CENTRAL UNIVERSITY

SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION

TECHNOLOGY

DESIGN AND IMPLEMENTATION OF A COMMUNITY-BASED LOCAL
SERVICES AND AMENITIES LOCATOR APP

DANIEL ADDOTEY ALLOTEY   &   MUNIRU MOHAMMED

2025

CENTRAL UNIVERSITY

DESIGN AND IMPLEMENTATION OF A COMMUNITY-BASED
LOCAL SERVICES AND AMENITIES LOCATOR APP

BY

DANIEL ADDOTEY ALLOTEY   &   MUNIRU MOHAMMED
CSC/21/01/1399   CSC/20/01/2455

A Long Essay submitted to the School Of Science and Technology, Central University in partial fulfillment of the requirements for the award of a Degree of Bachelor of Science (BSc.) in Computer Science.

AUGUST 2025

DECLARATION

Student's Declaration

We hereby declare that this project work is the result of our own original research and that none of it has been presented for another degree in this University or elsewhere. We are responsible for any errors and omissions detected.

Signature: _____   Signature: _____

Name: Daniel Addotey Allotey   Name: Mohammed Muniru

Date: _____   Date: _____

Supervisor's Certification

I hereby certify that the preparation of the project work was supervised in accordance to the guidelines of supervision of project work laid by Central University.

Supervisor's Signature: _____

Name: _____

Date: _____

ABSTRACT

Finding local services and businesses has become increasingly challenging, particularly when someone moves to a new area or visits an unfamiliar community. This project aimed to develop a mobile application that bridges the gap between community members and local service providers. The app serves both everyday users searching for nearby amenities and business owners wanting to showcase their services to the local community.

Our research approach involved building a cross-platform mobile application using Flutter framework, complemented by a web-based admin dashboard developed with React JS. Firebase provided the back-end infrastructure for real-time data management and user authentication. We gathered requirements through informal community observations and tested the system using various user scenarios.

The resulting application offers an intuitive interface where users can discover restaurants, schools, hospitals, and service providers within their neighborhood. Business owners can create profiles, update their information, and reach potential customers more effectively. The admin dashboard enables system oversight and content moderation. Our findings demonstrate that community-focused location-based applications can significantly improve local service discovery while supporting small business growth.

Future enhancements could include user review systems, push notifications for nearby services, and integration with public service information to create a more comprehensive community platform.

ACKNOWLEDGEMENT

All praises belong to God almighty, and thanks for this divine assistance and guidance, which made it possible for it to be possible for us to complete our academic program. We want to express our gratitude to all the people who helped complete our final year project work. Their support and encouragement have been priceless throughout the completion of our project work. We would also like to thank our project supervisor Dr. Kissi Mireku for his guidance and assistance. His expertise and mentorship have played a huge role in enhancing the quality of our project. We are also grateful for his support and the opportunities he provided us for our professional growth. We are grateful for our family's love and encouragement throughout our academic journey. Finally, to all who played a role and made this project a success, we are very grateful for all your support and contributions.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION

Background of study

Moving to a new neighbourhood or visiting an unfamiliar area often presents unexpected challenges. During experiences relocating within Ghana, individuals frequently find themselves struggling to locate basic services - from finding a reliable mechanic to discovering which school might be suitable for a family member. The usual approach involves asking strangers on the street, which sometimes leads to incorrect directions or outdated information.

This challenge is not unique to newcomers. Even long-time residents often remain unaware of businesses operating just a few blocks away. A skilled seamstress might work from her home nearby, but without proper visibility, potential customers never discover her services. Meanwhile, she struggles to attract new clients because she lacks an online presence.

Current solutions like Google Maps tend to favour established businesses and urban areas, often overlooking smaller enterprises in local communities. Popular review platforms like Yelp have limited penetration in many African markets and do not adequately serve rural or suburban areas. This gap became the motivation for developing a community-centred solution that could benefit both service seekers and local business owners.

Statement of the problem

Local communities face a dual challenge in service discovery and business visibility. Residents, particularly newcomers, struggle to identify available services within their neighbourhood, while small business owners lack accessible platforms to promote their services to local customers. Existing location-based applications primarily focus on established businesses or urban centers, leaving a significant gap in community-level service discovery.

This disconnect results in inefficient resource utilization where capable service providers remain underutilized while community members travel unnecessarily far for services available nearby. The absence of a centralize, community-focused platform perpetuates this information asymmetry, hindering both consumer convenience and local economic growth.

Objectives of the study

The main objective of this study was to develop a user-friendly mobile platform that facilitates local service discovery while empowering small businesses to manage their digital presence.

The specific objectives were to:
- Develop location-based service discovery functionality that displays businesses and amenities by community
- Implement user account management with the flexibility to switch between regular and business account types
- Create business profile management tools that allow owners to update their information independently
- Build an administrative interface for platform oversight and content management
- Ensure the system works reliably on various mobile devices with different performance capabilities

Research questions

- How can mobile technology improve the way people discover local services and businesses?
- What features would make such a platform appealing to both service seekers and business owners?
- Which development tools and approaches would best serve the technical requirements while maintaining user-friendliness?

Significance of the study

This research addresses real community needs while demonstrating practical applications of mobile technology for local development. The project contributes to understanding how digital platforms can support grassroots economic activity and improve community connectivity.

The work also showcases how modern development tools like Flutter and Firebase can be leveraged to create meaningful solutions without requiring extensive resources, making it relevant for similar projects in resource-constrained environments. The study provides insights into community-level technology adoption and the potential for mobile applications to bridge information gaps in local service markets.

Delimitation of the study

The scope of this study focused specifically on community-level service discovery within Ghanaian contexts. The application includes various business categories from restaurants and shops to professional services like electricians and tailors. It encompasses both social amenities such as schools and hospitals, and commercial enterprises.

The system consists of a mobile application for end-users and business owners, plus a web-based administrative dashboard. Geographic coverage depends on user location settings, allowing flexibility while maintaining community focus. The study was limited to the development and testing phases, without long-term user adoption analysis.

Organisation of the study

This documentation follows a structured approach across five chapters. Chapter One establishes the context and objectives of the study. Chapter Two examines related literature and theoretical foundations. Chapter Three explains the development methodology and testing procedures. Chapter Four presents the findings and analyses system performance. Chapter Five summarizes achievements and suggests future improvements.

Limitations

- The system requires consistent internet connectivity to function properly, limiting use in areas with poor network coverage.

- Each user can only maintain one business account, which might restrict some multi-business owners.

- Location-based features depend on users granting location permissions or manually selecting their community.

- The administrative structure relies on a single super admin account, which could create bottlenecks in larger deployments.

CHAPTER TWO: LITERATURE REVIEW

Introduction

Understanding existing solutions and theoretical frameworks proved essential for developing an effective community-focused platform. This review examines relevant research in location-based services, mobile application development, and community technology adoption. We also analyze existing applications to identify strengths, weaknesses, and opportunities for improvement.

