9/5/2025

ITMDA3-B23
Modelling

Prototyping a mobile engagement app: Evaluating push notification strategies for improved customer engagement

Benjamin Dähn, Kieran Horsford,
NicolaasHuisman, Luca Karsten,
Hayden Muller, Durotimi Samuel
Supervisor: Ntombesisa Mateyisi
EDUVOS – CAPE TOWN – TYGER VALLEY

Table of Contents

1. Topic selection and use case description

The problem being addressed is that traditional email-based systems are not sufficient for small businesses to keep communication with their customers. Many larger businesses can use resources to develop and use alternative methods of communication, however small businesses often lack the ability and resources to do this for themselves.

User and their need:

Businesses:

Businesses need the ability to post updates and announcements to the platform to assist with business-customer communication

Customers:

Customers need to be able to follow their favourite businesses and make personalization decisions about how they receive push notifications.

The system is intended to facilitate communications between businesses and their customers. The application will do so with notifications which allow customers to make personalization decisions about how they prefer to receive notifications. Customers will be able to subscribe/follow their favourite businesses and make decisions about what notifications they want from the business. Businesses will be able to post various types of updates to send to their customers. This is going to be done through a mobile app and the goal is to close the gap between larger businesses and smaller local businesses by providing small businesses with a platform which is easy to use with their customers.

2. Functional and non-functional requirements

Functional requirements			
ID	**Category**	**Requirement description**	**Linked user perspective**
FR-AUT-01	Authentication	Customers, businesses, and admins must be able to register and log in securely	Login
FR-AUT-02	Authentication	The system must distinguish between customers, business, and admin accounts with different permissions	Login
FR-CON-01	Content	Customers must be able to view a feed of the latest updates from businesses they follow	Viewing updates on feed
FR-CON-02	Content	Customers must be able to browse products uploaded by businesses	Browse products
FR-CON-03	Content	Customers must be able to open a product link that takes them to an external site for purchase	Browse products
FR-CON-04	Content	Customers must be able to search for businesses by name or category	Search for businesses
FR-CON-05	Content	Businesses must be able to create new posts	Post a new update
FR-CON-06	Content	The system must screen posts for basic unacceptable content	Post a new update
FR-PRO-01	Profiles	Customers must be able to view a business profile	View business profile
FR-PRO-02	Profiles	Customers must be able to follow or unfollow businesses	Follow or unfollow a business
FR-PRO-03	Profiles	Customers and businesses must be able to customize general user settings for the app	Customize app settings

Functional requirements			
ID	**Category**	**Requirement description**	**Linked user perspective**
FR-PRO-04	Profiles	The system or admins must be able to verify actual businesses with certain provided details	Verify business account
FR-PRO-05	Profiles	Users must be able to update their account information	Update profile
FR-REV-01	Reviews	Customers must be able to read client reviews and ratings	Read client reviews
FR-REV-02	Reviews	Businesses must be able to respond to client reviews	Respond to reviews
FR-REV-03	Reviews	Customers must be able to write business reviews and rate a business	Write business review
FR-MAN-01	Management	Business users must be able to add, update, or remove product information manually, or via file uploads, or via database connection	Add or update products
FR-MAN-02	Management	Admins must be able to manage customer and business profiles	Manage profiles
FR-MAN-03	Management	Admins must be able to view and manage reported posts	Manage reported posts
FR-NOT-01	Notifications	Customers must receive push notifications when followed businesses create a new update	Receive notifications
FR-NOT-02	Notifications	Clicking a notification must take the customer directly to the relevant post	Receive notifications
FR-NOT-03	Notifications	When a business posts, followers must automatically receive push notifications	Post a new update
FR-NOT-04	Notifications	Customers must be able to customize preferences of the notifications they receive	Received notification customization

Functional requirements			
ID	**Category**	**Requirement description**	**Linked user perspective**
FR-NOT-05	Notifications	Businesses must be able to customize the style and content of the notifications they send	Post a new update
FR-ANA-01	Analytics	Businesses must be able to view subscriber engagement with notifications	View subscriber engagement
FR-ANA-02	Analytics	Businesses must be able to view product performance	Analyse product popularity
FR-ANA-03	Analytics	Admins must be able to view overall analytics of the app usage and performance	View app usage and performance

