

Dynamic Contact App: The Case Study of JNEC Contact

Rudra Dungyel¹, Dendup Tshering², Younten Tshering³, Tashi Yangchen⁴
Department of Information Technology, Jigme Namgyel Engineering College
Royal University of Bhutan, Dewathang, Bhutan

*E-mail: 05220174.jnec@rub.edu.bt¹, 05220170.jnec@rub.edu.bt², yountentshering.jnec@rub.edu.bt³, tashiyangchen.jnec@rub.edu.bt⁴

Abstract — Even though efficient communication is crucial in today’s workplace, many businesses struggle with inconsistent contact management platforms and disjointed data sources. Inefficiencies are frequently caused by this fragmentation, since employees find it difficult to obtain available and correct contact information when they need it. Furthermore, juggling several communication channels can cause miscommunication and delays, which impedes teamwork and productivity. This highlights the necessity for a dedicated contact management application, named the “Contact App”. Using Flutter for cross-platform development and Firebase for real-time database and authentication, the application is made dynamic and compatible with both Android and iOS devices. This application simplifies the process of managing contacts, allowing users to store, update and access contact details with ease. This increases overall workplace productivity by making "Contact App" a complete and effective solution for the contemporary corporate communication. Moreover, the “Contact App” provides an interactive interface to users to directly call, email, and send messages. It has been demonstrated to be more comfortable, efficient, and user-friendly for finding a contact information compared to existing app.

Keywords — *application, dynamic, flutter, firebase*

I. INTRODUCTION

In an era where digital communication is pervasive, the need for a centralized and efficient system to manage contact information has become increasingly clear. Traditional methods of maintaining contacts, such as physical address books or scattered digital files, often result in disorganization, inefficiency, and missed opportunities. Whether it is for personal connections or professional networking, managing contact information efficiently is key to staying organized and building meaningful relationships.

As per [1], the “JNEC Contact” app was deployed by giving the Android Application Package (APK) file or setup file to all android phone users. Although android phone users found the contact application user-friendly and convenient in their daily lives. However, the current version has limited features based on its compatibility. The iPhone Operating System (iOS) users preferred the application to be compatible in their IOS system. Similarly, the feedback given for the improvement of the app was to make the application dynamic where the contact details can be modified when required and some users with technical background suggested integrating the database. Hence, this paper became the main pillar of reason to embark on this project to develop it to Dynamic “Contact App”.

Recognizing the challenges, this project is embarked on the development of the “Contact App”, a modern solution tailored to meet the evolving needs of contact organization and accessibility. Using Flutter for cross-platform development and Firebase for real-time database and authentication, the application is made dynamic and compatible with both Android and iOS devices.

II. LITERATURE REVIEW

As per [2], the mobile app development is made possible using different techniques and methodologies which add essence to the application. It’s a long-going process which cannot be achieved all at once. Iterative development is the way of breaking down the development of a large application software into small chunks so that all the features are brought out to the users with a lot more efficiency. The iterative approach to mobile app development provides the ease of getting it done in small steps. By repeatedly focusing on the entire process of app development, it helps in completing the application development with more efficiency. Therefore, the same method was implemented as mentioned above to design and develop the “Contact App”.

According to [3], Building an app in Flutter can be smooth, seamless, and cost-effective. As Flutter decreases the amount of coding, the overall cost is lower than when building separate native apps, where the code needs to be written twice. Besides the costs, apps built in Flutter perform similarly to native apps. [4] Flutter has some technology advantages such as built-in widgets and command line tools. It relies less on third-party tools, has great testing, CI/CD and maintenance tools which can make it easy to learn and quickly become productive. As time goes on, the community will grow and Google will put more emphasis on it which will also help its future. Thus, Flutter framework was used for cross-platform development for this project.

In the report [5], Firebase is considered as web application platform. It helps developers build high-quality apps. Firebase is NoSQL based. It stores the data in JavaScript Object Notation (JSON) format which doesn’t use query for inserting, updating, deleting or adding data to it. It is the backend of a system that is used as a database for storing data. Hence, the Firebase database was used for querying the data and to make the application dynamic.