Theoretical framework

Two primary theoretical concepts guided this project's development: Location-Based Services (LBS) and the Technology Acceptance Model (TAM).

Location-Based Services represent a technological approach that delivers information and functionality based on user location (Kang et al., 2018). Applications like Uber and food delivery platforms demonstrate LBS effectiveness in connecting users with nearby services. However, most implementations focus on urban environments and established businesses, creating opportunities for community-specific applications.

In our context, LBS enables users to discover nearby businesses and amenities without requiring extensive local knowledge. This particularly benefits newcomers and visitors who lack familiarity with their surroundings. The technology also helps business owners reach customers within their service radius more effectively.

The Technology Acceptance Model explains how users decide whether to adopt new technologies. According to TAM, adoption depends primarily on perceived usefulness and ease of use (Davis, 1989). Users are more likely to embrace applications that provide clear value without requiring significant learning effort.

We applied TAM principles by prioritizing intuitive navigation and immediately apparent functionality (Venkatesh & Bala, 2008). The application avoids complex features that might intimidate less tech-savvy users while ensuring that core functions deliver obvious benefits.

Conceptual framework

Our system architecture integrates multiple user roles within a single platform. Regular users can explore their community, save interesting listings, and share discoveries with others. Business owners can create and manage their profiles while maintaining regular user capabilities. The system administrator oversees platform operations through a dedicated web interface.

Firebase provides the underlying data infrastructure, enabling real-time updates and secure user authentication. Flutter handles the mobile interface, ensuring consistent performance across Android and iOS devices. React JS powers the administrative dashboard, offering comprehensive management capabilities.

The platform emphasizes community-level organization, allowing users to filter content by neighborhood or area. Categories help organize different business types while maintaining browsing simplicity. This structure supports both casual exploration and targeted searching.
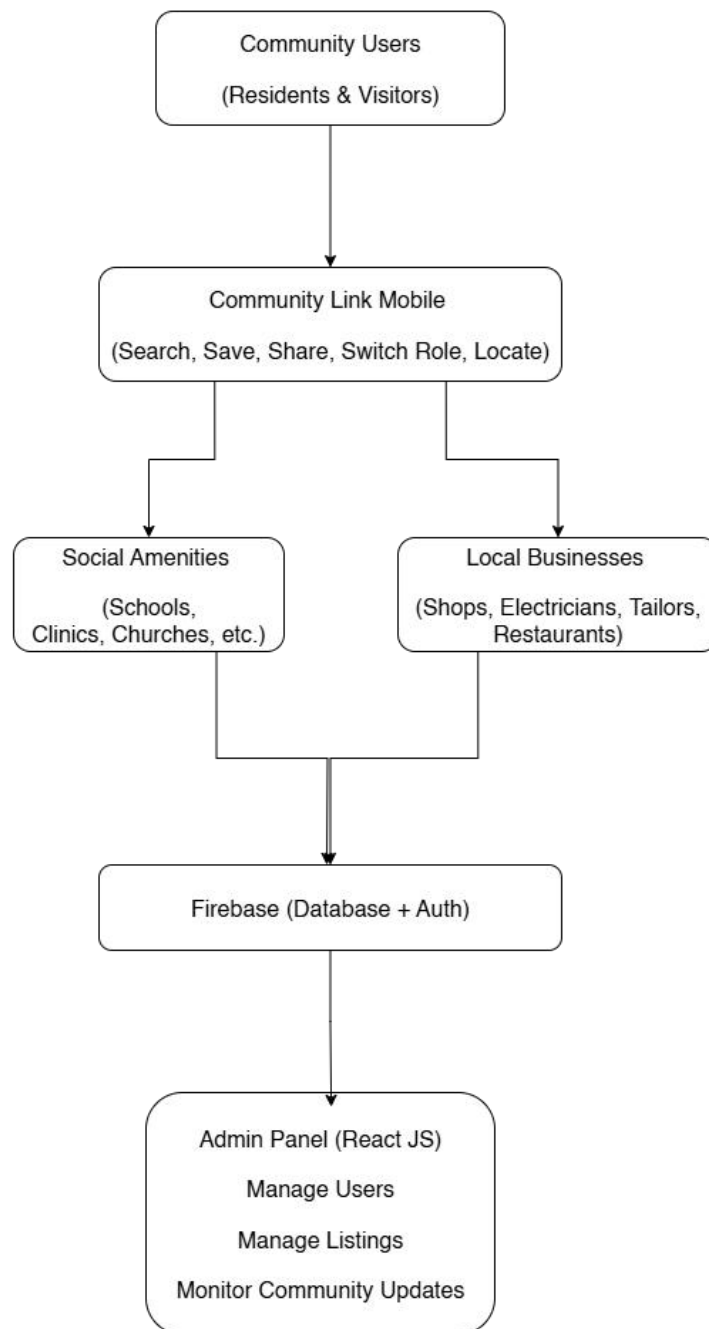
Figure 1: Conceptual Framework Diagram

Empirical review

Several existing platforms offer related functionality, each with distinct strengths and limitations.

Google Maps excels at navigation and finding established businesses but often misses smaller local enterprises (Mountain & Raper, 2001). Its review system works well for popular locations but provides limited coverage for community-level businesses. The platform's urban bias means rural and suburban areas receive less comprehensive coverage.

Yelp offers robust review functionality and detailed business information but has limited penetration in many African markets. Its focus on dining and entertainment doesn't adequately cover the full range of community services that residents need. The platform also requires significant user engagement to generate useful content, which can be challenging in smaller communities.

Local platforms like Jiji Ghana primarily serve marketplace functions rather than service discovery. While useful for buying and selling, they don't address the need for finding local service providers or community amenities.

GhanaPost GPS provides addressing solutions but lacks business discovery features. Users can create location pins but cannot easily search for nearby services or browse business categories.

Summary of literature

The literature review reveals a clear opportunity for community-focused service discovery platforms. Existing solutions either target urban markets, focus on specific business types, or lack the comprehensive approach needed for effective community service discovery.

Our application addresses these gaps by combining social amenities with commercial services, allowing users to switch between consumer and business roles, and providing administrative oversight capabilities. This comprehensive approach differentiates our solution from existing platforms while addressing identified community needs.

