**Technical Design Document (TDD)  
C&C CRM MVP Journey Planning**

**Version:** 1.0  
**Date:** 2025-05-21  
**Prepared By:** Solution Delivery Team

**1. Introduction**

This Technical Design Document (TDD) translates the requirements of the Functional Design Document (FDD) for the "C&C CRM MVP Journey Planning" solution into actionable technical specifications for implementation within Microsoft Dynamics 365 CE. It details all customizations, configurations, integration approaches, data migration, PCF opportunities, and security controls required for a robust and scalable solution.

**2. Solution Architecture Overview**

**2.1 Key Solution Components**

|  |  |  |
| --- | --- | --- |
| **Component** | **Description** | **Type** |
| Journey | Tracks customer journeys from onboarding to closure | Custom Entity |
| Task | Tasks/activities assigned within a journey | Custom Entity |
| Interaction | Logs of customer touchpoints (call, email, meeting, etc.) | Custom Entity |
| Escalation | Manages escalations & exceptions | Custom Entity |
| Customer (Account/Contact) | Customer information; extended for journey linkage | Standard Entity (Extended) |
| User | CRM Users (Planner, Agent, Supervisor, etc.) | Standard Entity |

**3. Entity & Attribute Details**

**3.1 Journey**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Required** |
| Journey Name | Text | Name of journey | Yes |
| Customer | Lookup (Account/Contact) | Linked customer | Yes |
| Status | Option Set | Not Started, In Progress, Completed, On Hold, Cancelled | Yes |
| Start Date | Date | Start of journey | Yes |
| End Date | Date | Completion date | No |
| Type | Option Set | Journey Type | Yes |
| Owner | User Lookup | Journey owner | Yes |
| Created On | DateTime | Record timestamp | Yes |

**3.2 Task**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Required** |
| Task Name | Text | Task title | Yes |
| Journey | Lookup (Journey) | Associated journey | Yes |
| Assigned To | User Lookup | Task assignee | Yes |
| Status | Option Set | Open, In Progress, Done | Yes |
| Due Date | Date | Deadline for task | Yes |
| Dependency | Lookup (Task) | Dependent task | No |
| Is Overdue | Calculated | True if Due Date < Today and not Done | No |

**3.3 Interaction**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Required** |
| Interaction Type | Option Set | Call/Email/Meeting/etc. | Yes |
| Interaction Date | DateTime | Date/time of interaction | Yes |
| Summary | Text | Brief description | Yes |
| Outcome | Text | Interaction result | No |
| Journey | Lookup (Journey) | Linked journey | Yes |
| Created By | User Lookup | Creator | Yes |

**3.4 Escalation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Required** |
| Escalation Reason | Option Set | Reason for escalation | Yes |
| Journey | Lookup (Journey) | Associated journey | Yes |
| Escalated To | User Lookup | Supervisor/manager | Yes |
| Status | Option Set | Open/Closed | Yes |
| Resolution Note | Text | Resolution details | No |

**4. Customization Analysis**

**4.1 Custom Entities, Fields, Business Rules, BPFs**

* **Journey Entity:** Custom entity with business rules for status transitions (e.g., cannot mark as Completed unless all tasks are Done).
* **Task Entity:** Custom entity with calculated field for "Is Overdue", plugin/workflow to escalate if overdue, business rule enforcing due date logic.
* **Interaction Entity:** Timeline control on journey form; business rule to require interaction on journey closure.
* **Escalation Entity:** Created by plugin or workflow when task is overdue; plugin to notify supervisor.
* **Business Process Flow:** Journey BPF with stages: Initiate → Plan → Execute → Close, each with mandatory and optional steps.

**4.2 Configuration Steps (Example: Journey Status Enforcement)**

1. Create custom entity "Journey" with required fields.
2. Define Option Sets for Status and Type.
3. Add business rule: Prevent status change to Completed unless all related tasks are Done.
4. Configure BPF with appropriate stages and required fields.

