# DAYS: A MODERN EVENT COUNTDOWN APP

## A MINI PROJECT REPORT

*Submitted By*


**MANOJ M G (210701149)**

**MERCY N (210701157)**

**MITESH A (210701158)**


*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**RAJALAKSHMI ENGINEERING COLLEGE**

**ANNA UNIVERSITY, CHENNAI**

April 2024

# BONAFIDE CERTIFICATE

Certified that this project report "**DAYS: A MODERN EVENT COUNTDOWN APP**" is the bonafide work of "**MANOJ M G, MERCY N, MITESH A**" who carried out the project under my supervision. Certified further to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.


**SIGNATURE**

Mrs. Ananthi S

**SUPERVISOR**

Assistant Professor (SG)

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105.

**SIGNATURE**

Dr. P. Kumar

**HEAD OF THE DEPARTMENT**

Department Of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai – 602 105.


**Submitted for Semester Mini-Project viva-voce examination held on _____**

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# ABSTRACT

Days App is an innovative Android application designed to provide users with a modern and visually appealing method to track countdowns to significant events. By utilizing Jetpack Compose and Kotlin, Days App offers a seamless and intuitive user experience, integrating the latest Android development practices. The core functionality revolves around presenting events as beautiful, customizable cards, allowing users to personalize their countdowns and enhance their engagement with the app.

One of the standout features of Days App is its ability to display events in the form of aesthetically pleasing cards. Each card can be customized by the user with a background image of their choice, ensuring that every event is represented in a unique and meaningful way. This level of customization not only improves the visual appeal of the app but also helps users to emotionally connect with their upcoming events.

In addition to the customizable event cards, Days App supports home screen widgets, providing users with the convenience of viewing their countdowns directly from their home screen. This feature enhances the app's usability by offering quick access to event information without the need to launch the application. The integration of widgets ensures that important dates are always within sight, further enriching the user experience.

Technically, Days App is built using Jetpack Compose, a modern toolkit for building native Android UI. This allows for a more efficient and flexible development process, enabling the creation of dynamic and responsive user interfaces. By leveraging Kotlin, the app benefits from a robust and concise programming language, ensuring high performance and maintainability. Overall, Days App combines aesthetic design with technical excellence to deliver a superior event countdown experience.

## ACKNOWLEDGEMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.,** for his sincere endeavour in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal, **Dr. S. N. Murugesan M.E., PhD.,** and **Dr. P. KUMAR M.E., PhD, Director computing and information science , and Head Of Department of Computer Science and Engineering** and our project coordinator **Mrs. S. Ananthi, M.TECH** for her encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all facultymembers and supp orting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

**MANOJ M G**

**MERCY N**

**MITESH A**

# TABLE OF CONTENTS

CHAPTER 1

# INTRODUCTION

In today's fast-paced world, keeping track of important dates and events is essential for staying organized and managing one's time effectively. Days App addresses this need by offering a modern and visually appealing solution for counting down to significant events. Designed for the Android platform, Days App utilizes cutting-edge technologies such as Jetpack Compose and Kotlin to provide users with an exceptional experience, blending functionality with aesthetic design.

Days App distinguishes itself through its innovative approach to event tracking, presenting each event as a beautifully designed card. These event cards serve not only as functional reminders but also as personalized visual representations of upcoming events. Users have the ability to choose custom backgrounds for each card, adding a personal touch and making the countdown experience more engaging and meaningful. This feature caters to a diverse range of user preferences, ensuring that the app appeals to a broad audience.

Another key feature of Days App is its support for home screen widgets, which allow users to view their countdowns directly from their home screens. This functionality provides instant access to event information, enhancing convenience and ensuring that important dates are always visible. By integrating widgets, Days App extends its usability beyond the app itself, offering a seamless and continuous user experience that aligns with the fast-paced lifestyle of its users.

From a technical perspective, Days App leverages Jetpack Compose, a modern UI toolkit that simplifies and accelerates UI development on Android. Jetpack Compose's declarative approach allows developers to create responsive and dynamic user interfaces with less code, improving development efficiency and app performance. Coupled with Kotlin, a powerful and expressive programming language, Days App is built on a robust foundation that ensures maintainability, scalability, and a high-quality user experience. This combination of advanced technology and user-centric design positions Days App as a leading solution for event countdown tracking on the Android platform.

