# 第一章 绪论

## 1.1 研究背景与意义

当前我国许多中小型生产企业在客户管理方面仍处于较为初级的阶段，广泛存在着流程缺乏标准化、客户资料分散、沟通不连续等现象。尤其在客户资料维护方面，不少企业仍依赖传统的Excel表格进行客户信息记录和销售跟进安排。这种方式在企业初创阶段或数据量较小时尚可应付，但随着客户数量和业务复杂度的提升，其低效、易错、难以协作的弊端日益显现。

此外，随着国内制造业逐步转型升级，企业面临日益复杂的销售链条与多样化的客户需求。客户个性化需求提升、销售周期拉长、协同环节增多等因素，使得传统的管理手段逐渐难以胜任。企业迫切需要一套能够灵活适配自身流程、实现销售过程数字化、客户数据结构化的客户关系管理系统（CRM）来支撑业务拓展与客户维护。

## 1.2 生产型企业客户关系管理现状及痛点

目前市面上的主流CRM产品多以通用型功能为主，虽然模块丰富、系统稳定，但在面对制造业尤其是新材料等高度定制化行业时，往往“水土不服”：一方面，这些系统难以贴合“从线索到订单、生产、发货、财务处理”的复杂流程，功能配置上存在冗余或缺失；另一方面，系统界面设计较为陈旧、操作复杂、部署实施周期长，增加了企业的使用门槛。此外，部分系统对AI智能的支持力度不足，无法根据销售人员的跟进行为或客户特征，提供个性化建议与预测辅助，降低了数据价值的转化效率。

调研发现，在某新材料生产企业的实际应用场景中，这些通用CRM系统在流程适配、操作便捷性、业务洞察能力等方面均存在显著短板。因此，亟需设计并构建一套更具垂直适配性、更注重用户体验、并能融合AI智能能力的生产型企业CRM解决方案。

## 1.3 研究目标与创新点

本研究以提升中小型生产企业客户管理水平为核心，设计并实现了一套智能化、轻量级、可灵活部署的CRM系统，旨在解决传统方式存在的“流程碎片化、数据利用率低、部署成本高”等问题。系统在设计中聚焦以下几个方面的创新：

1. **垂直化流程定制能力**：作者通过6个月的企业现场调研，深入参与客户管理、订单处理、发货协同等流程，全面梳理企业各部门协同逻辑，基于实际流程进行系统建模，确保系统能够与业务无缝衔接；
2. **AI能力集成**：系统内嵌销售预测分析模块，支持成交概率评估、个性化跟进建议生成等功能，助力销售科学决策；
3. **双后端架构支持**：系统支持Java Spring Boot与Supabase双模式运行，满足企业从快速上线到安全部署的不同阶段需求；
4. **轻量化部署能力**：前端可部署至GitHub Pages，后端主要功能通过Supabase提供，企业无需单独购置服务器即可实现完整CRM功能；
5. **跨部门协同设计**：支持销售、生产、财务等多角色联动，通过任务提醒、客户共享、分析报表等机制打通企业内部协作壁垒。

## 1.4 本项目优势概述

相比传统CRM解决方案，本项目在“垂直业务适配”、“部署成本控制”、“智能辅助决策”三个方面展现出明显优势。通过真实企业的调研与迭代开发，系统功能更贴近一线用户实际操作习惯；双后端设计让企业根据自身资源选择最适部署方案；AI能力则使得客户数据具备可解释性与行动指引，真正实现“数据驱动销售”的现代管理目标。

本项目不仅具备较强的应用价值和推广潜力，也可为其他垂直行业CRM系统的设计提供方法参考。

# 第二章 相关技术与理论基础

## 2.1 客户关系管理（CRM）理论框架

客户关系管理（Customer Relationship Management, CRM）是一种以客户为中心的管理理念，其核心目标是通过系统化管理客户信息、优化客户关系、提升客户满意度与忠诚度，从而实现企业的可持续盈利。CRM系统将销售、市场、客户服务等多个业务流程进行整合，实现客户数据的统一管理和深度挖掘。对于生产型企业而言，CRM不仅是销售支持工具，更是连接销售与生产、服务、物流等多部门的协同平台，是推动企业数字化转型的重要支撑。

