JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

MOBILE APPLICATION FOR APARTMENT MAINTENANCE

Mrs.D.Saritha[*1], Ch.Tarak[2], G.Sri Chaithanya Reddy[3], J.V Sai Agasthya[4], E.Tharun Kumar[5], G.V.S Ajith Kumar[6]

Assistant Professor[*1], UG Scholars[2,3,4,5&6]
Department of Computer Science and Engineering[1,2,3,4,5&6], Narayana Engineering College, Nellore, Andhra Pradesh, India.

Abstract: *The Apartment Maintenance mobile application designed to improve communication and management in apartment communities. This paper presents the android mobile application that streamlines tasks such as managing maintenance bills, reporting issues, and controlling visitor access. For tenants/owners, it provides features to pay and track maintenance charges, contact handyman services, and report problems easily. For administrators, it simplifies generating charges, scheduling reminders for meetings and events, and managing visitor access history. The gatekeeper function enhances security by managing visitor entries effectively. Additionally, this application supports community communication by facilitating announcements, meetings, and events. This application improves the overall living experience and management efficiency in apartment communities.*

Key Words: *Mobile Application, Android, Community Apartment, Gated Community, Gate Pass, Interaction, Visitor Access Management.*

1. INTRODUCTION

Nowadays, more people are living in community apartments and gated communities and in today's living, managing community apartments and gated communities efficiently is essential. Good communication between tenants/owners and admin, strong security, and easy visitor access are the key to a well-run community. This android mobile application is designed to meet these needs, providing a platform that makes life easier for tenants/owners and admin of the community. The Apartment Maintenance mobile application aims to help tenants/owners handle maintenance charge bills, helps in reminding community organized meetings, events and announcements and report issues easily. It generates maintenance bills and has a directory of handyman service providers for quick access to necessary services. The complaints feature allows tenants/owners to report problems easily, ensuring they are addressed promptly. A standout feature of our application is its gatekeeper function, which manages visitor access and enhances security and ensures smooth visitor management.

Built using Java and Kotlin, and firebase database for backend storage, the Apartment Maintenance application is developed in Android Studio to ensure compatibility with a wide range of Android devices. The development process follows a systematic approach, including requirements gathering, system design, module development, integration, and rigorous testing to ensure reliability and performance.

This paper details the existing system, proposed system and its design, results and conclusion of the Apartment Maintenance application, its key features and the benefits it brings to apartment management. By using this technology, this application aims to create a more secure, efficient, and connected living experience for tenants/owners in community apartments and gated communities. The paper also explores several potential enhancements to further improve and include more functionality and user experience will extend the application's capabilities.

1.1 Problem statement

People living in apartments face challenges in managing visitor access, maintenance charge bills, handyman services, complaint reporting and meeting, event reminders. Traditional communication methods, like notice boards or word-of-mouth, are often slow and unreliable, causing delays in important updates. Security is compromised by manual visitor logs and gate keeping. Managing bills and maintenance requests manually is time-consuming and error-prone. Reporting issues is difficult, leading to unresolved problems. Finding reliable handyman services quickly is also a common struggle.

1.2 Objective of the paper

This paper aims to enhance communication between tenants/owners and admin in the communities and to make it easier to provide security and giving access to visitors for entering into communities. The objective is to streamline a reliable mobile application for the people living in community apartment and gated communities, and to provide user-friendly interface, secure user data, visitor access management, enhanced communication, strengthen security, issue reporting, access to handyman service providers.

2. EXISTING WORK

The current management system in many community apartments and gated communities is often a combination of manual and semi-automated processes. These processes include handling maintenance requests, managing visitor access, communicating with tenants/owners, and organizing community events. While these methods have been in place for a long time, they frequently lead to inefficiencies, communication breakdowns, and security vulnerabilities.

Traditional Information Exchange System

The traditional information exchange includes some communities using physical notice boards to share important information such as scheduled meetings, events and announcements and some communities use emails or group messaging apps to communicate with tenants/owners. While this can be more immediate than notice boards, it can still result in missed messages and fragmented communication.

Maintenance Charge System

Maintenance charge bills are typically generated manually and then distributed to tenants or owners, a process that is prone to errors and inefficiencies. These manual methods can result in inaccuracies in billing, leading to confusion and disputes, as well as significant delays in the collection of payments.