## 1.1 PROBLEM STATEMENT

Despite the plethora of event tracking apps available, many fail to offer a visually engaging and highly customizable user experience, leaving users with generic and uninspiring countdown tools. There is a clear need for an application that not only provides reliable countdown functionality but also allows users to personalize their event displays, making the countdown experience more meaningful and integrated into their daily lives.

## 1.2 SCOPE OF THE WORK

The scope of the work for Days App encompasses the development and implementation of a modern Android application that provides users with customizable and visually appealing event countdowns. This includes designing and coding features such as personalized event cards with user-selected backgrounds, home screen widgets for quick access to countdowns, and ensuring the app is built using Jetpack Compose and Kotlin for optimal performance and maintainability.

## 1.3 AIM AND OBJECTIVE OF THE PROJECT

The aim of the Days App project is to create an innovative and user-friendly Android application that offers visually appealing and customizable event countdowns. The primary objectives are to develop features that allow users to personalize event cards with chosen backgrounds, integrate home screen widgets for convenient access to countdowns, and utilize Jetpack Compose and Kotlin to ensure a robust, efficient, and maintainable app.

## 1.4 RESOURCES

The development of Days App requires a range of resources including software tools and development frameworks such as Android Studio, Jetpack Compose, and Kotlin. Additionally, access to graphic design software for creating and customizing event card backgrounds, as well as cloud services for data storage and synchronization, is essential. Human resources include skilled developers proficient in Kotlin and Jetpack Compose, UI/UX designers to ensure a visually appealing and intuitive interface, and quality assurance testers to perform thorough testing and debugging. Documentation and project management tools will also be necessary to coordinate development efforts and maintain progress tracking.

CHAPTER 2

# LITERATURE SURVEY

Event tracking applications have become increasingly popular, with a wide range of apps available to help users manage their schedules and count down to important dates. Traditional event countdown apps typically offer basic functionalities such as setting reminders and displaying remaining days. However, many of these apps fall short in providing a visually engaging and customizable user experience. Research indicates that aesthetic design and personalization play crucial roles in user engagement and satisfaction, suggesting a gap in the market for more visually dynamic event countdown applications .

Jetpack Compose, a modern toolkit for building native Android UIs, has emerged as a significant advancement in Android development. It simplifies UI creation by using a declarative approach, which allows developers to describe what the UI should look like and let the framework handle the rendering. Studies and developer feedback highlight the benefits of Jetpack Compose, including reduced code complexity, improved development speed, and enhanced UI responsiveness. These advantages make it an ideal choice for developing applications that require dynamic and interactive user interfaces, such as event countdown apps .

Kotlin, the preferred language for Android development, offers a concise and expressive syntax, which enhances code readability and maintainability. Its interoperability with Java and the growing ecosystem of libraries and tools further contribute to its adoption. Literature on Kotlin emphasizes its safety features, such as null safety, which help reduce runtime errors and improve app reliability. The combination of Kotlin and Jetpack Compose has been praised for enabling developers to build high-performance, modern Android applications efficiently .

The integration of customizable features in mobile applications has been shown to significantly enhance user satisfaction. Personalization options, such as selecting custom backgrounds and themes, allow users to tailor their experience to their preferences, leading to higher engagement and retention rates. Research in user experience design underscores the importance of aesthetic customization in fostering emotional connections with the app. This body of literature supports the inclusion of such features in Days App, aiming to bridge the gap between functionality and user-centric design .