## 2.2 生产型企业业务流程分析

与通用型服务业不同，生产型企业的客户管理具有显著行业特性：客户需求定制化程度高、销售周期长、订单处理流程复杂，涉及合同签订、生产计划、发货安排、发票管理等多个环节。因此，传统CRM系统常常难以完全适配生产企业的流程需求。通过前期6个月的实地调研，我们梳理了某新材料企业的全流程业务链条，识别出信息孤岛、响应滞后、客户画像模糊等核心痛点。据此，本系统对业务流程进行了重构，实现从销售线索到发货开票的闭环覆盖。

## 2.3 前端技术选型：Vue3 与 Element Plus 在客户管理系统中的实践优势

Vue3 是一款轻量级、高性能的渐进式 JavaScript 框架，支持响应式数据绑定和组合式API编程，提升代码复用性和开发效率。Element Plus 是基于 Vue3 的企业级 UI 组件库，提供丰富的组件与主题配置，支持快速构建统一风格的企业系统界面。在本系统中，Vue3 + Element Plus 提供了流畅的用户交互体验、模块化开发方式和高可维护性，是前端界面构建的首选方案。 因此，Vue3 + Element Plus 的组合不仅提升了系统的界面现代化程度，还极大提高了前端开发效率，是实现企业级客户交互系统的理想选择。

## 2.4 后端架构支撑：Java Spring Boot 的稳定性与安全保障

Spring Boot 是基于 Spring 框架的快速开发平台，支持自动化配置、内嵌式服务运行、微服务架构拓展等特性，极大简化了后端开发流程。其高度模块化设计使得本系统能够灵活对接数据库、安全认证、业务逻辑处理等模块，保障后端系统的稳定性与可扩展性。此外，结合 Spring Security 实现用户身份认证与权限控制，为数据安全提供全面保护。 鉴于系统需保障数据一致性与接口响应速度，Spring Boot 的高度整合能力与企业级成熟度成为本系统首选后端架构。

## 2.5 快速开发与部署平台：Supabase 在敏捷开发中的作用

Supabase 是一个开源后端即服务（BaaS）平台，基于 PostgreSQL 提供数据存储、认证、实时订阅等能力。通过其内置的 RESTful API 与数据库管理工具，开发者可快速实现业务数据的创建、查询、更新与权限管理。在本项目中，Supabase 作为辅助平台用于构建原型、管理客户数据、开发实时数据接口，显著提升了开发效率与系统部署灵活性。 综合来看，Supabase 有效降低了系统原型的构建周期，并增强了系统的迭代开发能力，适配本项目快速上线与定制化需求。

尽管 Java 后端具备稳定性高、可扩展性强等显著优势，但其开发与部署流程相对复杂，对开发人员的技术要求较高，特别是在持续集成、配置管理、系统运维等方面，中小企业往往缺乏专职IT团队来进行支持。同时，许多中小生产企业的销售流程存在个性化强、迭代频繁的特点，亟需一种更为灵活、轻量、低门槛的技术方案来快速响应业务变化。

为此，本系统引入 Supabase 平台，作为快速开发和数据接口构建的重要补充手段。Supabase 基于 PostgreSQL 构建，具备以下优势：

* 提供开箱即用的 RESTful API 和数据库管理功能；
* 支持实时数据订阅、用户权限控制、文件存储等核心能力；
* 开发者可通过可视化界面和脚本方式快速构建原型和后端逻辑；
* 简化了系统的部署流程，大幅降低中小企业在早期阶段使用系统的技术门槛。

