**🧾 SalesSphere CRM – Core Product Requirements Document (PRD)**

**🧠 TL;DR**

SalesSphere CRM is a modern CRM for B2B mid-market sales teams that combines intuitive UX, workflow efficiency, and powerful, customizable reporting. This PRD defines the core features and requirements for the initial product release, focusing on the needs of Sales Reps, Sales Managers, and Admins.

**🎯 Goals**

**✅ Business Goals**

* Reduce onboarding time by **50%** vs. competitors like Salesforce and HubSpot.
* Improve **workflow efficiency** for sales reps, especially in lead management and activity logging.
* Deliver **robust, customizable reporting** to improve forecasting accuracy and team performance visibility.
* Maintain **>99.9% uptime** and ensure scalable, secure performance.
* Enable **native integrations** with top marketing automation tools (e.g., HubSpot, Mailchimp, Marketo).

**👤 User Goals**

**Sales Reps:**

* Quickly add/update leads and contacts with minimal clicks.
* Log calls/emails/meetings with 1–2 taps or via automation.
* View their own pipeline and next actions at a glance.

**Sales Managers:**

* Track team performance via dashboards (activity, deal stage conversion, forecast).
* Easily create, save, and share custom reports.
* Assign and reassign leads to reps efficiently.

**Admins:**

* Manage users, permissions, and fields with no-code interfaces.
* Configure basic workflows and automate lead assignment.
* Ensure secure access and backup data regularly.

**🚫 Non-Goals**

* Advanced marketing automation (e.g., campaigns, journey builders).
* Highly customized workflow engines or 3rd-party marketplace integrations.
* Support for consumer/B2C sales teams.

**👥 User Stories**

**As a Sales Rep,** I want to:

* Add a new lead in under 30 seconds.
* Update deal stage from my phone after a meeting.
* See all my tasks, pipeline, and upcoming meetings in one dashboard.

**As a Sales Manager,** I want to:

* Track team activities (calls, meetings, emails) in a single view.
* Forecast next quarter’s revenue based on current deal stages.
* Create a report showing win rates by rep over the past 3 months.

**As a System Admin,** I want to:

* Invite new users and assign roles with predefined permissions.
* Customize lead and contact fields without technical help.
* Set up data backups and enforce MFA.

**🧭 User Experience – Step-by-Step Flow**

**Sales Rep Flow – Lead Management**

1. Click “+ New Lead” (auto-suggest company info via domain).
2. Fill minimal required fields (Name, Email, Company).
3. Assign source/tag (optional), hit “Save.”
4. Lead appears in “My Pipeline” with drag-and-drop stage transitions.
5. Call/email logged via sidebar, with AI suggestions (optional for later).
6. Quick filters for “Overdue Tasks” or “Deals Stuck > 14 days.”

**Sales Manager Flow – Custom Report Builder**

1. Go to “Reports” → “Create New.”
2. Choose base object: Leads, Deals, Activities.
3. Drag and drop filters (e.g., Date, Rep, Stage).
4. Choose visualization (Bar, Funnel, Line).
5. Save and share with team or schedule recurring delivery.

**Admin Flow – User Management**

1. Navigate to “Settings” → “Users.”
2. Add by email → assign role (Rep, Manager, Admin).
3. Customize field-level permissions (toggle view/edit).
4. View access logs and export audit trail.

**📖 Narrative**

Imagine a sales rep, Maya, joining a new company using SalesSphere. By Day 1 afternoon, she's already logged her first three deals and scheduled follow-ups—without a single training session. The UI is intuitive, designed for motion and speed. She drags leads across pipeline stages and auto-logs calls from her phone.

Meanwhile, her manager, James, logs into his dashboard. In two clicks, he sees Maya's activity, projected revenue for Q2, and deal stage conversion trends. He spots a bottleneck—deals are stalling at “Proposal Sent.” He creates a quick report, shares it with the team, and assigns coaching time.

No clunky menus. No digging. Just clarity and action.

SalesSphere doesn’t overwhelm. It accelerates.

