Employee Task Assignment Android Application

K. Manoj[1], Krishna Raj[2], G .Chaitanya Bharathi[3]

UG Scholar, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana[1,2]

Assistant Professor, Guru Nanak Institutions Technical Campus, Hyderabad, Telangana [3]

ABSTRACT

An Android-based Employee Tracking System was created to monitor all employee activities. This system tracks scheduling and time-off requests, which are crucial for managers to determine when employees are expected to be in the office or working remotely. The system provides a convenient way for managers to oversee their team's performance via mobile phones. It was developed using the JAVA programming language, with data being stored in an SQLite database. The design followed an Object-Oriented Analysis and Design methodology, incorporating a structured, iterative approach. Data was gathered through document reviews and field methods, with analytical tools like bar charts applied to analyze the findings. The implemented system successfully improved productivity, reduced costs, and enabled instant access to employee attendance records**.**

INTRODUCTION

This Android application acts as a unified platform for managers to delegate tasks, prioritize them, and set deadlines, while providing employees with an easy way to view, monitor, and update their tasks. With real-time notifications, employees are promptly alerted about new assignments or changes, reducing delays and promoting responsibility. Moreover, the app incorporates a feature that allows managers to track the progress of tasks in real time, facilitating more effective planning and resource distribution.

OBJECTIVE

To create a user-friendly and efficient mobile application that simplifies task assignment, tracking, and management for employees. This app aims to optimize task delegation, boost communication, and increase productivity by allowing managers to assign tasks, set deadlines, and monitor progress, while giving employees a clear overview of their tasks and timelines.

SCOPE OF THE WORK

In this work, an Android application focused on employee task assignment includes the development of a platform that enables managers to set deadlines ,assign task and track the progress of work in real-time. The application will allow employees to view and update their tasks, ensuring their responsibilities and timelines. Features will include task prioritization, notifications for updates or new assignments, and a progress-tracking system for managers. The application will aim to enhance productivity, streamline communication, and improve task management efficiency for both managers and employees

RELATED WORK

Research Papers & Academic Work:

Numerous academic studies focus on task assignment and management systems, offering insights into various approaches for task allocation, human involvement, and optimization of performance. Some key research topics in this area include:

Task Assignment in Multi-Agent Systems: Research on algorithms that can efficiently allocate tasks to employees or agents, using criteria such as skill matching and workload distribution.

Mobile Task Management Systems: Studies focusing on mobile-based systems designed for task management, particularly in settings where teams are distributed or employees are working remotely. These systems emphasize the importance of effective task assignment and real-time tracking.

Human Resource Management Systems (HRMS): Some research explores how task assignment can be integrated into larger HRMS frameworks that include employee performance tracking, attendance, and payroll management.

Machine Learning for Task Optimization: Research exploring the application of machine learning to automate task scheduling, prioritize tasks based on past data, and optimize task allocation based on employee performance and availability.

Key Features for Employee Task Assignment Android App:

When developing an employee task assignment Android app, several essential features should be considered to ensure it meets organizational and employee needs. These features are:

Task Creation and Assignment: Managers should be able to create tasks, assign them to specific employees, set deadlines, and monitor the progress of task completion.

Notifications and Reminders: Push notifications can be used to alert employees about newly assigned tasks, upcoming deadlines, and any changes to their schedules.

Team Collaboration: Features that promote communication among team members, such as chat tools, file sharing, and task-related updates.

Time Tracking: Employees should be able to log the hours spent on tasks, which helps both in tracking productivity and for billing or payroll purposes.

Skill Matching: The app should incorporate algorithms that match tasks with the most qualified employees based on skills and experience.

Progress Monitoring and Reporting: Dashboards and reporting tools for managers to track task completion, monitor employee performance, and assess overall team progress.

Integration: The app should integrate seamlessly with other tools like HR management systems, calendars, and communication platforms to enhance overall efficiency.

Technologies Used:

Building an employee task assignment app involves several technical components that enable smooth functioning:

Back end: Technologies like Firebase, Node.js, or Django for back end development, ensuring real-time updates and synchronization. Databases such as MySQL or MongoDB store task data and user information.

Front End: The app's interface should be built using Android Studio, with Kotlin or Java as the programming languages for a native Android experience.

Push Notifications: Firebase Cloud Messaging (FCM) or One Signal can be employed for sending task reminders and notifications to employees' mobile devices.

Cloud Services: Cloud platforms like AWS or Google Cloud are used to host the back end, ensuring scalability, security, and reliability.

Data Security: Role-based access control (RBAC) should be implemented to secure employee data, restricting access to sensitive task information based on employee roles and responsibilities

METHODOLOGY-ALGORITHM USED

Object-Oriented Programming (OOP)

In the development of Android applications, object-oriented programming (OOP) is essential for structuring the app effectively. With programming languages like Java or Kotlin, OOP principles are employed to create scale able and maintainable code. The fundamental concepts of OOP include:

Encapsulation: This involves hiding the complexity of task management functions from the user, ensuring that the app's user interface (UI) remains clean and intuitive.