### 2.5.1 双接口支持的架构亮点

值得强调的是，本系统支持 **双模式接口结构**：一方面保留基于 Java Spring Boot 实现的后端主服务，确保高性能、大规模并发、高安全性的核心业务处理能力；另一方面集成 Supabase 作为轻量级子服务，用于快速上线、敏捷迭代、原型设计等业务场景。这种架构模式具备以下优势：

* **灵活性强**：当企业需求稳定、系统部署正式化时使用 Java 后端；当业务迭代快、资源有限时使用 Supabase 接口快速响应。
* **开发成本低**：部分低复杂度的功能（如客户留言、日志记录等）可直接通过 Supabase 开发，节约开发成本。
* **部署便捷**：支持在无运维人员情况下上线运行，提高中小企业落地系统的可行性。

因此，双接口架构不仅是本系统的一大技术亮点，也是其服务中小企业场景的重要体现，有效提升了系统的适配性、可维护性与推广价值。

## 2.6 智能化客户管理：AI 技术在 CRM 系统中的融合与实践

人工智能技术在现代CRM系统中扮演着日益关键的角色，特别是在客户行为分析、销售预测、个性化推荐等方面具有广泛应用前景。本系统引入机器学习模型实现客户跟进策略分析与复购预测功能。系统可根据客户历史沟通记录、订单行为、响应频率等多维度数据构建客户画像，进而生成个性化销售建议。同时，结合自然语言处理（NLP）技术辅助销售人员编写沟通纪要，提升客户沟通质量与数据结构化水平。

## 2.7 线上部署与轻量化运行优势

本系统的演示版本已上线至 GitHub Pages（访问地址：https://guomengtao.github.io/web-time-tracker/），采用 Supabase 作为主要后端服务。该架构显著降低了系统的运维成本与部署复杂度，尤其适合中小企业在缺乏专业 IT 团队的情况下独立运行和维护CRM系统。

通过 Supabase 提供的数据库视图（View）功能，系统实现了对复杂数据结构的高效处理与关联查询，有效弥补了传统关系型数据库在前端使用中的复杂性问题。系统无需配置服务器与传统 Java 后端，即可通过静态部署实现功能完整的客户管理与销售跟进流程。

此外，系统还扩展集成了企业内部沟通工具，包括自研的客服聊天系统与视频系统，支持客户与销售之间的实时沟通，销售部门与生产、财务部门之间的跨部门沟通与提醒。这些功能可作为在线CRM系统的重要补充，满足现代企业对在线协同办公的多样化需求。

本系统已集成“AI 跟进助理”功能，作为智能客户管理模块的重要组成部分。该助理可在销售跟进页面中自动分析客户历史行为数据、沟通频次、订单记录等信息，实时生成个性化的跟进建议、阶段判断和预计成交概率，为销售人员提供具体可执行的操作方案。此功能有效提升了销售效率与客户转化成功率，充分体现了本系统在 AI 智能化方向的深度融合成果。

# 第三章 系统需求分析与总体设计

## 3.1 需求调研方法与过程

本系统在开发前期，采用实地参与、访谈记录、流程图建模等多种方式，深入某新材料生产企业开展为期6个月的业务调研。调研内容涵盖客户开发、订单处理、物流协同、财务开票、客户回访等全流程环节。通过对业务痛点的归纳总结，形成系统需求文档，并在企业一线员工反馈基础上不断修正和细化功能模型。

## 3.2 功能性需求分析

根据业务流程调研结果，系统需满足以下核心功能：

* **线索管理模块**：用于登记客户初始信息、来源渠道、需求方向，支持线索分配与跟进记录。
* **客户信息管理模块**：实现客户档案的新增、编辑、查询，支持客户分类、历史行为查询等。
* **订单管理模块**：支持订单创建、审批、生产状态追踪、发货信息录入与发票开具。
* **智能分析模块**：集成AI算法，根据客户行为数据生成复购预测、成交概率评分等辅助决策信息。
* **权限与通知模块**：支持基于角色的访问控制，自动提醒销售人员待跟进客户、已发货订单等事项。
* **数据可视化模块**：以图表形式展示客户增长、销售趋势、复购周期等运营指标，辅助管理层决策。