**📊 Success Metrics**

* ⏱️ **Onboarding Time:** Time from first login to first deal logged ≤ 2 hours.
* 📈 **Workflow Efficiency:** Reduce avg. time to log call/email by 40%.
* 📊 **Manager Reporting Satisfaction:** ≥80% rate reporting tools 8/10+ in beta.
* 🔌 **Integration Readiness:** Support plug-and-play with 3 major MA platforms at launch.
* 🔐 **System Uptime:** ≥99.9% uptime over 90-day rolling period.

**⚙️ Technical Considerations**

* **Data Security:** Use field-level encryption for sensitive data (emails, notes).
* **Authentication:** Support OAuth & SSO (SAML) at launch; MFA required for Admins.
* **Scalability:** Handle 1,000 concurrent users with <300ms response time for all UI interactions.
* **Audit Logs:** Full access history for Admins.
* **Data Model:** Use relational structure; core objects: Lead, Contact, Deal, Activity, User.
* **Third-party Integrations:** REST-based connectors with HubSpot, Mailchimp, Marketo (OAuth 2.0).

**🗓️ Milestones & Sequencing**

**Phase 1 (0–6 weeks): Foundation**

* Define data model & UX wireframes
* Build core entities: Leads, Deals, Contacts
* Authentication, Role-based Access, Activity Logging

**Phase 2 (6–12 weeks): Sales Rep & Manager Tools**

* Drag-and-drop pipeline
* Activity log with inline actions
* Dashboard + default reports
* Custom Report Builder (v0)

**Phase 3 (12–18 weeks): Admin & Integration Layer**

* User & Role Management UI
* Custom Fields + Permissions
* Marketing integrations
* Audit logging + Backup system

**Phase 4 (18–22 weeks): Polish & Launch Prep**

* UI Polish + Empty State UX
* Performance Optimization
* QA + Beta rollout
* Launch Candidate

Question 1

How effectively did you identify the key functional requirements for the SalesSphere CRM based on the scenario's description of sales team workflows and goals? Provide specific examples of functional requirements documented in your PRD.

**How Effectively Were Key Functional Requirements Identified?**

I’d rate this as **highly effective**, because the PRD maps **each user persona’s core workflows and goals** into **specific, detailed functional requirements**. The connection between user needs and documented features is clear and intentional throughout.

Let’s look at **specific examples**:

**👤 Sales Reps – Workflow Efficiency & Simplicity**

**User Need (from scenario):**  
"Sales reps need an easy way to manage contacts and accounts, log activities, and track leads."

**Functional Requirements Documented:**

* **Add a lead in under 30 seconds:** Quick create modal with minimal required fields.
* **Drag-and-drop pipeline management:** Kanban-style UI for lead stages.
* **Activity logging with 1–2 taps:** Inline sidebar to log calls/emails/meetings fast.
* **Mobile-optimized interactions:** Step-by-step experience designed for reps on the go.

📌 *Why this is effective:* Each functional element helps reduce cognitive load and friction, which directly aligns with the business goal of cutting onboarding time by 50%.

**👩‍💼 Sales Managers – Reporting & Forecasting**

**User Need:**  
"Sales managers need clear dashboards, customizable reports, and forecasting tools."

**Functional Requirements Documented:**

* **Custom Report Builder:** Drag-and-drop fields and filters (e.g., by rep, by stage).
* **Saved & Scheduled Reports:** Ability to create recurring email reports.
* **Forecast View:** Visual breakdown of projected revenue based on pipeline stage.
* **Team Activity Tracker:** Dashboard with real-time insights into call/email volume and meeting frequency.

📌 *Why this is effective:* These features are built to *empower insight*, not overwhelm with data — the PRD emphasizes flexibility without sacrificing simplicity.

**🧑‍💻 System Admins – Control & Security**

**User Need:**  
"Admins need straightforward tools for permissions, field customization, and data security."

**Functional Requirements Documented:**

* **User & Role Management:** Add users, assign predefined roles (Rep, Manager, Admin).
* **Custom Field Creation (no-code):** UI to manage and reorder fields per object.
* **Field-Level Permissions:** Toggle view/edit access by role.
* **Audit Logs + MFA:** Admins can view access history and enforce secure login policies.

