

Phase 1: Problem Understanding & Industry Analysis

Overview

The first phase establishes a foundation by clearly understanding the problem, identifying stakeholders, analyzing the PlayStation gaming industry, and mapping existing business processes. This helps ensure the CRM solution aligns with real-world needs.

Activities Done

1. Requirement Gathering

- Conducted discussions with hypothetical stakeholders.
- Identified the following requirements:
 - Centralized gamer profile management (purchase history, subscriptions, preferences).
 - Subscription lifecycle management for PlayStation Plus & Premium.
 - Customer support case management & automated ticket routing.
 - Marketing automation for personalized promotions.
 - Developer/publisher partnership tracking (onboarding, revenue share).
 - Esports and event tracking for community engagement.

2. Stakeholder Analysis

- **Gamers** – End users; need personalized recommendations and faster support.
- **Customer Support Agents** – Manage cases and improve resolution time.
- **Marketing Teams** – Run campaigns for new game launches.
- **Developers & Publishers** – Manage contracts, game releases, and performance analytics.
- **PlayStation Management** – Require executive dashboards with KPIs on subscriptions, sales, and engagement.

3. Business Process Mapping

- Current: Multiple disconnected systems (payment, support, marketing).

- Issues: Delays in renewals, fragmented customer view, poor personalization.
- Future with Salesforce CRM: Single system to unify gamer data, automate renewals, support, and campaigns.

4. Industry-Specific Use Case Analysis

- Benchmarked Xbox Game Pass, Steam, Epic Games.
- Observed gaps PlayStation can solve: deeper personalization, loyalty tracking, AI-driven insights, and improved developer relations.

5. AppExchange Exploration

- Evaluated relevant tools:
 - Gamification Add-ons → to reward loyalty.
 - Survey Apps → collect gamer feedback.
 - Knowledge Base & Chatbot → reduce support burden.
 - Marketing Enhancements → improve campaign automation.

Phase 2: Org Setup & Configuration

Overview

In this phase, the Salesforce **Developer Edition** org was set up to provide the base environment for the PlayStation Gaming CRM project. The focus was on configuring only the most essential settings to keep the project simple, structured, and ready for development.

Activities Done

1. Salesforce Edition

- Selected **Developer Edition**, which provides essential CRM features such as custom objects, automation, reports, and dashboards.
- Chosen because it is free, easy to use, and suitable for student and training projects.

2. Company Profile Setup

- **Company Name:** Sony PlayStation CRM
- **Currency:** USD and INR
- **Locale/Time Zone:** English (United States), IST (GMT+05:30)

3. Business Hours & Holidays

- **Business Hours:** Configured as 24x7 (to simulate gamer support availability).
- **Holidays Added:** Christmas, New Year, Diwali.

4. Fiscal Year Settings

- Fiscal cycle configured as **April – March** to align with common financial reporting practices.

5. User Setup

- Added the following users for simulation:
 - **System Administrator** – Full access (project owner).
 - **Support Agent** – To handle gamer cases.
 - **Marketing Manager** – For campaigns and promotions.
 - **Publisher/Repair Team** – To manage game publisher/repair data.
- Assigned available licenses (limited in Developer Edition).

6. Profiles & Roles

- **Profiles Created:** Admin, Support, Marketing, Publisher/Repair.
- **Roles Defined:** CEO → Managers → Agents (basic hierarchy).

7. Security Settings

- **OWD (Organization-Wide Defaults):** Private for sensitive data such as subscriptions.

- **MFA (Multi-Factor Authentication):** Enabled for login security.

The screenshot shows the Salesforce Setup interface with the 'Roles' page selected. The left sidebar includes sections for 'Users' (with 'Roles' highlighted), 'Feature Settings', 'Sales', 'Service', and 'Case Teams'. A search bar at the top right says 'Search Setup'. The main content area is titled 'Creating the Role Hierarchy' and displays a tree view of roles under 'Sony PlayStation CRM'. The hierarchy includes 'CEO', 'Managers', 'SVP.Customer Service & Support', and 'SVP.Sales & Marketing'. Each node has 'Edit | Del | Assign' options and a 'Add Role' link.

The screenshot shows the Salesforce Setup interface with the 'Users' page selected. The left sidebar includes sections for 'Permission Set Groups', 'Permission Sets', 'Profiles', 'Public Groups', 'Queues', 'Roles', 'User Management Settings' (with 'Users' highlighted), and 'Feature Settings'. A search bar at the top right says 'Search Setup'. The main content area is titled 'Active Users' and displays a table of users. The table columns are 'Action', 'Full Name', 'Alias', 'Username', 'Role', 'Active', and 'Profile'. The table lists several users with their respective details and roles like 'CEO', 'Marketing_Team', and 'Customer Support, International'.

Action	Full Name	Alias	Username	Role	Active	Profile
<input type="checkbox"/> Edit Login	Admin_Himanshu	Owner	himanshu9981@gmail.com	CEO	<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/> Edit	Chatter_Expert	Chatter	chatty.00d0k000007bziduay.sxrribfc03s@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
<input type="checkbox"/> Edit Login	EPIIC_OrgFarm	OEPIC	epic.a72ed2b33ce5@ergfarm.salesforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/> Edit	Kumar_Himanshu	him	himanshu.kumar.aiml22673@agentforce.com		<input checked="" type="checkbox"/>	System Administrator
<input type="checkbox"/> Edit Login	Marketing_Sarah	smark	sarah_marketing@playstationcrm.com	Marketing_Team	<input checked="" type="checkbox"/>	Standard Platform User
<input type="checkbox"/> Edit Login	Support_Alex	asupp	alex.support@playstationcrm.com	Customer Support, International	<input checked="" type="checkbox"/>	Custom_Support Profile
<input type="checkbox"/> Edit	User_Integration	integ	integration@00d0k000007bziduay.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
<input type="checkbox"/> Edit	User_Security	sec	insightssecurity@00d0k000007bziduay.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

SETUP
Holidays

Holiday Detail

[Help for this Page](#)

Holidays are dates and times at which business hours are suspended. These dates and times, when associated with business hours, also suspend any escalation rules associated with business hours.

Add or remove business hours to holidays to suspend business hours and escalation rules during the holidays.

[Business Hours \[1\]](#)

Holiday Detail

[Edit](#) [Delete](#)

Holiday Name	Christmas
Description	
Date and Time	12/25/2025 All Day
Created By	Himanshu Kumar 9/22/2025, 9:20 AM

Last Modified By [Himanshu Kumar](#) 9/22/2025, 9:20 AM

[Edit](#) [Delete](#)

Business Hours

[Add/Remove](#)
[Business Hours Help](#)

Business Hours Name	Time Zone
24x7 Support	(GMT+05:30) India Standard Time (Asia/Kolkata)

[^ Back To Top](#)

Always show me [▼ more](#) records per related list

SETUP

Business Hours

Organization Business Hours

[Help for this Page](#)
[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#) [Other](#) [All](#)
[New Business Hours](#)

Action	Business Hours Name	Active	Time Zone	Default
Edit	24x7 Support	<input checked="" type="checkbox"/>	(GMT+05:30) India Standard Time (Asia/Kolkata)	<input checked="" type="checkbox"/>
Edit	Default	<input checked="" type="checkbox"/>	(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)	<input type="checkbox"/>
Edit	Gaming_CRM	<input checked="" type="checkbox"/>	(GMT-07:00) Pacific Daylight Time (America/Los_Angeles)	<input type="checkbox"/>

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#) [Other](#) [All](#)

P&O

Setup
Home
Object Manager

Search Setup
Star
Plus
Cloud
Question
Gear
Bell
Help

Company Settings
Company Information

Didn't find what you're looking for?
Try using Global Search.

The organization's profile is below.

[User Licenses \[10+\]](#) | [Permission Set Licenses \[10+\]](#) | [Feature Licenses \[11\]](#) | [Usage-based Entitlements \[10+\]](#)

Organization Detail		Edit	
Organization Name	Sony PlayStation CRM	Phone	+1-800-555-1234
Primary Contact	Himanshu	Fax	
Division		Default Locale	English (United States)
Address		Default Language	English
Fiscal Year Starts In	January	Default Time Zone	(GMT+05:30) India Standard Time (Asia/Kolkata)
Activate Multiple Currencies	<input type="checkbox"/>	Currency Locale	English (United States) - USD
Enable Data Translation	<input type="checkbox"/>	Used Data Space	384 KB (8%) [View]
Newsletter	<input checked="" type="checkbox"/>	Used File Space	125 KB (1%) [View]
Admin Newsletter	<input checked="" type="checkbox"/>	API Requests, Last 24 Hours	0 (15,000 max)
Hide Notices About System Maintenance	<input type="checkbox"/>	Streaming API Events, Last 24 Hours	0 (10,000 max)
Hide Notices About System Downtime	<input type="checkbox"/>	Restricted Logins, Current Month	0 (0 max)
Locale Formats	ICU	Salesforce.com Organization ID	00DgK000007bzID
		Organization Edition	Developer Edition

Phase 3: Data Modeling & Relationships

Overview

After completing Org Setup & Configuration (Phase 2), the focus of this phase was to design and implement the data model for the PlayStation Gaming CRM. The objective was to represent the key entities of the gaming ecosystem—gamers, games, subscriptions, and support cases—through structured objects and meaningful relationships.