Visitor Access Management System

Gatekeepers often rely on paper logs to record visitor details, maintaining a manual record of each visitor's information and their entry times. These logs typically involve the visitor signing in with their name, contact information, purpose of visit, and the resident they are visiting. Identification is sometimes checked and noted down. This system is often inefficient, error-prone, and less secure compared to digital solutions, which can provide more accuracy, security, and convenience.

Issue Reporting System

When issues are reported directly, the handling process often lacks consistency and systematic tracking. Reporting issues verbally or through informal notes can result in miscommunication or misunderstanding of the problem, leading to inconsistent handling, lack of accountability, and inefficient problem resolution, ultimately impacting the overall management and satisfaction within the community.

Limitations of Existing System

The following are the limitations for the existing work:

- Time consuming process and requires manual work.
- Important updates may not reach all tenants/owners in a timely manner.

- Physical notice boards and paper notices are not accessible to tenants/owners who are away from the premises, leading to a lack of real-time information.
- Manual visitor logs are prone to human error and manipulation, compromising the accuracy and reliability of visitor records.
- Manual generation and distribution of maintenance bills leads to errors and delays, leading to potential disputes and late payments.
- Tenants/owners find difficulty in accessing handyman service providers due to lack of connections.

3. PROPOSED SYSTEM

The proposed work is an Android mobile application designed to enhance communication, security, and management within community apartments and gated communities. The application aims to streamline processes, provide real-time updates, and improve overall tenant/owner and admin experiences. The proposed work involves several key phases, each focusing on critical aspects of the system's functionality and performance.

a. Requirements Analysis and System Design

Objectives: Clearly define the requirements and scope of the application, ensuring all needs are met.

Activities:

- Conduct requirement gathering through references taken from the Google.
- Manually draw screen designs for the user interface and system architecture for better understanding.
- Develop use case diagrams, class diagrams, and sequence diagrams to outline system interactions and workflows.

b. Development of Core Modules

Admin Module:

Features: Secure login and registration, generation of maintenance charge bills, creating login credentials for gate keeper and newly joining tenants/owners, adding directory of handyman service providers, scheduling timely-updates of meetings, events and announcement, receive complaints generated by tenants/owners and viewing gate passes.

Tenant/Owner Module:

Features: Viewing maintenance charge bills and its status(paid/pending), view directory and interactions scheduled by admin and report issues through complaints, generate gate pass and access to his created gate passes history.

Gate keeper Module:

Features: Scanning QR code from thegate pass provided to the visitor by a tenant/owner living in that building and other operation is verifying gate pass code from the gate pass provided to the visitor.

c. Integration and Data Management

Objectives: Ensure seamless interaction and data flow between the admin, tenant/owner and gate keeper modules.

Activities:

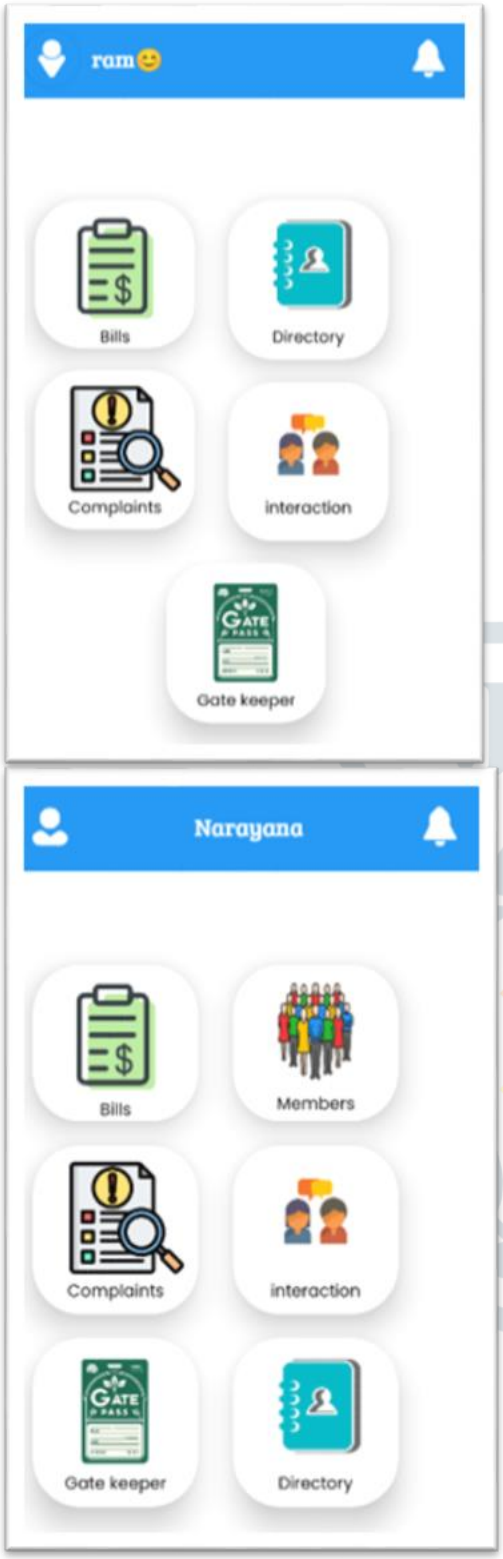
- Develop and implement API's for communication between the frontend and backend.
- Ensure data consistency and integrity through proper validation and error handling.