📌 *Why this is effective:* It anticipates core IT and compliance needs without adding bloat.

**🔗 Linking Back to Strategic Goals**

The PRD consistently ties functional requirements to the original **project goals**, including:

* ✂️ **Onboarding time** → Simple UI, fast workflows for reps
* ⚙️ **Workflow efficiency** → Streamlined logging and pipeline movement
* 📊 **Reporting depth** → Flexible reports and dashboards for managers
* 📶 **Reliable performance** → Uptime goals and scalability targets
* 🔌 **Ecosystem integration** → Marketing platform connectors

**📌 Conclusion**

The PRD **translates user needs into specific, practical functionality**. It balances depth and simplicity — just like the CRM itself aims to do. Every requirement can be traced directly to a scenario-driven user need or business objective.

Question 2

How well did you define important non-functional requirements (NFRs) such as usability, performance, integration needs, or basic security/data handling in your PRD? Provide specific examples from your PRD for at least two different NFR categories and explain their importance for the CRM's success based on the scenario.

**How Well Were Non-Functional Requirements (NFRs) Defined?**

The PRD includes clear, **specific non-functional requirements (NFRs)** across multiple critical categories — most notably **usability**, **performance**, **integration**, and **security/data handling**.

Let’s go through **two key NFR categories** with examples and explain why they matter deeply for the CRM’s success.

**🧠 1. Usability**

**Why it matters:**  
The *core differentiator* for SalesSphere is **ease of use** — the PRD must deliver on the promise to cut onboarding time by 50% and reduce friction for busy sales reps.

**Examples from the PRD:**

* **Onboarding Metric:**

“Time from first login to first deal logged ≤ 2 hours.”  
This defines a measurable usability outcome — new reps should be productive *fast*.

* **Sales Rep Flow Design:**

“Add a new lead in under 30 seconds.”  
“Drag-and-drop pipeline stages.”  
“Log calls/emails with 1–2 taps.”

* **No training required for reps to complete common tasks.**  
  This is implied in the narrative and reinforced by the step-by-step flows.

📌 *Why this is important:*  
Usability isn’t vague here — it’s tied to **specific actions** (logging activities, updating pipeline) and **timing** (how fast reps can do it). Without this, reps won’t adopt the tool — which tanks product stickiness and customer satisfaction.

**⚙️ 2. Performance, Security & Data Handling**

**Why it matters:**  
Sales data is sensitive and business-critical. System reliability and security are **non-negotiable** for adoption in mid-market B2B environments.

**Examples from the PRD:**

**Performance:**

* “≥99.9% uptime over a 90-day rolling period.”
* “Handle 1,000 concurrent users with <300ms response time for all UI interactions.”

These ensure the CRM stays responsive and dependable even during peak usage. Performance affects **trust**, especially for sales managers reviewing real-time dashboards or reps logging activities in rapid succession.

**Security/Data Handling:**

* “Use field-level encryption for sensitive data (emails, notes).”
* “Support OAuth & SSO (SAML) at launch; MFA required for Admins.”
* “Full access history available via audit logs.”

These are **baseline requirements** for any enterprise-leaning SaaS product. The PRD also includes **role-based permissions** and **regular backups**, helping ensure compliance, access control, and recovery — all critical for customer trust.

📌 *Why this is important:*  
Without strong NFRs around performance and data handling, SalesSphere won’t be taken seriously by mid-market buyers — especially those in regulated or data-sensitive industries (e.g., financial services, healthcare tech).

**🧩 Summary**

The PRD **clearly defines critical NFRs** across usability and technical reliability, and ties them directly to the product’s strategic positioning and user expectations.

| **NFR Category** | **Example from PRD** | **Why It’s Important** |
| --- | --- | --- |
| Usability | Lead creation in <30 sec; onboarding in 2 hrs | Delivers on core promise of ease-of-use |
| Performance | <300ms response time; 99.9% uptime | Builds user trust and supports team workflows |
| Security | MFA, audit logs, encryption | Essential for adoption in B2B & compliance-heavy orgs |
| Integration | Native connectors to 3 marketing tools | Keeps SalesSphere embedded in customers’ tech stack |

Question 3

How did using Copilot (if applicable) assist in identifying or refining requirements for your PRD? What specific requirements documented in your PRD were influenced by Copilot? If you did not use Copilot, explain your reasoning.