CHAPTER THREE: RESEARCH METHODS

Introduction

This chapter outlines our development methodology, target user identification, feature selection process, and testing procedures. We employed a user-centered design approach, focusing on real community needs while leveraging appropriate technical tools.
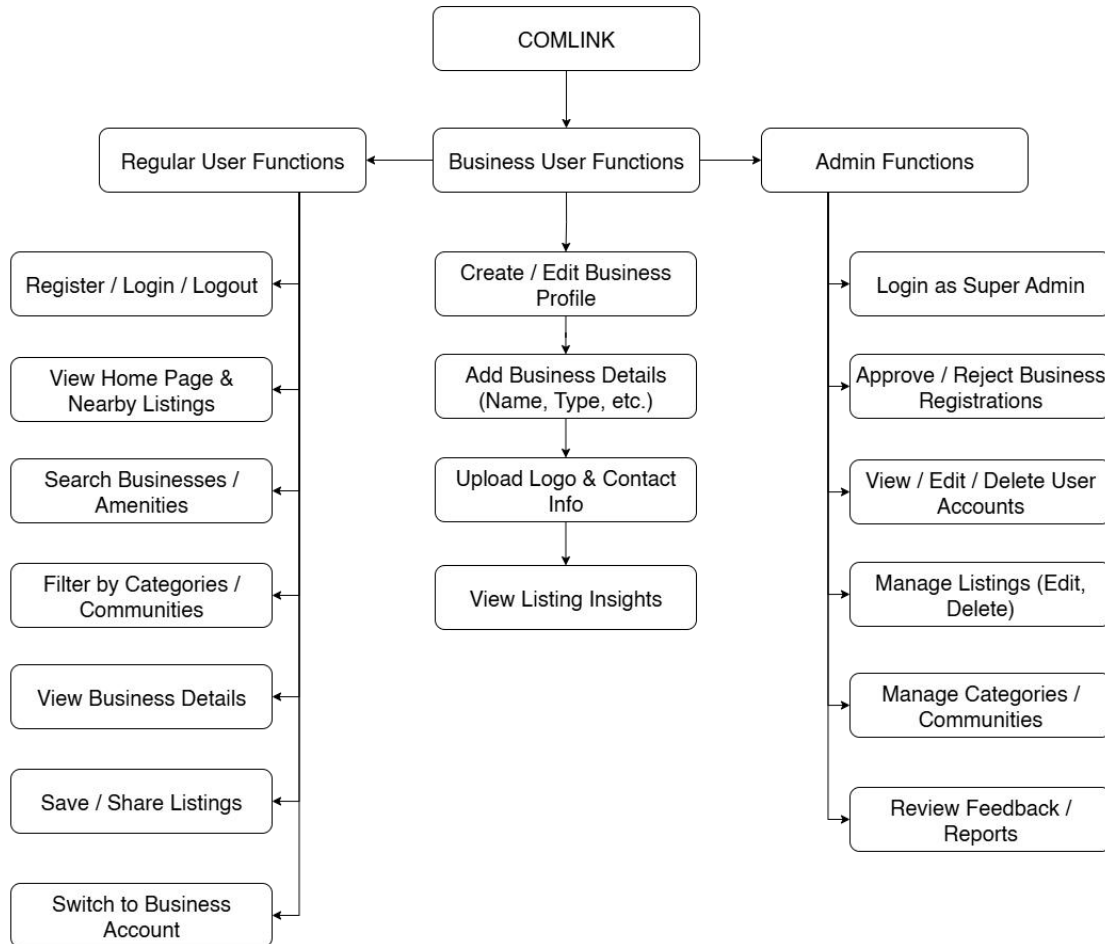
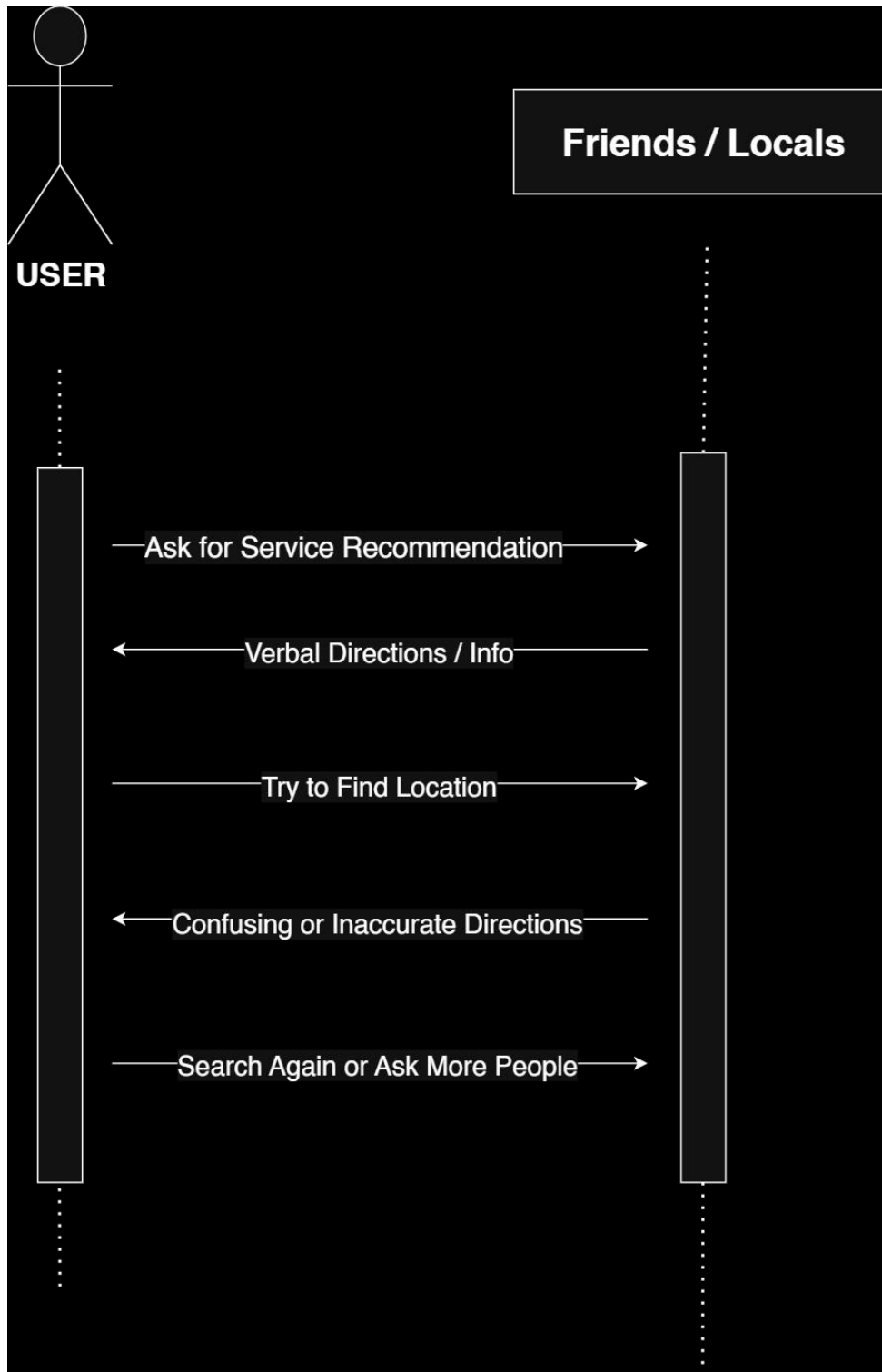