III. METHODOLOGY

The “Contact App” was developed with iterative approach in application development where the development process is divided into small, manageable chunks called iterations. This process is repeated until the final product meets the desired requirements for the deployment of the application. “Contact App” development include the following phases:

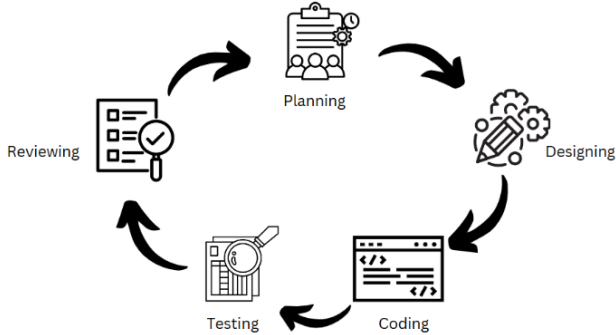


Figure 1: Iterative Process Model

A. Planning

During the planning phase, all the requirements for the “Contact App” development were collected before getting started with the designing and coding phase or next phase. For the collection of requirements, brainstorming and analysis review of the existing application and documentation were conducted. After determining the requirements, the structuring of functional requirements and non-functional requirements were collected and analyzed based on the need of the system and user satisfaction.

B. Designing

For the “Contact App” development, Draw.io(diagrams.net), a free, open-source, and versatile diagramming application was used for creating the use case diagram to describe the functional requirements of a system. This phase made it convenient for us in understanding and documenting system behavior from the user’s perspective.

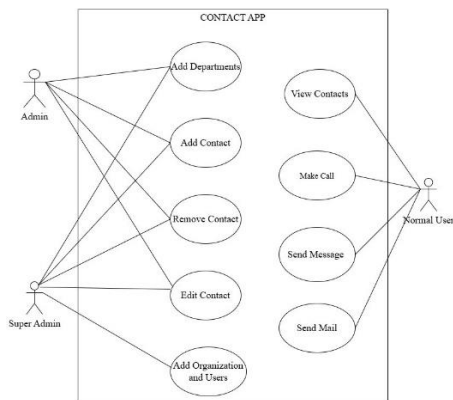


Figure 2: Use Case Diagram

C. Coding

With the completion of the design, coding was initiated with the pair programmer concept using Flutter and Dart as the programming language in the Visual Studio Code. Wherein the driver focused on coding and the navigator focused on fixing the error simultaneously with the roles periodically swapped, to

ensure both programmers remain actively engaged and share a deep understanding of the code. In the due course of app development, functional testing was carried out to ensure that it was as per the design.

D. Testing

Unit testing, white box testing and integration testing were a concern during the development and implementation of the application. For the successful operation of the application, it was tested at the time of coding to check and ensure that it functions properly. When the coding was done with the final version, the complete system testing of the app was done on a different level of the android version in emulators.

E. Reviewing

With constant feedbacks and insightful reviews of the testing outcomes by the project guide, it formed an essential part of the iterative approach’s feedback loop, promoting agility, adaptability, and responsiveness to changing needs throughout the software development lifecycle. Each cycle or sprint aims to incrementally improve the app by adding new features or refine the existing ones. The insights and feedback from each review phase was used for the next planning phase, ensuring continuous improvement and alignment with user needs.

IV. RESULT

When any user successfully login to the app with the authenticated email and password, the users are categorized into three Users based on the roles that are assigned to the authorized email as shown in Figure 3. The following Users are:

1. Normal User – can view and has access to contact information of their organization and make call, text messages and send email.
2. Admin – has authority over their organization to modify contact information.
3. Super Admin – has complete authority over any organizations and modify the contact information, and the privilege to create new organizations and the users in the app.

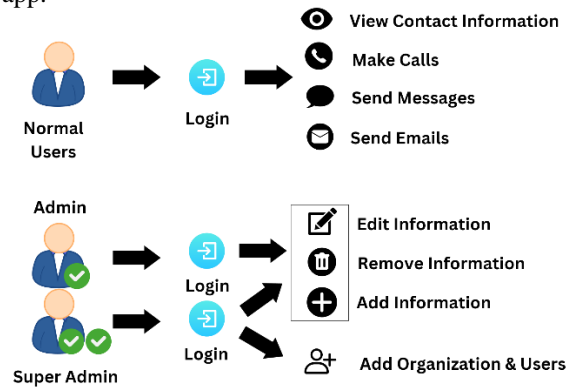


Figure 3: System Functionality

To run an application on android devices, developer options are enabled on android devices to debug. Then, the application is installed on an android device. “Contact App” is launched and used.

