# Week 1 Development Plan: VeroSuite CRM

## \*\*Week 1 Goal:\*\* Complete Scheduling & Dispatch Module

### \*\*Daily Schedule (12 hours/day, 6 days/week)\*\*

---

## \*\*Day 1: Monday - Project Setup & Foundation Review\*\*

### \*\*Morning (6 hours)\*\*

\*\*8:00 AM - 12:00 PM\*\*

- \*\*Hour 1-2:\*\* Install dependencies, set up development environment

- \*\*Hour 3-4:\*\* Review current codebase, understand architecture

- \*\*Hour 5-6:\*\* Set up local development, test existing functionality

### \*\*Afternoon (6 hours)\*\*

\*\*1:00 PM - 7:00 PM\*\*

- \*\*Hour 1-2:\*\* Study Prisma schema for scheduling domain

- \*\*Hour 3-4:\*\* Plan scheduling module architecture

- \*\*Hour 5-6:\*\* Create scheduling module structure

\*\*Deliverables:\*\*

- ✅ Development environment running

- ✅ Understanding of current codebase

- ✅ Scheduling module folder structure created

---

## \*\*Day 2: Tuesday - Work Orders Foundation\*\*

### \*\*Morning (6 hours)\*\*

\*\*8:00 AM - 12:00 PM\*\*

- \*\*Hour 1-2:\*\* Create WorkOrder model and database migrations

- \*\*Hour 3-4:\*\* Implement WorkOrder service (CRUD operations)

- \*\*Hour 5-6:\*\* Add validation schemas for work orders

### \*\*Afternoon (6 hours)\*\*

\*\*1:00 PM - 7:00 PM\*\*

- \*\*Hour 1-2:\*\* Create WorkOrder controller and API endpoints

- \*\*Hour 3-4:\*\* Implement multi-tenant filtering for work orders

- \*\*Hour 5-6:\*\* Test work order creation and retrieval

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | No description | uuid | uuid |  |  |
| tenant\_id | No description | uuid | uuid |  |  |
| account\_id | No description | uuid | uuid |  |  |
| location\_id | No description | uuid | uuid |  |  |
| service\_type | No description | character varying | varchar |  |  |
| description | No description | text | text |  |  |
| recurrence\_rule | No description | character varying | varchar |  |  |
| estimated\_duration | No description | integer | int4 |  |  |
| service\_price | No description | numeric | numeric |  |  |
| special\_instructions | No description | text | text |  |  |
| status | No description | character varying | varchar |  |  |
| created\_at | No description | timestamp with time zone | timestamptz |  |  |
| updated\_at | No description | timestamp with time zone | timestamptz |  |  |