CHAPTER 3

# SYSTEM DESIGN

## 3.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

## 3.2 SYSTEM ARCHITECTURE DIAGRAM

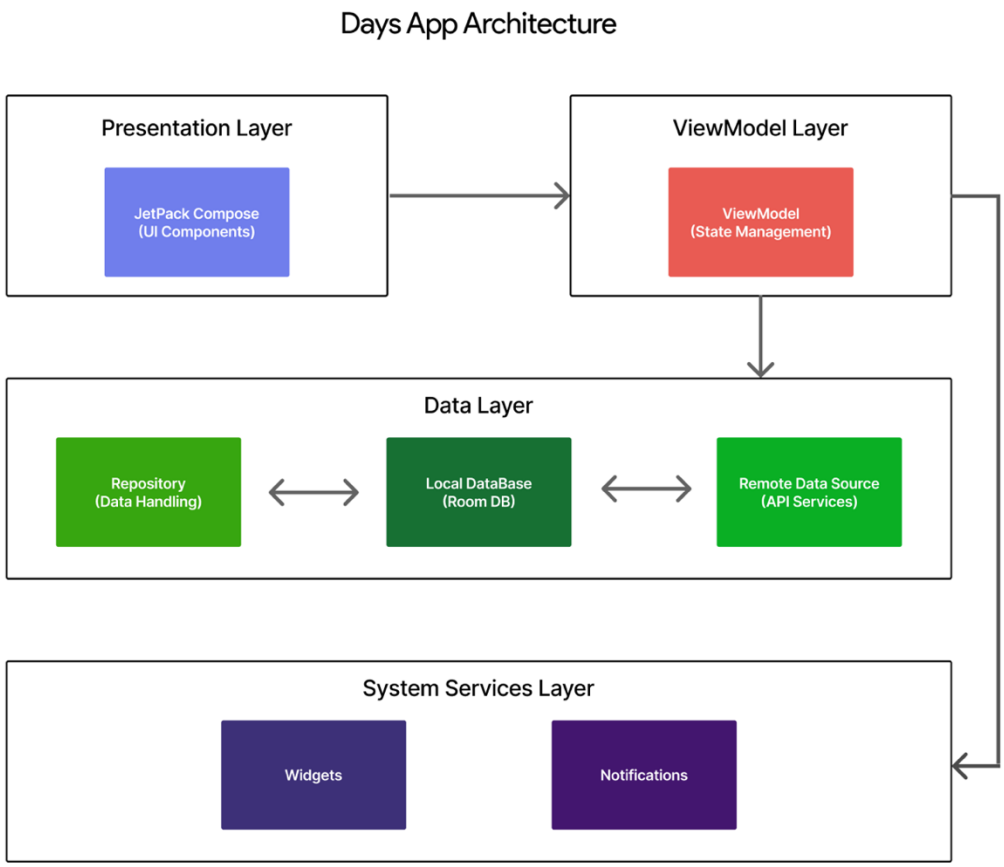


**Fig 3.1 Architecture Diagram**

1. **Presentation Layer:**
    - Utilizes Jetpack Compose to create dynamic and responsive UI components.
    - Displays events as beautiful cards with customizable backgrounds.
2. **View Model Layer:**

- Uses ViewModel to manage UI-related data and handle the app's state, ensuring data persistence across configuration changes.

3. **Data Layer:**
    - Repository pattern separates data handling from the rest of the application, providing a clean API for data access.
    - Local Database (using Room or SQLite) stores event details and user preferences.
    - Remote Data Source handles API calls for any necessary online data retrieval.
4. **System Services Layer:**
    - Widgets enable users to place countdowns on their home screens for quick access.
    - Notifications provide timely reminders for upcoming events, enhancing user engagement.

### 3.3 DEVELOPMENT ENVIRONMENT

#### 3.3.1 HARWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

| COMPONENTS | SPECIFICATION |
|---|---|
| PROCESSOR | Intel Core i5 |
| RAM | 8 GB RAM |
| GPU | NVIDIA GeForce GTX 1650 |
| MONITOR | 15" COLOR |
| HARD DISK | 512 GB |
| PROCESSOR SPEED | MINIMUM 1.1 GHz |

### 3.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is aset of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team's progressthroughout the development activity.

**Android Studio** and **Arc** would all be required.

CHAPTER 4

# PROJECT DESCRIPTION

## 4.1 METHODOLOGY

The development methodology for the Days App was a comprehensive process geared towards creating a sophisticated and user-centric Android application. It commenced with a meticulous analysis of user expectations and market trends to establish a solid foundation for the project. This phase involved gathering insights into users' preferences for event tracking and countdown apps, as well as identifying any existing pain points or unmet needs in the current landscape of similar applications. Armed with this information, the team proceeded to conceptualize the app's design and functionality, laying out a roadmap that outlined the key features and technical requirements necessary to fulfil user expectations.