## 3.3 非功能性需求分析

* **系统响应性**：在并发访问情况下，关键功能响应时间控制在2秒以内。
* **系统安全性**：基于Spring Security和RBAC权限模型，保障数据访问安全。
* **可维护性与可扩展性**：系统模块划分清晰，支持后续功能迭代与新模块接入。
* **跨平台兼容性**：前端系统支持PC和移动端响应式访问，提升使用场景适配性。

## 3.4 系统总体架构设计

为实现灵活部署与高效响应，系统采用前后端分离架构，并支持双后端接口方式：

* **前端**：基于 Vue3 + Element Plus 构建，负责用户交互、界面渲染。
* **后端主服务**：Java Spring Boot 提供核心业务逻辑、数据库访问、安全控制等功能。
* **后端辅助服务**：Supabase 提供轻量级数据管理与原型接口，适配快速部署场景。
* **数据库**：采用MySQL作为主要数据存储，部分功能集成Supabase PostgreSQL接口。

系统架构划分为四层结构：

1. 表现层（前端）：负责用户界面与交互逻辑。
2. 控制层（Controller）：接收前端请求，调用业务逻辑层处理。
3. 业务逻辑层（Service）：实现业务规则、数据流转与AI推理服务调用。
4. 接口访问层（API 调度逻辑）：前端通过统一的 API 管理模块封装接口请求逻辑，支持双后端模式切换（Java Spring Boot 与 Supabase）。系统可配置默认优先访问主后端，当接口未响应或不可用时，自动切换至备用接口，确保系统的高可用性与弹性。该机制在前端 Vue 中实现，提升了系统部署灵活性与多场景适配能力。

系统整体架构设计如图所示（略），为后续模块设计与系统实现提供坚实基础。

# 第四章 系统功能模块设计与实现

## 4.1 模块划分概述

本系统基于业务流程的调研成果与系统总体架构设计，采用模块化开发方式，划分为以下核心功能模块：

1. **线索管理模块**：负责潜在客户信息采集、来源追踪、初步需求记录与分配机制。
2. **客户管理模块**：管理客户基本资料、联系人信息、交互记录、客户分级与标签管理。
3. **订单管理模块**：从订单创建到发货与开票的全流程业务处理，支持状态跟踪与审批流程。
4. **智能分析模块**：AI驱动的复购预测、销售漏斗分析、客户评分系统。
5. **报表与数据可视化模块**：运营数据展示与导出，支持多维度查询与图表分析。
6. **权限与通知模块**：基于RBAC模型实现细粒度权限控制，提供操作日志与任务提醒功能。

每个模块都由前端Vue组件、后端接口服务（Java或Supabase）、数据库表结构及辅助工具构成，模块之间通过统一API进行集成与通信。

## 4.2 前端模块实现设计

前端采用 Vue3 框架构建，组件组织清晰，按功能模块分布在 src/views 与 src/components 中。主要页面包括：

* LeadsAdd.vue / LeadsList.vue：线索录入与管理界面。
* Customers.vue / CustomerAddPage.vue：客户管理与详情查看。
* OrdersList.vue：订单信息查询、状态更新、发货登记等操作。
* SalesFunnel.vue / Reports.vue：智能分析与销售报表呈现。
* Login.vue / User.vue：用户身份认证与管理。

所有页面均通过 router/index.js 实现路由导航，配合 Layout.vue 和 TopBar.vue 提供统一布局风格，提升整体用户体验。

## 4.3 后端接口与逻辑实现

### 4.3.1 Java Spring Boot 接口实现

核心业务流程由 Spring Boot 提供服务，包括：

* 客户信息 CRUD 接口（Controller + Service + Repository 层分离）
* 订单处理流程（含状态管理、发票逻辑等）
* 用户登录与权限校验（基于 Spring Security）