After clicking on the app to launch, the app will open the login page interface for the user to login as shown in Figure 4. The users are only able to login to the app through the

authenticated email and password that are shared by the Super Admin of the app or the Admin of the organization.

When the user logged in as the Super Admin, the app will open the Super Admin interface. Figure 5 shows the Super Admin Page to select any organization from the organization lists and modify the information embedded in it. It also provides an interface to create an organization and the users (Admin and Normal User) for that organization.



Figure 4: Login Page

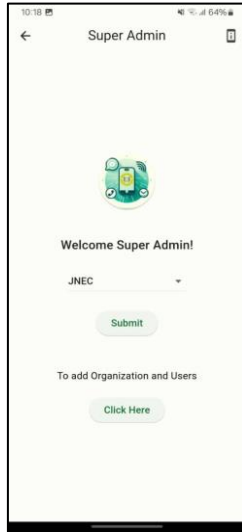


Figure 5: Super Admin Page

If the user logged in is Super Admin or Admin, the Department Page provide a comprehensive set of features to manage the department information as shown in Figure 6. The user is able to perform add, edit and delete operations to the department list.

When clicked on any department from the department list, the app will navigate to the Staff Page. If the user logged in is Super Admin or Admin, the Staff Page provide a comprehensive set of features to manage the contact information details. The user is able to perform add, edit and delete operations to the contact details.

However, if the user logged in is Normal User, the Department Page will provide just the list of departments of the organization and will not be able to perform add, edit and delete operations to the department list like Super Admin or Admin.

When clicked on any department from the department list, the app will navigate to the Staff Page. If the user logged in is Normal User, the Staff Page will provide just the list of contact information details of that department. The user will not be able to perform add, edit and delete operations to the contact details like Super Admin or Admin.

When the user clicks on any contact details, the app provides the user with an interface option to call, send message or send emails as shown in Figure 7. This is an important functional feature of this application that provides an interactive interface for users to directly call, email, and send messages.

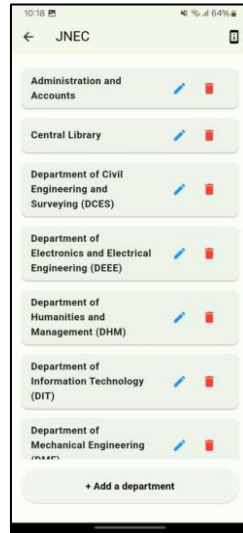


Figure 6: Department Page

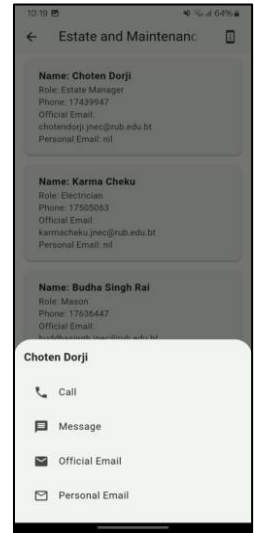


Figure 7: Contact Options Interface

V. DISCUSSION

The “Contact App” was successfully developed as a dynamic and user-responsive application, replacing the existing static application and eliminating the conflicts of accessing outdated data and limited compatibility features to enhance overall user satisfaction. It provides an intuitive and user-friendly interface that enables users to navigate the application effortlessly, providing easy access to contact information and a comprehensive suite of features to simplify contact management tasks.

Moreover, the “Contact App” replaces the existing android-based static application with a robust dynamic application, developed to be compatible with both Android and iOS devices. Additionally, this app is integrated with real-time database to streamline the contact information management process to provide an efficient, accurate, reliable, real-time and user-friendly platform.

VI. CONCLUSION

The "Contact App" marks a pivotal shift in corporate communication, ensuring efficiency and inclusivity across multiple operating systems and addressing issues with disjointed contact management. Developed iteratively, it adapts to stakeholders' evolving needs, refining its user-centric features through continuous analysis, design, execution, and assessment.

The app's comprehensive features cater to diverse user needs, from robust user identification to intuitive contact management and communication tools. Super Admin and Admin roles facilitate organizational management by allowing contact additions, updates, and removals, ensuring scalability and adaptability.

Emphasizing non-functional requirements such as compliance, user-friendliness, cross-platform compatibility, security, performance, and reliability, the "Contact App" demonstrates a commitment to high-quality mobile development. With stringent data protection, speed optimization, and privacy measures, it sets a new standard in mobile applications.

