Parking Slot Reservation System

Problem Statement:

In many companies, employees face daily challenges in finding available parking spaces. This leads to wasted time, employee frustration, and inefficiencies in managing parking resources. Currently, there is no automated way to check slot availability or reserve a space in advance.
To solve this, we will implement a Salesforce-based Parking Slot Reservation System that allows employees to easily book a parking slot, automatically update its availability, and provide management with real-time dashboards for monitoring utilization.

Project Implementation Phases

Phase 1: Problem Understanding & Industry Analysis

Goal: Understand what we're building and why.

1. Requirement Gathering

- Talk to stakeholders (facility manager, employees, security team, admin staff).
- Example requirements:
    - Track all parking slots with availability status.
    - Allow employees to reserve parking slots in advance.
    - Prevent double-booking of the same slot.
    - Generate utilization and availability reports.

2. Stakeholder Analysis

- **Admin** (you, managing system setup).

•

- **Employees** (create/manage reservations).
- **Manager/Facilities Head** (approves special reservations, monitors reports).
- **Security/Support Staff** (handles slot issues, validations, entry checks).

3. Business Process Mapping

- Flow:
  Employee requests slot → System checks availability → Reservation created →
  If special approval needed → Manager approves →
  Confirmation email sent to employee.

4. Industry-specific Use Case Analysis

- Parking slots are **limited resources**.
- Different slots may be reserved for different roles (VIP, guest, employee).
- Maintenance or blocked slots affect availability.
- So we need to:
  - Track slot availability.
  - Automate approvals.
  - Notify employees.

5. AppExchange Exploration

- Look for "Parking Management" apps. Some exist, but we'll build a simpler **custom Salesforce solution** to learn.

Phase 2: Org Setup & Configuration

Goal: Prepare Salesforce environment.

1. **Salesforce Editions**

- Use **Developer Edition Developer Org** (free dev org).

2. **Company Profile Setup**
    - Add company info, local time zone.
    - Set currency to INR/USD as per project.
3. **Business Hours & Holidays**
    - Define working hours (e.g., 8am–8pm).
    - Add public holidays (no reservations allowed).
4. **Fiscal Year Settings**
    - Standard (Jan–Dec) → good for reporting.
5. **User Setup & Licenses**
    - Create users: Employee, Manager, Admin. Assign Salesforce licenses.
6. **Profiles**
    - Employees: Can create reservations, view slots.
    - Manager: Full access.
7. **Roles**
    - Manager on top, employees below → visibility rolls up.
8. **Permission Sets**
    - For extra access (e.g., Reports).
9. **OWD (Org-Wide Defaults)**
    - Parking Slot object: Public Read Only.
    - Reservation object: Private (only owner/manager sees).
10. **Sharing Rules**
    - If reservation needs to be visible to security staff, add rules.

•

11. **Login Access Policies**
    - Restrict login hours (e.g., 7am–9pm for employees).
12. **Dev Org Setup**
    - Sandbox for building/testing.
13. **Deployment Basics**
    - Move config/code using **Change Sets** from Sandbox → Production.

Phase 3: Data Modeling & Relationships

Goal: Build data structure.

- **Standard & Custom Objects**
  - Standard: Contact (employees).
  - Custom: Parking Slot, Reservation.
- **Fields**
  - Parking Slot: Slot No, Location, Status (Available/Occupied), Type (VIP/Guest/Employee).
  - Reservation: Start Time, End Time, Employee, Slot, Status, Total Duration.
- **Record Types**
  - Reservation → "Employee Reservation" vs "Guest Reservation."
- **Page Layouts**
  - Parking Slot page shows reservation history.
  - Reservation page shows related slot & employee.
- **Compact Layouts**
  - Mobile: Slot No, Location, Status.
- **Schema Builder**
  - Visual tool → draw relationships.
- **Lookup vs Master-Detail**
  - Reservation ↔ Parking Slot → Lookup (slots are not owned by reservations).
  - Reservation ↔ Employee → Lookup.

•

- **Junction Objects**
  - Not required (one slot = one reservation at a time).
- **External Objects**
  - Could connect to external facility DB if needed.

Phase 4: Process Automation (Admin)

Goal: Automate tasks.

- **Validation Rules**
  - End Time must be after Start Time.
- **Workflow Rules (legacy)**
  - Auto-send email when reservation is created (use Flow instead).
- **Process Builder (legacy)**
  - Auto-update reservation status → replaced by Flow now.
- **Approval Process**
  - If VIP/Guest slot → send to Manager for approval.
- **Flow Builder**
  - Record-triggered Flow → calculate reservation duration.
  - Screen Flow → reservation form.
- **Email Alerts**
  - Confirmation email after approval.
- **Field Updates**
  - After approval, Reservation Status = “Confirmed.”
- **Tasks**
  - Create task for security staff to validate entry.
- **Custom Notifications**
  - Notify employee & security after approval.

Phase 5: Apex Programming (Developer)

Goal: Add advanced logic.

1. **Classes & Objects**
   - Create ParkingService class for reusable logic.
2. **Apex Triggers**

-
    - On Reservation Insert → prevent overlapping bookings for the same slot.

3. **Trigger Design Pattern**
    - Use handler class instead of writing directly in trigger.
4. **SOQL & SOSL**
    - Query: Available slots WHERE status = ‘Available’.
5. **Collections (Set, Map, List)**
    - Store multiple slot IDs to avoid duplicates.
6. **Control Statements**
    - If reservation times overlap → throw error.
7. **Batch Apex**
    - Night job to mark expired reservations as “Completed.”
8. **Queueable Apex**
    - Async notifications for bulk reservations.
9. **Scheduled Apex**
    - Morning job → email Manager daily reservation summary.
10. **Future Methods**

- Call external APIs async (e.g., gate entry system).

11. **Exception Handling**

- Catch errors if booking overlaps.

12. **Test Classes**

- Insert reservations, test validation.

13. **Asynchronous Processing**

- Batch + Queueable + Future jobs.

•

Phase 6: User Interface Development

Goal: Make it user-friendly.

1. **Lightning App Builder**
    - Create “Parking Reservation” app.
2. **Record Pages**
    - Parking Slot → list of reservations.
3. **Tabs**
    - Add Parking Slots & Reservations tabs.
4. **Home Page Layouts**
    - Dashboard of slot utilization.
5. **Utility Bar**
    - Quick “New Reservation” button.
6. **LWC (Lightning Web Components)**
    - Component: Search slots by date/time.
    - Show results in a datatable.
7. **Apex with LWC**
    - Imperative Apex call → create reservation.
8. **Events in LWC**
    - Child (search form) → sends results to parent.
9. **Wire Adapters**
    - Wire available slots list.
10. **Imperative Apex Calls**
    - Call Apex when “Reserve Now” clicked.
11. **Navigation Service**
    - After reservation → navigate to record page.

Phase 7: Integration & External Access

Goal: Connect with outside systems.

1. **Named Credentials**
    - Store gate/entry API credentials securely.
2. **External Services**
    - Integration with access control or facility system.

3. **Web Services (REST/SOAP)**
    - REST callout → notify security system.
4. **Callouts**
    - Triggered when reservation is created.
5. **Platform Events**
    - Publish event if slot breakdown/maintenance reported.
6. **Change Data Capture**
    - Notify external service if reservation updated.
7. **Salesforce Connect**
    - If external DB stores slot info.
8. **API Limits**
    - Monitor calls/day.
9. **OAuth & Authentication**
    - Employee login via portal.
10. **Remote Site Settings**

- Allow external calls.

.

Phase 8: Data Management & Deployment

Goal: Manage data and move changes.

1. **Data Import Wizard**
    - Import demo slot records.
2. **Data Loader**
    - Import bulk reservations.
3. **Duplicate Rules**
    - Prevent duplicate slot entries.
4. **Data Export & Backup**
    - Weekly backup.
5. **Change Sets**
    - Move from Sandbox → Production.
6. **Managed vs Unmanaged Packages**
    - Unmanaged for learning, Managed for publishing.
7. **ANT Migration Tool**
    - Command-line deployment.
8. **VS Code & SFDX**
    - Dev-friendly deployments.

Phase 9: Reporting, Dashboards & Security Review

Goal: Monitor business & secure data.

1. **Reports**
    - Slot Utilization (how many hours used).
    - Reservations by Employee/Department.
2. **Report Types**
    - Parking Slot + Reservation.
3. **Dashboards**
    - Slot Utilization Dashboard.
    - Manager's Reservation Dashboard.
4. **Dynamic Dashboards**
    - Each employee sees only their reservations.
5. **Sharing Settings**
    - Reservations private, slots public.
6. **Field Level Security**

•

    - Hide employee personal ID from others.
7. **Session Settings**
    - Timeout after 30 mins.
8. **Login IP Ranges**
    - Restrict to office IP.
9. **Audit Trail**
    - Track changes.

Phase 10: Final Presentation & Demo Day

Goal: Wrap up like a real project delivery.

1. **Pitch Presentation**
    - Problem → Solution → Benefits.
2. **Demo Walkthrough**
    - Show slot search, reservation, approval, notifications, dashboard.
3. **Handoff Documentation**
    - Share system design doc, user guide.
4. **LinkedIn/Portfolio Project Showcase**
    - Post demo video & highlights.

==========