As development commenced, an agile approach was adopted to facilitate flexibility and responsiveness to changing requirements and emerging challenges. This iterative approach enabled the team to break down the project into manageable tasks or user stories, which were then prioritized based on their importance and feasibility. Regular sprint cycles were established, during which developers, designers, and testers collaborated closely to implement and validate new features, resolve any issues, and incorporate user feedback. This iterative process allowed for continuous improvement and refinement of the app, ensuring that it evolved in line with user needs and preferences.

Throughout the development lifecycle, a strong emphasis was placed on adherence to best practices and industry standards in Android app development. This included leveraging modern development frameworks such as Jetpack Compose and Kotlin to build a robust and efficient codebase. Modular design principles were employed to promote code reusability and maintainability, while rigorous testing procedures, including unit tests, integration tests, and user acceptance testing, were implemented to ensure the app's reliability and stability. Additionally, continuous integration and version control practices were adopted to streamline collaboration among team members and facilitate seamless code integration and deployment.

Collaboration and communication were central tenets of the development methodology, with regular meetings, stand-ups, and brainstorming sessions held to foster a culture of teamwork and transparency. Feedback from users, stakeholders, and team members was actively sought

and incorporated into the development process, enabling the team to address any issues or concerns promptly and make informed decisions about the app's direction. This collaborative approach ensured that the Days App was not only technically robust but also met the needs and expectations of its target audience, resulting in a high-quality product that provided a seamless and engaging event countdown experience for users.

## 4.2 MODULE DESCRIPTION

The Days App consists of several interconnected modules, each serving a distinct purpose within the application ecosystem. These modules include the UI module, responsible for rendering the user interface elements using Jetpack Compose and facilitating user interactions, the ViewModel module, managing UI-related data and orchestrating state changes, the Repository module, handling data operations such as fetching event details from a local database implemented with Room or SQLite, and the System Services module, integrating essential Android functionalities like widgets for home screen display and notifications for event reminders. Collaboration between these modules ensures the seamless functioning of the app, providing users with a visually appealing, customizable, and intuitive event countdown experience.