This data model ensures that information is well-organized, interconnected, and easy to manage, forming the foundation for future automation, workflows, and reporting.

Activities Completed

1. Standard Objects Utilized

- **Contacts** → Represent gamers.
- **Cases** → Log and track gamer support issues.
- **Campaigns** → Manage PlayStation marketing campaigns and promotions.

2. Custom Objects Created

- **Games**  → Stores details such as Game Name, Genre, Release Date, and Price.
- **Subscriptions**  → Tracks gamer subscriptions with fields for Plan Type, Start Date, End Date, and Status.
- **Repairs**  → Records repair or installation service requests raised by gamers.

3. Relationships Defined

- **Contact → Subscriptions (One-to-Many)**: A single gamer can have multiple subscriptions.
- **Contact → Cases (One-to-Many)**: A single gamer can raise multiple support issues.
- **Contact ↔ Games (Many-to-Many)**: Gamers can play multiple games, and each game can be linked to multiple gamers through a junction object (*Game Participation*).

4. Page Layouts & Compact Layouts

- **Page Layouts**: Customized for Games and Subscriptions to capture detailed data fields.
- **Compact Layouts**: Configured to display key highlights such as Subscription Plan, Status, and End Date at a glance.

Setup Home Object Manager

SETUP > OBJECT MANAGER Games

Games Compact Layouts

Compact Layout Assignment

[Edit Assignment](#)

Primary Compact Layout

The primary compact layout defines the fields that display when this object's records appear as list view items in the mobile app.

Primary Compact Layout: Games Compact Layout

[Edit Assignment](#)

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules

Help for this Page ?

SETUP > OBJECT MANAGER Repairs

Repairs

Details
Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules
Conditional Field Formatting

Save Quick Save Preview As... Cancel Undo Redo Layout Properties

Quick Find Field Name Service Type

Repair Name GEN-2004-001234

Created By Sample Text Last Modified By Sample Text

Request Date 9/22/2025

Repair Request Info

Service Type Sample Text Status Sample Text Request Date 9/22/2025

Gamer Info Gamer Sample Text

Mobile Cards (Salesforce mobile only)

Open Activities

Subject	Name	Task	Due Date	Status	Priority	Assigned To
Sample Text	Sample Text	✓	9/22/2025, 11:59 AM	Sample Text	Sample Text	Sarah Sample

SETUP > OBJECT MANAGER Subscriptions

Subscriptions

Details
Fields & Relationships

Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Restriction Rules
Scoping Rules
Object Access
Triggers
Flow Triggers
Validation Rules

Fields & Relationships 10 Items. Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Contact	Contact_c	Lookup(Contact)		✓
Created By	CreatedById	Lookup(User)		
End Date	End_Date_c	Date		
Gamer	Gamer_c	Lookup(Contact)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Plan Type	Plan_Type_c	Picklist		
Start Date	Start_Date_c	Date		
Status	Status_c	Picklist		
Subscription Name	Name	Auto Number		✓

Phase 4: Process Automation & Business Logic

Overview

After establishing the **Org Setup & Configuration** in Phase 2 and designing the **Data Model & Relationships** in Phase 3, Phase 4 focused on embedding **automation and business logic** into the PlayStation Gaming CRM.

The main objective of this phase was to ensure that the system could automatically handle **repetitive business processes**, enforce **data integrity through validations**, and implement **approval workflows** where managerial intervention is required. This not only reduces manual workload but also improves the **efficiency, accuracy, and reliability** of the CRM.

Automation plays a key role in providing gamers with **timely updates**, ensuring that **support teams respond quickly**, and allowing managers to **maintain control over critical decisions** like discounts and high-cost repairs.

Activities Done

1. Workflow Rules

Workflow rules were used to implement **straightforward, rule-based automations** that run in the background whenever conditions are met.

- **Subscription Expiry Reminder:**
 - Automatically triggers an email notification to the gamer before their subscription expires.
 - Ensures gamers are aware in advance and can renew on time.
 - **Repair Update Notification:**
 - Sends an email or in-app notification to gamers when the status of their repair request changes (e.g., *Received* → *In Progress* → *Completed*).
 - Keeps gamers informed without manual follow-ups.
-

2. Process Builder

Process Builder allowed us to automate **multi-step logic and updates** beyond simple workflows.

- **Auto Case Assignment:**
 - New support cases raised by gamers are automatically assigned to the *Support Agent* role.

- This eliminates the need for manual routing and ensures faster response times.
 - **Auto Field Updates:**
 - Subscription records are automatically updated to *Expired* once the End Date passes.
 - Removes dependency on manual monitoring and ensures accurate reporting.
-

3. Approval Processes

Approval processes introduced **managerial oversight** into business workflows.

- **Discount Approval:**
 - If a gamer requests a subscription discount, the request is routed to the *Marketing Manager* for approval.
 - Prevents unauthorized discounts and maintains consistency.
 - **Repair Cost Approval:**
 - Repair requests exceeding a predefined cost limit are sent to the *System Administrator* (Admin) for approval.
 - Ensures budget control and prevents unnecessary expenses.
-

4. Validation Rules

Validation rules were implemented to maintain **data accuracy and completeness**.

- **Mandatory Subscription Dates:**
 - Users cannot save a subscription record unless both the Start Date and End Date are filled.
 - Prevents incomplete records and improves reporting accuracy.
 - **Repair Completion Validation:**
 - A repair request cannot be marked as *Completed* unless a valid *Service Date* is entered.
 - Ensures that closed repairs always have complete and reliable information.
-

5. Flows

Flows provided **guided, interactive automation** to help users carry out tasks more efficiently.

- **New Gamer Onboarding Flow:**
 - Step-by-step process for creating a new gamer (Contact), assigning them a subscription, and linking their games.
 - Simplifies onboarding and reduces errors for new records.

- **Repair Logging Flow:**

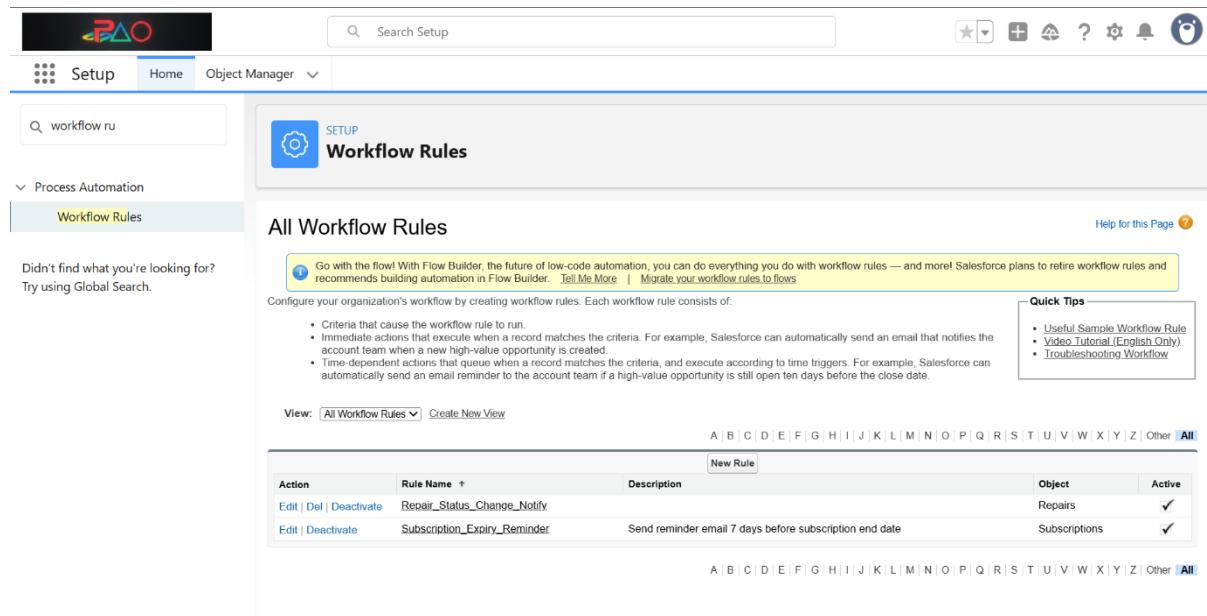
- Guides support staff through the repair request creation process by prompting for required details.
- Ensures all necessary data is captured at the time of logging.

Outcome

At the end of Phase 4, the PlayStation Gaming CRM became **intelligent and automated** rather than just a static data storage system. Key achievements include:

- **Routine tasks automated** → Reduces manual work and ensures efficiency.
- **Business rules enforced** → Approvals and validations guarantee accuracy and accountability.
- **Improved gamer experience** → Gamers receive timely updates on subscriptions and repairs.
- **Operational efficiency increased** → Support and marketing teams can focus on important tasks instead of repetitive manual updates.

This phase successfully transformed the CRM into a **smart system** capable of handling business processes on its own. The environment is now ready for **Phase 5: Reports & Dashboards**, where structured data and automation will be converted into meaningful business insights.