Figure 2: Functional Requirements Diagram

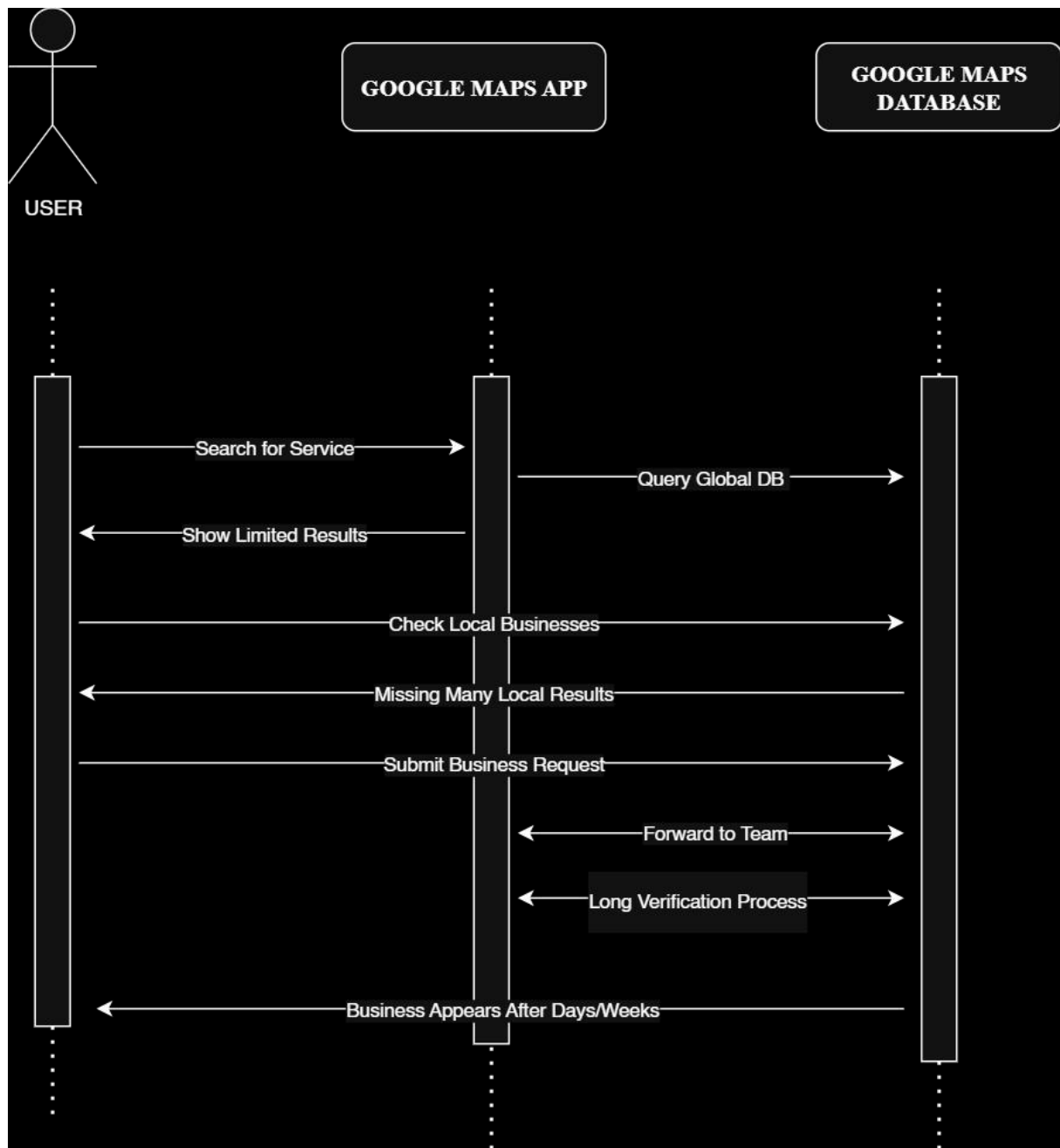**Figure 3: Existing System (Traditional Method) Diagram**

Figure 4: Existing System (Google Maps Service Discovery) Diagram
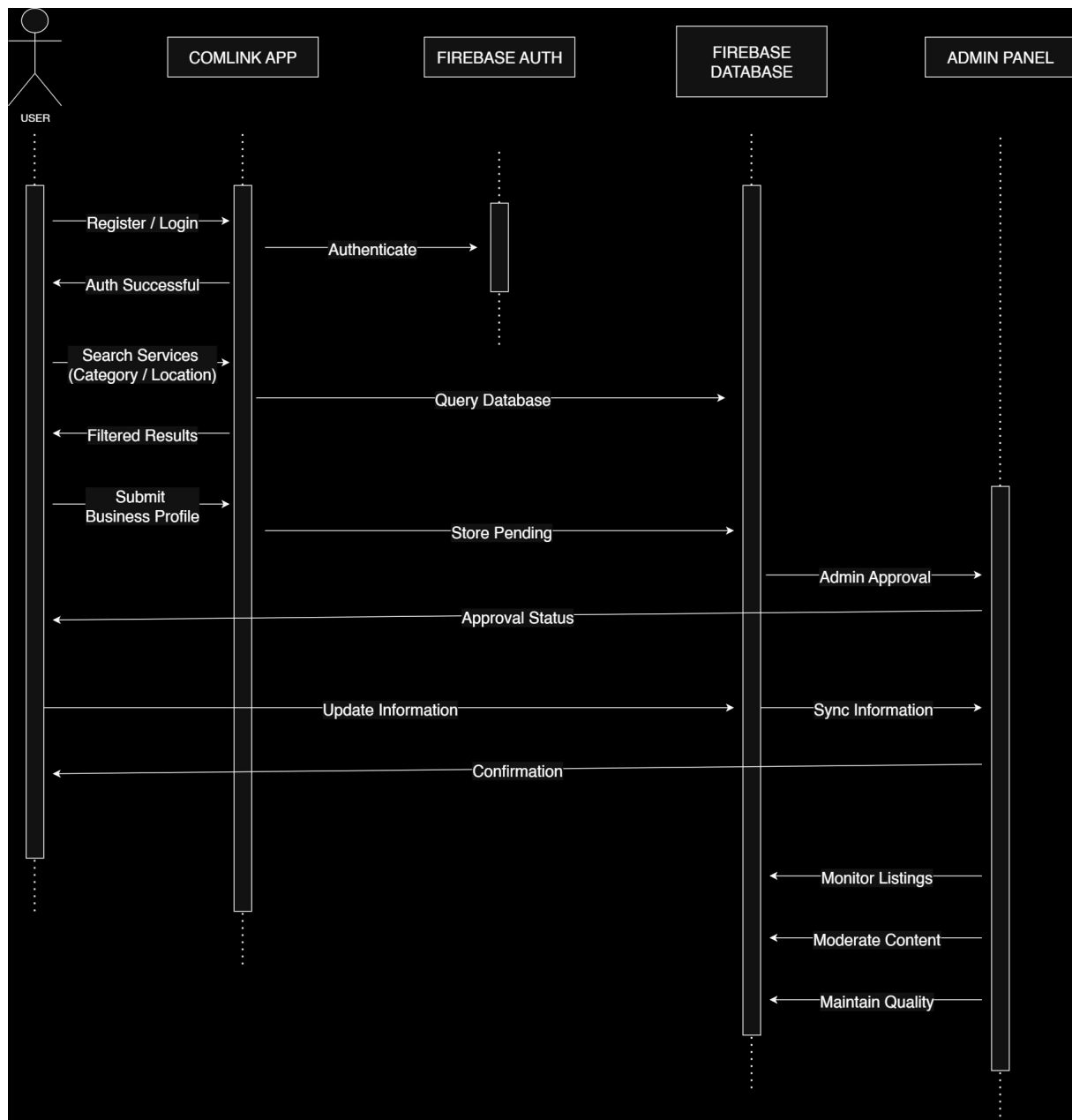
Figure 5: Proposed System Diagram