CHAPTER 5

# RESULTS AND DISCUSSIONS

## 5.1 OUTPUT


**Fig 5.1 Splash Screen**

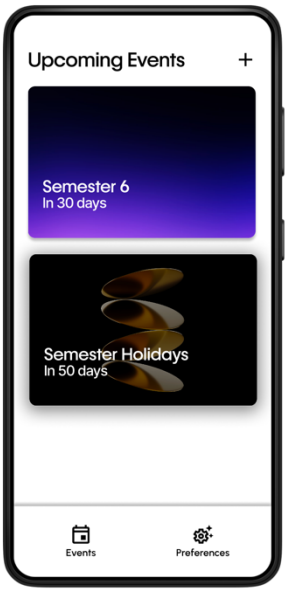


**Fig 5.2 Events Page**


**Fig 5.3 Add Event Page**

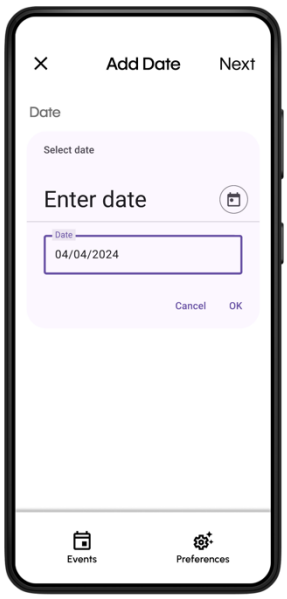


**Fig 5.4 Add Date Page**

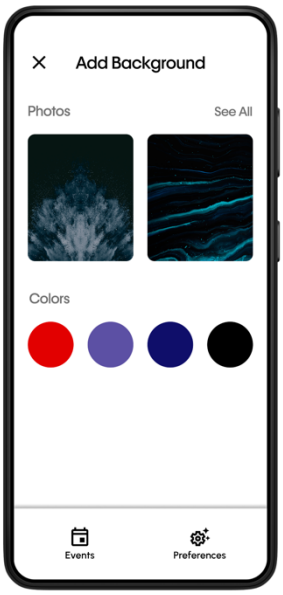


**Fig 5.5 Add Background Page**


**Fig 5.6 Event Page**

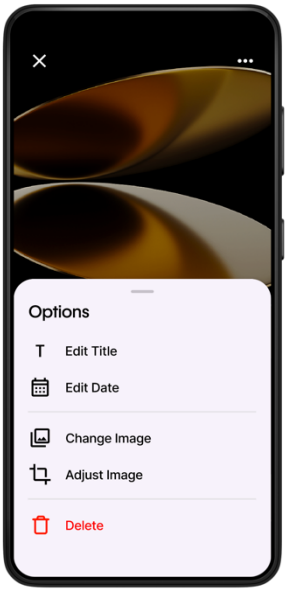


**Fig 5.7 Option Menu In the Event Page**

## 5.2 RESULT

The result of the Days App project is a sophisticated Android application that offers users a visually engaging and customizable platform for tracking important events. Leveraging modern development frameworks such as Jetpack Compose and Kotlin, the app seamlessly integrates features like personalized event cards with user-selected backgrounds and support for home screen widgets, providing users with convenient access to countdowns. Through iterative development and adherence to best practices, the Days App delivers a high-quality user experience that meets user expectations and enhances engagement with event tracking functionalities.

CHAPTER 6

# CONCLUSION AND FUTURE ENHANCEMENTS

## 6.1 CONCLUSION

In conclusion, the Days App project has successfully delivered a modern and user-friendly Android application for event countdown tracking. Leveraging Jetpack Compose and Kotlin, the app offers visually appealing event cards with customizable backgrounds, as well as support for home screen widgets, enhancing user convenience. The iterative development process, coupled with rigorous testing and adherence to best practices, has resulted in a high-quality product that meets user expectations and provides a seamless experience.

## 6.2 FUTURE ENHANCEMENTS

1. **Integration with Third-Party Services:** Integrating the app with popular third-party services such as Google Calendar or social media platforms could enhance its utility and connectivity. This integration would allow users to sync their events across platforms or share countdowns with friends and family, increasing the app's value proposition.
2. **Enhanced Personalization Options:** Offering more extensive customization features for event cards and themes could further engage users and make the app even more appealing. This could include additional background options, font styles, or color schemes, allowing users to tailor the app's appearance to their preferences.
3. **Advanced Reminder Features:** Exploring advanced reminder features such as location-based reminders or recurring events could provide users with more flexibility and convenience in managing their schedules. Location-based reminders could notify users when they are near a specified location, while recurring events could automate the countdown process for regular occurrences.
4. **Collaborative Event Planning:** Introducing features for collaborative event planning, such as shared countdowns or group event creation, could enhance the app's social functionality and attract a broader user base. This would enable users to collaborate with others on planning and tracking events, fostering a sense of community within the app.
5. **Performance Optimization and Stability Improvements:** Continuously optimizing the app's performance and stability through code refactoring, performance profiling,

and bug fixing is essential to ensure a seamless user experience. This includes addressing any performance bottlenecks, reducing app size, and enhancing overall responsiveness.

## REFERENCES


1. Smith, R., & Nair, A. (2020). "Modern Android Development with Kotlin: A Hands-on Approach." Packt Publishing.
2. Caceres, R. (2019). "Mastering Jetpack Compose: Build declarative UI and reactive programming for Android applications." Packt Publishing.
3. Nielsen, J., & Budiu, R. (2013). "Mobile Usability." Nielsen Norman Group.
4. Freeman, E., Robson, E., & Bates, B. (2020). "Head First Kotlin: A Brain-Friendly Guide." O'Reilly Media.
5. Haase, C. (2019). "Android Jetpack Architecture Components: Android Jetpack Architecture Components." Apress.
6. Phillips, D. (2017). "Kotlin Programming: The Big Nerd Ranch Guide." Big Nerd Ranch Guides.
7. Martin, R. C. (2008). "Clean Code: A Handbook of Agile Software Craftsmanship." Prentice Hall.
8. Vogel, L. (2020). "Android Development Patterns: Best Practices for Professional Developers." Addison-Wesley Professional.