<u>Cresta Sales Documentation</u>

Demonstration Video:
https://drive.google.com/file/d/1_zUcLIfAIFImaEQ15rXNLyXGN24UIXtt/view?usp=sharing

- Lead Management Module Documentation

Overview

The **Lead Management Module** provides tools to manage and track sales leads effectively. It includes features for organizing lead data, tracking progress, and generating customer quotations.

Models

1. Lead

The primary model for managing sales leads.

Fields

Basic Fields

- `name` **(Char)**: The name of the lead. Automatically converted to uppercase on creation.
- `email` **(Char)**: Email address of the lead.
- `phone` **(Char)**: Phone number of the lead.
- `description` **(Text)**: Additional details or notes about the lead.
- `budget` **(Float)**: Estimated budget for the project.
- `create_date` **(Datetime)**: Date and time when the lead was created. Defaults to the current date and time.
- `write_date` **(Datetime)**: Last modification date and time.
- `meeting_date` **(Datetime)**: Scheduled date and time for a meeting with the lead.
- `requirements` **(Html)**: Project requirements in HTML format.

Related Fields

- `project_scope` **(Many2one)**: Links to the **Project Scope** model, indicating the scope of the project.

- `assigned_to` **(Many2one)**: The user assigned to the lead.
- `campaign` **(Many2one)**: Links the lead to a specific **Sales Campaign**.
- `tags` **(Many2many)**: Tags associated with the lead for categorization.
- `attachment_ids` **(Many2many)**: Attachments linked to the lead.
- `communication_ids` **(One2many)**: Communication records related to the lead.

Selection Fields

- `status` **(Selection)**: Tracks the lead's progress. Options:
  - New (default)
  - In Progress
  - Meeting
  - Quotation
  - Lost
- `lead_source` **(Selection)**: Source of the lead. Options:
  - Website (default)
  - Social Media
  - Campaign
  - Other
- `qualification_status` **(Selection)**: Computed field representing lead qualification. Options:
  - Cold
  - Warm
  - Lost
  - Very Likely

Methods

Computed Fields

- `_compute_qualification_status(self)`:
  Automatically computes the lead's qualification status based on the presence of a budget, campaign, and project scope:
  - **Very Likely**: If all three fields are set.
  - **Warm**: If budget and campaign are set.
  - **Cold**: If only the campaign is set.
  - **Lost**: If none are set.

Group Expansion

- `_group_expand_states(self, states, domain, order)`:
  Ensures all status options are displayed in grouped views, regardless of existing records.

Override Create

- `create(self, vals)`:
  Converts the `name` field to uppercase before saving the record.

Actions

- `action_mark_as_done(self)`:
  Updates the lead's status to **In Progress**.
- `action_mark_as_lost(self)`:
  Updates the lead's status to **Lost**.
- `action_make_quotation(self)`:
  Creates a customer quotation and quotation lines based on the lead's data, linking it to the project scope.

Workflow

1. Lead Lifecycle

1. **Creation**: Leads are created with a default status of **New**.
2. **Progress Tracking**:
   - Update the status to **In Progress**, **Meeting**, or **Quotation** as the lead advances.
   - Mark leads as **Lost** if they are no longer viable.
3. **Qualification**: Qualification status updates automatically based on criteria.

2. Generating Quotations

- Use the **Generate Quotation** action to create a linked customer quotation directly from the lead.
- The generated quotation includes:
  - Lead requirements as proposal details.
  - Project scope and pricing as a line item.

Dependencies

- **Project Scope**: Links leads to predefined project scopes for quoting and planning.
- **Sales Campaign**: Tracks the campaign from which a lead originated.
- **Customer Quotation**: Enables automated creation of customer quotations from leads.

- Customer Quotation Module Documentation

Overview

The **Customer Quotation Module** streamlines the process of generating and managing customer quotations. It allows users to create, send, and track quotations while integrating seamlessly with project management workflows.

Models

1. Quotation

Represents a customer quotation.