Study design

We adopted an iterative development approach combined with user-centered design principles (Pressman & Maxim, 2014). This methodology allowed us to refine features based on testing feedback while maintaining focus on core objectives.

The development process involved several cycles: initial requirement gathering, prototype creation, feature implementation, testing, and refinement. This approach proved flexible enough to accommodate insights gained during development while ensuring systematic progress toward project goals.

Population

Our target users include three distinct groups, each with specific needs and expectations:

Community members seeking local services represent the primary user group. This includes long-time residents looking for new options, newcomers needing to establish local connections, and visitors requiring temporary services.

Business owners wanting to increase their visibility comprise the second group. These range from established businesses seeking better digital presence to home-based service providers hoping to reach more customers.

System administrators need tools to maintain platform quality and manage user conflicts. This role requires comprehensive oversight capabilities while remaining manageable for non-technical users.

Sample and sampling procedure

Instead of testing on real people, I picked the main features that users would need most and focused on building those (Preece et al., 2015). The main features chosen were:

- User registration and login
- Search by community and category
- Save/share business listings
- Switch to business account
- Admin dashboard to manage everything

These features were chosen because they solve common problems faced by people in communities — like finding a good school or a nearby electrician. Basic sample users were created to test these features, with different roles (regular, business, admin).
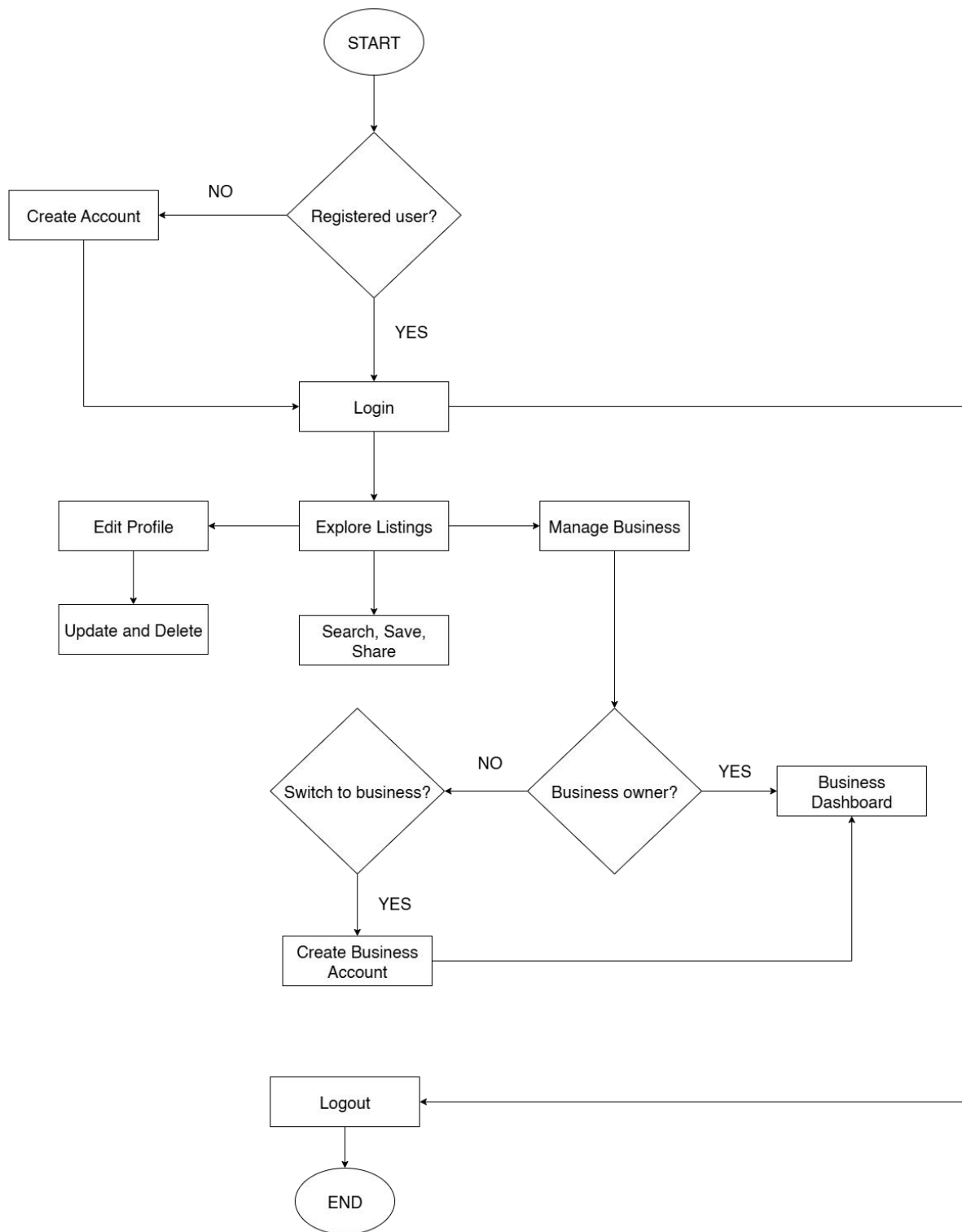

Figure 6: Decision Tree for System Flow Diagram

Instruments

Our development relied on several industry-standard tools, each chosen for specific advantages:

Flutter provided cross-platform mobile development capabilities, ensuring consistent performance on both Android and iOS devices (Windmill et al., 2020). Its single code base approach reduced development complexity while maintaining native performance characteristics.

Firebase offered comprehensive back-end services including user authentication, real-time database functionality, and cloud storage (Moroney, 2017). This eliminated the need for custom server infrastructure while providing enterprise-grade reliability and security.

