ISSN 2277-2685
IJESR/April-June. 2025/ Vol-15/Issue-2s/866-874
Korani Smyrna *et. al.,* / International Journal of Engineering & Science Research

Mobile Application for Enhanced Time Capsule

Ms. K. Shireesha, Korani Smyrna, Takemal Sindhuja
[1]Associate Professor, Department of CSE, Bhoj Reddy Engineering College for Women, India
[2,3]B.Tech Student, Department of CSE, Bhoj Reddy Engineering College for Women, India

Abstract:
Many individuals like working professionals, students, and house makers often find lack of your time and time management as problems for successful post accomplishment. One among the key reasons for failure in post accomplishment is inefficient planning of the posts. It's vital to seek out a tool that may help in schedule and manage the time, meetings, and appointments. The importance of this project work comes from the requirement of employing a single React Native Application that may be accustomed control many of the operations like posts, appointments and meetings for people and also for teams and tiny companies. This application uses Flow Time technique for post management. This tool will reduce the buying costs of multiple applications.

Keywords—Post, react native, meeting, scheduling, firebase, Flow Time technique.

INTRODUCTION

In our daily operations, time is exceedingly valuable for employees, as we cannot afford to squander it on superfluous tasks; the answer is in using a post management application. Post management is the instrument for generating and overseeing posts during the project's lifespan. It include requirements collecting, planning, progress monitoring, testing, and the generation of final reports upon project completion.

Individuals use post management tools like as notebooks or software applications to systematically arrange and achieve personal objectives for daily tasks. Posts may possess a status, start date, due date, assigned personnel, comments, tags, and associated files. Advanced post management systems facilitate dependencies, repetition, priority, and complexity.

React Native integrates the optimal elements of native development with React, a premier JavaScript toolkit for constructing user interfaces. React components encapsulate existing native code and interface with native APIs using React's declarative UI paradigm and JavaScript. This facilitates native application creation for whole new teams of developers and allows current native teams to operate with increased efficiency.
The use of cloud computing features, such as Firebase, a Google product, facilitates developers in building, managing, and expanding applications with ease. It is an excellent option for developers to construct applications more rapidly and securely. It offers services for Android, iOS, Web, and Unity.

The Flow Time Technique is a time management strategy developed by Zoe Read-Bivens. The Flow Time approach is beneficial for creatives, developers, students, and anyone whose tasks need intense focus. It involves selecting a post, engaging with it until fatigue sets in, and thereafter taking a respite. Continue the procedure until the post is completed.

This article examines the specifics of the native application PostAtFlow, along with its scheduling mechanism.

Recognizing Firebase and React Native facilitates the comprehension of developing a cost-effective and scalable solution. The real intricacy of the

solution lies in its scheduling mechanism, which segments the user post into many sub-posts. It does this by using machine learning to automatically schedule and reschedule a day for the user. Scheduling is conducted according to the user's chosen time slots, determined from pending postings and the user's progress.

PROPOSED SYSTEM

The Pomodoro technique is effective for tasks that are undesirable or do not need profound contemplation. The issue emerges when one must engage in tasks necessitating creativity, invention, and problem-solving capabilities. A duration of 25 minutes often proves insufficient, disrupting your productivity and train of thought. Moreover, 25-minute intervals may be too lengthy for certain postings. The Flow time approach addresses these issues by allowing you to choose your own time frames, ensuring they correspond with your workflow.

Journal of Cardiovascular Disease Research, ISSN: 0975-3583, 0976-2833, Volume 12, Issue 07, 2021, Page 2268

The flow time approach is an adaptation of the Pomodoro technique, whereby instead of a timer for each specific task, the individual determines the duration of work sessions and breaks. It promotes the user to attain a flow state, characterized by complete engagement and immersion in a task. This approach is characterized by few regulations; the key aspect is to heed your inner voice. The simplicity is among its foremost merits.

PostAtFlow is a post management tool using the FlowTime approach, enabling users to improve their daily activities. It is developed using React Native, a JavaScript framework used for creating native applications. The most significant feature of PostAtFlow is post scheduling. A user only has to input their posts, choose a priority (if wanted), and a daily schedule will be prepared for them. The integration with personal calendars is a significant aspect of PostAtFlow.