Table 2.1 – Functional requirements table

Non-functional requirements		
ID	**Category**	**Requirement description**
NFR-USA-01	Usability	Customers must be able to change notification preferences in 4 or less steps
NFR-USA-02	Usability	Businesses must be able to publish posts in 5 or less steps
NFR-MAI-01	Maintainability	The app must be built with reusable cross-platform code
NFR-MAI-02	Maintainability	Backend code must be modular
NFR-MAI-03	Maintainability	The system must be able to log errors
NFR-REL-01	Reliability	The system should aim to deliver 95% or more of notifications under normal device and network conditions
NFR-REL-02	Reliability	The system should aim for 98% or more uptime

Non-functional requirements		
ID	**Category**	**Requirement description**
NFR-SEC-01	Security	Businesses must not be able to send notifications to customers that do not follow them
NFR-SEC-02	Security	Customers and businesses must not be able to access any admin actions
NFR-SEC-03	Security	The system must use encrypted connections to keep all data transfers secure
NFR-SEC-04	Security	The system must use secure authentication methods for login and remembering user logged in devices
NFR-SEC-05	Security	The system should aim to detect at least 90% or more of basic unacceptable post content
NFR-PER-01	Performance	The system should aim to deliver notifications in 4 or less seconds under normal network conditions
NFR-PER-02	Performance	The system should aim to handle 300 or more concurrent users during prototyping
NFR-PER-03	Performance	The system should aim to load a normal user feed in 3 or less seconds

Table 2.2 – Non-functional requirements table

3. User stories or use case narrative

Subscriber Perspective

- Viewing updates on feed

  Narrative: As a subscriber, I'd open the app and see an updated feed with the latest news from the businesses I follow. I can scroll through and view new posts.

  The aim is for subscribers to stay up to date with the businesses they follow.

- Receive Notifications

  Narrative: I receive push notifications when a business I follow posts a new update. Clicking on this notification will take me to the update.

  The aim here is to get alerts of important updates

- View Business Profile

  Narrative: I tap on a business name from the feed to view its profile which will show the business description, products, and client reviews.

  The aim here is to learn more about a specific business.

- Browse Products

  Narrative: As a subscriber, I can navigate the business profile to browse through its product list where each product will have a brief description, and client reviews

  The aim is for the client to review offers from a specific business

- Read Client Reviews

  Narrative: I read reviews and ratings left by other clients to decide whether I would like to engage with a business or its products.

  The aim is to assist clients in making informed decisions based on other clients' experiences

- Follow or Unfollow a Business

  Narrative: I can follow a business I would like to receive updates from or unfollow a business I am no longer interested in.

  The aim of this is to allow clients to personalize their content feed

- Search for businesses

  Narrative: I use a search feature to find businesses. The results will allow me to view their profile and follow the business.

  The aim is to allow subscribers to discover new businesses

- Login / Register

  Narrative: I can securely register and log in so I can access my personalized account.

The aim is to authenticate and differentiate between account types

- Update Profile
  Narrative: As a user, I can update my profile to keep my account accurate and up to date.

  The aim is to allow users to maintain relevant account information.

- Write Business Review
  Narrative: As a subscriber, I can write a review and rate a business so that other users can benefit from my experience.

  The aim is to help the community make informed choices through shared feedback.

Business User Perspective

- Post a New Update
  Narrative: I can create a new post on my profile to share updates or promotions. Subscribers get notifications about the posts automatically.

  The aim of this is to inform subscribers of the latest news or promotions

- Add or Update Products
  Narrative: As a business user, I add or update products, prices, and images so subscribers may see the latest offers

- View Subscriber Engagement
  Narrative: As a business user, I can review nuances such as the number of followers, views, and comments to gauge how my updates are being engaged with. The goal is to understand customers' interactions and interests.

- Respond to Reviews
  Narrative: I respond to client reviews by thanking them for their feedback, or concerns, which improves the relationship between the business and client

- Analyse Product Popularity
  Narrative: I can review product performance nuances, such as views, clicks, or purchases to identify popular products and make less popular offers better. The purpose of this would be to assist me in making informed business decisions.

- Login / Register
  Narrative: I can securely register and log in so I can access my personalized account.