React JS powered the administrative interface, offering flexibility for creating complex data management screens. Its component-based architecture facilitated rapid development of administrative features.

Adobe XD supported user interface design and prototyping, allowing us to visualize and test interface concepts before implementation. GitHub provided version control and code backup capabilities.

Data collection and development

Development proceeded through systematic feature implementation and testing. Each major component underwent individual testing before integration with the broader system.

We created test user accounts representing different roles and scenarios. Business profiles were populated with realistic data to simulate actual usage conditions. Navigation flows were tested to ensure intuitive user experiences.

Firebase security rules were implemented and tested to prevent unauthorized access while allowing appropriate functionality for each user type. Error handling was implemented to provide helpful feedback when operations failed.

Data analysis procedure

Testing focused on functional verification rather than statistical analysis (Sommerville, 2011). Each feature was evaluated against its intended functionality, with bugs documented and resolved before proceeding to dependent features.

Firebase console provided real-time monitoring of database operations, allowing us to verify that user actions resulted in correct data updates. Flutter's debugging tools helped identify and resolve interface issues.

We maintained detailed logs of testing results, noting both successful operations and identified issues. This systematic approach ensured comprehensive feature validation before considering components complete.

CHAPTER FOUR: RESULTS AND DISCUSSION

Introduction

This chapter presents the outcomes of our development effort, analyzing how well the implemented system meets project objectives. We examine feature performance, user interface effectiveness, and system reliability while discussing implications for community service discovery.
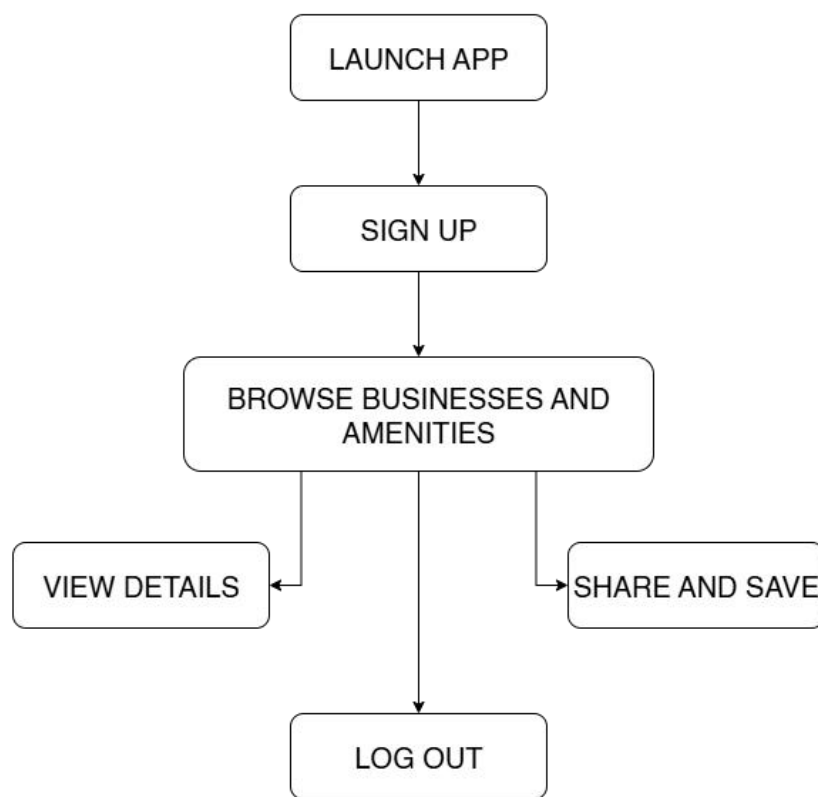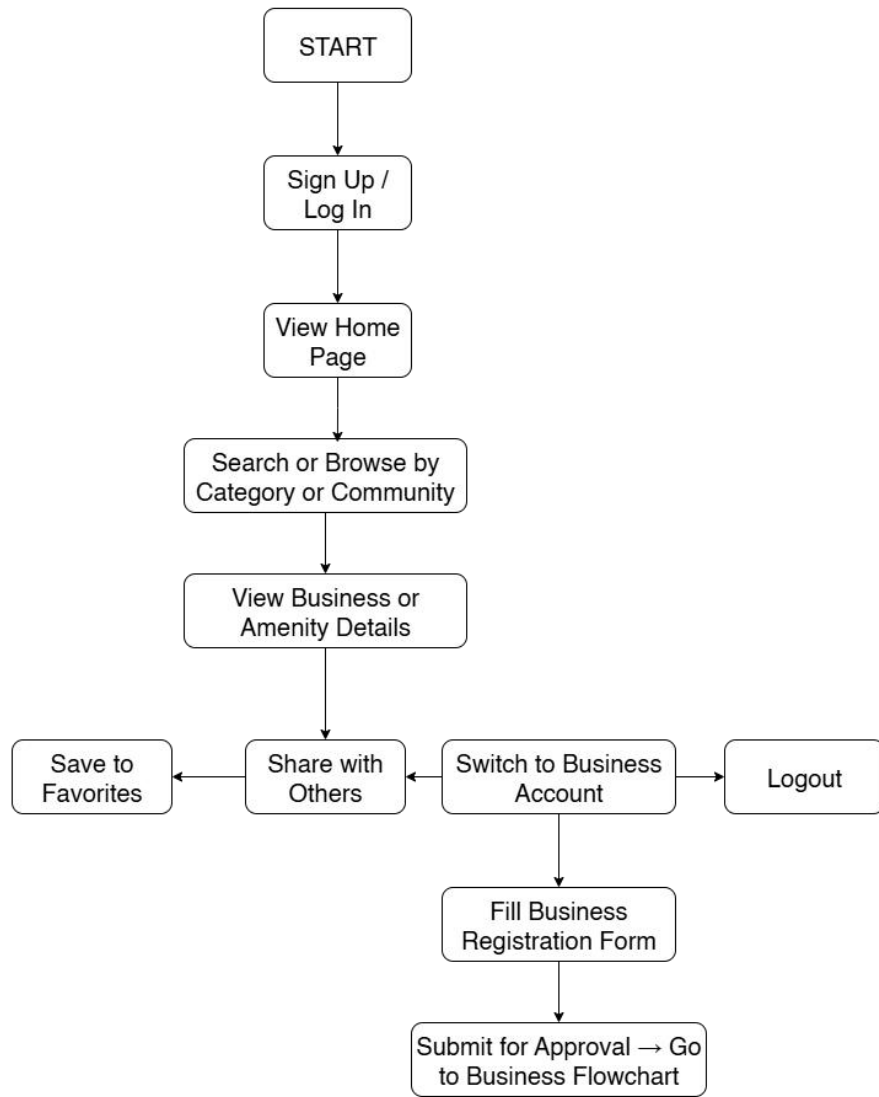