Fields

Basic Fields

- `name` **(Char)**: Unique identifier for the quotation. Auto-generated if not provided.
- `lead_id` **(Many2one)**: Links the quotation to a **Lead Management** record.
- `requirements` **(Html)**: Details of the customer's requirements.
- `proposal_details` **(Text)**: A textual summary of the proposal details.
- `quotation_date` **(Date)**: Date of the quotation. Defaults to the current date.
- `total_amount` **(Float)**: Total cost of the quotation, computed from line items.
- `asign_to` **(Many2one)**: The user assigned to handle the quotation.
- `state` **(Selection)**: Tracks the status of the quotation. Options:
  - Draft (default)
  - Sent
  - Accepted
  - Signed
  - In Project
  - Rejected
- `date_start` **(Date)**: Start date for the proposed project.
- `date_end` **(Date)**: End date for the proposed project.
- `quotation_lines` **(One2many)**: Line items associated with the quotation.

Methods

Computed Fields

- **`_compute_total_amount(self)`**:
  Calculates the `total_amount` as the sum of subtotals from all associated **QuotationLine** records.

Group Expansion

- **`_group_expand_states(self, states, domain, order)`**:
  Ensures all status options are displayed in grouped views, regardless of the presence of records.

Override Create

- **`create(self, vals)`**:
  Automatically generates a unique quotation reference using Odoo's sequence API if the `name` field is not provided.

Actions

- **`action_send(self)`**:
  Marks the quotation as **Sent** and triggers the quotation report.
- **`action_mark_accepted(self)`**:
  Updates the status to **Accepted**.
- **`action_send_contract(self)`**:
  Changes the status to **Signed** and generates a contract report.
- **`action_mark_rejected(self)`**:
  Updates the status to **Rejected**.
- **`action_create_project(self)`**:
  Creates a linked project in the **Project Management** module, updates the status to **In Project**, and redirects to the newly created project.

2. QuotationLine

Represents individual items or services included in a quotation.

Fields

- **`quotation_id`** **(Many2one)**: Links the line item to a specific quotation.
- **`project_id`** **(Many2one)**: Links the line item to a **Project Scope**.
- **`quantity`** **(Float)**: Quantity of the item or service. Defaults to `1.0`.
- **`unit_price`** **(Float)**: Price per unit of the item or service.
- **`subtotal`** **(Float)**: Computed field representing the total cost of the line (`quantity * unit_price`).

Methods

- `_compute_subtotal(self)`:
  Computes the `subtotal` for each line item based on the quantity and unit price.

Workflow

1. Quotation Lifecycle

1. **Creation**: A quotation is created in **Draft** state, with optional details such as requirements and line items.
2. **Sending**:
   - Mark the quotation as **Sent** to notify the customer and generate a report.
3. **Acceptance**:
   - Update the status to **Accepted** or **Rejected** based on customer feedback.
4. **Project Creation**:
   - If accepted, create a project from the quotation, linking relevant details.

2. Generating Reports

- Quotations generate reports when marked as **Sent** or **Signed**.

Dependencies

- **Lead Management**: Links quotations to customer leads for better tracking.
- **Project Scope**: Provides predefined scopes for quotation line items.
- **Project Management**: Facilitates project creation directly from quotations.

- Simple Invoice Module Documentation

Overview

This module defines two main models, **Invoice** and **InvoiceLine**, which together represent a basic invoicing system.

Models

1. Invoice

The primary model for managing invoices.

Fields

- **`name`** **(Char)**: Unique invoice number. Auto-generated if not provided.
- **`customer_name`** **(Char)**: Name of the customer associated with the invoice.
- **`invoice_date`** **(Date)**: Date of the invoice. Defaults to the current date.
- **`due_date`** **(Date)**: Due date for payment.
- **`total_amount`** **(Float)**: Total amount of the invoice, computed from the sum of line subtotals.
- **`state`** **(Selection)**: Status of the invoice. Options are:
  - Draft (default)
  - Sent
  - Paid
- **`line_ids`** **(One2many)**: Relationship to **InvoiceLine**, representing individual items in the invoice.
- **`project_id`** **(Many2one)**: Links the invoice to a specific project in the **project.management** model.

Methods

- **`create(vals)`**:
  Overrides the `create` method to auto-generate the invoice number using Odoo's sequence API.
- **`_compute_total_amount(self)`**:
  Computes the total amount by summing the `subtotal` field from all related **InvoiceLine** records.
- **`action_send(self)`**:
  Changes the state of the invoice to **Sent** and triggers the invoice report generation.

- `action_mark_paid(self)`:
  Changes the state of the invoice to **Paid**.

2. InvoiceLine

Represents individual line items in an invoice.