**4.3 JavaScript Web Resource (Due Date Validation)**

On Task form save:  
If Due Date is before Assignment Date:  
  Show error "Due date cannot precede assignment date"  
  Prevent save

**4.4 Plugin (Auto-create Tasks on Journey Creation)**

On Journey record create:  
  If journey type has default tasks:  
    For each default task:  
      Create Task linked to Journey  
      Assign to user/role as per configuration

**4.5 Plugin (Escalate Overdue Tasks)**

On Task status change or scheduled check:  
  If Task is overdue and not Done:  
    Create Escalation record  
    Notify supervisor/manager

**5. Role & Privilege Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role** | **Journey** | **Task** | **Interaction** | **Escalation** | **Customer** |
| Journey Planner | Create/Read/Update/Delete | Create/Read/Update | Read/Create | Read/Create | Read |
| Customer Service Agent | Read | Create/Read/Update | Create/Read | Read | Read |
| Supervisor/Manager | Read/Update | Read/Update | Read | Create/Read/Update/Close | Read |
| Admin | Full | Full | Full | Full | Full |

**6. Integration Strategy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Integration Point** | **External System** | **Data Flow** | **Integration Method** | **Security/Handling** |
| ERP Data Sync | ERP System | Customer & Journey info (bi-directional) | REST API + Plugin | OAuth2, field mapping, error retries, logs |
| Marketing Automation | Marketing Platform | Journey outcome, customer segments | Outbound API/Webhook | Token auth, queue on fail, admin notification |
| Telephony/Email | Exchange/Telephony API | Interaction logs (inbound) | OOB Connector, Scheduled Import | Service account, error notification |

**Data Mapping Example (ERP)**

|  |  |  |
| --- | --- | --- |
| **D365 Field** | **ERP Field** | **Transformation** |
| Journey Name | ERP\_JourneyDesc | String copy |
| Customer | ERP\_CustomerID | Lookup by unique ID |
| Status | ERP\_Status | Option set mapping |

**Integration Pseudocode**

On data change (Journey/Customer):  
  Prepare payload  
  Authenticate with external system  
  POST/PUT data via API  
  If error:  
    Log error  
    Retry up to 3 times  
    If still failing, notify admin

**7. Data Migration Approach**

|  |  |
| --- | --- |
| **Aspect** | **Description** |
| Source Systems | Legacy CRM, ERP, CSV files |
| Data Cleansing | Deduplicate, validate required fields, standardize phone/email |
| Transformation | Map source to D365 attributes, convert option sets |
| Migration Approach | KingswaySoft/SSIS Data Migration Tool, custom scripts if needed |
| Validation | Reconcile record counts, run QA reports, spot checks |

**Migration Pseudocode**

For each source record:  
  Clean data  
  Map to D365 schema  
  Validate required fields  
  If valid, create/update record  
  Else, log error  
After migration:  
  Run validation queries/reports  
  Reconcile totals

**8. PCF Component Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case** | **Component Name** | **Description** | **Input/Output** | **References** |
| Journey Timeline Visualization | JourneyTimelinePCF | Visualizes journey stages, tasks, and interactions on a timeline | Input: Journey ID; Output: Timeline events | [Sample Timeline PCF](https://learn.microsoft.com/power-apps/developer/component-framework/sample-controls/timeline-control) |
| Task Dependency Graph | TaskDependencyVisualizer | Graphical view of task dependencies for a journey | Input: Journey ID; Output: Task graph | [PCF Gallery](https://pcf.gallery/) |

**PCF Pseudocode**

On component load:  
  Fetch Journey/Tasks data  
  Render visual elements based on stage/dependency  
  Handle user clicks for details

**9. Security Considerations**

* **Customizations:** Use field-level security for sensitive data; expose JavaScript/plugins only to authorized security roles; restrict BPF stage transitions based on role.
* **Integrations:** Use OAuth2 or service principals, encrypt all data in transit, enable logging for all integration calls and errors.
* **Data Migration:** Mask sensitive fields in logs, restrict migration tool access to authorized users only, always test in non-production environments first.
* **PCF Controls:** Ensure only authorized users can trigger updates via PCF; validate input before processing.