The screenshot shows the Salesforce Setup interface with the following details:

- Header:** P&AO, Search Setup, various icons (star, plus, question, gear, bell).
- Left Navigation:** Setup, Home, Object Manager, Workflow Rules (selected).
- Search Bar:** Q workflow ru.
- Section Header:** SETUP Workflow Rules.
- Message Bar:** Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with workflow rules — and more! Salesforce plans to retire workflow rules and recommends building automation in Flow Builder. [Tell Me More](#) | [Migrate your workflow rules to flows](#).
- Text:** Didn't find what you're looking for? Try using Global Search.
- Help:** Help for this Page.
- Quick Tips:**
 - Useful Sample Workflow Rule
 - Video Tutorial (English Only)
 - Troubleshooting Workflow
- Content Area:**
 - All Workflow Rules**
 - Configure your organization's workflow by creating workflow rules. Each workflow rule consists of:
 - Criteria that cause the workflow rule to run.
 - Immediate actions that execute when a record matches the criteria. For example, Salesforce can automatically send an email that notifies the account team when a new high-value opportunity is created.
 - Time-dependent actions that queue when a record matches the criteria, and execute according to time triggers. For example, Salesforce can automatically send an email reminder to the account team if a high-value opportunity is still open ten days before the close date.
 - View:** All Workflow Rules ▾ Create New View
 - Action** **Rule Name** **Description** **Object** **Active**
 - Repair_Status_Change_Notify (Repairs) ✓
 - Subscription_Expiry_Reminder (Subscriptions) ✓

Workflow Rules

Subscription_Expiry_Reminder

Workflow Rule Detail

Rule Name	Subscription_Expiry_Reminder	Object	Subscriptions
Active	✓	Evaluation Criteria	
Description	Send reminder email 7 days before subscription end date	Evaluate the rule when a record is created, and any time it's edited or subsequently meets criteria	
Rule Criteria	AND: NOT(ISBLANK(End Date) End Date <= TODAY())		
Created By	Himanshu Kumar	Modified By	Himanshu Kumar, 9/23/2025, 7:56 AM

Workflow Actions

Immediate Workflow Actions

No workflow actions have been added.

Time-Dependent Workflow Actions [See an example](#)

Type	Description
Email Alert	Send expiry email to gamer

⚠️ You cannot add new time triggers to an active rule. [Deactivate This Rule](#)

Email Alerts

Send expiry email to gamer

Email Alert Detail

Description	Send expiry email to gamer	Email Template	Subscription_Expiry_Notice
Unique Name	Send_expiry_email_to_gamer	Object	Subscriptions
From Email Address	Current User's email address		
Recipients	User: Himanshu Admin Related Contact: Gamer		
Additional Emails			
Created By	Himanshu Kumar, 9/23/2025, 7:50 AM	Modified By	Himanshu Kumar, 9/23/2025, 7:50 AM

Rules Using This Email Alert

Action	Rule Name	Description	Object	Active
Edit Deactivate	Subscription_Expiry_Reminder	Send reminder email 7 days before subscription end date	Subscriptions	✓

Approval Processes Using This Email Alert

This alert is currently not used by any approval processes

Entitlement Processes Using This Email Alert

This alert is currently not used by any entitlement processes

Flows Using This Email Alert

Flow Name	Version	Description	Object	Active
Always show me fewer more records per related list				

[^ Back To Top](#)

Setup **Home** **Object Manager**

Approval Proces

Process Automation

Approval Processes

Subscriptions

Didn't find what you're looking for?
Try using Global Search.

Try Flow Approval Processes!

Introducing **Flow Approval Processes**, an enhanced alternative to classic Approval Processes.

- Flexible, User-Friendly Interface: Create everything your approval process needs in Flow Builder.
- Record-Change Triggers: Automate approvals based on specific record updates.
- Apex Extensibility: Customize with Apex for advanced functionality.
- Detailed Logging: Ensure compliance with comprehensive audit trails.
- Dynamic Routing: Route approvals based on data and business rules.

Get started with Flow Approval Processes in the Approval app where you can manage approval submissions, approval work items, and flow approval processes in one location.

Open Approval App

Approve any complex business processes that require information gathering and planning before implementing. It is recommended that you follow the instructions below before getting started.

- Read the help topic
- View the checklist
- Create a new approval process using hierarchical relationship field
- Create email templates
- Configure approval process using either the Jump Start or Standard Wizard
- Add Approval History Related List to all page layouts
- Activate the process to deploy to your users

Manage Approval Processes For: Subscriptions

A listing of both active and inactive approval processes for Subscriptions is displayed below. To create a new approval process, click Create New Approval Process then select Use Jump Start Wizard to set up your approval process in a few short steps. Or, select Use Standard Wizard to configure all approval options.

Create New Approval Process

Active Approval Processes

Action	Process Order	Approval Process Name	Description
Edit Deactivate	1	Subscription_Discount_Approval	Approval process for discount requests on Subscriptions

Inactive Approval Processes

Setup **Home** **Object Manager**

Approval Proces

Process Automation

Approval Processes

Subscriptions: Subscription Discount Approval

Process Definition Detail

Process Name	Subscription_Discount_Approval	Active
Unlocked By	Subscription_Discount_Approval	Read authorized approver determined by
Approval Type	Subscription_Discount_Approval	Allow submitters to Recall Approval Requests
Entry Criteria	Subscription_Discount_Approval	Requested discount from 0
Record Categorizability	Administrator ONLY	
Approval Assignment Criteria	Marketing Manager	Approved by
Initial Submitter	Subscription Owner	Created By
Final Approver	Himanshu Kunte	Modified By
Created By	Himanshu Kunte	Created Date
Modified By	Himanshu Kunte	Modified Date

Initial Submission Actions

Action	Type	Description
Edit Remove	Field Update	Lock the record from being edited
Edit Remove	Field Update	Set Approval Status Pending
Edit Remove	Email Alert	Notify Marketing Manager of new discount request

Approval Steps

Action	Step Header	Name	Description	Owner	Assigned Approver	Record Behavior
Show Actions	Edit 1	Marketing Manager Approval Step	Step for Marketing Manager to approve subscription discount requests.	User:Subit Marketing	User:Subit Marketing	Final Rejection

Final Approval Actions

Action	Type	Description
Edit Remove	Field Update	Lock the record from being edited
Edit Remove	Field Update	Set Status Approved
Edit Remove	Field Update	Reset Discount Requested
Edit Remove	Field Update	Subscription_Discount_Request_After

Final Rejection Actions

Action	Type	Description
Edit Remove	Field Update	Unlock the record for editing
Edit Remove	Field Update	Set Status Rejected
Edit Remove	Field Update	Reset Discount Requested
Edit Remove	Email Alert	Subscription_Discount_Rejected_After

Recall Actions

Action	Type	Description
Edit Remove	Record Lock	Unlock the record for editing

Back To Step

Always show this footer 1 more records per related list

Subscriptions

Validation Rules

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
Mandatory_Dates	Top of Page	Error Detected	✓	Himanshu Kumar, 9/24/2025, 12:59 AM

Subscriptions Validation Rule

Validation Rule Detail

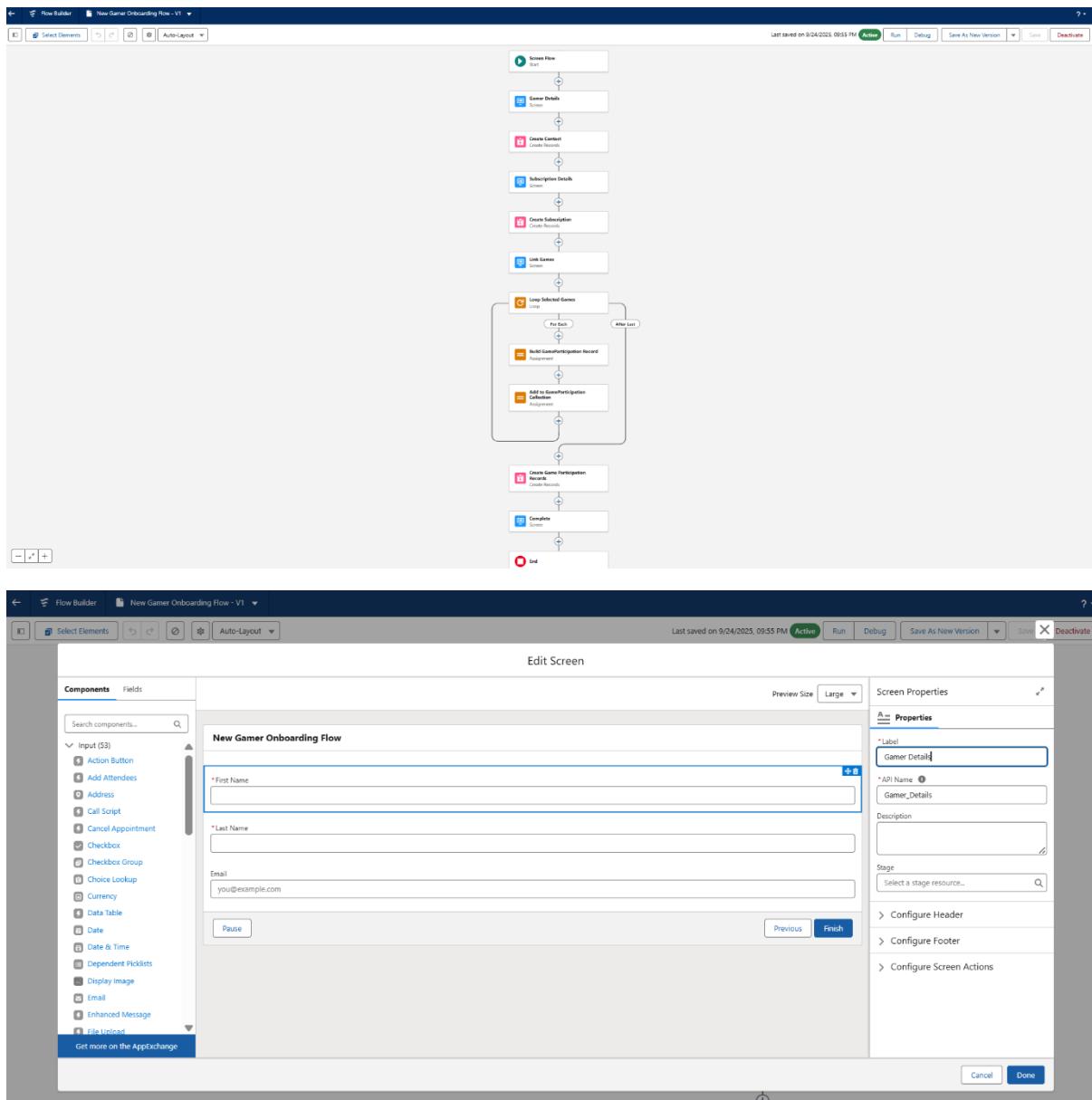
Rule Name	Mandatory_Dates	Edit	Close	Active	✓
Error Condition Formula	OR (ISBLANK(Start_Date__c) (ISBLANK(End_Date__c))				
Error Message	Error Detected	Error Location	Top of Page		
Description	Prevent saving Subscription unless Start Date and End Date are provided.	Created By	Himanshu Kumar, 9/24/2025, 12:59 AM	Modified By	Himanshu Kumar, 9/24/2025, 12:59 AM