d. User-friendly Interface and Experience Enhancement

Objectives: Create an intuitive, responsive, and user-friendly interface for all users.

Activities:

- Design and implement responsive user interfaces for admin, tenant/owner and gate keeper.
- Ensure the application is accessible and easy to navigate.
- Incorporate real-time notifications for new bill, new interaction and new complaint.



e. Testing and Validation

Objectives: Validate the functionality, performance, and security of the application.

Activities:

- Conduct unit testing for validating individual components.
- Perform integration testing to ensure different parts of the system work together.
- Perform black box testing to validate the system's functionality. Conduct white box testing to verify the internal logic, code structure, and data flow of the application.

4. EXPERIMENTAL RESULTS

The admin menu screen is a part of the mobile application that provides the admin of the building to access his intended functionalities. Here, in this application the admin menu consists of six different options with their respective features. The following are included in the admin menu screen:

- **Profile Management:** Allows the admin to update their personal and building profile details.
- **Bills:** Access to create, view, and manage maintenance charge bills for tenants/owners.
- **Members:** Access to create tenant/owner login credentials.
- **Complaints:** For viewing and addressing complaints submitted by tenants/owners.
- **Interaction:** Access to create and send meetings, events and announcements to the community.
- **Gate keeper:** Access to view gate passes created by tenant and create gate keeper login credentials.
- **Directory:** Access to the directory of handyman service providers and update their contact information.

Fig: 4.1 Admin menu screen

The tenant/owner menu screen is a part of the mobile application that provides the tenant/owner living in the building to access his intended functionalities. Here, in this application the tenant/owner menu consists of five different options with their respective features. The following are included in tenant/owner menu screen.

- **Profile Management:** Allows the tenant/owner to view their personal details.
- **Bills:** Access to view and update maintenance charge bills.
- **Complaints:** For creating, viewing and addressing complaints submitted.
- **Interaction:** Access to view and being notified of meetings, events and announcements to the community.
- **Gate keeper:** Access to create gate passes for visitors.

- **Directory:** Access to view the directory of handyman service providers.

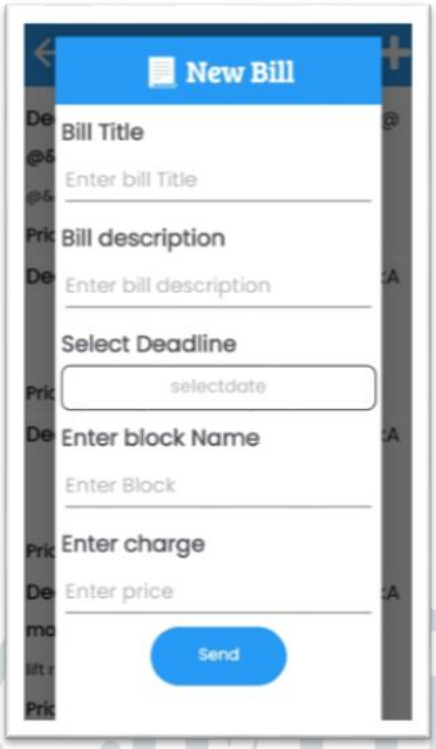
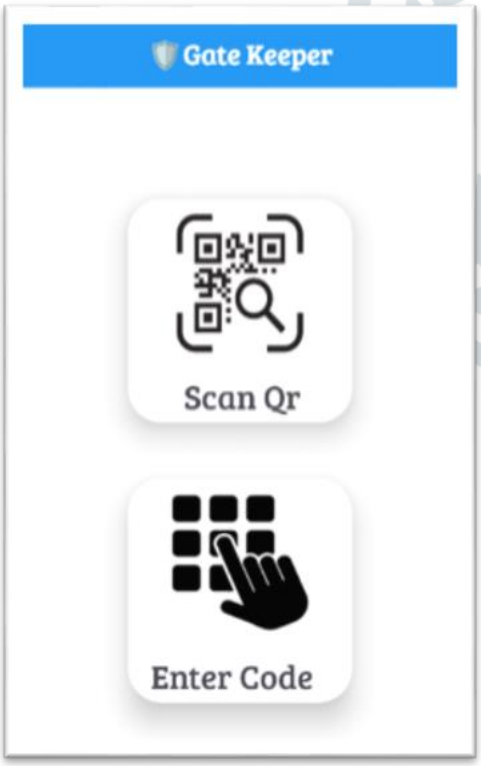