  The aim is to authenticate and differentiate between account types

- Update Profile

  Narrative: As a business user, I can update my profile to keep my account accurate and up to date.

  The aim is to allow users to maintain relevant account information.

Admin Perspective

- Manage Profiles

  Narrative: I can view and manage customer and business profiles to ensure accurate information and compliance with platform policies.

  The aim is to maintain the platform's integrity by ensuring all accounts are valid and up to date.

- Verify Business Accounts

  Narrative: I can verify businesses by checking the details they provide so that subscribers know they are engaging with legitimate businesses.

  The aim is to build trust between businesses and subscribers.

- Manage Reported Posts

  Narrative: I can review posts flagged by subscribers or the system and take appropriate action.

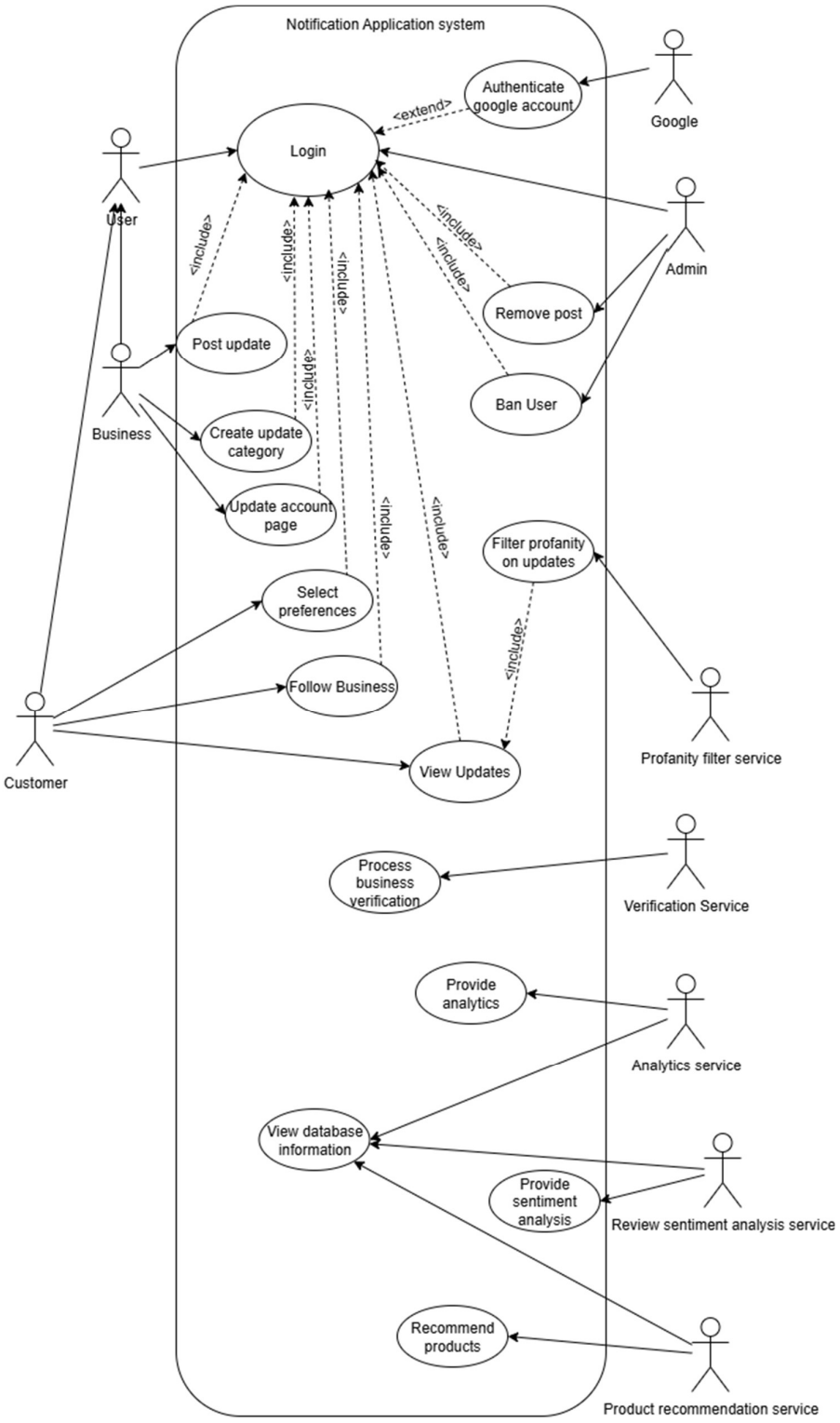
  The aim is to keep the platform safe, reliable, and free from inappropriate content.