**How Did Copilot Assist in Identifying or Refining PRD Requirements?**

**Yes, Copilot (in this case, ChatPRD) was used** to generate and refine the requirements for the SalesSphere CRM PRD. The assistant played a strategic role in **translating user and business goals into specific, actionable requirements**, and in ensuring that **no critical user group or need was overlooked**.

Here’s a breakdown of **how Copilot helped**, with **specific examples** from the PRD and their impact:

**🧠 1. Refining Functional Requirements from Vague to Precise**

**Example:**

“Sales reps need to quickly add and manage leads.”

**How Copilot refined it:**

* Translated into specific functional flows:

“Add a new lead in under 30 seconds.”  
“Drag-and-drop pipeline stages.”  
“Log calls/emails with 1–2 taps.”

📌 *Impact:* This took a general user goal and broke it down into **measurable, testable product behaviors** — crucial for design and dev alignment.

**🔐 2. Ensuring Inclusion of NFRs Often Overlooked in Early Drafts**

**Example NFRs added via Copilot:**

* “≥99.9% uptime”
* “<300ms UI response time”
* “MFA required for Admins”
* “Field-level encryption for sensitive data”

📌 *Impact:* These were not always top-of-mind in early ideation but are essential for enterprise readiness — especially for mid-market CRM buyers. Copilot ensured the PRD covered **performance, security, and scalability**, not just features.

**📊 3. Structuring Custom Reporting Requirements for Managers**

**Prompted by Copilot’s flow suggestion:**

“Sales managers need custom dashboards and forecasting tools.”

**Translated into:**

* “Drag-and-drop custom report builder”
* “Save and schedule recurring reports”
* “Filter by rep, deal stage, activity type”

📌 *Impact:* This ensured the **reporting system isn’t static or hardcoded**, but flexible and manager-driven — a key product differentiator from clunky legacy CRMs.

**✍️ 4. Narrative & Storytelling Layer**

The “Maya and James” user journey narrative was **entirely Copilot-generated**, based on the user personas.

📌 *Impact:* This section helps stakeholders emotionally connect to the product — especially execs, sales leaders, and marketing. It brings the functional requirements to life in a way that raw specs can’t.

**🔚 Summary**

Copilot directly influenced multiple layers of the PRD:

| **Area** | **Copilot Contribution** | **Impact** |
| --- | --- | --- |
| Functional Requirements | Converted high-level needs into detailed flows (e.g., lead creation, activity logging) | Clearer developer and designer direction |
| Non-Functional Requirements | Suggested uptime, response time, security specs | Better product quality and trust |
| Reporting System | Structured dynamic report builder concept | Stronger value for sales managers |
| Narrative | Wrote “Maya and James” story | Better exec/stakeholder alignment |

**Without Copilot**, the PRD likely would’ve been **less structured, less complete, and slower to produce.** Copilot helped move from vague requirements to *execution-ready detail*, making it a powerful tool for product managers working fast and under pressure.

Question 4

Choose one key functional or non-functional requirement documented in your PRD. Explain its justification – how does it directly support a specific project goal (e.g., user efficiency, intuitive UI, reporting needs) mentioned in the scenario and likely captured in your PRD's introduction or goals section?

**Selected Requirement:**

**“Add a new lead in under 30 seconds”** (Functional Requirement)

**✅ Justification:**

This requirement **directly supports the project goal** of:

**“Provide a highly intuitive user interface that reduces onboarding time by 50% compared to major competitors.”**

And more specifically, the **Sales Rep user goal** from the PRD:

“Quickly add/update leads and contacts with minimal clicks.”

**📌 Why This Requirement Matters:**

Adding leads is **one of the most frequent actions** a sales rep performs in a CRM. In legacy systems (like Salesforce), this is often slow and bloated — requiring navigating multiple screens, completing unnecessary fields, and waiting for page loads.