Architectural Design:

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components is built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

MVC Components

Following are the components of MVC −

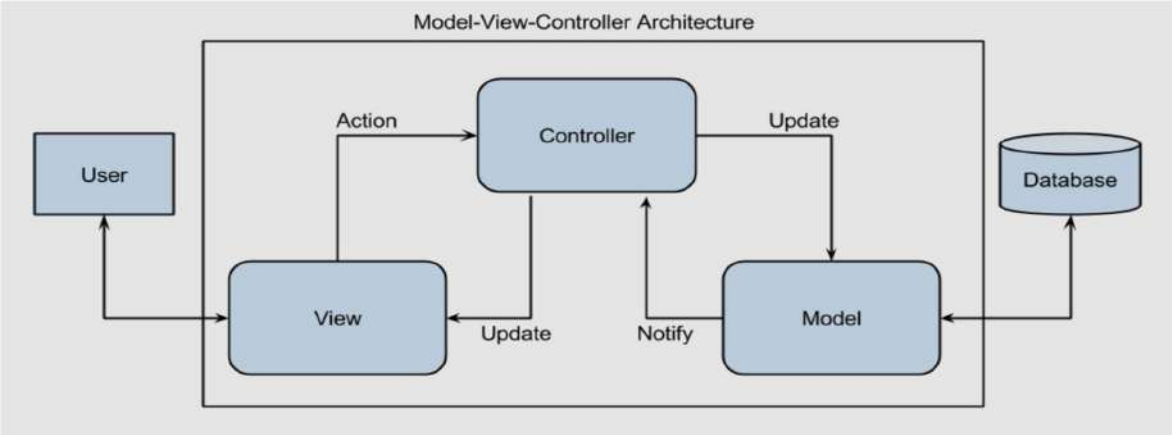


Fig: Technical Architecture

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

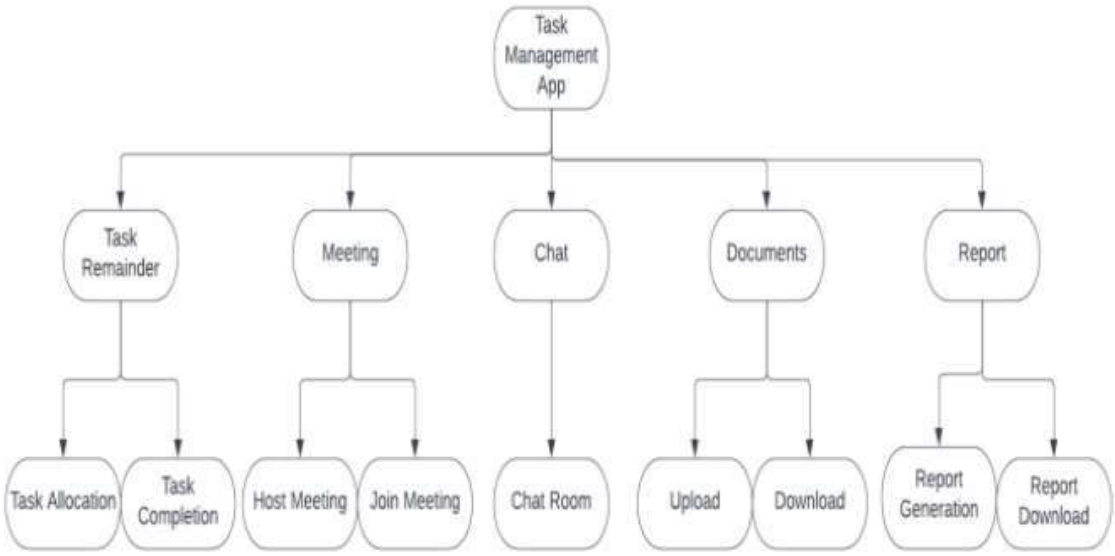
View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

System Architecture:



Implementation:

PostAtFlow is a mobile application that employs the React Native framework, a widely used JavaScript framework for creating applications on both Android and iOS platforms from a single codebase. It is used to create apps for Android, Android TV, iOS, macOS, tvOS, Web, Windows, and UWP by allowing developers to leverage the React framework in conjunction with native platform functionalities.

The application's backend utilizes Firebase, a Google tool that facilitates developers in building,

managing, and expanding their applications efficiently. Firebase offers features such as authentication, database management, machine learning, file storage supported by Google Cloud Storage, and hosting services. This makes the application appropriate for development on Firebase. It enables developers to construct their applications more rapidly and securely. No programming is necessary on the Firebase side, facilitating more effective use of its functionalities. It offers services for Android, iOS, web, and Unity.

The Firebase Realtime Database is a cloud-based database. Data is kept in JSON format and synced in real-time with all connected clients. When developing cross-platform applications with our Apple iOS, Android, and JavaScript SDKs, all clients access a singular Real-time Database instance and immediately get updates with the latest data. The JSON format has syntactical resemblance to the code used for constructing JavaScript objects. Consequently, a JavaScript application can easily transform JSON data into JavaScript objects. Due to its text-only structure, JSON data may be effortlessly sent between computers and used by any programming language.

Firebase Authentication offers backend services, user-friendly SDKs, and pre-built UI frameworks for user authentication in your application. It facilitates authentication by passwords, phone numbers, and prominent federated identity providers such as Google, Facebook, and Twitter, among others. Firebase Authentication seamlessly connects with other Firebase services and utilizes industry standards like as OAuth 2.0 and OpenID Connect, facilitating straightforward integration with your own backend.