所有接口统一遵循 RESTful 规范，数据格式采用 JSON，前后端通过 Axios 进行通信。

### 4.3.2 Supabase 接口适配逻辑

对于低复杂度、迭代频繁的功能，如留言管理、日志记录、快速原型验证，系统支持直接通过 supabaseClient.js 接口进行数据操作：

* 使用 Supabase 提供的 API 快速实现数据读写；
* 利用 Supabase 的 Row Level Security 配置，实现前端可控的数据权限管理；
* 使用 Vue 中封装的接口选择逻辑，根据配置项动态选择 Java 或 Supabase 接口。

该机制极大提升了系统开发效率，特别适用于资源受限的中小企业场景。

## 4.4 数据访问层（DAO）与接口封装

* Java后端采用 JPA/MyBatis 进行数据访问封装，数据库连接池采用 HikariCP，提高并发处理能力；
* Supabase 接口通过 supabaseClient.js 封装调用逻辑，并在 apiClient.js 中统一出口；
* 所有 API 封装均支持异常捕捉与统一提示，提高系统健壮性与用户体验。

特别设计的 **双数据访问机制** 允许开发者或管理员配置默认数据源，当前端访问一个模块接口失败时，系统自动尝试备用数据源（例如优先访问 Java 后端，若不可用则自动 fallback 至 Supabase）。

## 4.5 模块功能示例：客户管理模块

以客户管理模块为例，其实现流程如下：

1. 客户信息由销售人员在处理销售线索时转化生成，并在转化过程中需同时登记至少一个订单；
2. 表单数据提交至后端接口（默认调用 Java 接口）；
3. 后端验证字段合法性后，将数据持久化至数据库；
4. 系统自动识别客户行业与历史标签，返回个性化客户画像建议；
5. 页面展示录入成功提示，跳转至客户详情页。

## 4.6 模块功能示例：智能分析模块

智能分析模块基于历史数据与机器学习算法，实现销售预测与行为洞察：

* 销售漏斗：展示各阶段客户转化率；
* 客户评分：结合复购频率、响应速度、订单金额等打分；
* 成交建议：结合AI模型与历史相似案例，提供成单路径提示；
* 复购预测：基于时间序列预测客户下一次下单时间。

上述分析结果通过图表或标签展示在 Dashboard.vue 与 Reports.vue 中，销售人员可直观获取业务状态。

第4章对各模块设计与实现进行详细说明，为后续测试与部署提供功能依据。

# 第五章 系统实现

## 5.1 开发环境与技术选型说明

系统采用前后端分离架构，开发环境配置如下： • 前端技术：Vue3 + Element Plus + Vite + JavaScript • 后端技术：Spring Boot（Java）、Supabase（PostgreSQL） • 部署环境：GitHub Pages（前端）、Supabase 云平台（后端） • 开发工具：VS Code、IntelliJ IDEA、Postman、Supabase Studio

此组合兼顾了开发效率、系统性能与部署灵活性，特别适配中小企业轻量级CRM需求。

## 5.2 前端实现（Vue3 + Element Plus）

前端采用 Vue3 框架构建，配合 Element Plus 提供一致、清晰、美观的交互界面。结构划分如下： • views/：存放业务页面（如 Customers.vue、OrdersList.vue） • components/：通用组件（如 CustomerAdd.vue） • layout/：页面布局组件（如 SideMenu.vue、TopBar.vue） • router/index.js：配置页面路由 • apiClient.js：统一封装所有 API 请求，支持动态选择后端（Java 或 Supabase）

所有页面基于模块化开发，支持响应式布局与移动端适配。

## 5.3 后端实现（Spring Boot）

后端核心服务由 Spring Boot 提供，主要功能包括： • 客户与订单的增删改查 • 销售线索到客户的转化流程 • 用户登录与权限控制（基于 Spring Security） • 销售数据分析接口

