

Customer Enquiry & Follow-up Architecture

Turn scattered conversations into assigned work, visible follow-ups, and a connected customer history.

Capture

Assign

Follow up

Track

Review

Abdul Rehman

Solution Architect | SaaS Architecture & .NET Engineering

Implemented foundations, target architecture and planned validation are distinguished throughout.

Every enquiry needs an owner

The initial focus connects enquiry capture, assignment, follow-up and visibility for smaller businesses.

Where the handoff breaks

Enquiries arrive through messaging, calls and social channels. Disconnected chats, spreadsheets and notebooks make lead ownership unclear and follow-ups easy to miss.

A focused first workflow

Capture the enquiry as a structured lead. Assign responsibility, schedule the next action, and show the owner which customers and tasks are still waiting.

What needs to be visible

The customer and source; the assigned employee or team; the next action and due time; pipeline progress; and the connected activity history.

Delivery direction

Start with a narrow WhatsApp lead workflow. Validate it through customer pilots before expanding into broader ERP capabilities and additional integrations.

Conversion, response time and follow-up activity are planned validation measures, not reported customer outcomes.

Make the next action visible

The proposed workflow connects customer communication, CRM, staff responsibility and owner visibility.

01 Receive an enquiry

A customer contacts the business through messaging, web, phone or another channel.

02 Create a lead

Connect contact details, source and conversation context to a structured record.

03 Assign ownership

Give an employee or team responsibility, manually or through an automation rule.

04 Schedule follow-up

Create a task, reminder or response deadline for the next action.

05 Update the pipeline

Record progress through configurable sales stages.

06 Retain activity

Keep notes, messages, appointments and changes with the customer record.

07 Surface pending work

Show unattended leads, overdue actions and pipeline movement to the owner.

08 Review the outcome

Use response, conversion and activity measures to evaluate the workflow.

Example stages: New, Contacted, Qualified, Won or Lost. Workflow and pilot validation remain part of staged delivery.

A modular SaaS foundation

Business modules share tenant, identity, audit and integration capabilities; external providers remain replaceable.

Design boundary

Keep the customer and lead domain independent of one messaging provider. External channels connect through adapters and APIs.

Scope of this diagram

A target architecture containing planned capabilities. Logical modules do not imply separately deployed microservices or a complete delivered feature set.

Channels

Messaging providers, web, phone, social channels and manual entry.

Experience

Business portal, owner dashboard, staff workspace and mobile-friendly UI.

Business modules

CRM, contacts, leads, pipeline, tasks, appointments, products, campaigns, automation, analytics, users and billing.

Shared platform capabilities

Identity, tenant management, notifications, audit, integration APIs and background jobs.

Technology foundation

.NET, Angular, PostgreSQL, REST APIs, cloud deployment patterns and CI/CD.

Subscription and configuration models are planned alongside the initial enquiry workflow.

A workspace for each business

Tenant, role and permission inform authorization; adapters define the boundary with external systems.

Tenant workspace

- Business-specific users and roles
- Branches and teams
- Configurable sales pipelines
- Tenant-specific automation
- Data isolation and planned subscription boundaries

Security and history

- Authenticated access
- Role-based permissions
- Activity and change history
- Protected integration credentials
- Controlled administration

Integration design

- Messaging and social adapters
- Webhooks
- Email and notifications
- Payment-provider boundaries
- Future accounting and ERP APIs

Engineering considerations

Translate provider events into business activity without coupling the domain to provider details. Define how reassignment and stage changes affect pending tasks. Preserve tenant context in configuration and administration.

A foundation for a focused pilot

My contribution combines product architecture, domain design and hands-on engineering.

Implemented foundation

Developed .NET backend services, integrated Angular experiences, worked with PostgreSQL and relational modelling, implemented authentication and authorization, and built reusable platform capabilities.

Architecture and design

Defined modular SaaS, CRM and pipeline domains, tenant boundaries, and the integration model. Designed REST APIs, application workflows, automation concepts, background processing and cloud deployment patterns.

Planned validation

Prioritize customer pilots, review missed leads and follow-up activity, and validate real operations before broader feature expansion. Subscription and configuration models remain planned work.

Scope of the evidence

Foundations are implemented. Customer pilots are planned to validate the workflow; conversion, response time and follow-up activity are intended evaluation measures.

Technologies and concepts

.NET, C#, ASP.NET Core, REST APIs, Angular, TypeScript, PostgreSQL, relational modelling, modular SaaS, tenant-aware authorization and background processing.

Architecture, engineering decisions, and delivery scope.