The document upload and download functionality utilizes Firebase Cloud Storage for the storage, retrieval, and deletion of documents. Cloud Storage for Firebase is a robust, user-friendly, and economical object storage solution designed for Google-scale applications. The Firebase SDKs for Cloud Storage enhance file uploads and downloads for your Firebase applications with Google security, irrespective of network quality. The SDKs may be used to store photos, music, video, or other user-generated material. Google Cloud Storage APIs may be used on the server to retrieve identical files.

The application's meeting functionality utilizes the integration of the Zoom SDK into our mobile application. The Zoom Meeting SDKs enable both new and current apps to include a comprehensive array of functionality found in the Zoom Client application.

The program has a chat function that enables individuals to pose inquiries to their colleagues and managers. The functionality is implemented using the React Native Gifted Chat package. This is the React Native package containing pre-built components for usage in our project. As Firebase offers cloud-based messaging inside its SDK, implementation is straightforward.

The primary function is to facilitate the scheduling and rescheduling of postings inside the system. Scheduling occurs when a user adds a new project or post and at the moment of PostAtFlowcompletion. Furthermore, the postings are adjusted anytime a user modifies the connected calendar. Posts are scheduled and rescheduled only upon the user's execution of certain triggers.

A. Scheduling Algorithm

The scheduling algorithm is a crucial element of PostAtFlow. The scheduling algorithm manages two actions: creating a new schedule and rescheduling a post.

The scheduling process is executed in the following steps:

Identify vacant time intervals in the user's calendar.

2. Evaluate the number of posts that can be accommodated during the designated timeframe for the user.
Create database records for each post.
4. Optionally use machine learning to determine the extent to which the post may be scheduled inside the user's time window.

In the first phase of the scheduling process, the tool generates an object that encompasses available time slots, including start and end timestamps. Step 2 involves creating an object that specifies the quantity of posts that may be scheduled in each available time slot, along with brief (5-minute) and extended (20-minute) pauses. The concluding phase retrieves all projects and postings the user is engaged in and allocates them into available time slots according to the user's designated priority.

Rescheduling occurs when users choose to bypass a post, seek more time for a post, or include a new post or calendar event. Rescheduling employs a modified version of the aforementioned stages, while also monitoring the number of posts required for a task's completion.

If a post is omitted, all following posts are advanced by one position, and the current post is repositioned to the end of the queue. If an additional post is required for the current post, all posts are displaced downward by one position, and a new lot is generated immediately after the current post in process. Upon the addition of a new post to the list, a vacant slot is designated for it, and the post is included into the timetable.

When a user modifies the calendar by including meetings, events, or time off, the time slots are recorded in the database with a personal tag, enabling PostAtFlow to bypass those periods during the scheduling procedure.

The most challenging aspect of scheduling is creating the rescheduling code, since it involves rearranging postings among existing items. The challenge is in monitoring the brief and extended gaps, as well as the need to adjust postings to the subsequent or preceding day. The start and end timings inside the post object alleviate this challenge, since these values enable the adjustment of posts in the schedule without necessitating a recalculation of the start and finish times. Nonetheless, it is crucial to consider the user's workday preferences and the quantity of posts while changing post objects. When rescheduling takes place, the postings must be precisely moved to the next day if the present day lacks available time slots. The rescheduling code is validating that the post's start and finish hours align with the user's workday choices. Furthermore, to guarantee the right allocation of breaks, it is essential that rescheduling maintains an accurate tally of modified postings. This guarantees that each post receives a brief intermission and an extended pause after every four postings.

B. Comparison of the Existing System with the Proposed System

The primary distinction is that the former system operates as a web application, while the current system functions as a mobile application. The current system employs the Pomodoro approach to schedule posts on the user timeline, which has several drawbacks.

The Pomodoro approach is suited for positions that need extended periods of focus or those with excessively brief work intervals.

The timer serves as a persistent interruption that obstructs the capacity to achieve a flow state, which is characterized by complete engagement and immersion in a task.

Integration and Implementation Tools:

Software may need to be integrated with the libraries, databases and other program(s). This stage of SDLC is involved in the integration of software with outer world entities.

Implementation this means installing the software on user machines. At times, software needs post-installation configurations at user end. Software is tested for portability and adaptability and integration related issues are solved during implementation.