Inheritance: By allowing reusable components (e.g., task templates or employee profiles), inheritance facilitates the creation of customization elements without the need to modify the entire code base.

Polymorphism: This allows the app to handle different task types (such as admin tasks or employee tasks) using a single, unified method for assignment.

By leveraging these OOP principles, the app becomes easier to maintain and extend, allowing for the addition of new features without disrupting existing functionality

Model-View-ViewModel (MVVM) Architecture:

The Model-View-View Model **(MVVM)** design pattern is commonly utilized in Android app development. It divides the app into three core components:

Model: Represents the data layer, containing information about tasks, employees, and task assignments.

View: The user interface (UI) that displays relevant information like tasks, deadlines, and updates.

View Model: Functions as the bridge between the Model and the View, managing data and ensuring the UI is updated with any changes.

MVVM helps maintain a clear separation of concerns, making the app easier to develop, test, and maintain. This structure is particularly valuable for an employee task assignment app where tasks and data are updated regularly, requiring robust synchronization between components.

User-Centered Design (UCD)

User-Centered Design (UCD) focuses on developing the app by prioritizing the needs and behaviors of the users. In the context of an employee task assignment app, this methodology involves several key practices:

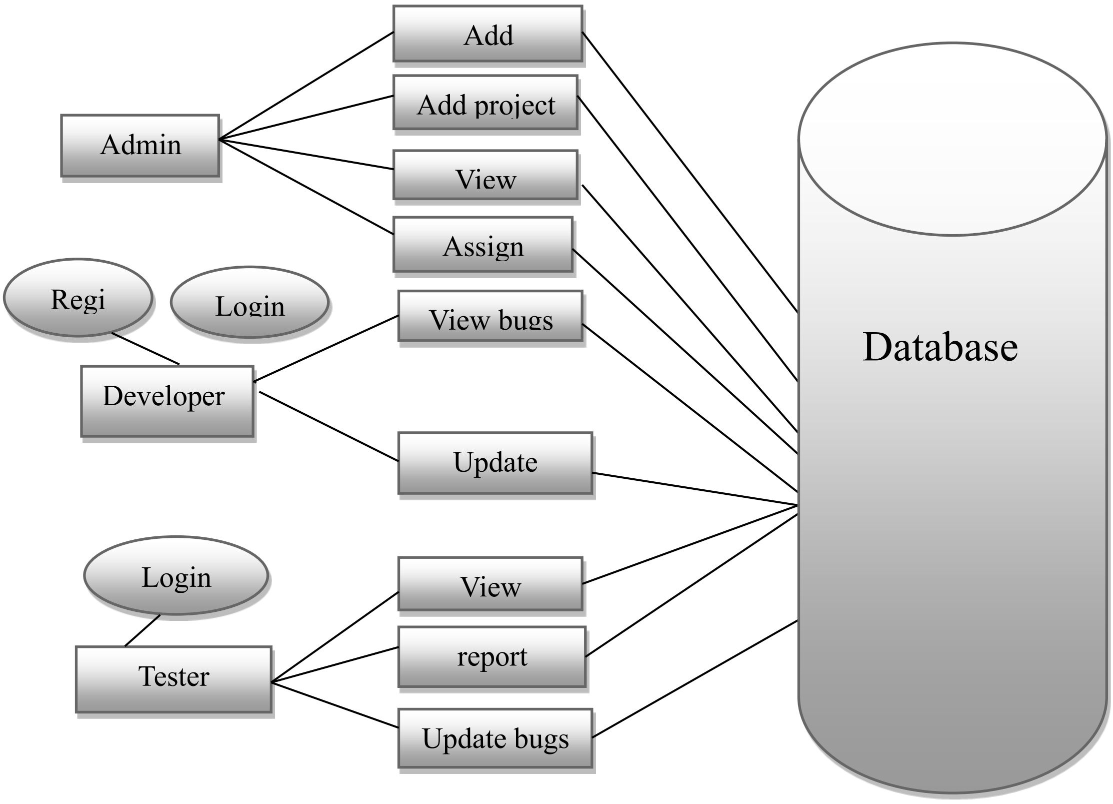
User Research:Gathering insights about the requirements of managers, employees, and other stakeholders who will interact with the app. This may involve surveys, interviews, and direct observations.

Personas: Creating detailed user profiles (such as a team manager, junior employee, or HR administrator) to guide design decisions and ensure the app serves the unique needs of different users.

Prototyping and Testing: Early versions of the app are developed and tested with real users. Their feedback is then used to refine the design, ensuring the final product is user-friendly and meets functional requirements.

By emphasizing the user's experience, this approach ensures that the app is intuitive, efficient, and tailored to the needs of its users.

SYSTEM ARCHITECTURE



EXPLANATION OF SYSTEM ARCHITECTURE

The system architecture shown in the image represents an Employee Task Assignment Android Application. Here is the explanation

1.Actors in the System:

Admin: Responsible for managing employees, projects, and tasks in the system.