\*\*Deliverables:\*\*

- ✅ WorkOrder entity with full CRUD operations

- ✅ API endpoints for work order management

- ✅ Multi-tenant data isolation working

---

## \*\*Day 3: Wednesday - Technician Management\*\*

### \*\*Morning (6 hours)\*\*

\*\*8:00 AM - 12:00 PM\*\*

- \*\*Hour 1-2:\*\* Extend User model for technician-specific fields

- \*\*Hour 3-4:\*\* Create technician service with skills and availability

- \*\*Hour 5-6:\*\* Implement technician assignment logic

### \*\*Afternoon (6 hours)\*\*

\*\*1:00 PM - 7:00 PM\*\*

- \*\*Hour 1-2:\*\* Create technician API endpoints

- \*\*Hour 3-4:\*\* Add technician to work order relationships

- \*\*Hour 5-6:\*\* Test technician assignment and management

\*\*Deliverables:\*\*

- ✅ Technician management system

- ✅ Skills and availability tracking

- ✅ Work order assignment functionality

---

## \*\*Day 4: Thursday - Scheduling Logic\*\*

### \*\*Morning (6 hours)\*\*

\*\*8:00 AM - 12:00 PM\*\*

- \*\*Hour 1-2:\*\* Create Schedule entity and relationships

- \*\*Hour 3-4:\*\* Implement basic scheduling algorithms

- \*\*Hour 5-6:\*\* Add scheduling service with conflict detection

### \*\*Afternoon (6 hours)\*\*

\*\*1:00 PM - 7:00 PM\*\*

- \*\*Hour 1-2:\*\* Create scheduling API endpoints

- \*\*Hour 3-4:\*\* Implement calendar view functionality

- \*\*Hour 5-6:\*\* Test scheduling and conflict resolution

\*\*Deliverables:\*\*

- ✅ Scheduling system with conflict detection

- ✅ Calendar view API endpoints

- ✅ Basic scheduling algorithms working

---

## \*\*Day 5: Friday - Status Tracking & Workflow\*\*

### \*\*Morning (6 hours)\*\*

\*\*8:00 AM - 12:00 PM\*\*

- \*\*Hour 1-2:\*\* Implement work order status workflow

- \*\*Hour 3-4:\*\* Create status transition logic and validation

- \*\*Hour 5-6:\*\* Add status history tracking

### \*\*Afternoon (6 hours)\*\*

\*\*1:00 PM - 7:00 PM\*\*

- \*\*Hour 1-2:\*\* Create status update API endpoints

- \*\*Hour 3-4:\*\* Implement notification system for status changes

- \*\*Hour 5-6:\*\* Test complete work order lifecycle

\*\*Deliverables:\*\*

- ✅ Work order status workflow

- ✅ Status transition validation

- ✅ Status change notifications

---

## \*\*Day 6: Saturday - Integration & Testing\*\*

### \*\*Morning (6 hours)\*\*

\*\*8:00 AM - 12:00 PM\*\*

- \*\*Hour 1-2:\*\* Integrate scheduling with existing CRM module

- \*\*Hour 3-4:\*\* Create comprehensive test suite

- \*\*Hour 5-6:\*\* Performance testing and optimization

### \*\*Afternoon (6 hours)\*\*

\*\*1:00 PM - 7:00 PM\*\*

- \*\*Hour 1-2:\*\* API documentation and Swagger updates

- \*\*Hour 3-4:\*\* Bug fixes and code cleanup

- \*\*Hour 5-6:\*\* Week 1 review and Week 2 planning

\*\*Deliverables:\*\*

- ✅ Complete scheduling module integrated

- ✅ Comprehensive test coverage

- ✅ API documentation updated

- ✅ Week 2 plan ready

# Complete 13-Week Development Timeline: VERO Ops Platform

## \*\*Your Schedule:\*\* 12 hours/day, 6 days/week (72 hours/week)

---

## \*\*Phase 1: Core MVP (Weeks 1-6) - 6 Weeks\*\*

### \*\*Week 1: Scheduling & Dispatch Module\*\*

\*\*Goal:\*\* Complete work order and technician management system

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Project setup, work orders foundation

- \*\*Days 3-4:\*\* Technician management, scheduling logic

- \*\*Days 5-6:\*\* Status tracking, integration, testing

\*\*Deliverables:\*\*

- ✅ Work order CRUD operations

- ✅ Technician management with skills/availability

- ✅ Scheduling system with conflict detection

- ✅ Status workflow management

- ✅ Calendar view functionality

\*\*API Endpoints:\*\* 10+ scheduling endpoints

\*\*Database:\*\* WorkOrder, Schedule tables, enhanced User model

---

### \*\*Week 2: Field Service Mobile Backend\*\*

\*\*Goal:\*\* Mobile API for technicians to manage work orders

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Mobile API structure, work order endpoints

- \*\*Days 3-4:\*\* Photo upload, signature capture

- \*\*Days 5-6:\*\* Offline sync, mobile-specific features

\*\*Deliverables:\*\*

- ✅ Mobile-optimized API endpoints

- ✅ Photo upload to S3/MinIO

- ✅ Digital signature capture

- ✅ Basic offline data sync

- ✅ Mobile authentication

\*\*API Endpoints:\*\* 8+ mobile-specific endpoints

\*\*Features:\*\* File upload, offline capability, mobile auth

---

### \*\*Week 3: Billing & Payments Foundation\*\*

\*\*Goal:\*\* Invoice generation and payment processing

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Invoice model, generation logic