- View Overall Analytics

  Narrative: I can access analytics showing app usage, performance, and engagement trends across all users.

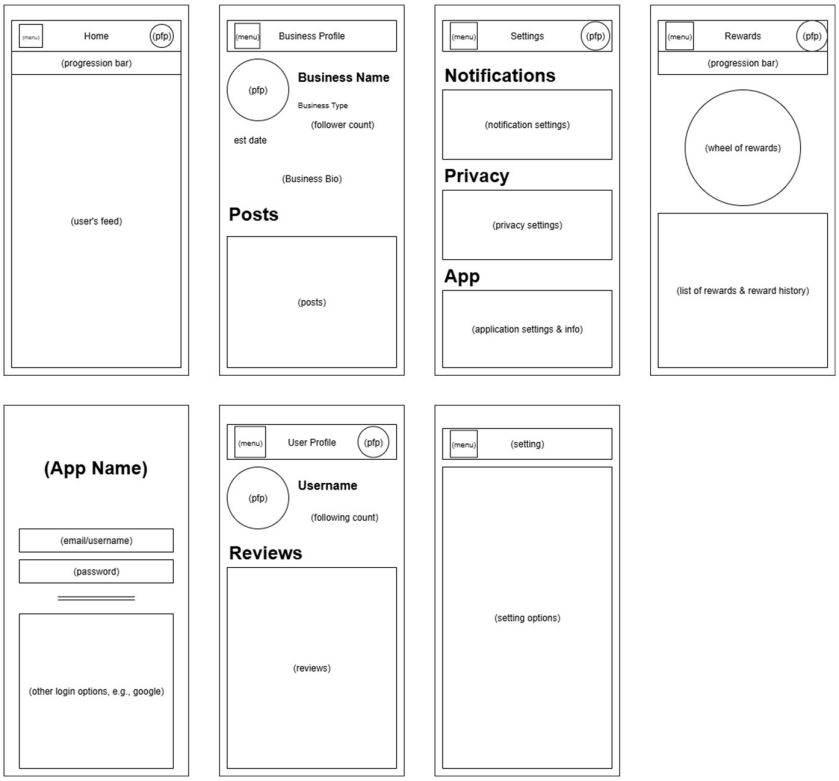
  The aim is to monitor platform health and support future improvements.

4. Use case diagram

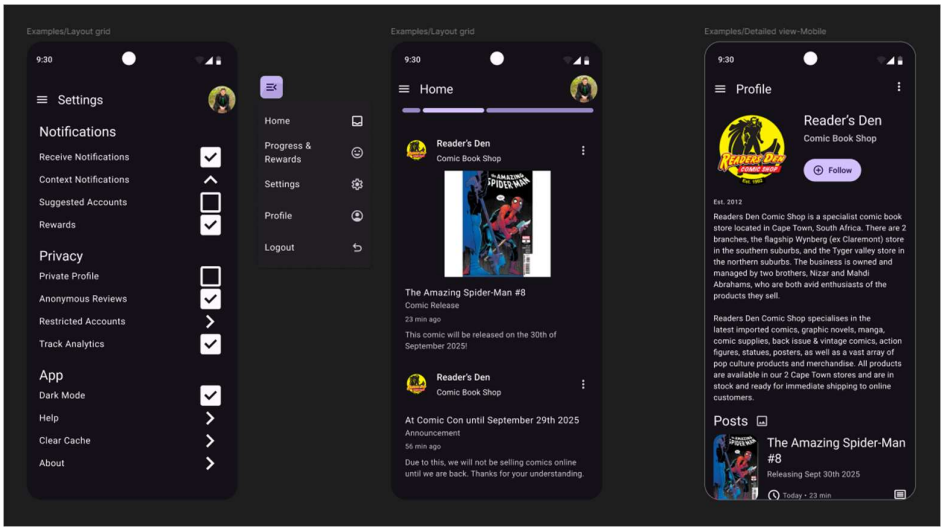


5. Wireframes and screen mockups

5.1. Wireframes



5.2. Screen mockups



6. Architecture design

The system is designed with the principles of mobile-first and serverless architecture, Flutter (Dart) is being used for the frontend and Firebase will be used for the backend and infrastructure services.