Developer: Assigned specific tasks (bugs) by the admin and updates their status.

Tester: Reports bugs or issues in projects and updates the bugs once resolved.

2. Functionalities:

***Admin Functions:**

Add Employee: Allows the admin to add employee details to the system.

Add Project: Enables the admin to register new projects in the database.

View Employee: Lets the admin view employee details.

Assign Bugs: The admin assigns specific bugs or tasks to developers.

Database Access: All admin actions are stored and retrieved from the database.

***Developer Functions:**

Register/Login: Developers must register and log in to the application.

View Bugs: Developers can view the bugs assigned to them.

Report: Developers report their progress or any issues in the project.

Update Status: Developers update the status of the bugs or tasks (e.g., fixed, in progress).

Database Access: All developer actions interact with the database for updates and retrieval.

***Tester Functions:**

Login: Testers log in to the application to perform their tasks.

View Projects: Testers can view the project details to identify bugs.

Report Bugs: Testers report bugs they encounter during testing.

Update Bugs: Testers update bug details once resolved or reassign them.

Database Access: Tester actions are also recorded and fetched from the database.

3. Database:

• The central component where all the data related to employees, projects, bugs, reports, and updates are stored.

• Facilitates seamless communication between Admin, Developers, and Testers.

4. Data Flow:

• Admin interacts with the database for CRUD (Create, Read, Update, Delete) operations on employees, projects, and bug assignments.

• Developers and Testers log in, retrieve their respective tasks or projects, and update the database based on their activities (status updates, bug reports, etc.).

This architecture provides a collaborative platform for effective task assignment, bug tracking, and resolution between Admins, Developers, and Testers.

CONCLUSION

The Employee Task Assignment Android Application provides a comprehensive solution for managing tasks, tracking progress, and ensuring smooth communication among admins, developers, and testers. By incorporating a user-friendly interface and efficient backend processes, the system streamlines task assignments, bug reporting, and status updates, ultimately enhancing productivity and ensuring that project timelines are met.

With well-defined roles for each actor—Admin, Developer, and Tester—the system ensures clear responsibilities, task management, and accountability. The use of a central database facilitates seamless data synchronization, ensuring that all changes and updates are reflected in real-time. The application also provides key features such as task status updates, bug reporting, and progress tracking, which improve transparency and communication across the team.

In conclusion, the Employee Task Assignment Android Application is an effective tool for task and project management in a collaborative environment. It simplifies the process of task allocation, bug tracking, and resolution, leading to improved efficiency and better project outcomes for all team members.

REFERENCE

[1] J. Anvik, L. Hiew, and G. C. Murphy. Who should fix this bug?

In ICSE’06: Proceedings of the 28th International Conference on

Software engineering, pages 361–370, 2006.

[2] J. Aranda and G. Venolia. The secret life of bugs: Going past the

errors and omissions in software repositories. In ICSE’09: Proceedings of the 31st International Conference on Software Engineering

(to appear), 2009.

[3] S. Artzi, S. Kim, and M. D. Ernst. Recrash: Making software failures reproducible by preserving object states. In ECOOP’08: Proceedings of the 22nd European Object-Oriented Programming Conference, pages 542–565, 2008.

[4] N. Bettenburg, S. Just, A. Schroter, C. Weiss, R. Premraj, and ¨

T. Zimmermann. What makes a good bug report? In FSE’08:

Proceedings of the 16th International Symposium on Foundations of Software Engineering, pages 308–318, November 2008.

[5] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim. Duplicate

bug reports considered harmful ... really? In ICSM'08: Proceedings of the 24th IEEE International Conference on Software Maintenance, pages 337–345, 2008.

[6] S. Breu, J. Sillito, R. Premraj, and T. Zimmermann. Frequently

asked questions in bug reports. Technical report, University of Calgary, March 2009.

[7] P. Fritzson, T. Gyimothy, M. Kamkar, and N. Shahmehri. Generalized algorithmic debugging and testing. In PLDI'91: Proceedings

of the ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 317–326, 1991.

[8] P. Hooimeijer and W. Weimer. Modeling bug report quality. In

ASE'07: Proceedings of the 22nd International Conference on Automated Software Engineering, pages 34–43, 2007.

[9] S. Just, R. Premraj, and T. Zimmermann. Towards the next generation of bug tracking systems. In VL/HCC'08: Proceedings of the

2008 IEEE Symposium on Visual Languages and Human-Centric

Computing, pages 82–85, September 2008.

[10] A. J. Ko and B. A. Myers. Debugging reinvented: asking and answering why and why not questions about program behavior. In

ICSE'08: Proceedings of the International Conference on Software

Engineering, pages 301–310, 2008.

[11] B. Liblit, M. Naik, A. X. Zheng, A. Aiken, and M. I. Jordan. Scalable statistical bug isolation. In PLDI'05: Proceedings of the 2005

ACM SIGPLAN Conference on Programming Language Design and

Implementation, pages 15–26, 2005.