**Development Sprint 1: Core Functionality**

**Member Management Module**

1. **Database Schema Design**
	* Design and create the Member table with fields: id, first\_name, last\_name, email, join\_date, status, created\_at, updated\_at.
	* Design and create the MembershipStatus table if needed to store statuses.
2. **Model Implementation**
	* Implement Member model with appropriate fields and relationships.
	* Implement MembershipStatus model if needed.
3. **Views and CRUD Operations**
	* Implement views for creating, reading, updating, and deleting members.
	* Ensure views handle all CRUD operations securely and efficiently.
4. **Forms and Templates**
	* Develop forms for member management (e.g., MemberForm).
	* Create templates for member list, detail, create, update, and delete views.

**Case Management Module**

1. **Database Schema Design**
	* Design and create the Case table with fields: id, title, description, member\_id (foreign key), status, created\_at, updated\_at.
	* Design and create the CaseDocument table with fields: id, case\_id (foreign key), uploaded\_by (foreign key), document, description, uploaded\_at.
2. **Model Implementation**
	* Implement Case model with appropriate fields and relationships.
	* Implement CaseDocument model with appropriate fields and relationships.
3. **Views and CRUD Operations**
	* Implement views for creating, reading, updating, and deleting cases.
	* Implement views for uploading, viewing, and managing case documents.
4. **Forms and Templates**
	* Develop forms for case management (e.g., CaseForm, CaseDocumentForm).
	* Create templates for case list, detail, create, update, and delete views.

**User Authentication**

1. **User Registration and Login**
	* Implement user registration and login functionality using Django's built-in authentication system.
	* Create custom user models if needed to support additional user roles.
2. **Role-Based Access Control**
	* Implement role-based access control to restrict access to certain views and actions based on user roles.
	* Ensure only authorized users can perform specific actions (e.g., only admins can delete member profiles).

**Initial Testing**

1. **Unit and Integration Tests**
	* Write unit tests for Member and Case models.
	* Write integration tests for member and case CRUD operations.

**Development Sprint 2: Advanced Features**

**Communication Management Module**

1. **Database Schema Design**
	* Design and create the CommunicationRecord table with fields: id, subject, message, recipient, sent\_at, created\_at.
2. **Model Implementation**
	* Implement CommunicationRecord model with appropriate fields.
3. **Views and Communication Operations**
	* Implement views for sending mass emails and SMS to members.
	* Implement views for viewing communication history.
4. **Forms and Templates**
	* Develop forms for creating and sending communications (e.g., CommunicationForm).
	* Create templates for communication list and detail views.

**Event Management Module**

1. **Database Schema Design**
	* Design and create the Event table with fields: id, name, description, date, location, organizer\_id (foreign key), created\_at, updated\_at.
	* Design and create the EventParticipant table with fields: id, event\_id (foreign key), member\_id (foreign key), registered\_at.
2. **Model Implementation**
	* Implement Event model with appropriate fields and relationships.
	* Implement EventParticipant model with appropriate fields and relationships.
3. **Views and CRUD Operations**
	* Implement views for creating, reading, updating, and deleting events.
	* Implement views for registering members for events and tracking attendance.
4. **Forms and Templates**
	* Develop forms for event management (e.g., EventForm).
	* Create templates for event list, detail, create, update, and delete views.

**Initial Reporting and Analytics**

1. **Basic Reporting**
	* Implement functionality to generate basic reports on member and case statistics.
	* Create views and templates for displaying reports.

**Notification Engine**

1. **Database Schema Design**
	* Design and create the Notification table with fields: id, recipient\_id (foreign key), message, created\_at, read\_at.
2. **Model Implementation**
	* Implement Notification model with appropriate fields.
3. **Views and Notification Operations**
	* Implement views for creating and sending notifications.
	* Implement views for viewing and managing notifications.
4. **Forms and Templates**
	* Develop forms for creating notifications (e.g., NotificationForm).
	* Create templates for notification list and detail views.

**Development Sprint 3: Finalization and Enhancements**

**Workflow Engine**

1. **Database Schema Design**
	* Design and create the Workflow table with fields: id, name, description, created\_at, updated\_at.
	* Design and create the WorkflowStep table with fields: id, workflow\_id (foreign key), name, description, order, created\_at, updated\_at.
2. **Model Implementation**
	* Implement Workflow and WorkflowStep models with appropriate fields and relationships.
3. **Views and Workflow Operations**
	* Implement views for defining and managing workflows.
	* Implement views for tracking workflow progress.
4. **Forms and Templates**
	* Develop forms for creating workflows and workflow steps (e.g., WorkflowForm, WorkflowStepForm).
	* Create templates for workflow list, detail, and management views.

**Secure Document Management**

1. **Enhance Document Security**
	* Implement document encryption for secure storage.
	* Implement access logs to track document access.
2. **Document Versioning**
	* Implement functionality to manage and track different versions of documents.
3. **Forms and Templates**
	* Develop forms for document versioning and access management.
	* Create templates for managing document versions and access logs.

**Meeting Management Module**

1. **Database Schema Design**
	* Design and create the Meeting table with fields: id, title, agenda, date, time, location, created\_at, updated\_at.
	* Design and create the MeetingParticipant table with fields: id, meeting\_id (foreign key), member\_id (foreign key), registered\_at.
2. **Model Implementation**
	* Implement Meeting model with appropriate fields and relationships.
	* Implement MeetingParticipant model with appropriate fields and relationships.
3. **Views and CRUD Operations**
	* Implement views for creating, reading, updating, and deleting meetings.
	* Implement views for registering members for meetings and tracking attendance.
4. **Forms and Templates**
	* Develop forms for meeting management (e.g., MeetingForm).
	* Create templates for meeting list, detail, create, update, and delete views.

**Advanced Reporting and Analytics**

1. **Customizable Reports**
	* Implement functionality to create customizable reports based on various criteria.
2. **Visual Dashboards**
	* Integrate Chart.js for visual dashboards.
	* Create views and templates for displaying visual dashboards.

**Performance Optimization**

1. **Optimize Database Queries**
	* Review and optimize database queries for performance improvements.
2. **Application Performance**
	* Optimize application performance through code optimization and efficient resource management.

**Final Testing**

1. **Comprehensive Testing**
	* Conduct comprehensive unit and integration tests for all modules.
	* Conduct user acceptance testing (UAT) with end-users.

**Deployment Preparation**

1. **Prepare for Deployment**
	* Prepare deployment scripts and documentation.
	* Conduct final reviews and adjustments before deployment.

These technical requirements provide a clear and detailed outline of the technical tasks and specifications needed for each development sprint, ensuring the successful implementation of the Labor Union CRM System.