- \*\*Days 3-4:\*\* Stripe integration, payment processing

- \*\*Days 5-6:\*\* Customer portal, payment tracking

\*\*Deliverables:\*\*

- ✅ Invoice generation from completed work orders

- ✅ Stripe payment processing integration

- ✅ Payment tracking and history

- ✅ Basic customer portal for payments

- ✅ Invoice PDF generation

\*\*API Endpoints:\*\* 12+ billing/payment endpoints

\*\*Integrations:\*\* Stripe, PDF generation

---

### \*\*Week 4: QuickBooks Integration & Advanced Billing\*\*

\*\*Goal:\*\* Complete billing system with accounting integration

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* QuickBooks API integration

- \*\*Days 3-4:\*\* Sync invoices, payments, customers

- \*\*Days 5-6:\*\* Error handling, reconciliation

\*\*Deliverables:\*\*

- ✅ QuickBooks customer sync

- ✅ Invoice synchronization

- ✅ Payment reconciliation

- ✅ Error handling and retry logic

- ✅ Data consistency checks

\*\*API Endpoints:\*\* 6+ QuickBooks sync endpoints

\*\*Integrations:\*\* QuickBooks API, webhook handling

---

### \*\*Week 5: Basic Reports & Dashboard\*\*

\*\*Goal:\*\* Analytics and reporting system

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Report models, data aggregation

- \*\*Days 3-4:\*\* Dashboard endpoints, key metrics

- \*\*Days 5-6:\*\* Chart data, export functionality

\*\*Deliverables:\*\*

- ✅ Revenue reports and analytics

- ✅ Job completion metrics

- ✅ Technician performance reports

- ✅ Customer analytics

- ✅ Data export functionality

\*\*API Endpoints:\*\* 8+ reporting endpoints

\*\*Features:\*\* Data aggregation, chart generation, exports

---

### \*\*Week 6: Admin Panel & MVP Polish\*\*

\*\*Goal:\*\* Complete MVP with admin functionality

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Admin user management, settings

- \*\*Days 3-4:\*\* System configuration, tenant management

- \*\*Days 5-6:\*\* Testing, bug fixes, MVP completion

\*\*Deliverables:\*\*

- ✅ Admin user management

- ✅ System settings and configuration

- ✅ Tenant management interface

- ✅ Comprehensive testing

- ✅ MVP ready for deployment

\*\*API Endpoints:\*\* 10+ admin endpoints

\*\*Features:\*\* User management, system configuration

---

## \*\*Phase 2: Enhanced Features (Weeks 7-10) - 4 Weeks\*\*

### \*\*Week 7: Route Optimization Foundation\*\*

\*\*Goal:\*\* Basic routing and distance calculations

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Google Maps integration, distance matrix

- \*\*Days 3-4:\*\* Basic routing algorithms (greedy)

- \*\*Days 5-6:\*\* ETA calculations, route visualization

\*\*Deliverables:\*\*

- ✅ Google Maps API integration

- ✅ Distance matrix calculations

- ✅ Basic route optimization

- ✅ ETA calculations

- ✅ Route visualization data

\*\*API Endpoints:\*\* 6+ routing endpoints

\*\*Integrations:\*\* Google Maps API

---

### \*\*Week 8: Advanced Route Optimization\*\*

\*\*Goal:\*\* Sophisticated routing algorithms

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Advanced algorithms (2-opt, genetic)

- \*\*Days 3-4:\*\* Time window constraints, vehicle capacity

- \*\*Days 5-6:\*\* Route optimization API, performance tuning

\*\*Deliverables:\*\*

- ✅ Advanced routing algorithms

- ✅ Time window and capacity constraints

- ✅ Route optimization API

- ✅ Performance optimization

- ✅ Route comparison tools

\*\*API Endpoints:\*\* 8+ advanced routing endpoints

\*\*Algorithms:\*\* 2-opt, genetic algorithm, constraint handling

---

### \*\*Week 9: Communications Engine\*\*

\*\*Goal:\*\* SMS and email notification system

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Twilio SMS integration, SendGrid email

- \*\*Days 3-4:\*\* Template system, message scheduling

- \*\*Days 5-6:\*\* Delivery tracking, notification preferences

\*\*Deliverables:\*\*

- ✅ SMS notifications (Twilio)