Flows

Flow Definitions

All Flows

Flow Label	Process Type	Active	Template	Package State	Packag...	Last Modified By	Last Modified Date
New Gamer Onboarding Flow	Screen Flow	✓	□	Unmanaged		Himanshu Kumar	9/24/2025, 9:25 AM
Auto Expire Subscriptions	Autolaunched Flow	✓	□	Unmanaged		Himanshu Kumar	9/23/2025, 10:55 AM



Phase 5: Apex Programming (Developer)

5.1 Overview

This phase concentrated on developing and integrating **Apex programming logic** within the Salesforce environment. While previous phases (Org Setup, Data Modeling, and Declarative

Automation) established the system's foundation, this phase ensured that the CRM could handle **complex business requirements** which are not achievable through declarative features alone.

Apex, being Salesforce's proprietary, strongly typed, object-oriented language, was leveraged to implement:

- **Custom automation** beyond workflow rules and Process Builder.
- **Robust data processing** using governor-limit friendly patterns.
- **Scalable asynchronous operations** for long-running tasks.
- **Test-driven development** ensuring deployment readiness.

This phase transformed the PlayStation Gaming CRM into a **customizable, efficient, and enterprise-ready platform**.

5.2 Key Apex Components Implemented

1. Apex Classes & Objects

- **Apex Classes** were developed to encapsulate business logic into reusable components.
- Core functionalities like subscription validation, repair request handling, and game-to-gamer mappings were written inside **classes** for modularity.
- Salesforce **sObject types** (standard and custom objects) were used directly in Apex for **CRUD operations** (Create, Read, Update, Delete).
- Example: A SubscriptionHandler class was created to auto-calculate subscription expiry and send notifications.

2. Apex Triggers

- **Before Triggers**: Used for data validation (e.g., ensuring subscription start date < end date).
- **After Triggers**: Used for post-save actions (e.g., logging support case history or sending alerts).
- Implemented using the **Trigger Handler Pattern** (one trigger per object, delegating logic to a handler class).
- Benefits: Cleaner structure, easier maintenance, scalable when new requirements arise.

3. SOQL (Salesforce Object Query Language) & Collections

- **SOQL queries** were optimized to fetch only necessary fields to avoid governor limit exceptions.
- **Collections (List, Set, Map)** were used for bulk processing of records in loops.
- Examples:

- Lists to store multiple subscriptions.
 - Sets to check for unique game IDs.
 - Maps to quickly link gamer IDs with their active subscriptions.
 - Bulkification ensured that triggers/processes handled **hundreds of records at once** without failing.
-

4. Asynchronous Processing

Certain tasks needed to run **outside normal transaction limits** or on schedules. These were implemented as:

- **Future Methods:** For lightweight asynchronous tasks like sending email/SMS alerts.
- **Queueable Apex:** For chaining complex background jobs (e.g., recalculating subscription usage stats).
- **Scheduled Apex:** To automatically run jobs like subscription expiry checks or monthly data cleanup.
- **Batch Apex:** For handling **large datasets** such as recalculating thousands of repair requests or gamer activity logs.

This ensured the system could scale for **enterprise-level data volumes**.

5. Control Statements & Exception Handling

- Used **control statements (IF-ELSE, SWITCH, loops)** to build dynamic business logic.
 - Implemented **try-catch blocks** to handle unexpected errors gracefully.
 - Exception handling ensured system reliability — errors were logged in custom error objects or sent as admin notifications instead of breaking transactions.
 - Example: If a Repair record failed due to missing gamer info, an error log was generated instead of halting the entire batch.
-

6. Test Classes & Deployment Readiness

- Created **Apex Test Classes** to validate triggers, classes, and asynchronous logic.
- Achieved **>75% code coverage** (Salesforce minimum requirement for deployment).
- Tests covered:
 - Positive scenarios (valid subscription creation).
 - Negative scenarios (invalid dates, missing fields).
 - Bulk scenarios (inserting/updating hundreds of records).

- Benefits: Reliable deployments, easier debugging, long-term maintainability.
-

5.3 Integration with Previous Phases

- **From Phase 2 & 3 (Setup & Data Modeling):** Apex logic extended the objects and relationships by adding **dynamic automation** that complements Page Layouts and Relationships.
 - **From Phase 4 (Testing):** Test classes verified that the Apex logic works correctly across all modules and remains governor-limit compliant.
 - Apex ensured that the system's functionality aligned with **business requirements**, moving it beyond simple configurations into a **robust enterprise application**.
-

5.4 Conclusion

Phase 5 demonstrated the power of **Apex programming** in making the PlayStation Gaming CRM a scalable, intelligent, and production-ready solution. By combining classes, triggers, SOQL, and asynchronous processes, the system gained:

- **Efficiency** (bulkified operations, optimized queries).
- **Automation** (custom business logic beyond declarative tools).
- **Reliability** (exception handling, test coverage).
- **Scalability** (batch jobs, scheduled jobs for large data sets).

SCHEDULED JOBS

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

All Scheduled Jobs

[Help for this Page](#)

The All Scheduled Jobs page lists all of the jobs scheduled by your users. Multiple job types may display on this page. You can delete scheduled jobs if you have the permission to do so.

Percentage of Scheduled Jobs Used: 1%
 You have currently used 1 type scheduled Apex jobs out of an allowed organization limit of 100 active or scheduled jobs. To learn about how this limit is calculated and what contributes to it see the [Lightning Platform Apex Limits](#) topic.

View: [All Scheduled Jobs](#) [Create New View](#)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other [All](#)

Schedule Apex								
Action	Job Name	Submitted By	Submitted	Started	Next Scheduled Run	Type	Cron Trigger ID	
Del	Metalytics Data Loader Job for Org : 00DgK000007bzID	User, Integration	7/17/2025, 8:10 PM	9/24/2025, 10:30 PM	9/25/2025, 10:30 PM	Autonomous Data Loader Job	08egK000007dnmT	
	Program Milestone Computation Cron Job	Process, Automated	7/17/2025, 8:10 PM	9/24/2025, 4:59 PM	9/24/2025, 11:59 PM	Program Milestone Computation Cron Job	08egK000007dnmR	
	Program Status Update Cron Job	Process, Automated	7/17/2025, 8:10 PM	9/24/2025, 8:00 PM	9/25/2025, 5:00 AM	Program Status Update Cron Job	08egK000007dnmS	
Manage Del Pause Job	Repair_Batch_Scheduler_Daily	Kumar, Himanshu	9/24/2025, 11:40 PM		9/25/2025, 2:00 AM	Scheduled Apex	08egK00000C5W1y	

APEX JOBS

Click here to go to the new batch.jobs.page

Apex Jobs

[Help for this Page](#)

Monitor the status of all Apex jobs, and optionally, abort jobs that are in progress.

Percent of Asynchronous Apex Used: 0%
 You have currently used 0 asynchronous Apex operations out of an allowed 24-hour organization limit of 250,000. To learn about how this limit is calculated and what contributes to it, see the [Lightning Platform Apex Limits](#) topic.