Test Cases:

Test Case ID	Feature	Test Description	Input Data	Expected Result	Actual Result	Status
TC001	Post Creation	Verify user can create a new post	Post name, description, due date	Post is successfully created and displayed in the post list		
TC002	Automatic Scheduling	Verify automatic scheduling of posts	Post priority, available time slots	Posts are scheduled based on priority and user availability		
TC003	Rescheduling Posts	Verify posts are rescheduled when a post is skipped	Skip a scheduled post	Skipped post is moved to the end, and subsequent posts are shifted forward		
TC004	Post Notification	Verify user receives notification for scheduled posts	Scheduled post time	Notification is received at the scheduled time		
TC005	Post Analytics	Verify user can view post analytics and data	Multiple completed posts	Post analytics (e.g., time spent, completion rate) are displayed correctly		
TC006	Meeting Integration	Verify integration with meeting platforms (e.g., Zoom)	Schedule a meeting with post	Meeting link is generated and accessible from the app		
TC007	Chat Room	Verify the chat room allows discussion among team	Send a message in chat room	Message appears in the chat room for all participants		
TC008	Document Upload/Download	Verify documents can be uploaded and downloaded	Upload a document	Document is uploaded and available for download		
TC009	Personal To-Do List	Verify posts can be added to a personal to-do list	Add a post to personal to-do	Post appears in the personal to-do list		

Test Case ID	Feature	Test Description	Input Data	Expected Result	Actual Result	Status
TC010	Flow Time Technique Timer	Verify the timer functionality for flow time technique	Start and stop the timer	Timer starts/stops correctly and logs the time		
TC011	Calendar Integration	Verify integration with personal calendar	Add events/posts to calendar	Events/posts are synced with the personal calendar		
TC012	Firebase Authentication	Verify user login using Firebase authentication	Valid login credentials	User is successfully authenticated and redirected to the dashboard		
TC013	Offline Functionality	Verify app functionality when offline	Turn off internet and create a post	Post is saved locally and synced when internet is back		
TC014	Multi-Platform Support	Verify app works on both Android and iOS devices	Install app on Android and iOS	App functions correctly on both platforms		
TC015	Break Scheduling (Flow Time)	Verify correct scheduling of short and long breaks	Schedule multiple posts	Breaks are scheduled after every 4 posts (long break) and after each post (short break)		

Operation and Maintenance

This phase confirms the software operation in terms of more efficiency and less errors. If required, the users are trained on, or aided with the documentation on how to operate the software and how to keep the software operational. The software is maintained timely by updating the code according to the changes taking place in user end environment or technology. This phase may face challenges from hidden bugs and real-world unidentified problems.

Disposition

As time elapses, the software may decline on the performance front. It may go completely obsolete or may need intense up gradation. Hence a pressing need to eliminate a major portion of the system arises. This phase includes archiving data and required software components, closing down the system, planning disposition activity and terminating system at appropriate end-of-system time.

RESULT

CONCLUSION

This work presents a mobile post management application that assists people and organizations in efficiently organizing their projects and postings. This project will serve as a calendar for appointments. Oversee projects using dashboards among team members. This project will diminish the reliance on other tools, therefore transforming the application into a comprehensive solution that encompasses the specified capabilities. The PostAtFlow is designed using React Native, including an interactive user interface system. This application, written in React Native, is compatible with both Android and iOS platforms.

REFERENCES

[1] Parvizsho Amnov, Navjot Bola, Dipti Shiralkar, Meghana Yoganarasimha “Cloud Based Algorithm for
Post Management” IEEE International Conference on Computational Science 2020

[2] Nikhil Sai, Santosh Gurram, Mohammad Amin Kuhail “PostDo: A Daily Post Recommender System” IEEE
Second International Conference on Computational Intelligence in Data Science 2020.

[3] Robert Gazda, Yang Guo, Li-Tse Hsieh, Hang Liu “Post Management for Cooperative Mobile Edge Computing” IEEE/ACM Symposium on Edge Computing 2020

[4] Kaviya P, Deva Jothi J, Hamsavarthini J, Suwathi P “Mobile Application for Automatic Post Management andReport Generation” Compliance Engineering Journal Volume 11, Issue 5, 2020

[5] G. Uma Maheswari, B. Persis Urbana IVY, P.J. Kumar and P. Suganya “Android Based Post Scheduler and
Indicator” International Journal of Applied Engineering Research Volume 9, Number 22 (2014)

[6] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, “Computing resource allocation in three-tier IoT
fog networks: A joint optimization approach combining stackelberg game and matching,” IEEE Internet

[7] Shiqi, et al. "Front-end and Back-end Separation-React Based Framework for Networked Remote Control
Laboratory." 2018 37th Chinese Control Conference (CCC).

[8] Reichert, M., Weber, B.: Enabling Flexibility in Process- Aware Information Systems: Challenges, Methods,
Technologies. Springer, Heidelberg (2012)

[9] Pryss, R., Tiedeken, J., Kreher, U., Reichert, M.: Towards Flexible Process Support on Mobile Devices.
In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 150–165. Springer, Heidelberg (2011).