- ✅ Email notifications (SendGrid)

- ✅ Template system

- ✅ Message scheduling

- ✅ Delivery tracking

\*\*API Endpoints:\*\* 10+ communication endpoints

\*\*Integrations:\*\* Twilio, SendGrid

---

### \*\*Week 10: Inventory Management\*\*

\*\*Goal:\*\* Chemical and equipment tracking system

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Inventory models, stock tracking

- \*\*Days 3-4:\*\* Chemical usage, safety data sheets

- \*\*Days 5-6:\*\* Compliance reporting, low stock alerts

\*\*Deliverables:\*\*

- ✅ Inventory tracking system

- ✅ Chemical usage logging

- ✅ Safety data sheet management

- ✅ Compliance reporting

- ✅ Low stock alerts

\*\*API Endpoints:\*\* 12+ inventory endpoints

\*\*Features:\*\* Chemical tracking, compliance reporting

---

## \*\*Phase 3: Production Ready (Weeks 11-13) - 3 Weeks\*\*

### \*\*Week 11: Security & Compliance\*\*

\*\*Goal:\*\* Production security and compliance features

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* PCI compliance, data encryption

- \*\*Days 3-4:\*\* Audit logging, security testing

- \*\*Days 5-6:\*\* Penetration testing, security hardening

\*\*Deliverables:\*\*

- ✅ PCI compliance implementation

- ✅ Data encryption (at-rest, in-transit)

- ✅ Comprehensive audit logging

- ✅ Security testing and hardening

- ✅ Compliance documentation

\*\*Security:\*\* PCI DSS, encryption, audit trails

---

### \*\*Week 12: Performance & Scalability\*\*

\*\*Goal:\*\* Optimize for production scale

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* Database optimization, query tuning