接口遵循 RESTful 风格，数据格式采用 JSON。使用 MyBatis 或 JPA 实现数据库访问，配合 HikariCP 连接池提升并发性能。

## 5.4 AI成单分析功能实现

本系统集成了 AI 跟进助理功能，当前已初步实现以下功能： • 销售跟进页面中，系统根据客户历史数据、订单记录、沟通频次等生成个性化跟进建议； • 显示预测成交概率、复购可能性； • 提供类似客户案例的处理建议，辅助销售人员决策。

该模块结合规则引擎与浅层机器学习算法，可持续扩展为深度智能推荐引擎。

## 5.5 Supabase 快速开发与部署

为提升开发效率并降低部署成本，系统引入 Supabase： • 提供可视化数据库管理与实时 API； • 快速创建表结构与权限控制； • 实现了部分子模块（如留言记录、简易订单查询）的快速开发； • 通过 PostgreSQL View 实现多表联合查询，提高前端数据整合效率； • 当前主系统运行基于 Supabase，配合 GitHub Pages 构建免费 CRM 平台。

## 5.6 系统集成与部署流程

### 5.6.1 双后端架构

系统支持 Java + Supabase 双后端接口机制，前端通过 apiClient.js 动态判断接口可用性： • 默认优先访问 Java 接口； • 如网络不可达或接口关闭，自动切换至 Supabase API； • 可在配置文件中设定默认后端类型，支持动态切换。

### 5.6.2 部署流程

• 前端使用 vite 打包，部署至 GitHub Pages，访问地址为：https://guomengtao.github.io/web-time-tracker/  
• Supabase 后端使用 PostgreSQL 提供数据库与认证服务，自动管理 API；  
• Java 后端在测试环境通过 Docker Compose 启动，后期可选私有化部署至云服务器；  
• 前后端均可实现开箱即用，无需额外服务器。

### 5.6.3 运维特点

• 无需专职 IT 运维，部署与升级简便；  
• 数据备份可通过 Supabase 控制台导出；  
• 支持多终端访问（桌面与移动浏览器）。

# 第六章 系统测试与部署

## 6.1 测试目标与原则

为确保本系统具备良好的稳定性、可用性与用户体验，测试工作遵循以下目标：

* 验证系统核心功能（客户管理、订单处理、线索转换、智能分析等）是否按预期工作；
* 测试前后端双接口（Java与Supabase）在不同网络与权限环境下的切换稳定性；
* 检查系统在多用户并发、数据大批量处理等场景下的性能；
* 保障用户数据访问的安全性与操作权限隔离；
* 验证跨平台适配性，包括PC端、移动端页面响应效果；
* 对系统部署过程进行复现性与可维护性测试。

## 6.2 测试环境与配置

| 项目 | 配置说明 |
| --- | --- |
| 前端 | Vue3 + Element Plus，部署于 GitHub Pages |
| 后端服务1 | Java Spring Boot（本地测试，使用PostgreSQL数据库） |
| 后端服务2 | Supabase 云端平台（PostgreSQL + RESTful API） |
| 浏览器环境 | Chrome、Edge、Safari 最新版本 |
| 测试工具 | Postman、Supabase Studio、JMeter、Lighthouse |
| 模拟设备 | Windows PC、Mac、Android 平板、iPhone 浏览器 |

## 6.3 测试类型与覆盖

### 6.3.1 功能测试

功能测试覆盖所有模块，包括线索管理、客户信息管理、订单处理、智能分析、权限控制等。

**测试示例：客户转化流程** - 测试目的：验证从线索添加、客户转化、订单登记是否流程闭环； - 测试步骤： 1. 添加销售线索； 2. 转化为客户信息； 3. 同时创建订单信息； 4. 查询客户详情，验证订单数据已关联； - 预期结果：客户信息正确生成，订单数据完整写入，界面跳转无误。