View: [All](#) [Create New View](#)

Action	Submitted Date	Job Type	Status	Status Detail	Total Batches	Batches Processed	Failures	Submitted By	Completion Date	Apex Class	Apex Method	Apex Job ID
	9/24/2025, 11:40 PM	Scheduled Apex	Queued		0	0	0	Kumar, Himanshu		RepairBatchScheduler		707gK0000

APEX CLASSES

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 0.08%
 You are currently using 5,076 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage [i](#)

Compile all classes [i](#)

View: [All](#) [Create New View](#)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other [All](#)

Developer Console									New	Generate from WSDL	Run All Tests	Schedule Apex
Action	Name	Namespace Prefix	Api Version	Status	Size Without Comments	Last Modified By	Has Trace Flags					
Edit Del Security	ApexErrorLogger		64.0	Active	448	Himanshu Kumar	9/24/2025, 11:30 PM	<input type="checkbox"/>				
Edit Del Security	RepairBatchScheduler		64.0	Active	206	Himanshu Kumar	9/24/2025, 11:33 PM	<input type="checkbox"/>				
Edit Del Security	RepairCostBatch		64.0	Active	1,005	Himanshu Kumar	9/24/2025, 11:32 PM	<input type="checkbox"/>				
Edit Del Security	SubscriptionNotifierQueueable		64.0	Active	1,453	Himanshu Kumar	9/24/2025, 11:30 PM	<input type="checkbox"/>				
Edit Del Security	SubscriptionTriggerHandler		64.0	Active	1,410	Himanshu Kumar	9/24/2025, 11:30 PM	<input type="checkbox"/>				

orgfarm-ba15bf8906-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apextrigger

```

1  @isTest
2  private class SubscriptionTriggerHandlerTest {
3
4  @isTest static void testValidInsertAndQueueable() {
5      Contact c = new Contact(FirstName='T', LastName='User', Email='test+1@example.com');
6      insert c;
7
8      Test.startTest();
9      Subscriptions__c s = new Subscriptions__c(
10         Contact__c = c.Id,
11         Start_Date__c = Date.today(),
12         End_Date__c = Date.today().addDays(30),
13         Status__c = 'Active'
14     );
15     insert s;
16     Test.stopTest();
17
18     Subscriptions__c s2 = [SELECT Id, Contact__c FROM Subscriptions__c WHERE Id = :s.Id];
19     System.assertEquals(c.Id, s2.Contact__c);
20 }
21
22 @isTest static void testInvalidDatesProducesError() {
23     Contact c = new Contact(FirstName='Bad', LastName='Date', Email='bad@example.com');
24     insert c;
25
26     Subscriptions__c s = new Subscriptions__c(
27         Contact__c = c.Id
28     );
}

```

Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems

File • Edit • Debug • Test • Workspace • Help • < >

SubscriptionTrigger.apxc • SubscriptionTriggerHandler.apxc • SubscriptionNotifierQueueable.apxc • ApexErrorLogger.apxc • RepairCostBatch.apxc • SubscriptionTriggerHandlerTest.apxc

Code Coverage: None • API Version: 64

```

1 global class RepairBatchScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4         RepairCostBatch b = new RepairCostBatch();
5         Database.executeBatch(b, 200);
6     }
7

```

File • Edit • Debug • Test • Workspace • Help • < >

SubscriptionTrigger.apxc • SubscriptionTriggerHandler.apxc • SubscriptionNotifierQueueable.apxc • ApexErrorLogger.apxc • RepairCostBatch.apxc • Saving: RepairCostBatch.apxc

Code Coverage: None • API Version: 64

```

3     return Database.getQueryLocator(
4         'SELECT Id, Repair_Cost__c, Status__c FROM Repairs__c WHERE Repair_Cost__c != null'
5     );
6
7
8     public void execute(Database.BatchableContext bc, List<Repairs__c> scope) {
9         List<Repairs__c> updates = new List<Repairs__c>();
10        for (Repairs__c r : scope) {
11            if (r.Repair_Cost__c != null && r.Repair_Cost__c > 500) {
12                r.Status__c = 'Needs Approval';
13                updates.add(r);
14            }
15        }
16        if (!updates.isEmpty()) {
17            try {
18                update updates;
19            } catch (Exception ex) {
20                ApexErrorLogger.logException('RepairCostBatch.execute', ex);
21            }
22        }
23    }
24
25    public void finish(Database.BatchableContext bc) {
26        System.debug('RepairCostBatch finished');
27    }
28 }

```

Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems

Developer Console - Google Chrome

orgfarm-ba15bf8906-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage?action=selectExtent&extent=apexttrigger

File • Edit • Debug • Test • Workspace • Help • < >

SubscriptionTrigger.apcpx • SubscriptionTriggerHandler.apcpx • SubscriptionNotifierQueueable.apcpx • ApexErrorLogger.apcpx

Code Coverage: None • API Version: 64 • Go To

```

1 * public class ApexErrorLogger {
2 *     public static void logException(String context, Exception ex) {
3 *         try {
4 *             Apex_Error__c e = new Apex_Error__c();
5 *             e.Message__c = ex.getMessage();
6 *             e.Stack_Trace__c = ex.getStackTraceString();
7 *             e.Context__c = context;
8 *             insert e;
9 *         } catch (Exception loggingEx) {
10 *             System.debug('Failed to log exception: ' + loggingEx);
11 *         }
12 *     }
13 * }
14 |

```

Logs Tests Checkpoints Query Editor View State Progress Problems

Name Line Problem

File • Edit • Debug • Test • Workspace • Help • < >

SubscriptionTrigger.apcpx • SubscriptionTriggerHandler.apcpx • SubscriptionNotifierQueueable.apcpx

Code Coverage: None • API Version: 64 • Go

```

1 * public class SubscriptionNotifierQueueable implements Queueable {
2 *     private List<Id> subscriptionIds;
3 *
4 *     public SubscriptionNotifierQueueable(List<Id> ids){
5 *         this.subscriptionIds = ids;
6 *     }
7 *
8 *     public void execute(QueueableContext ctx) {
9 *         try {
10 *             List<Subscriptions__c> subs = [
11 *                 SELECT Id, Name, Contact__c, End_Date__c, Status__c
12 *                 FROM Subscriptions__c
13 *                 WHERE Id IN :subscriptionIds
14 *             ];
15 *
16 *             List<Messaging.SingleEmailMessage> mails = new List<Messaging.SingleEmailMessage>();
17 *             for (Subscriptions__c s : subs) {
18 *                 if (s.Contact__c != null) {
19 *                     Messaging.SingleEmailMessage m = new Messaging.SingleEmailMessage();
20 *                     m.setTargetObjectId(s.contact__c);
21 *                     m.setSaveAsActivity(false);
22 *                     m.setSubject('Subscription update: ' + (s.Name==null ? '' : s.Name));
23 *                     m.setPlainTextBody('Hello, your subscription status: ' +
24 *                         (s.Status__c==null ? '' : s.Status__c) +
25 *                         ', End date: ' + (s.End_Date__c==null ? '' : string.valueOf(s.End_Date__c)));
26 *                     mails.add(m);
27 *                 }
28 *                 if (!mails.isEmpty()) Messaging.sendEmail(mails);
29 *             } catch (Exception ex) {
30 *                 ApexErrorLogger.logException('SubscriptionNotifierQueueable.execute', ex);
31 *             }

```

```

1 public with sharing class SubscriptionTriggerHandler {
2     public static void beforeInsert(List<Subscriptions__c> newList) {
3         validateDates(newList);
4     }
5
6     public static void beforeUpdate(List<Subscriptions__c> newList, Map<Id, Subscriptions__c> oldMap) {
7         validateDates(newList);
8     }
9
10    private static void validateDates(List<Subscriptions__c> records){
11        for (Subscriptions__c s : records) {
12            if (s.Start_Date__c != null && s.End_Date__c != null) {
13                if (s.Start_Date__c > s.End_Date__c) {
14                    saddError('Start Date must be earlier than End Date.');
15                }
16            }
17        }
18    }
19
20    public static void afterInsert(List<Subscriptions__c> newList) {
21        List<Id> ids = new List<Id>();
22        for (Subscriptions__c s : newList) ids.add(s.Id);
23        if (!ids.isEmpty()) {

```



```

1 trigger SubscriptionTrigger on Subscriptions__c (
2     before insert, before update,
3     after insert, after update
4 ) {
5     if (Trigger.isBefore) {
6         if (Trigger.isInsert) SubscriptionTriggerHandler.beforeInsert(Trigger.new);
7         if (Trigger.isUpdate) SubscriptionTriggerHandler.beforeUpdate(Trigger.new, Trigger.oldMap);
8     }
9     if (Trigger.isAfter) {
10        if (Trigger.isInsert) SubscriptionTriggerHandler.afterInsert(Trigger.new);
11        if (Trigger.isUpdate) SubscriptionTriggerHandler.afterUpdate(Trigger.new, Trigger.oldMap);
12    }
13 }
14

```

Phase 6: User Interface Development

6.1 Overview

While earlier phases focused on building the **data model (Phase 3)**, **automation (Phase 4)**, and **Apex programming logic (Phase 5)**, this phase focused on the **user-facing interface** of the PlayStation

Gaming CRM.

The objective was to design record pages, layouts, and Lightning Web Components (LWCs) that present all business processes clearly to **Support Agents, Marketing Managers, and Admins**, ensuring smooth adoption and usability.

By combining **Salesforce Lightning App Builder** with **custom LWCs integrated with Apex**, the CRM now provides a **dynamic, interactive, and user-friendly interface**.

6.2 Activities Done

1. Lightning App Builder

- A custom **Lightning App** named *PlayStation CRM* was created.
- Navigation items were structured to include:
 - **Gamers (Contacts)** 
 - **Games** 
 - **Subscriptions** 
 - **Repairs** 
 - **Cases** 
 - **Campaigns** 

This app acts as the single workspace where all PlayStation CRM data and features can be accessed.