In the dynamic digital market, the "Contact App" exemplifies innovation, enabling businesses to implement advanced communication strategies and enhance productivity. Its dynamic nature and cross-platform accessibility make it a valuable tool for improving workplace efficiency, collaboration, and communication.

ACKNOWLEDGMENT

We would like to extend our sincere appreciation to Mr. Tashi Tobgay, Associate Instructor of Department of Information Technology, for his unwavering support, advice, assistance, and expertise in shaping our project and ensuring its excellence. We would also like to express our deepest gratitude to all those who have contributed to the successful completion of our project. Further, we would like to extend our heartfelt gratitude to Mr. Tshewang Dhendup, Assistant Lecturer, for his valuable support and providing the necessary infrastructure.

REFERENCES

- [1] Y. Tshering, S. Dorji, and T. Wangmo, "User Friendly Android-based Contact Application: Case of Accessible Contact in Educational Organization," iJRASET, vol. 10, no. 9, Sep. 2022. [Online]. Available: <https://www.ijraset.com/best-journal/user-friendly-androidbased-contact-application-case-of-accessible-contact-in-educational-organization>
- [2] V. Shah, "Top reasons to opt for iterative approach to mobile app development," dev, Apr. 28, 2018. [Online]. Available: <https://dev.to/vaibhavshah/top-reasons-to-opt-for-iterative-approach-to-mobile-app-development-173d>
- [3] B. Szypelow, "Why is Flutter a good choice for cross-platform projects?," Netguru, Dec. 6, 2023. [Online]. Available: <https://www.netguru.com/blog/flutter-for-cross-platform-app-development>
- [4] D. Erickson, "Cross Platform Development Technology Comparison: React Native vs. Flutter," SCREAMINGBOX, Mar. 27, 2024. [Online]. Available: <https://www.screamingbox.net/blog/cross-platform-development-technology-comparison-react-native-vs-flutter>
- [5] C. Khawas and P. Shah, "Application of Firebase in Android App Development-A Study," International Journal of Computer Applications, vol. 179, no. 46, Jun. 2018. [Online]. Available: <https://www.ijcaonline.org/archives/volume179/number46/khawas-2018-ijca-917200.pdf>
- [6] A. Jaiswal, "Contact Management System," Galgotias University. [Online]. Available: <http://103.47.12.35/bitstream/handle/1/1603/contact%20management%20system%28report%29%20-%20Abhishek%20Jaiswal.pdf?sequence=1&isAllowed=y>
- [7] A. Krishnan, "Flutter vs React Native in 2023: A Detailed Comparison," hacker, Jan. 4, 2023. [Online]. Available: <https://hackr.io/blog/react-native-vs-flutter>
- [8] HeyFlutter, "Flutter Firebase Setup using Flutter Fire CLI [2023] Easy and Fastest Way!," YouTube, 2023. [Online]. Available: <https://www.youtube.com/watch?v=G-mbqiE87Lw>
- [9] S. U. Sambhe and S. Murab, "Literature Review on Smart Social Contact Management System for Better Memory Recall," IJEIT, vol. 3, no. 9, Mar. 2014. [Online]. Available: https://www.ijeit.com/Vol%203/Issue%209/IJEIT1412201403_42.pdf
- [10] P. R. Saraf, "A Review on Firebase (Backend as A Service) for Mobile Application Development," ResearchGate, Jan. 2022. [Online]. Available: https://www.researchgate.net/publication/358244355_A_Review_on_Firebase_Backend_as_A_Service_for_Mobile_Application_Development
- [11] T. K. Chawla and A. Kajala, "Transfiguring of an Android App Using Reverse Engineering," International Journal of Computer Science and Mobile Computing, vol. 3, no. 4, Apr. 2014. [Online]. Available: <https://www.ijcsmc.com/docs/papers/April2014/V3I4201499b43.pdf>
- [12] freeCodeCamp, "Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial," YouTube, 2022. [Online]. Available: <https://www.youtube.com/watch?v=VPvVD8t02U8>
- [13] C. Fanchi, "The importance of iterative prototyping in application development," backendless, Feb. 7, 2023. [Online]. Available: <https://backendless.com/the-importance-of-iterative-prototyping-in-application-development/#:~:text=Iterative%20prototyping%20can%20lead%20to,an%20functionality%20of%20the%20app.>
- [14] Net Ninja, "Flutter tutorial for beginners," YouTube, 2020. [Online]. Available: <https://www.youtube.com/watch?v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ>