# **Campus Help Desk CRM: A Comprehensive Salesforce Platform for Student Support**

# **Project Implementation Phases Documentation** **PREPARED BY: Budidha Saanvi Reddy**

# 

## **Phase 1: Problem Understanding & Industry Analysis**

>**Goal:** Understand what we’re building and why.

1. **Requirement Gathering**
   * Talk to stakeholders (students, faculty, admin, IT support).
   * Example requirements:  
     + Students can raise help tickets (IT, hostel, academics, library).
     + Departments should get assigned tickets automatically.
     + Track ticket progress and resolution time.
     + Generate reports on student satisfaction and issue trends.
2. **Stakeholder Analysis**
   * **Admin** (you, managing Salesforce setup).
   * **Students** (create queries/tickets).
   * **Faculty/Staff** (resolve tickets in their department).
   * **Campus Manager** (monitor dashboards, approve escalations).
3. **Business Process Mapping**
   * Flow:  
      Student raises ticket → System auto-assigns to department → Staff works on issue → If unresolved, escalate → Ticket closed → Student notified.
4. **Industry-Specific Use Case Analysis**
   * In education, issues span multiple departments (IT, hostel, academics).
   * Key focus: transparency, speed, and accountability in resolving student problems.
   * Solution must track cases, automate escalations, and provide a knowledge base.
5. **AppExchange Exploration**
   * Look for “Help Desk” or “Education Cloud” apps.
   * We’ll build a simpler custom version for learning.

## **Phase 2: Org Setup & Configuration**

> **Goal:** Prepare Salesforce environment.

1. Salesforce Edition – Use Developer Org.
2. Company Profile Setup – University/College info, IST timezone.
3. Business Hours & Holidays – Define help desk timings (9 AM – 5 PM).
4. Fiscal Year – Standard Jan–Dec.
5. User Setup – Create roles: Student, Faculty, Admin, Manager.
6. Profiles – Faculty: resolve tickets only; Manager: full access.
7. Roles – Manager on top, Faculty/Staff below.
8. Permission Sets – Special access for generating reports.
9. OWD – Tickets private, Knowledge Articles public.
10. Sharing Rules – Tickets shared within same department.
11. Login Access – Restrict staff login to working hours.
12. Dev Org Setup – Sandbox for testing.
13. Sandbox Usage – Build/test before production.
14. Deployment Basics – Use Change Sets.

## **Phase 3: Data Modeling & Relationships**

>**Goal:** Build data structure.

1. **Objects**
   * Standard: Contact (students).
   * Custom: Department, Help Ticket.
2. **Fields**
   * Ticket: Category (IT/Hostel/Library/Academic), Priority, Status, Resolution Time.
3. **Record Types**
   * Ticket: IT Issue, Hostel Issue, Academic Issue, General.
4. **Page Layouts**
   * Ticket page shows student info + department assignment.
5. **Compact Layouts**
   * Ticket shows Category, Priority, Status.
6. **Schema Builder**
   * Visualize relationships (Student ↔ Ticket ↔ Department).
7. **Relationships**
   * Ticket → Student (Lookup).
   * Ticket → Department (Lookup).

## **Phase 4: Process Automation (Admin)**

## **>** **Goal:** Automate tasks.

1. Validation Rules – Ticket must have category & priority.
2. Workflow Rules – Auto-send email when ticket created.
3. Process Builder (legacy) – Replaced by Flow.
4. Approval Process – Escalate if ticket unresolved after X hours.
5. Flow Builder – Auto-assign tickets based on category.
6. Email Alerts – Notify student when ticket is resolved.
7. Field Updates – Status auto-updates to “Closed” after resolution.
8. Tasks – Assign faculty task to follow up.
9. Notifications – Send in-app notification to staff on new ticket.

## **Phase 5: Apex Programming (Developer)**

> **Goal:** Add advanced logic.

1. Classes – TicketService class.
2. Apex Triggers – Prevent duplicate tickets for same issue by same student.
3. Trigger Handler Pattern – For clean code.
4. SOQL/SOSL – Query: Tickets WHERE Status = “Open.”
5. Collections – Store multiple ticket IDs.
6. Control Statements – If overdue → escalate.
7. Batch Apex – Auto-close old tickets weekly.
8. Queueable Apex – Async notifications.
9. Scheduled Apex – Daily summary of pending tickets emailed to Manager.
10. Future Methods – Call external survey API for feedback.
11. Exception Handling – Handle ticket assignment errors.
12. Test Classes – Ensure triggers and logic work.
13. Async Processing – Batch + Queueable + Future.

## **Phase 6: User Interface Development**

> **Goal:** Make it user-friendly.

1. Lightning App – “Campus Help Desk CRM.”
2. Record Pages – Ticket record page with student + department details.
3. Tabs – Tickets, Departments, Students, Knowledge Base.
4. Home Page – Dashboard of open vs resolved tickets.
5. Utility Bar – Quick “New Ticket” action.
6. LWC – Component: Submit new ticket.
7. Apex + LWC – Search past tickets.
8. Events – Update ticket list after submission.
9. Wire Adapters – Show open tickets for student.
10. Imperative Apex Calls – Create ticket.
11. Navigation Service – Redirect student to ticket details.

## **Phase 7: Integration & External Access**

> **Goal:** Connect with outside systems.

1. Named Credentials – Store feedback API keys.
2. External Services – Integrate with student survey tool.
3. Web Services – REST callout for campus announcements.
4. Callouts – Notify external SMS service.
5. Platform Events – Publish event if critical issue raised.
6. Change Data Capture – Notify system when ticket is updated.
7. Salesforce Connect – Pull external student data if needed.
8. API Limits – Monitor calls.
9. OAuth – Student login via portal.
10. Remote Site Settings – Allow callouts.

## **Phase 8: Data Management & Deployment**

>**Goal:** Manage data & move changes.

1. Import Wizard – Import student records.
2. Data Loader – Bulk import tickets.
3. Duplicate Rules – Prevent duplicate student records.
4. Data Backup – Weekly exports.
5. Change Sets – Move configs to production.
6. Managed Packages – Optional for publishing.
7. ANT Tool – For deployment.
8. VS Code + SFDX – Dev-friendly builds.

## **Phase 9: Reporting, Dashboards & Security Review**

> **Goal:** Monitor performance & secure system.

1. Reports – Open vs Closed Tickets, Avg. Resolution Time.
2. Report Types – Ticket + Department.
3. Dashboards – Manager Dashboard for student issues.
4. Dynamic Dashboards – Each faculty sees only their tickets.
5. Sharing Settings – Tickets private.
6. Field Security – Hide sensitive student details from staff.
7. Session Settings – Timeout after 30 mins.
8. Login IP Ranges – Restrict staff access.
9. Audit Trail – Track system changes.

## **Phase 10: Final Presentation & Demo Day**

> **Goal:** Deliver project like real implementation.

1. Pitch – Problem → Solution → Benefits.
2. Demo – Student creates ticket → Faculty resolves → Notification sent → Reports generated.
3. Handoff Docs – System design + user guide.
4. Portfolio – Add to LinkedIn/Resume as Salesforce project.