2. Record Pages

- **Gamers (Contacts)** → customized record page showing:
 - Related **Subscriptions** list view.
 - Related **Games** (via junction object Game Participation).
 - Related **Cases** to view support history.
- **Games** → record page highlighting:
 - Game details (Name, Genre, Release Date, Price).
 - Related Gamers who play the game.
- **Subscriptions** → record page showing:
 - Plan Type, Start/End Date, Status.
 - Approval process fields (Discount Requested, Approval Status).
 - Alerts and automation outcomes from earlier phases.

These record pages provide **360° visibility** into each entity and its connections.

3. Tabs & Navigation

- Created **custom tabs** for:
 - **Games_c**
 - **Subscriptions_c**
 - **Repairs_c**
- Tabs were added to the **PlayStation CRM app navigation bar**, making it simple for support staff to switch between modules.

This improved accessibility and minimized clicks for end users.

4. Home Page Layouts & Utility Bar

- **Home Page** customized for Support Agents:
 - Dashboard of **open Cases**.
 - List of **Repairs pending approval**.
 - Alerts for **upcoming Subscription expiries**.
- **Utility Bar** added to the app:
 - **Notes** → quick updates during calls.
 - **History** → access to recently viewed records.

This empowers agents to stay productive without leaving their workspace.



Search



Setup

Home

Object Manager ▾

App Launcher X playstr ×

Apps



PlayStation CRM

Items

No results

[View All](#)

Hyperforce Assistant

2

Analytics Studio

SETUP

lightning

orted by App N

pp Name ↑

I Tabs

Lightning App Builder | Games Record Page

Components Fields

Standard (42)

- Accordion
- Action Launcher
- Actions & Recommendations
- Activities
- Approval Trace
- Assessment List
- CRM Analytics Collection
- CRM Analytics Dashboard
- Dynamic Related List - Single
- Einstein Next Best Action
- Flow
- Flow Orchestration Work Guide
- Highlights Panel
- Invoice Preview
- Launchpad
- List View
- LWC CRM Analytics Dashboard

Get more on the AppExchange

Games Fifa 25

Related Details

Created By: Himaresh Kumar, 8/25/2025, 3:30 AM Last Modified By: Himaresh Kumar, 8/25/2025, 3:30 AM

Basic Info

Game Name: Fifa 25 Genre: Sports Release Date: 7/1/2025

Pricing

Price: \$70.00

Activity

Filters: All time • All activities • All types

Upcoming & Ongoing

No activities to show. Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

Page

Label: Games Record Page

API Name: Games_Record_Page

Page Type: Record Page

Object: Games

Template: Header and Right Sidebar

Description

Enable page-level dynamic actions for the Salesforce mobile app

PlayStation CRM Subscriptions Games Contacts Repairs Cases Campaigns Home

Copy of Open Cases by Status

We can't draw this chart because there is no data.

View Report As of Today at 9:38 AM

All

All 1 item, sorted by Repairs Name

REP-001

Copy of Subscriptions Expiring Soon

Record Count

End Date: 10/11/2025

View Report As of Today at 9:40 AM

PlayStation CRM Subscriptions Games Contacts Repairs Cases Campaigns Home

Repairs Console Repair

New Contact Edit New Opportunity

Approval Info

Cost Approved:

Approval Status:

Repair Request Info

Service Type: Console Repair

Status: In Progress

Request Date: 9/16/2025

Repair Cost: \$25.00

Console Info

History

The image displays three separate screenshots of the PlayStation CRM application, illustrating its user interface and various features:

- Subscriptions Page:** Shows a detailed view of a "Basic" plan. It includes fields for Plan Type (Basic), Status (Active), Start Date (9/23/2025), and End Date (10/11/2025). Below this, sections for Approval Info (Discount Requested: \$15.00, Approval Status: Pending Approval) and Gamer Info (Contact: Bruno Fernandez) are visible. A sidebar on the right contains activity and communication icons.
- Contacts Page:** Displays a contact record for "Mr. Bruno Fernandez". Key details include Title, Account Name, Phone (2), Email, and Contact Owner (Himanshu Kumar). The "Repairs" tab is currently selected, showing one repair entry named "REP-001".
- Home Page:** A general overview page with navigation links for Subscriptions, Games, Contacts, Repairs, Cases, Campaigns, and Home. It includes a search bar and a header with the PlayStation logo and various system status indicators.

Phase 7: Integration & External Access

Overview

With the **user interface development completed in Phase 6**, the focus of this phase was to extend the PlayStation CRM into a **connected ecosystem**. Instead of being an isolated system, Salesforce now interacts with **external applications and services**.

The integrations were designed around the **core business needs of the gaming company**:

- **Gamers (Contacts)** → view consistent subscription and repair status across platforms.
- **Subscriptions** → validated against external payment systems.
- **Repairs** → updates shared with third-party repair centers.
- **Games** → catalog synchronized from external sources.
- **Cases** → monitored through Salesforce but can trigger notifications outside Salesforce.

This ensures that PlayStation CRM not only manages data internally but also participates in the **broader service ecosystem**.

Activities Done

1. Named Credentials & Remote Site Settings

To establish secure communication with external systems, **Named Credentials** and **Remote Site Settings** were configured:

- A **Named Credential** was created for the payment gateway. This allows Salesforce to securely authenticate and make API calls to verify subscription renewals or process payments.
- A **Remote Site Setting** was added to whitelist the gateway's domain, ensuring callouts are permitted only to trusted endpoints.

This step ensures **subscription data in CRM is validated against actual payment information**, avoiding discrepancies.

2. REST API Callouts

Using the Apex programming concepts from Phase 5, **REST API callouts** were implemented to enable direct communication with external applications:

- **Subscription Renewal Callouts:** Apex methods fetch renewal details from an external billing service, ensuring that Salesforce always reflects the latest subscription status (active, expired, or renewed).
- **Repair Status Push:** Whenever a repair record is updated in Salesforce, an API call is made to a third-party repair tracking app, keeping external agents and service centers informed in real-time.

This provides **two-way communication**: Salesforce can both **retrieve** and **send** information to external apps.

3. External Services & Salesforce Connect

Two integration tools were explored to expand data availability:

- **External Services:** A PlayStation Store API schema was defined in Salesforce, allowing actions like "*fetch game details*" or "*check store availability*" to be invoked directly from Salesforce Flows without custom code.

- **Salesforce Connect:** Configured to link to an external game catalog via OData. This allows users to **see the game library directly in Salesforce** without duplicating or storing that data internally.

This ensures the **Games object in CRM stays synchronized** with the official external catalog.

4. Platform Events & Change Data Capture (CDC)

To support real-time communication with external systems:

- **Platform Events:** A custom event was created to notify external systems whenever a **Repair record** is created or updated. This allows third-party service apps to immediately pick up repair requests without polling Salesforce.
- **Change Data Capture (CDC):** Enabled for the **Subscriptions object**, ensuring external analytics tools receive real-time updates whenever a subscription is created, renewed, or expires.

Together, these enable **event-driven integrations**, making Salesforce proactive in sharing changes.

5. API & Authentication

Finally, secure external access was set up:

- A **Connected App** was created with OAuth 2.0 authentication, enabling trusted third-party apps to access Salesforce data.
- Access was restricted using **scopes** to ensure only necessary data (like Subscription and Repair details) is exposed.
- API limits and governor constraints were reviewed to ensure integrations are sustainable under Salesforce usage caps.

This ensures that **external apps can fetch gamer data securely**, without compromising the system's performance or integrity.

orgfarm-ba15bf8906-dev-ed.lightning.force.com/lightning/setup/NamedCredential/page?address=%2F0XAgK000000969...

Search Setup

SETUP Named Credentials

Named Credential: PaymentGateway

Specify the callout endpoint's URL and the authentication settings that are required for Salesforce to make callouts to the remote system.

Back to Named Credentials

Label: PaymentGateway
Name: PaymentGateway
URL: https://api.paymentgateway.com

Authentication

Certificate
Identity Type: Named Principal
Authentication Protocol: Password Authentication
Username: user

Callout Options

Generate Authorization Header: checked
Allow Merge Fields in HTTP Headers: unchecked
Allow Merge Fields in HTTP Body: unchecked
Outbound Network Connection: unchecked

orgfarm-ba15bf8906-dev-ed.lightning.force.com/lightning/setup/SecurityRemoteProxy/page?address=%2F0rpgK000001ZBxh...

Search Setup

SETUP Remote Site Settings

Remote Site Details

Help for this Page

Custom Code

Remote Access

Security

Remote Site Settings

Didn't find what you're looking for?
Try using Global Search.