**10. Code Snippets & Visuals**

**Task Overdue Calculation (Calculated Field)**

If (Due Date < Today) and (Status != Done)  
  Then IsOverdue = true  
Else  
  IsOverdue = false

**Process Flow Diagram**

**Security Considerations  
C&C CRM MVP Journey Planning – Technical Design**

This section describes the security measures and best practices necessary for the safe and compliant implementation of all technical components in the C&C CRM MVP Journey Planning solution. Security controls are outlined for customizations, integrations, data migration, and Power Apps Component Framework (PCF) controls. The recommendations below ensure data integrity, privacy, least privilege access, and regulatory compliance (e.g., GDPR).

**1. Customizations (Entities, Plugins, Workflows, JavaScript)**

* **Role-based Security:**
  + Assign granular privileges to custom entities (Journey, Task, Interaction, Escalation) using security roles.
  + Restrict Create, Read, Update, Delete actions based on user role (e.g., Planner, Agent, Supervisor, Admin).
* **Field-level Security:**
  + Protect sensitive fields (e.g., escalation notes, customer details) with field-level security profiles.
  + Ensure only authorized users can view or edit confidential information.
* **Business Logic Security:**
  + Plugins and workflows must check user context and permissions before performing data changes.
  + JavaScript must not expose privileged operations to unauthorized users (validate role before enabling UI actions).
* **Audit & Logging:**
  + Enable auditing on all custom entities and fields where compliance or traceability is required.
  + Log all plugin executions that impact critical business processes.
* **Data Validation & Input Sanitization:**
  + Validate all inputs (including those via JavaScript) to prevent injection or malformed data.
  + Enforce server-side validation in plugins and workflows.
* **Secure Solution Deployment:**
  + Deliver managed solutions to production environments to prevent unauthorized changes.

**2. Integrations**

* **Authentication & Authorization:**
  + Use OAuth 2.0, Client Secret, or Certificate-based authentication for all API integrations.
  + Limit integration credentials to only the permissions needed (least privilege principle).
* **Data in Transit:**
  + All integration traffic must use HTTPS/TLS 1.2+ encryption.
* **Data Scoping & Filtering:**
  + Integrations should only exchange the minimum necessary fields/data (avoid oversharing).
* **Error Handling & Monitoring:**
  + Catch and log all integration errors with enough detail for troubleshooting, but never log raw secrets/tokens.
  + Alert administrators of failed syncs or repeated errors.
* **Audit & Logging:**
  + Log all data exchanges with external systems for traceability and compliance.
* **API Rate Limiting & Throttling:**
  + Implement retry logic and respect partner system rate limits to prevent lockouts or service degradation.

**3. Data Migrations**

* **Access Control:**
  + Restrict migration tool usage to authorized admins only. Never perform migrations with user accounts having excessive privileges.
* **Data Privacy & Masking:**
  + Mask or redact sensitive data in migration logs and reports (e.g., PII, credentials).
  + Encrypt migration files at rest and in transit (use SFTP, Azure Blob Storage with encryption, etc.).
* **Validation & Reconciliation:**
  + Perform post-migration validation to verify no unauthorized data exposure or corruption.
* **Audit:**
  + Maintain audit trails for all migration runs (who, when, what).
* **Non-production Testing:**
  + Conduct all test migrations with masked/anonymized data in non-production environments before any live migration.

**4. Power Apps Component Framework (PCF) Controls**

* **Data Access:**
  + PCF controls must use the current user's security context and never elevate privileges client-side.
  + Restrict PCF visibility and actions based on user role and form context.
* **Input Validation:**
  + Sanitize and validate all user input handled by PCF controls before processing or saving.
* **Secure Communication:**
  + Ensure that any data exchanged with web resources or external APIs is encrypted (HTTPS).
* **Deployment:**
  + Deploy PCF controls via managed solutions and validate that only authorized users have access to add/update/remove them.

**5. General Security Practices**

* Enforce Multi-Factor Authentication (MFA) for all users, especially admins and those with integration/data migration privileges.
* Adhere to GDPR and organizational data protection policies at all times.
* Review and update security roles, field security profiles, and audit settings regularly as part of operational governance.
* Document and communicate all security controls to stakeholders and administrators.