**测试示例：智能分析模块** - 验证 AI 推荐结果在客户数据变化后的实时更新； - 验证销售漏斗图、复购预测图表是否正确渲染；

### 6.3.2 性能测试

* 使用 JMeter 模拟50个用户并发提交客户数据与订单；
* 前端页面打开平均耗时低于1.2秒；
* 后端接口响应时间控制在2秒以内，Supabase接口响应基本稳定在1秒内；
* GitHub Pages 静态资源部署响应良好。

### 6.3.3 安全性与权限测试

* 角色划分测试：普通销售不可访问“用户管理”模块；
* 登录认证测试：错误密码3次自动锁定用户；
* 接口保护：通过 Postman 模拟SQL注入攻击被系统拦截。

## 6.4 部署方式与运维策略

本系统采用轻量部署策略，适配中小企业低成本运行需求。

### 6.4.1 前端部署

* 使用 vite 构建项目；
* 项目通过 gh-pages 分支发布到 GitHub Pages；
* 优势：无需服务器成本，免备案，适合个人与小团队快速部署。

### 6.4.2 后端部署方式

* Supabase 接口：使用官方平台进行注册、创建数据库、权限设置；
* Java接口（可选）：部署于本地服务器或云主机，可用于高负载业务处理。

### 6.4.3 部署流程简述

1. 配置前端环境变量，指向 Supabase 项目 URL 与 KEY；
2. 通过命令 npm run build 打包静态资源；
3. 执行 npm run deploy 自动部署至 GitHub Pages；
4. Supabase 提供数据接口，不需要额外部署服务器；
5. Java服务在测试环境使用 Docker Compose 启动，包含后端服务与数据库。

## 6.5 部署效果与用户反馈

系统上线后在测试企业中部署使用，得到以下反馈：

* 无需专职IT人员即可上线系统；
* 客户资料集中管理，业务协同效率明显提升；
* 智能推荐功能减少销售盲目跟进行为；
* 报表分析提升管理层对销售趋势的把控能力；
* GitHub Pages部署稳定，适配多设备访问，访问速度快，界面友好。

# 第七章 总结与展望

## 7.1 项目总结

本系统围绕生产型企业客户管理的实际痛点，采用现代化技术架构与智能化分析手段，设计并实现了一套具备高度垂直行业适配性的客户关系管理（CRM）系统。系统从客户线索采集、客户信息维护、订单管理、智能分析到跨部门沟通，全面覆盖生产企业销售与服务的核心环节。

在开发过程中，系统突出以下亮点：

* 深度垂直定制，依据实际企业流程打造功能结构；
* AI智能辅助决策，推动销售科学化转型；
* 支持双后端架构（Java + Supabase），适应不同资源环境；
* 前后端分离架构，支持敏捷开发与快速部署；
* 支持在线部署（GitHub Pages + Supabase），显著降低运维门槛。

系统在真实企业中试运行后反馈良好，已初步验证其功能适配性、稳定性与实用价值。

## 7.2 项目不足

尽管系统实现了预期功能，但仍存在以下不足：

1. **AI分析模型较为基础**：当前AI模块以规则与统计为主，缺乏深度学习支持；
2. **移动端优化不足**：界面以PC端为主，移动端适配体验仍需增强；
3. **深度集成有限**：与ERP、财务等系统尚未打通，系统协同程度需提升；
4. **权限管理仍需细化**：多角色精细化权限策略有待扩展。

## 7.3 未来展望

后续工作可从以下方向进行深化与拓展：

1. **AI智能化升级**：引入深度学习模型进行客户行为预测、成交概率建模；
2. **移动端适配**：开发专属移动App，支持销售人员移动办公与远程协同；
3. **业务流程集成**：与ERP、MES、OA等系统对接，形成全业务数字化闭环；
4. **可视化增强**：引入图形化工作流引擎、拖拽式报表工具提升管理效率；
5. **生态开放平台构建**：构建开放API，支持第三方服务快速接入；
6. **国际化与多语言支持**：为系统海外部署做好准备；
7. **数据合规与安全**：支持多租户与企业数据隔离，增强数据合规能力。