Remote Site Detail

Remote Site Name	PaymentGateway	Modified By	Himanshu Kumar, 9/25/2025, 10:31 AM
Remote Site URL	https://api.paymentgateway.com		
Disable Protocol Security	<input type="checkbox"/>		
Description			
Active	<input checked="" type="checkbox"/>		
Created By	Himanshu Kumar, 9/25/2025, 10:31 AM		

Developer Console - Google Chrome

SubscriptionIntegration.apxc

```

1  public with sharing class SubscriptionIntegration {
2
3      // Asynchronous callout (future method allows callouts after DML)
4      @future(callout=true)
5      public static void fetchRenewal(String subscriptionId) {
6          try {
7              // Build HTTP request
8              HttpRequest req = new HttpRequest();
9              req.setEndpoint('callout:PaymentGateway/renewals/' + subscriptionId);
10             req.setMethod('GET');
11
12             // Send the request
13             Http http = new Http();
14             HttpResponse res = http.send(req);
15
16             // Process response if successful
17             if (res.getStatusCode() == 200) {
18                 // Parse JSON (example format: {"status":"Active","endDate":"2025-12-31"})
19                 Map<String, Object> result = (Map<String, Object>) JSON.deserializeUntyped(res.getBody());
20
21                 String newStatus = (String) result.get('status');
22                 String newEndDate = (String) result.get('endDate');
23
24                 // Update Subscription record in Salesforce
25                 Subscriptions__c sub = [SELECT Id FROM Subscriptions__c WHERE Id = :subscriptionId LIMIT 1];
26                 sub.Status__c = newStatus;
27                 sub.End_Date__c = Date.valueOf(newEndDate);
28                 update sub;
29             } else {
30                 System.debug('Error: ' + res.getStatusCode() + ' ' + res.getBody());
31             }
32         } catch (Exception e) {
33             System.debug('Callout failed: ' + e.getMessage());
34         }
35     }
}

```

Search Setup

Setup Home Object Manager

PlayStationStoreService

External Services

Service name: PlayStationStoreService
Creation source: From API specification
Description: External service for fetching PlayStation Store game details and availability.
Type: OpenApi3

Named credentials: PaymentGateway
Created by: Himanshu Kumar
Created date: September 26, 2025 at 12:05 AM
Last modified by: Himanshu Kumar
Last modified date: September 26, 2025 at 12:08 AM

Operations 1 Items · Sorted by Operation Name

Operation Name ↑	Description	Input parameters	Output parameters
getGamesById	Returns details for a specific game.	id	responseCode, default, 200

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-ba15bf8906-dev-ed.lightning.force.com/lightning/setup/ExternalDataSource/page?address=%2FOXCgK0000000JdI>. The page title is "External Data Sources". The left sidebar shows "Integrations" and "External Data Sources" under "Setup". The main content area displays a table for the "External Games Catalog" data source, with columns for Name (ExternalGamesCatalog), Type (Salesforce Connect: OData 4.0), and various parameters like URL, Connection Timeout, and Request Row Counts.

The screenshot shows the Salesforce Setup interface with the URL <https://orgfarm-ba15bf8906-dev-ed.lightning.force.com/lightning/setup/EventObjects/page?address=%2F01gK000002TWWP%3Fsetupid%...>. The page title is "Platform Events". The left sidebar shows "MuleSoft", "Einstein", "Custom Code", "Integrations", "Security", and "Platform Events" under "Setup". The main content area displays a table for the "Platform Events" object, showing fields like Created By, Created Date, Event UUID, and Replay ID. It also includes sections for "Custom Fields & Relationships", "Triggers", and "Subscriptions".

This screenshot is identical to the one above, showing the "Platform Events" page in the Salesforce Setup interface. The URL is the same: <https://orgfarm-ba15bf8906-dev-ed.lightning.force.com/lightning/setup/EventObjects/page?address=%2F01gK000002TWWP%3Fsetupid%...>. The page title is "Platform Events". The left sidebar shows "MuleSoft", "Einstein", "Custom Code", "Integrations", "Security", and "Platform Events" under "Setup". The main content area displays a table for the "Platform Events" object, showing fields like Created By, Created Date, Event UUID, and Replay ID. It also includes sections for "Custom Fields & Relationships", "Triggers", and "Subscriptions".

The screenshots illustrate the configuration of Change Data Capture and the management of external client applications in the Salesforce Setup.

Change Data Capture

- Available Entities:**
 - Account (Account)
 - Account Clean Info (AccountCleanInfo)
 - Account Contact Role (AccountContactRole)
 - Agent Work (AgentWork)
 - Apex Error (Apex_Error__c)
- Selected Entities:** Subscriptions (Subscriptions__c)

Manage External Client Apps

PlayStationIntegrationApp

- Contact Email: himanshu9981@gmail.com
- App Authorization: All users can self-authorize
- Type: Local
- App Status: Enabled

Policies (selected tab) | Settings | Package Defaults

Configure policies to customize the external client app and plugins for this Salesforce organization.

App Policies

- Start Page: None

Phase 8: Data Management & Deployment

Overview

After completing **integration and external access in Phase 7**, the next step was to ensure the **PlayStation CRM data is reliable, consistent, and secure**, while also preparing for **safe deployments** of all the customizations and logic built in earlier phases.

This phase had two major objectives:

1. **Data Management** – make sure gamer data, subscription information, repair records, and related objects are **accurate, deduplicated, backed up, and portable**.
2. **Deployment**

By completing this phase, the PlayStation CRM became not just functional and integrated but also **enterprise-ready** — capable of scaling with reliable data and controlled deployments.

Activities Done

1. Data Import & Export

Managing gamer-related records at scale required bulk operations and backup strategies:

- **Data Import Wizard** was used for simple imports:
 - Imported new **Gamers (Contacts)** along with their subscription data.
 - Ensured relationships (e.g., which subscription belongs to which gamer) were preserved using lookup fields.
- **Data Loader** was configured for larger datasets, such as bulk uploading **Repair_c** records and exporting large subscription history logs.
 - This tool was particularly useful for mass updates when subscription statuses changed during testing.
- **Weekly Data Export Service** was scheduled as a **disaster recovery strategy**, generating compressed export files of all CRM data (Contacts, Games, Subscriptions, Repairs, Cases).

Result: All gamer and subscription data is backed up and can be restored or migrated if needed.

2. Data Quality & Duplicate Rules

Clean data was critical for CRM usability:

- **Duplicate Rules** were created to prevent multiple records of the same gamer being entered (e.g., matching on Email + Phone). This ensured each gamer had a single unified profile.
- **Validation Rules** were fine-tuned to enforce data integrity:
 - Subscriptions must always have a **Start Date earlier than End Date**.
 - Repair records cannot be marked as **Closed** without entering **Repair Cost**.

Result: Subscriptions, repairs, and gamer records remain consistent and trustworthy.

3. Deployment Using Change Sets

Since this project was built in a Developer Edition org without Sandbox support, all deployments were configured directly. In a production scenario, Outbound and Inbound Change Sets would be used.

The screenshot displays two overlapping Salesforce interfaces. The top interface is the 'Edit Field Mapping: Subscriptions' page, which shows a table of mappings between CSV headers and Salesforce fields. The bottom interface is the 'Load Inserts' step of the Salesforce Data Loader, where a user is selecting the 'Repairs' object to import.

Edit Field Mapping: Subscriptions

Your file has been auto-mapped to existing Salesforce fields, but you can edit the mappings if you wish. Unmapped fields will not be imported.

Edit	Mapped Salesforce Object	CSV Header	Example	Example	Example
Change	Start Date	Start_Date__c	2023-01-01	2023-03-15	2023-06-10
Change	End Date	End_Date__c	2023-12-31	2024-03-14	2024-06-09
Change	Status	Status__c	Active	Expired	Active
Change	Plan Type	Plan_Type__c	Premium	Basic	Premium
Change	Record Owner	ContactId	003XXXXXXXXXXXXX	003XXXXXXXXXXXXX	003XXXXXXXXXXXXX

Salesforce Data Loader 64.1.0 - orgfarm-ba15bf8906-dev-ed.develop.my.salesforce.com

File Settings View Help Cancel Previous Next

Step 2: Select Salesforce object

Import batch size: 200 Start at row: 0 Current API usage for the org: 278

Select Salesforce object to import:

Insert Update

Repairs (Repairs__c)

Show all Salesforce objects

Import from (CSV file): Browse...

< Back Next > Finish Cancel

SETUP Data Export

Monthly Export Service

[Help for this Page](#)

Data Export lets you prepare a copy of all your data in salesforce.com. From this page you can start the export process manually or schedule it to run automatically. When an export is ready for download you will receive an email containing a link that allows you to download the file(s). The export files are also available on this page for 48 hours, after which time they are deleted.

Next scheduled export:
None

[Export Now](#) [Schedule Export](#)

Scheduled By	Himanshu Kumar
Schedule Date	9/25/2025
Export File Encoding	ISO-8859-1 (General US & Western European, ISO-LATIN-1)

Action	File Name	File Size
download	WE_00DgK000007bzIDUAY_1.ZIP	8.9K

SETUP Duplicate Rules

Contact Duplicate Rule
Prevent_Duplicate_Gamer

[Help for this Page](#)

Duplicate Rule Detail		Edit Delete Clone Deactivate
Rule Name	Prevent_Duplicate_Gamer	Order 3 of 3 [Reorder]
Description		
Object	Contact	
Record-Level Security	Enforce sharing rules	
Action On Create	Allow	Operations On Create <input checked="" type="checkbox"/> Alert <input checked="" type="checkbox"/> Report
Action On Edit	Allow	Operations On Edit <input checked="" type="checkbox"/> Alert <input checked="" type="checkbox"/> Report
Alert Text	Use one of these records?	
Active	<input checked="" type="checkbox"/>	
Matching Rule	<input checked="" type="checkbox"/> Standard Contact Matching Rule <input checked="" type="checkbox"/> Mapped	Matching Criteria Matching rule for contact records. More info
Conditions		
Created By	Himanshu Kumar, 9/25/2025, 2:58 PM	Modified By Himanshu Kumar, 9/25/2025, 2:58 PM
Edit Delete Clone Deactivate		

Subscriptions Validation Rule

[Back to Subscriptions](#)

[Help for this Page](#)

Validation Rule Detail		Edit Clone
Rule Name	Validate_Dates	Active <input checked="" type="checkbox"/>
Error Condition Formula	End_Date__c < Start_Date__c	
Error Message	End Date must be later than Start Date.	Error Location Top of Page
Description		
Created By	Himanshu Kumar, 9/25/2025, 3:00 PM	Modified By Himanshu Kumar, 9/25/2025, 3:00 PM
Edit Clone		

Phase 9: Reporting, Dashboards & Security Review

Overview

After completing **data management and deployment (Phase 8)**, the focus of Phase 9 was on **turning CRM data into insights** and ensuring **gamer information is protected through security controls**.