Figure 7: Regular User Flowchart

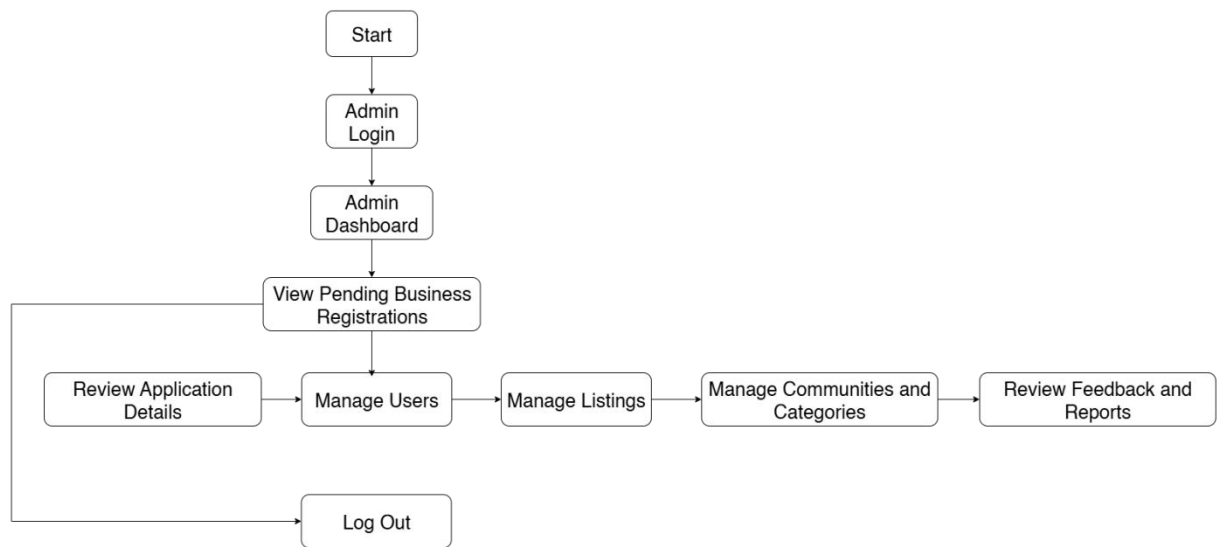**Figure 8: Business Owner Flowchart**

Figure 9: Super Admin Flowchart

Overview of features

The completed application provides several core capabilities that address identified community needs:

User registration and authentication allow secure account creation with email verification. The system supports both individual users and business accounts, with seamless switching between roles when appropriate.

Location-based discovery enables users to find nearby businesses and amenities by selecting their community from a predefined list. Categories help organize different service types, making browsing more efficient.

Business management functionality allows owners to create profiles, upload images, and update contact information. The approval process ensures quality control while remaining accessible to non-technical users.

Administrative oversight provides comprehensive platform management through a web-based dashboard. Administrators can monitor user activity, approve business listings, and manage system content.

Results Based on Research Questions

How can mobile technology improve local service discovery?

Our implementation demonstrates that mobile platforms can significantly enhance community service discovery when designed with local contexts in mind (Dey et al., 2001). The application successfully addresses common challenges like finding nearby services and discovering new businesses.

Testing revealed that users appreciate having both social amenities and commercial services in one platform. This comprehensive approach reduces the need to consult multiple sources when exploring community resources. The category-based organization helps users find specific service types efficiently.

Location-based filtering proved particularly valuable for newcomers who lack familiarity with their surroundings. By focusing on community-level discovery rather than city-wide searching, the application provides more relevant results for typical user needs.

Can mobile applications improve business visibility in local communities?

The business management features successfully provide small enterprises with digital presence capabilities that were previously difficult to access. Business owners can create and maintain their profiles independently, reducing barriers to online visibility.

The system's community focus benefits small businesses that might be overlooked by larger platforms. Local service providers like electricians, tailors, and home-based businesses gain equal visibility with established enterprises.

Business account functionality allows entrepreneurs to control their information directly rather than depending on third-party listings that might contain errors or become outdated. This ownership approach improves information accuracy while empowering business owners. Which development tools best support community-focused applications?

Flutter proved excellent for creating consistent user experiences across different mobile platforms. Its single code base approach reduced development complexity while maintaining performance standards expected by mobile users.

Firebase provided reliable back-end infrastructure without requiring extensive server management expertise. Real-time data synchronization ensures that business information updates appear immediately for all users.

React JS facilitated rapid administrative interface development with sophisticated data management capabilities. The combination of these tools created a robust, scalable platform suitable for community-level deployments.

Challenges faced

Several technical and design challenges emerged during development:

- Data synchronization occasionally failed in poor network conditions, requiring robust error handling and user feedback systems. We implemented retry mechanisms and clear error messages to address these situations.

- User interface design needed to balance functionality with simplicity, particularly for less tech-savvy users. Multiple design iterations were required to achieve appropriate complexity levels.

- Firebase security rules required careful configuration to prevent unauthorized access while maintaining necessary functionality. Testing these rules thoroughly proved time-consuming but essential for platform security.

Functional Testing

Functional testing was done to make sure all the main parts of the app worked as expected. It was based on what the app is supposed to do (its requirements), like signing in, registering businesses, switching accounts, and managing listings.

We tested each function one by one, to check if the right thing happens when users interact with it. Here's how the testing was done for each key area:

User Registration and Login

- Test if new users can sign up with email and password
- Check if existing users can log in successfully
- Check if wrong credentials show error messages

Switching Between User and Business Account

- Confirm that a regular user can switch to a business account
- Confirm that business users can access their business dashboard
- Make sure users who haven't registered a business can't access the dashboard

Business Registration

- Check if all required fields are filled before allowing registration
- Business info should save in the database
- Business must appear under the correct community

Listing and Discovery

- Users can search for nearby businesses and amenities
- Listings are grouped into categories like Schools, Hospitals, etc.
- Users can save, view, and share listings

Admin Dashboard

- Admin can log in using email/password

- Admin can see and manage all users and businesses

- Admin can delete inappropriate listings or ban users

Navigation and UI

- All buttons work and go to the correct pages

- All error messages show when needed (e.g., missing input)

- App responds well on different screen sizes

Table 1: Functional Testing – Test Case Table

Functionality	Test Result
User Sign Up	Pass
User Login	Pass
Switch to Business Account	Pass
Business Registration	Pass
Listing Search	Pass
Save & Share Listings	Pass
Admin Login & Management	Pass
Error Handling	Pass
UI Navigation	Pass
Firebase Integration	Pass

Limitations

Several constraints affect the current system implementation:

- Internet connectivity requirements limit functionality in areas with poor network coverage. While this reflects current technological constraints, it may restrict user adoption in some communities.
- Single business account limitation prevents users from managing multiple enterprises. This design decision simplified initial implementation but may require revision for broader deployment.
- Administrative structure relies on single super admin account, which could create management bottlenecks in larger communities. Expanding admin capabilities would improve scalability.

