**Admissions-Elevate-CRM Project – Full Explanation**

# Phase 1: Problem Understanding & Industry Analysis

## 1. Requirement Gathering

* + Track prospective students from their first inquiry through to the final admission decision.
  + Manage a formal, multi-stage application review process (e.g., Submitted, Under Review, Accepted).
  + Automate key communications, such as sending an email to confirm an application has been received.
  + Provide a central place to view all applicant data and related documents.

## 2. Stakeholder Analysis

* **Admissions Director (Manager):** Needs to monitor application volumes and team performance through reports and dashboards.
* **Admissions Officers (End Users):** Need an efficient system to manage their assigned applicants and review application details.
* **Faculty Reviewers (Specialized Users):** Need limited access to only see and provide feedback on applications assigned to them.

## 3. Business Process Mapping

* **Current Process:** A manual system using spreadsheets to track applicants and email to communicate. This is slow, prone to data entry errors, and provides no clear visibility into the admissions pipeline.
* **Future Process:** A prospective student (Lead) is captured from the university website. When they apply, an Application record is created. This record moves through a structured review process, automatically notifying the applicant of status changes.

## 4. Industry-specific Use Case Analysis

* In higher education, application deadlines are strict, and different academic programs often have unique requirements. The system must be able to manage these complexities and ensure all required documents are submitted and verified.

## 5. AppExchange Exploration

* Researched form-building tools like **FormAssembly** and document generation apps like **Conga**.
* **Decision:** To ensure the project is completely **cost-free**, all functionality will be built using native Salesforce features.

.

# Phase 2: Org Setup & Configuration

1. **Salesforce Editions**

○ A free **Developer Edition Org** has been created and is the active environment for all development.

## 2. Company Profile Setup

* The university's core information, including name, address, default time zone, and primary language, has been configured in Setup > Company Information.

## 3. Business Hours & Holidays

* The admissions office's standard operating hours (e.g., 9:00 AM - 5:00 PM, Monday-Friday) and key university holidays have been set.

1. **Fiscal Year Settings**

○ A **Standard Fiscal Year** beginning in January has been configured for reporting purposes..

1. **User Setup & Licenses**

○ Example user records for the primary roles (Admissions Director, Admissions Officer) have been created and assigned **Salesforce** user licenses..

## 6. Profiles

## 

* A custom profile named **"Admissions Officer"** has been cloned from a base profile to provide specific, limited access to the system. The System Administrator profile is assigned to the Director.

7. **Roles**

○ A role hierarchy has been established with the Admissions Director at the top and Admissions Officers reporting to them to ensure proper data visibility for reporting.

## 8. OWD (Org-Wide Defaults)

* The Organization-Wide Default for the custom Application object has been set to **Private** to ensure applicant data is secure and confidential by default.

1. **Sharing Rules**

* No sharing rules have been created yet. They will be added in a later phase if specific users need access to records they do not own.

1. **Login Access Policies**

* The default login and password policies for the org have been reviewed and confirmed.

1. **Dev Org Setup**

* The Developer Org has been created and is fully configured.

## 12. Deployment Basics

* The Developer Org is connected to a VS Code project. All configurations from Phase 1 and 2 have been successfully retrieved and committed to a public GitHub repository for version control

# Phase 3 : Data Modeling & Relationships

1. **Standard & Custom Objects**

* + **Standard Objects:** We will use the standard Contact object to store information about the applicants/students. This allows us to leverage built-in features for managing person details.
  + **Custom Objects:**
* **Application (Application\_\_c):** This is the central object of the project. It will hold all the details related to a single student's application to the university.
* **Review (Review\_\_c):** This object will store the feedback and recommendations submitted by faculty members for a specific application.

.

## 2. Fields

* **Application Object Fields:**
* Status (Picklist): Tracks the application's current stage (e.g., Submitted, Under Review, Accepted, Rejected).
* Program of Interest (Picklist): The academic program the student is applying for (e.g., Computer Science, Business Administration).
* Submission Date (Date): The date the application was officially submitted.
* **Review Object Fields:**
* Recommendation (Picklist): The reviewer's official recommendation (e.g., Strongly Recommend, Waitlist, Reject).
* Reviewer Comments (Long Text Area): Detailed feedback from the faculty reviewer.

## 3. Record Types

* **Application Record Types:** Two record types have been created on the Application object: "Undergraduate Admission" and "Graduate Admission." This allows the university to have different page layouts, picklist values, and business processes for each application type in the future.

1. **Page Layouts**

* The Application record page has been customized to show the most relevant information at a glance.
* Crucially, the Reviews related list has been added to the Application page layout, so admissions officers can easily see all feedback associated with an application in one place.

1. **Compact Layouts**

## The compact layout for the Application object has been configured to show the Applicant's Name, Status, and Program of Interest. This ensures key information is visible in list views and on mobile devices.

## Schema Builder

## 

* The Schema Builder tool was used to visualize the relationships between the Contact, Application, and Review objects, confirming the data model is structured correctly.

1. **Lookup vs. Master-Detail vs. Hierarchical Relationships**

* **Application to Contact:** This is a **Lookup Relationship**. An application is related to a contact (the applicant), but they are independent records.
* **Review to Application:** This is a **Master-Detail Relationship**. A Review is a direct child of an Application. If an application record is deleted, all of its associated review records are automatically deleted as well. This ensures data integrity.

## 8. Junction Objects

* A junction object is not required for the current data model. This would be needed if, for example, one review could be linked to multiple applications, which is not the case in our process.

1. **External objects**

* External objects are not in scope for this project. They could be used in a future phase to connect Salesforce with an external Student Information System (SIS) without migrating the data directly.