This application will allow businesses to make post, post updates regarding stock or offers, manage products, and respond to reviews, while subscribers follow businesses, view a personalized feed, like posts and add reactions and receive real-time notifications

Frontend

For the frontend Flutter is being used since it supports cross platform support development for Android and IOS.

Subscriber UI: feed of businesses posts, user/business profiles, product browsing, following, liking, reactions, reviews

Business UI: Create post, update products, view analytics (followers, comments, like, reviews), create notifications

Firebase SDKs

- Firebase Auth & Flutter Google Login plugin – For Login
- Firestore – NoSql Database
- Firebase Cloud Messaging – For Notifications
- Firebase Analytics – For logging in-app events such as post views, likes, and notification interactions (later exported to BigQuery for reporting
- Cloud Storage for Firebase – Saving Media

Real-time sync via Firestore listeners.

Backend

The backend will be made up out of firebase serverless services and External Microservices.

Firestore a NoSql (Document) Database will be used.

Real-time sync supports instant feed updates and engagement tracking.

Firebase Authentication

- Users/Businesses can sign in with email and password
- Can Login with Flutter Google Sign-in plugin
- Roles for users of app: users (subscribers), businesses and admins

- Firebase Auth handles the sign-up, and the Verification External service needs to approve new business accounts before the business role is activated

Firebase Cloud Functions

- Sending notifications for new posts
- Updating counters for followers and likes, reactions
- Enforce frequency of notifications and quiet times for notifications

Firebase Cloud Messaging

- Send push notifications for new posts from businesses
- Can send notifications to businesses regarding follows, reviews, like/reactions

Cloud storage

- Stores businesses posts images, products images and profile images

Firebase Analytics

- Tracks in app events such as post likes, reactions, reviews
- Tracks interaction with push notifications
- Data is exported to BigQuery and processes by Engagements analytics service

External Services

While Firebase provides the core backend infrastructure, additional services are added to enhance functionality and analytics:

Java – Business Recommendation Engine

- Input: Product catalogue (from Firestore products collection) + customer interactions (likes, views, followed business accounts).
- Output: Recommended: related posts, related products, related businesses.
- Purpose: Makes browsing better and more efficient for subscribers.
- Integration: Queried directly from Flutter as an API call, since recommendations must be returned in real time with low latency.

C# (ASP.NET Core Web API) – Engagement Analytics Service

- Input: Business ID.
- Output: JSON metrics like number of posts, average views per post, notification click-through rates, average likes per post etc.
- Purpose: Gives businesses insights into how effective their posts and notifications are.
- Integration: Queries BigQuery via a Web API service layer, as engagement analytics can be processed in batches and do not require real-time delivery

C# – Business Verification Admin Tool

- Input: Business verification requests (business name, email domain, documents).
- Output: Pass verifying business or fail, and business accounts rejected
- Purpose: Helps administrators validate businesses, for example, the API can auto-check that the business email domain matches the registered business name.
- Integration: Internal API that is called by the Admin Dashboard, Cloud Function is triggered marking new signups as pending until verified.

Python – Review Sentiment Analysis

- Input: Customer review text.
- Output: Positive/Negative/Neutral label for review
- Purpose: Overview of reviews to help see what customer feelings towards business is
- Integration: Triggered by a Cloud Function whenever a new review is submitted, so sentiment is analysed and stored in Firestore immediately.

Python – Profanity/Spam Filter for Posts

- Input: Business post text (title + body).
- Output: “Clean” or “Flagged” result.
- Purpose: Provides basic content moderation, preventing spam, offensive language, or inappropriate posts before publishing to feed.
- Integration: Triggered by a Cloud Function whenever a post is created, ensuring flagged posts are blocked or sent to moderation before appearing on the user’s feed.