The system currently lacks user review functionality, which limits community feedback mechanisms. Future versions should incorporate rating and review capabilities.

Summary

Testing results demonstrate that the application successfully addresses core project objectives while identifying areas for future enhancement. The platform provides meaningful improvements in local service discovery while empowering small businesses with digital presence capabilities. Technical choices proved appropriate for community-scale deployments, though some limitations suggest directions for continued development.

CHAPTER FIVE SUMMARY, CONCLUSION AND RECOMMENDATION

Summary of the study

This project addressed a persistent challenge in many communities: the difficulty of discovering local services and the limited visibility of small businesses. We developed a mobile application that serves both service seekers and business owners, creating a platform for improved community connectivity.

The development process utilized modern tools including Flutter for mobile interfaces, Firebase for backend services, and React JS for administrative capabilities. Our approach emphasized simplicity and reliability, ensuring the platform could serve users with varying technical expertise.

Key achievements include successful implementation of location-based service discovery, user account management with business capabilities, comprehensive administrative oversight, and cross-platform mobile compatibility. Testing demonstrated that these features effectively address identified community needs.

Conclusion

Our research confirms that community-focused mobile applications can meaningfully improve local service discovery while supporting small business growth. The platform successfully bridges information gaps that traditional solutions often overlook.

The project demonstrates that accessible development tools can create sophisticated solutions without requiring extensive technical resources. This accessibility suggests potential for similar initiatives in other communities facing comparable challenges.

Technical implementation proved robust and scalable, with chosen tools providing reliable performance under various conditions. User interface design achieved appropriate balance between functionality and simplicity, making the platform accessible to diverse user groups.

Most importantly, the application creates value for multiple stakeholders: community members gain easier access to local services, business owners receive improved visibility, and administrators can maintain platform quality efficiently.

Recommendation

Future development should consider several enhancements based on our experience:

- Expanding administrative capabilities to support multiple moderators would improve scalability for larger communities. This could include role-based permissions and regional management assignments.

- Implementing user review and rating systems would provide valuable feedback mechanisms while helping users make informed decisions about services. This feature should include moderation capabilities to maintain quality.

- Adding push notification functionality could alert users about new businesses in their area or special promotions from saved businesses. This would increase user engagement while providing marketing opportunities for business owners.

- Map integration would provide visual service location information, complementing the current text-based discovery approach. This enhancement would particularly benefit users unfamiliar with community geography.

- Offline functionality development would extend platform usefulness in areas with inconsistent internet connectivity. Basic information browsing and business contact details could remain accessible without active connections.

- Consider conducting formal user experience studies to identify additional improvement opportunities. Direct user feedback would provide valuable insights for interface refinement and feature prioritization.

Future researchers might explore specialized versions for specific community types or investigate integration possibilities with existing municipal services or community organizations.

The platform foundation provides strong basis for continued enhancement while demonstrating immediate value for community service discovery and small business support.

REFERENCES

Björk, B. C., Roos, A., & Lauri, M. (2009). Scientific journal publishing: yearly volume and open access availability. Information Research, 14(1), 391.

Bojanova, I., Hurlburt, G., & Voas, J. (2014). Imagineering an Internet of anything. Computer, 47(6), 72-77.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13(3), 319-340.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction, 16(2-4), 97-166.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., & Zhu, X. X. (2018). Building instance classification using street view images. ISPRS Journal of Photogrammetry and Remote Sensing, 145, 44-59.

Khawas, C., & Shah, P. (2018). Application of firebase in android app development-a study. International Journal of Computer Applications, 179(46), 49-53.

Kumar, A., Adlakaha, A., & Mukherjee, K. (2021). The effect of perceived security and grievance redressal on continuance intention to use M-wallets in a developing country. International Journal of Bank Marketing, 36(7), 1170-1189.

Li, Q., & Longley, P. (2006). Geodemographic classification from user-generated content. In Proceedings of the 14th annual ACM international symposium on advances in geographic information systems (pp. 275-282).

Moroney, L. (2017). The definitive guide to Firebase. Apress.

Mountain, D., & Raper, J. (2001). Positioning techniques for location-based services. In Location-based services (pp. 33-53). Morgan Kaufmann.

Naudé, W. (2020). Artificial intelligence vs COVID-19: limitations, constraints and pitfalls. AI & Society, 35(3), 761-765.

Preece, J., Sharp, H., & Rogers, Y. (2015). Interaction design: Beyond human-computer interaction. John Wiley & Sons.

Pressman, R., & Maxim, B. (2014). Software engineering: a practitioner's approach. McGraw-Hill.

Schiller, J., & Voisard, A. (2004). Location-based services. Morgan Kaufmann.

Sommerville, I. (2011). Software engineering. 9th edition. Pearson.

UNCTAD. (2019). Digital Economy Report 2019: Value Creation and Capture: Implications for Developing Countries. United Nations Conference on Trade and Development.

Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. Decision Sciences, 39(2), 273-315.

Windmill, E., Zuniga, R., & Bejan, A. I. (2020). Learning Flutter SDK for creating Android and iOS mobile applications. In Proceedings of the 2020 ACM Southeast Conference (pp. 251-254).

World Bank. (2019). World Development Report 2019: The Changing Nature of Work. The World Bank.

Wu, W. (2018). React native vs flutter, cross-platforms mobile application frameworks. Master's thesis, Metropolitan College of New York.

APPENDIX

Appendix A: Code Repository and System Setup Guide

Code Location

The complete source code for the ComLink project is hosted on GitHub:
https://github.com/danielallotey/Final_Year_Project

Repository Structure

- .admin_panel/   # React JS admin dashboard code
- .mobile_app/   # Flutter mobile application source code
- documentation/   # Project documentation, possibly draft report
- ui_ux_prototype/   # Mockups, wireframes, design assets
- README.md   # Setup instructions and repository overview

Requirements

- Firebase Project (Firestore, Authentication, Storage)

- Node.js (for Admin Panel)

- Flutter SDK (for Mobile App)

- Android Studio or VS Code

Setup Instructions

- Clone the Repository
- Configure Firebase
- Run the Admin Dashboard
- Launch the Mobile App

Appendix B: Turnitin Similarity Report

Appendix C: Firebase Firestore Database Schema

Collections and Key Fields

Users

- uid (string)
- name (string)
- email (string)
- avatarUrl (string)
- accountType (string: 'regular' or 'business')
- savedItems (array of business IDs)
- communityId (string)

Businesses

- businessId (string)
- name (string)
- ownerUid (string)
- categoryId (string)
- communityId (string)
- profileImageUrl (string)
- phone (string)
- description (string)
- createdAt (timestamp)

Categories

- categoryId (string)
- name (string)
- iconUrl (string)

Communities

- communityId (string)
- name (string)

Admins

- uid (string — hardcoded in the React app for Super Admin access)