Fig: 4.2 Tenant/Owner menu screen



The gate keeper menu screen is a part of the mobile application that provides the gate keeper of the building to access his intended functionalities. Here, in this application the gate keeper menu consists of only two options with their respective features. The following are included in gate keeper menu screen.

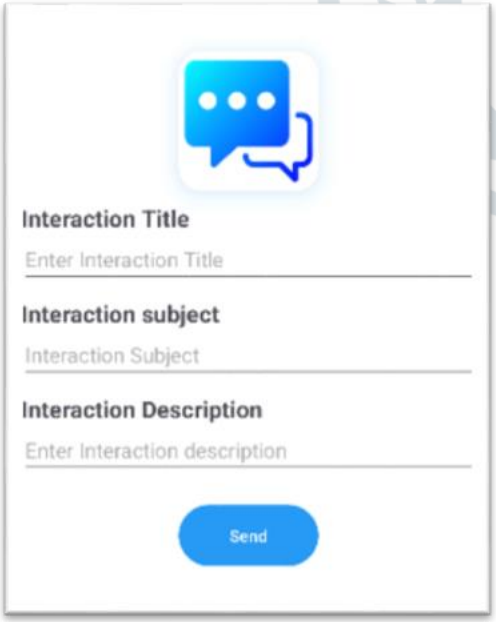
- **Scan QR:** Allows to scan the QR code provided by gate pass from the visitor and to view visitor details for entry purpose.
- **Enter code:** Allows to enter the gate pass code provided by gate pass from the visitor and view visitor details for entry purpose.

Fig: 4.3. Gate keeper menu screen

The new bill generation screen is a part of the mobile application that provides the admin of the building to access bill generation functionality and the generated bill is displayed on tenant/owner side. Here, in this application fills the admin in the following details to generate a new bill:

- **Bill title:** Name of the bill.
- **Bill description:** Description of the bill to be more specific.
- **Select deadline:** Date before the bill needs to be paid.
- **Enter block name:** Name of the block.
- **Enter charge:** Amount to be paid.

Fig: 4.4. New bill generation screen

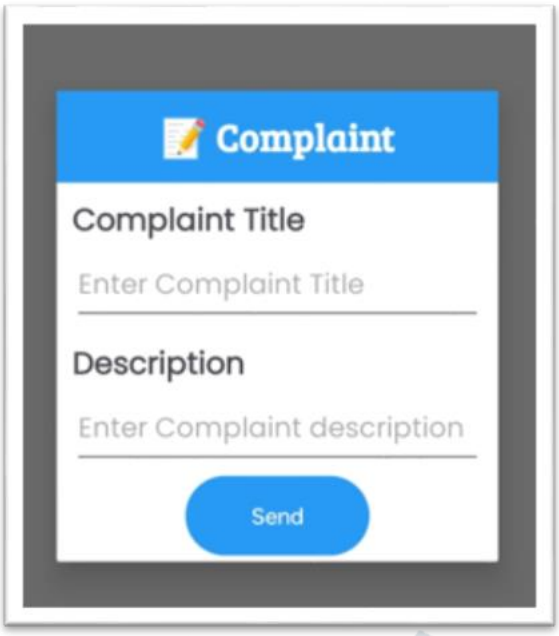


The login credential creation screen is a part of the mobile application that provides the admin of the building to access login credential creation functionality for a new tenant/owner. Here, in this application the admin fills in the following details to create a login credential for new tenant/owner:

- **Name:** Name of the tenant/owner.
- **Enter mobile number:** Mobile number to be set for the tenant/owner login.
- **Enter password:** Password to be set for the tenant/owner login.