Communication between Components

Login:

- Users or Business sign in using Firebase Auth or Google Sign-in
- Firestore saves profile and assigns user a role of subscriber or business

Business Verification:

- A new business account signs up with Firebase Auth
- Firestore saves account as pending verification form external service
- Verification Admin Tool service verifies the request
- If approves the business account is activated in Firebase Auth

Business Posting:

- Business creates post and it is stored in Firestore database
- Profanity/Spam Filter service checks post

- If post is clean continue, if not send to moderation
- Cloud function sends notifications
- Feed updates for users

User engagement:

- Users can view posts on feed: add reactions, like posts
- Cloud updates for like count and reactions

Business Reviews:

- On businesses profiles users can leave a review of the business
- Businesses can respond to reviews left on page

Notifications:

- Users can Receive notifications for new posts, limited time offers or replies to reviews
- Businesses get notifications for reviews, follows, likes

Analytics:

- Firebase Analytics log engagement data with notifications and posts
- C# Analytics Service processes this data via BigQuery
- Businesses have analytics view dashboard with all engagement information.

Infrastructure Considerations

Scalability

- Firestore auto-scales with follower counts and engagement.
- Cloud Functions scale horizontally and is event driven.
- Cloud Tasks queue ensures smooth push notification fan-out.

Security

- Firestore Security Rules restrict access by role.
- Firebase ID tokens secure communication with external services
- Business Verification tool verifies if it is a legitimate business reducing possible fraud
- Profanity/Spam filter prevents potential inappropriate content and spam.
- Subscribers can only modify their own comments/likes/reviews.
- Business users restricted to managing their own posts/products.

Performance

- Indexed queries for fast feed loading
- Aggregation counters for like/comment counts to avoid expensive queries.
- Batched writes for high-volume notifications.

Reliability

- Notifications delivered via at-least-once model with deduplication.
- Cloud Logging & Monitoring for observability.

7. Data design

Data Model:

The system is going to use Firebase Firestore to manage users, posts, products, follows, reviews, notifications and admin features. The data model will also support Google sign-in, role-based access, ratings as well as optional comments and responses, post moderation, business verification, notifications, event logging and audit trails.

Data Collection and Document Structure:

Collection	Document ID	Key Fields
users	{uid}	name, email, role, status, fcmToken, preferences, consent, auth, timezone, createdAt, businessProfile, verified, verifiedAt
Users/{uid}/{devices}	{deviceId}	token, platform, notifiationsEnabled, lastSeenAt, appVersion, topics
posts	{postId}	businessId, title, content, media, categoryId, tags, metrics, status, createdAt
posts/{postId}/reactions	{uid}	type, createdAt
categories	{categoryId}	businessId, name, description, createdAt
follows	{followId}	customerId, businessId, createdAt
products	{productId}	businessId, name, description, price, imageUrl, categoryId, createdAt
reviews	{reviewId}	customerId, businessId, productid, rating, comment, response, createdAt

Collection	Document ID	Key Fields
notifications	{nId}	businessId, title, body, deepLink, channel, target, sendAt, maxPerDay, experimentId, createdBy, createdAt
notificationEvents	{eventId}	uid, nid, event, ts, latencyMs
policies	global	maxPerUserPerDay, minGapMinutes, quietHoursDefault, channelCaps, updatedAt
auditLogs	{logId}	actorUserId, action, details, ts

Relationships:

Business – Content

- A business user can create many posts, categories and products.
- Posts/products may optionally link to a category.

Followers

- Users can follow many businesses, and each business can have many followers.
- Follows make up the customer feed and define the notification audience.

Reviews

- Users can leave one review per business; a star rating with an optional comment.
- Reviews will link the user to the business.
- Businesses cannot be reviewed at the post level meaning no product/post reviews are available.

Post Reactions

- Users can add a reaction to a post along with a reaction type.

Devices

- Each user may have multiple devices under users/{uid}/devices.
- Notification will deliver to all eligible devices for a followed user.

Notifications and Events

- Businesses author notifications; each notification belongs to a business.
- Audience is followers of that business only.
- Delivery and user interactions produce notificationEvents tied to both notification {nid} and recipient {uid}.

Administration and Audit

- Admins can verify businesses, moderate posts, ban users and edit policies
- All admin actions are recorded in auditLogs for traceability.