Fields

- `invoice_id` **(Many2one)**: Links the line item to a specific invoice.
- `description` **(Char)**: Description of the product or service.
- `quantity` **(Float)**: Quantity of the item. Defaults to `1.0`.
- `unit_price` **(Float)**: Unit price of the item.
- `subtotal` **(Float)**: Computed field. Product of `quantity` and `unit_price`.

Methods

- `_compute_subtotal(self)`:
  Computes the `subtotal` as `quantity * unit_price`.

Workflow

1. Creating an Invoice

1. Navigate to the **Invoices** menu in the Odoo application.
2. Fill in details like:
   - **Customer Name**
   - **Invoice Date**
   - **Due Date**
3. Add line items in the **Invoice Lines** tab:
   - Specify **Description**, **Quantity**, and **Unit Price**.

2. Invoice States

- **Draft**: The initial state when an invoice is created.
- **Sent**: Use the **Send** button to change the state to **Sent** and generate a report.
- **Paid**: Use the **Mark as Paid** button to mark the invoice as paid.

3. Generating Reports

Invoices automatically generate reports when the **Send** action is triggered.

Dependencies

- **ir.sequence**: Used for generating unique invoice numbers.
- **project.management**: (Optional) Links invoices to projects if this module is installed.

- Project Management Module Documentation

Overview

The **Project Management Module** allows organizations to track projects, manage tasks, and link projects to quotations and invoices. This module supports project lifecycle management and includes task assignment and tracking features.

Models

1. Project

Represents a project created from an accepted quotation, including its lifecycle, associated tasks, and invoicing.

Fields

Basic Fields

- `name` *(Char)*: Unique identifier for the project. Auto-generated if not provided.
- `quotation_id` *(Many2one)*: Links the project to a customer quotation.
- `invoice_id` *(Many2one)*: Links the project to an associated invoice.
- `project_manager` *(Many2one)*: User assigned as the project manager. Defaults to the current user.
- `client` *(Char)*: Name of the client associated with the project.
- `start_date` *(Date)*: Start date of the project.
- `end_date` *(Date)*: End date of the project.
- `total_amount` *(Float)*: Total project amount, used in invoicing.

- `state` *(Selection)*: Lifecycle state of the project. Possible values:
  - **Draft** (default)
  - **In Progress**
  - **Completed**
- `task_ids` *(One2many)*: Tasks associated with the project.

Group Expand

- `_group_expand_states(self, states, domain, order)`:
  Ensures all states are displayed in grouped views, even if no records exist for specific states.

Methods

Override Create

- `create(self, vals)`:
  Automatically generates a unique sequence for the project reference (`name`) if not provided.

Actions

- `action_start_project(self)`:
  Updates the state to **In Progress**, signaling the project's initiation.
- `action_complete_project(self)`:
  Changes the state to **Completed**, marking the project as finished.
- `action_make_payment(self)`:
  Generates an invoice for the project and links it to the project.
  - **Invoice Fields:**
    - Customer name: Fetched from the project.
    - Due date: Set as the project's end date.
    - Total amount: Project's `total_amount`.
  - Creates an **Invoice Line** for payment, with a description of "Project Payment."

2. Task

Represents tasks associated with a project, supporting individual task management and assignment.

Fields

- `name` *(Char)*: Name of the task.
- `project_id` *(Many2one)*: Links the task to its parent project.
- `assigned_to` *(Many2one)*: User assigned to the task.

- `deadline` *(Date)*: Deadline for completing the task.
- `description` *(Text)*: Detailed description of the task.
- `state` *(Selection)*: Lifecycle state of the task. Possible values:
  - To Do (default)
  - In Progress
  - Done

Methods

Actions

- `action_start_task(self)`:
  Updates the state to **In Progress**, signaling that the task is being worked on.
- `action_complete_task(self)`:
  Changes the state to **Done**, marking the task as finished.

Workflow

1. Project Lifecycle

1. **Draft Creation**:
   Projects begin in the **Draft** state.
2. **Start Project**:
   Change the state to **In Progress** using the **Start Project** action.
3. **Task Management**:
   Assign tasks, update their states, and track progress.
4. **Completion**:
   Mark the project as **Completed** once all tasks and deliverables are finalized.

2. Invoicing

- Generate an invoice using the **Make Payment** action.
- Automatically creates an invoice record with line items for the project's total amount.

Dependencies

- **Quotation Management**: Links projects to customer quotations.
- **Invoice Management**: Supports invoicing and payment tracking for projects.