Fig: 4.5. Login credential creation screen

The new interaction generation screen is a part of the mobile application that provides the admin of the building to access interaction creation functionality and the generated interaction is displayed on tenant/owner side. Here, in this application the admin fills in the following details to create a new interaction:



- **Interaction title:** Title of the interaction.
- **Interaction subject:** Type of the interaction (meeting/event/announcement).
- **Interaction description:** Description of the event to be more specific.

Fig: 4.6. New interaction generation screen

The new complaint generation screen is a part of the mobile application that provides the tenant/owner of the building to access complaint generation functionality and the generated complaint is displayed on admin side. Here, in this application the tenant/owner fills in the following details to generate a new complaint:

- **Complaint title:** Name/title of the complaint.
- **Complaint description:** Description of the complaint to be more specific.

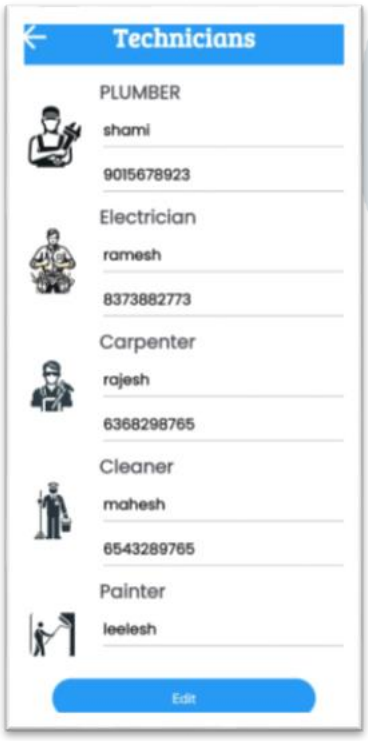


Fig: 4.7. New complaint generation screen

The directory screen is a part of the mobile application that provides the admin of the building to update directory functionality. Here, in this application the admin updates the service provide name and his mobile number in the given data field.

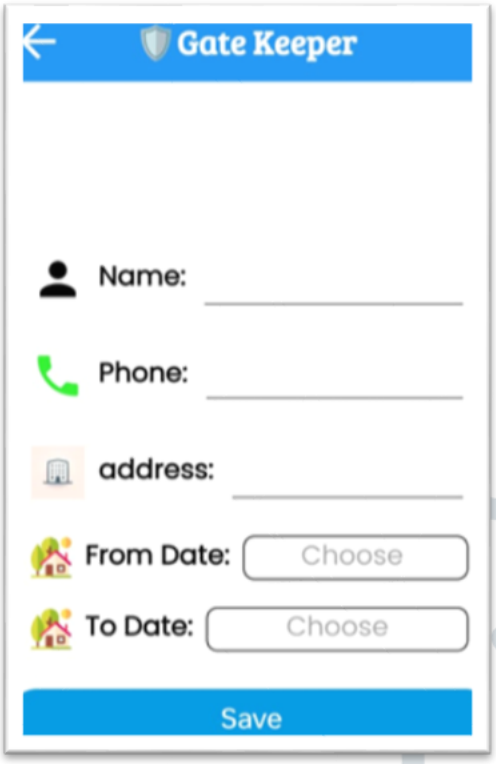


Fig: 4.8. Directory screen

The gate pass generation screen is a part of the mobile application that provides the tenant/owner of the building to access gate pass generation functionality for a visitor and the gate pass that is generated is displayed on both admin and tenant/owner side. Here, in this application the tenant/owner fills in the following details to generate a new gate pass:

- **Name:** Name of the visitor.
- **Phone:** Mobile number of the visitor
- **Address:** Address of the visitor.
- **From Date:** Starting date for allowing into apartment.
- **To Date:** Ending date for allowing into apartment.

5. CONCLUSION

In conclusion, our Android application addresses the complexities of managing apartment complexes by providing a centralized platform for admin, tenants/owners, and gatekeepers. Developed using Firebase, Kotlin, and Android Studio, the application streamlines operations such as billing management, complaint management, directory services, and visitor entry management. Its clean and intuitive interface ensures a seamless user experience, enhancing communication and operational efficiency. By simplifying apartment management tasks and fostering a connected community, this mobile application significantly improves the quality of life for tenants/owners and reduces managing tasks for admin of the community.

6. REFERENCES

[1] https://upkeep.com/learning/apartment-maintenance/
[2] https://www.mrhandyman.com/tips-ideas/checklists-resources/apartment-maintenance-checklist/
[3] https://adda.io/blog/2021/03/apartment-maintenance-charges/
[4] https://www.gofmx.com/facilities-management-software/
[5] https://www.homee.com/blog/on-demand-apartment-maintenance-services