By setting a **time-bound target ("under 30 seconds")**, this requirement ensures:

* Minimal required fields (e.g., Name, Email, Company only)
* Smart defaults or auto-fill (e.g., suggesting company info from domain)
* A clean, mobile-friendly modal instead of a full-page flow

This kind of efficiency:

* **Reduces training needs** (reps don’t need a walkthrough to add a lead)
* **Drives adoption** (if reps love how fast it is, they'll actually use it)
* **Directly reduces onboarding time**, fulfilling the business goal of being 50% faster to adopt

**🔗 Clear Link to Project Goals:**

| **Requirement** | **Supports Which Goal?** | **How** |
| --- | --- | --- |
| Add new lead in <30 sec | 🧠 "Reduce onboarding time by 50%" | Removes friction in top workflow |
|  | ⚡ "Improve sales workflow efficiency" | Speeds up lead intake |
|  | 👤 Sales Rep user goal | Makes the interface feel responsive and effortless |

**✅ Conclusion:**

This requirement is **laser-focused** on what makes SalesSphere stand out: *speed, simplicity, and usability*. It's not just a nice-to-have — it’s a **strategic enabler** of faster onboarding, better UX, and stronger user satisfaction.

Question 5

What were the most valuable insights you gained about defining requirements and creating a PRD for business software through this activity? How specifically can you apply these insights (e.g., about balancing different user needs, defining NFRs clearly within a PRD structure, ensuring testability) to future product management tasks?

**Most Valuable Insights from This PRD Activity**

**1. Balancing Simplicity for Users with Power for Admins/Managers Is Critical**

**Insight:**  
A single product (like a CRM) must serve *very different users* — frontline sales reps want **speed and ease**, while managers need **depth and insight**, and admins require **control and stability**. Designing for one without undermining the others is a nuanced balancing act.

**How I’ll apply this:**  
In future PRDs, I’ll always:

* Define **separate user stories and flows** for each persona
* Call out **conflicting needs early** (e.g., “simplicity vs. flexibility”) to drive intentional tradeoffs
* Prioritize **default views** that are simple, while allowing deeper customization via secondary menus

**2. Non-Functional Requirements (NFRs) Deserve Just as Much Attention as Features**

**Insight:**  
NFRs — like system uptime, data handling, or performance — often get ignored early, but they're **deal-breakers** for B2B buyers. Without explicitly stating things like “<300ms response time” or “MFA required for Admins,” the team may underbuild the technical backbone.

**How I’ll apply this:**

* I’ll **always include NFRs as a first-class section** in my PRDs, not an afterthought
* I’ll **make NFRs testable** (e.g., response times, security protocols, uptime targets)
* I’ll **tie NFRs to buyer personas** (e.g., admins and IT reviewers) so engineering sees why they matter

**3. Good Requirements Are Measurable and Specific**

**Insight:**  
Statements like “make onboarding faster” or “intuitive UI” are too vague. The PRD got sharper when we said “Add a lead in under 30 seconds” or “onboarding complete within 2 hours.”

**How I’ll apply this:**

* Push every requirement through a **“Can we test this?”** lens
* Work with design/dev to ensure each one is **buildable and verifiable**
* Use **real-world task timings** (e.g., time to log a call) to define benchmarks

**4. Storytelling Makes a PRD Memorable and Persuasive**

**Insight:**  
The narrative (“Maya and James”) added emotion, relevance, and clarity — it helped align stakeholders by showing *why* the features mattered in real-life terms.

**How I’ll apply this:**

* Include a **short scenario-based narrative** in PRDs for internal alignment
* Use storytelling during product reviews or stakeholder demos to ground the work in user outcomes

**📌 Summary of What I’ll Do Differently**

| **Insight** | **How I’ll Apply It** |
| --- | --- |
| Balance user needs | Explicitly document and reconcile tradeoffs early |
| Prioritize NFRs | Make them measurable and aligned to user types |
| Make requirements testable | Add timing, success criteria, and constraints |
| Use storytelling | Help stakeholders empathize and engage with the product vision |