- \*\*Days 3-4:\*\* Caching strategy, Redis optimization

- \*\*Days 5-6:\*\* Load testing, performance monitoring

\*\*Deliverables:\*\*

- ✅ Database performance optimization

- ✅ Redis caching strategy

- ✅ Load testing results

- ✅ Performance monitoring

- ✅ Scalability improvements

\*\*Performance:\*\* Query optimization, caching, load testing

---

### \*\*Week 13: Deployment & Launch\*\*

\*\*Goal:\*\* Production deployment and go-live

\*\*Daily Focus:\*\*

- \*\*Days 1-2:\*\* AWS infrastructure setup

- \*\*Days 3-4:\*\* CI/CD pipeline, automated deployment

- \*\*Days 5-6:\*\* Final testing, documentation, launch

\*\*Deliverables:\*\*

- ✅ AWS production infrastructure

- ✅ CI/CD pipeline

- ✅ Automated deployment

- ✅ Production monitoring

- ✅ Launch documentation

\*\*Infrastructure:\*\* AWS ECS, RDS, S3, CloudFront

---

## \*\*Weekly Progress Tracking\*\*

### \*\*Success Metrics by Week:\*\*

| Week | Module | API Endpoints | Key Features | Integration Status |

|------|--------|---------------|--------------|-------------------|

| 1 | Scheduling | 10+ | Work orders, technicians, calendar | ✅ Complete |

| 2 | Mobile Backend | 8+ | Photo upload, signatures, offline | ✅ Complete |

| 3 | Billing | 12+ | Invoices, Stripe payments | ✅ Complete |

| 4 | QuickBooks | 6+ | QB sync, reconciliation | ✅ Complete |

| 5 | Reports | 8+ | Analytics, dashboards | ✅ Complete |

| 6 | Admin Panel | 10+ | User management, settings | ✅ Complete |

| 7 | Route Optimization | 6+ | Maps, distance, basic routing | ✅ Complete |

| 8 | Advanced Routing | 8+ | Advanced algorithms, constraints | ✅ Complete |

| 9 | Communications | 10+ | SMS, email, templates | ✅ Complete |

| 10 | Inventory | 12+ | Chemical tracking, compliance | ✅ Complete |

| 11 | Security | - | PCI compliance, encryption | ✅ Complete |

| 12 | Performance | - | Optimization, caching | ✅ Complete |

| 13 | Deployment | - | AWS infrastructure, CI/CD | ✅ Complete |

### \*\*Monthly Milestones:\*\*

\*\*Month 1 (Weeks 1-4):\*\* MVP Core Features

- ✅ Scheduling and dispatch

- ✅ Mobile backend

- ✅ Billing and payments

- ✅ QuickBooks integration

\*\*Month 2 (Weeks 5-8):\*\* Enhanced Features

- ✅ Reports and analytics

- ✅ Admin panel

- ✅ Route optimization

- ✅ Advanced routing

\*\*Month 3 (Weeks 9-13):\*\* Production Ready

- ✅ Communications engine

- ✅ Inventory management

- ✅ Security and compliance

- ✅ Performance optimization

- ✅ Production deployment

---

## \*\*Risk Mitigation Timeline:\*\*

### \*\*High Risk Areas:\*\*

- \*\*Week 2:\*\* Mobile offline sync complexity

- \*\*Week 4:\*\* QuickBooks API integration challenges

- \*\*Week 8:\*\* Advanced routing algorithm performance

- \*\*Week 11:\*\* PCI compliance requirements

### \*\*Buffer Time Built In:\*\*

- \*\*Week 6:\*\* MVP polish and testing

- \*\*Week 10:\*\* Inventory system completion

- \*\*Week 13:\*\* Final deployment and testing

---

## \*\*Final Deliverable (Week 13):\*\*

✅ \*\*Complete VERO Ops Platform\*\*

- Multi-tenant pest control operations platform

- CRM, scheduling, mobile, billing, routing, communications

- Production-ready with security and compliance

- Deployed on AWS with monitoring and CI/CD

\*\*Total Timeline:\*\* 13 weeks (3.25 months)

\*\*Total Development Hours:\*\* ~936 hours (72 hours/week × 13 weeks)

\*\*This timeline gives you clear weekly goals and success metrics to track your progress!\*\*

---

## \*\*Weekly Deliverables:\*\*

### \*\*By End of Week 1:\*\*

✅ \*\*Complete Scheduling & Dispatch Module\*\*

- Work order management (CRUD operations)

- Technician management with skills and availability

- Scheduling system with conflict detection

- Status tracking and workflow management

- Calendar view functionality

- Multi-tenant data isolation

- API endpoints for all scheduling operations

- Integration with existing CRM module

### \*\*API Endpoints Created:\*\*

- `GET /api/v1/scheduling/work-orders` - List work orders

- `POST /api/v1/scheduling/work-orders` - Create work order

- `GET /api/v1/scheduling/work-orders/:id` - Get work order details

- `PUT /api/v1/scheduling/work-orders/:id` - Update work order

- `DELETE /api/v1/scheduling/work-orders/:id` - Delete work order

- `POST /api/v1/scheduling/work-orders/:id/assign` - Assign technician

- `POST /api/v1/scheduling/work-orders/:id/status` - Update status

- `GET /api/v1/scheduling/technicians` - List technicians

- `GET /api/v1/scheduling/calendar` - Calendar view

- `POST /api/v1/scheduling/schedule` - Create schedule

### \*\*Database Schema Updates:\*\*

- WorkOrder table with all required fields

- Schedule table for calendar management

- Enhanced User table for technician data

- Status history tracking

---

## \*\*Success Metrics for Week 1:\*\*

### \*\*Technical Metrics:\*\*

- ✅ All API endpoints return correct responses

- ✅ Multi-tenant data isolation working

- ✅ Database queries optimized

- ✅ Error handling implemented

- ✅ Validation working for all inputs

### \*\*Functional Metrics:\*\*

- ✅ Can create and manage work orders

- ✅ Can assign technicians to work orders

- ✅ Can schedule work orders on calendar

- ✅ Can track work order status changes

- ✅ Can view technician availability

- ✅ Can detect scheduling conflicts

---

## \*\*Week 2 Preview:\*\*

\*\*Focus:\*\* Field Service Mobile Backend

- Mobile API for work orders

- Photo upload functionality

- Signature capture

- Offline data sync

\*\*This plan gives you a clear roadmap for Week 1 with specific deliverables and success criteria!\*\*