reference: See User Flow Diagram and Component Diagram (if available).

**7. Data & Backend Integration**

* Purpose: Manages all persistent data, user authentication, and synchronization.
* Features:
  + Firebase Firestore for storing user data (ideas, tasks, analytics, settings).
  + Firebase Authentication for secure login/logout.
  + Real-time sync between app and cloud backend.
* Diagram Reference: See ER Diagram and High-Level Architecture Diagram.

**Summary Table: Core Components**

|  |  |  |
| --- | --- | --- |
| **Component** | **Main Function** | **Related Diagrams** |
| Dashboard | Summary, quick actions, stats | Dashboard Wireframe, User Flow |
| Idea Bank | Idea CRUD management | Idea Bank Wireframe, Use Case |
| Calendar | Task/event scheduling & management | Calendar Wireframe, User Flow |
| Analytics | Manual analytics tracking & visualization | Analytics Wireframe, ER Diagram |
| Settings | Profile, preferences, theme, auth | Settings Wireframe |
| Navigation | In-app section navigation | User Flow Diagram, Component |
| Data/Backend | Data persistence & sync | ER Diagram, Architecture |

This modular design allows each feature to be developed, tested, and updated independently, supporting a robust and scalable application architecture. For detailed UI layouts, see the respective wireframes included in the appendix. For data flow and entity relationships, refer to the ER and architecture diagrams.

# **Implementation Plan/ Schedule**

This project will be developed and delivered over an eight-week period. The implementation is organized into phases to ensure steady progress and timely completion of all core features.

**Week-by-Week Schedule:**

1. **Week 1: Planning & Setup**

* Finalize project requirements and technical specifications
* Set up project repositories and development environment
* Create initial mockups and wireframes

1. **Weeks 2–3: Core Functionality Development**

* Implement user authentication (sign up, login, password reset)
* Build the idea bank for adding, editing, and deleting content ideas
* Develop basic UI components and layouts

1. **Week 4: Content Calendar & Workflow**

* Develop the content calendar for scheduling and viewing planned content
* Implement customizable workflow checklists for content production stages

1. **Week 5: Analytics & Notifications**

* Enable manual analytics logging and display basic performance summaries
* Integrate reminder and notification features

1. **Week 6: Responsive Design & Cross-Platform Testing**

* Refine UI for consistent experience across mobile and web
* Conduct testing and debugging on different devices and browsers

1. **Week 7: Polish & Documentation**

* Address remaining bugs and optimize performance
* Finalize project documentation, including user guide and system notes

1. **Week 8: Final Testing & Presentation Preparation**

* Conduct comprehensive testing and quality assurance
* Prepare project presentation and deployment for demonstration

**Milestones:**

* Core features (authentication, idea bank) functional by end of Week 3
* Calendar and workflow features integrated by end of Week 4
* Full MVP completed and tested by end of Week 6
* Documentation, polish, and presentation-ready deployment by end of Week 8

# **Testing Plan**

A comprehensive testing approach will be employed to ensure the reliability, usability, and quality of the content creator productivity app. Testing will occur throughout the development lifecycle and cover core functionalities as well as the overall user experience.

## **1. Unit Testing:**

* **Objective:** Validate individual components and functions (e.g., idea bank logic, authentication).
* **Approach:**
  + Write unit tests for key modules and utility functions.
  + Use testing frameworks such as Jest for JavaScript/React Native (Jest, n.d.).

## **2. Integration Testing:**

* **Objective:** Ensure multiple components work together as intended (e.g., adding an idea updates the calendar and workflow).
* **Approach:**
  + Test integration between frontend components and backend services (Firebase).
  + Verify data flow between user actions, storage, and notifications.

## **3. User Interface (UI) Testing**

* **Objective:** Confirm that the user interface behaves correctly and remains consistent across devices.
* **Approach:**
  + Manually test screens for responsiveness and usability on both mobile and web.
  + Address layout or navigation issues on different screen sizes.

## **4. End-to-End (E2E) Testing**

* **Objective:** Validate complete user workflows from login to idea creation, scheduling, workflow tracking, and analytics entry.
* **Approach:** Simulate real user interactions using tools such as Detox (for React Native) or manual step-by-step testing.

## **5. User Acceptance Testing (UAT)**

* **Objective:** Ensure the app meets user expectations and functional requirements.
* **Approach:**
  + Conduct feedback sessions with a small group of potential users or peers.
  + Collect feedback on usability, feature completeness, and overall experience.

## **6. Bug Tracking and Resolution**

* All discovered bugs or issues will be documented in a shared tracking system (e.g., GitHub Issues or Trello) (GitHub, Inc., n.d.; Trello, n.d.).
* Bugs will be prioritized and resolved before the final release.

## **7. Regression Testing**

* After fixing bugs or adding new features, retest critical paths to ensure no existing functionality is broken.

# **Conclusions & Limitations**

# Due to time and development constraints, several advanced features were excluded from the initial release of Momentum. Notably, the app does not support real-time analytics integration with external content platforms. Users are required to manually enter performance metrics for their content, which, while effective, can be time-consuming and less accurate. Integrating APIs to automatically retrieve analytics from social media platforms would streamline this process and provide users with up-to-date, actionable insights. Although not feasible for the current timeline, this remains a key area for future enhancement to further empower content creators.

# **References**

Expo. (n.d.). *Expo documentation*. https://docs.expo.dev/

Firebase. (n.d.). *Firebase documentation*. https://firebase.google.com/docs

GitHub, Inc. (n.d.). *GitHub Docs*. https://docs.github.com/

Jest. (n.d.). *Jest documentation*. https://jestjs.io/docs/getting-started

React Native. (n.d.). *React Native documentation*. https://reactnative.dev/docs/getting-started