通过上述拓展方向，本系统有望演进为面向中小制造业的行业级智能客户运营平台，助力企业迈向数字化与智能化新阶段。

# 结束语

本研究面向生产型企业客户管理的实际需求，构建了一套智能化、轻量化、垂直化的CRM系统，显著提升了企业销售管理效率与客户关系维护水平。系统采用现代化技术栈，兼容双后端架构，集成AI智能分析与跨部门协同能力，具有良好的实用性与推广潜力。项目开发全过程结合真实企业场景调研，验证了其在中小制造业的落地能力。后续将继续优化AI能力、扩展系统集成、完善移动端体验，推动系统向更智能、更高效的方向演进。

# 参考文献

[1] 胡金宇.基于Spring Boot和Vue框架的企业绩效考核系统设计与实现[D].湖北师范大学,2023. [2] 杨东.基于Spring Boot的学生信息管理系统的设计与实现[D].武汉轻工大学,2021. [3] 何海明.基于Spring Boot的血管外科信息管理系统的设计与实现[D].华中科技大学,2021. [4] 陈新府豪.基于SpringBoot和Vue框架的创新方法推理系统的设计与实现[D].浙江理工大学,2022. [5] 王智伟.基于B/S架构的咨询项目管理系统的设计与实现[D].内蒙古科技大学,2022. [6] 宇玉瑜.企业物业云平台系统设计与实现[D].西安电子科技大学,2020. [7] 朱静.基于SpringBoot和Vue的全能型供电所管理系统设计与实现[D].华东师范大学,2024. [8] 付强.基于SpringBoot和Vue框架的农村信用信息管理系统的设计[D].河北工程大学,2023. [9] 唐双林.基于Vue和SpringBoot架构的智能推荐农产品团购销售系统[D].重庆三峡学院,2023. [10] 王娟.基于Spring Boot的矿产品智能追溯系统设计与实现[D].华东师范大学,2022. [11] 肖濛飞.X芯片公司关键客户识别与管理研究[D].电子科技大学,2024. [12] 韩锐.V公司社交客户管理系统建设方案研究[D].电子科技大学,2024. [13] 何冬梅.M公司客户关系管理改进研究[D].哈尔滨工业大学,2021. [14] 郭政男.通信行业客户资源管理系统的设计与实现[D].山东大学,2020. [15] 董超.基于SSH的保险客户管理系统设计与实现[D].内蒙古大学,2020.

# 致谢

本论文是在国家开放大学实验学院及指导教师的悉心指导下完成的。在整个系统设计与开发的过程中，得到了许多宝贵的帮助与支持，在此谨向所有给予帮助的人表示诚挚的感谢。

首先，感谢我的导师周朋红老师在选题确定、技术路线、论文结构等方面给予我全面的指导。导师严谨的治学态度、丰富的实践经验、耐心细致的指导，让我在论文写作与项目开发的过程中受益匪浅。

其次，感谢国家开放大学实验学院与魏公村学习中心提供的良好学习平台和资源环境，为项目调研、实验开发提供了必要的支持。

特别感谢参与调研企业的相关人员，尤其是某新材料生产企业的销售、生产、财务等部门的同仁，是他们在调研期间对企业业务流程的耐心讲解与配合，使我得以深入理解生产型企业的客户关系管理实情，为本系统的定制开发打下了坚实基础。

此外，感谢家人和朋友在整个学习和开发期间给予我的理解与鼓励。他们的支持是我坚持完成论文与系统的最大动力。

最后，感谢所有为本论文查阅过的参考文献作者以及开源社区的开发者们，是他们的研究成果和无私共享精神为本系统开发提供了重要的理论与技术支撑。

谨以此文，致谢所有给予我帮助与启发的人。

靳佩佩  
2025年6月21日