This phase had two main goals:

1. Build **reports and dashboards** that allow support agents, marketing teams, and managers to analyze data such as gamer activity, subscription performance, and support cases.
2. Conduct a **security review** to ensure sensitive PlayStation gamer data remains private, while still allowing teams to collaborate effectively.

By the end of this phase, PlayStation CRM became a **decision-making tool**, not just a data repository, while also meeting **security and compliance standards**.

Activities Done

1. Reports

Multiple report types were created to provide insights from different angles:

- **Tabular Report:**
A simple list showing all **Contacts (Gamers)** who currently have an active subscription. This helps the support team quickly find who is eligible for services.
- **Summary Report :**
Grouped Subscriptions by **Plan_Type_c** (e.g., Basic, Premium). This gave management a clear view of which subscription tiers are most popular.
- **Matrix Report:**
Rows = **Games**, Columns = **Regions** → values = total subscriptions or cases. This report highlighted which games were trending in specific regions, guiding marketing campaigns.

Together, these reports connected the CRM's **data model (Phase 3)** with real-world analysis.

2. Dashboards

Dashboards were built on top of these reports for visual insights:

- **Subscription Dashboard:**
 - Chart: Subscription Counts by Plan.
 - Chart Game Popularity by Genre.
 - Chart: Subscriptions Expiring Soon.

- **Support Dashboard:**
 - Chart: Open Cases by Status.
- **Dynamic Dashboards:**

Configured so that managers automatically see **only their team's data**, without having to create separate dashboards for each user.

These dashboards provided **at-a-glance visibility**, ensuring leadership can monitor business health in real time.

3. Security Review

To protect sensitive gamer data, several security layers were reviewed and updated:

- **Sharing Settings:**
 - **Contacts (Gamers)** and **Subscriptions** set to **Private** → only record owners and admins can view or edit.
 - **Campaigns** set to **Public Read-Only** → marketing teams can collaborate without editing rights.
- **Field-Level Security:**
 - Sensitive fields such as payment references and discount approvals are only visible to **System Administrators**.
 - Support agents see only the fields they need to assist gamers.
- **Login Security:**
 - Defined **Login IP Ranges** for admin users, ensuring only trusted networks are used for system administration.
 - Adjusted **Session Settings** (timeouts, logout policies) for additional protection.
- **Audit Trail:**
 - Setup Audit Trail enabled to track configuration and user changes.
 - This provides transparency and compliance evidence in case of disputes.

The security review ensured the CRM complies with **data protection principles** while still being usable by day-to-day teams.

PlayStation CRM Reports Recent

Report Name	Description	Folder	Created By	Created On	Subscribed
Active Gamers with Subscriptions	Private Reports	Himanshu Kumar	9/25/2025, 3:54 PM		
Game Popularity by Genre	Private Reports	Himanshu Kumar	9/25/2025, 4:01 PM		
Subscription Counts by Plan	Private Reports	Himanshu Kumar	9/25/2025, 3:57 PM		

Subscription Dashboard

As of Sep 25, 2025, 4:15 PM · Viewing as Himanshu Kumar

Subscription Counts by Plan

Record Count: 5

Plan Type: Basic (selected)

[View Report \(Subscription Counts...\)](#) As of Sep 25, 2025, 4:15 PM

Game Popularity by Genre

Value: 60

[View Report \(Game Popularity by ...\)](#) As of Sep 25, 2025, 4:15 PM

Subscriptions Expiring Soon

Record Count: 1

Status: Active (selected)

[View Report \(Subscriptions Expi...\)](#) As of Sep 25, 2025, 4:17 PM

Support Dashboard

As of Sep 25, 2025, 4:16 PM · Viewing as Himanshu Kumar

Open Cases by Status

Record Count: 1

Status: Working (selected)

[View Report \(Open Cases by Stat...\)](#) As of Sep 25, 2025, 4:16 PM

Profile System Administrator

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Action	IP Start Address	IP End Address	Description
Edit Del	122.175.200.159	122.175.200.255	User License: Salesforce Created By: salesforce.com, inc., 7/17/2025, 8:08 PM Modified By: Himanshu Kumar, 9/25/2025, 7:34 AM

Page Layouts

Standard Object Layouts

Global	Global Layout [View Assignment]	Location Group	Location Group Layout [View Assignment]
Email Application	Not Assigned [View Assignment]	Location Group Assignment	Location Group Assignment Layout [View Assignment]

View Setup Audit Trail

The last 20 entries for your organization are listed below. You can download your organization's setup audit trail for the last six months (Excel .csv file).

Date	User	Source Namespace Prefix	Action	Section	Delegate User
9/25/2025, 4:29:46 PM PDT	himanshu.kumar.aim22673@agentforce.com		Changed Session Timeout Value from 120 to 30 minutes	Session Settings	
9/25/2025, 4:29:46 PM PDT	himanshu.kumar.aim22673@agentforce.com		Session Security Level for Multi-Factor Authentication was set to High Assurance	Session Settings	
9/25/2025, 4:29:46 PM PDT	himanshu.kumar.aim22673@agentforce.com		Session Security Level for Passwordless Login was set to Standard	Session Settings	
9/25/2025, 4:29:46 PM PDT	himanshu.kumar.aim22673@agentforce.com		Session Security Level for Lightning Login was set to Standard	Session Settings	
9/25/2025, 4:29:46 PM PDT	himanshu.kumar.aim22673@agentforce.com		Session Security Level for Activation was set to Standard	Session Settings	
9/25/2025, 4:29:46 PM PDT	himanshu.kumar.aim22673@agentforce.com		Session Security Level for Delegated Authentication was set to Standard	Session Settings	

Phase 10: Final Presentation & Demo Day

Overview

This phase focused on **presenting the PlayStation Gaming CRM project** to stakeholders in a clear and structured way. The goal was to showcase the problem addressed, the solution developed, its benefits, and possible future enhancements.

1. Pitch Presentation

1. Problem Statement

- Gamers face difficulties in **managing subscriptions, repairs, and support requests**.
 - PlayStation lacks a **centralized CRM system** to track gamers, games, and service requests.
 - Support agents and marketing teams struggle to get **real-time insights** into gamer activities.
-

2. Solution

- Built a **Salesforce-based PlayStation Gaming CRM** to manage:
 - **Gamers (Contacts)** → Store gamer details.
 - **Games & Subscriptions** → Track purchases and active plans.
 - **Cases & Repairs** → Handle gamer issues and service requests.
 - **Campaigns & Marketing** → Run targeted promotions.
 - Added **automation (workflows, approvals, flows)** to reduce manual effort.
 - Created **reports & dashboards** for actionable insights.
 - Ensured **data security** with role hierarchy, field-level security, and audit trail.
-

3. Benefits

- **For Gamers:**
 - Easy tracking of subscriptions and repairs.
 - Faster support through automation.
 - **For Support Team:**
 - Automatic case assignment & updates.
 - Centralized data access for solving gamer issues.
 - **For Marketing Team:**
 - Better campaign targeting using reports.
 - Improved engagement insights with dashboards.
 - **For Management:**
 - Secure data visibility with role-based access.
 - Real-time performance monitoring.
-

4. Future Enhancements

- **Gamification Features:** Reward points and loyalty tracking for gamers.
- **AI-Powered Recommendations:** Suggest games or subscription upgrades.
- **Chatbot Integration:** Automated gamer support through Salesforce Einstein Bot.
- **Mobile App Access:** Extend CRM features to gamers and staff via mobile.
- **Third-Party Payment Integration:** Enable direct billing and payment tracking.

2. Demo Walkthrough

The demo was structured around a **realistic support and management scenario**:

1. App Launcher & Navigation

- From the App Launcher, opened **PlayStation CRM**.
- Navigation bar displayed all required objects: Contacts, Subscriptions, Games, Repairs, Cases, and Campaigns.

2. Gamer Record Page

- Opened a Contact record (e.g., *Alex Johnson*).
- Viewed gamer details in the Highlights Panel.
- Tabs displayed related Subscriptions, Repairs, and Cases.

3. Subscriptions & Repairs

- Opened a Subscription record to review fields (Plan Type, Status, Dates).
- Demonstrated validation rule: End Date must be later than Start Date.
- Opened a Repair record → attempted to close without cost (error triggered).
- Entered cost → record successfully closed.

4. Reports & Dashboards

- Showed **Tabular Report**: Active Gamers with Subscriptions.
- Showed **Summary Report**: Subscriptions grouped by Plan Type.
- Showed **Matrix Report**: Game popularity by Region.
- Showed **Joined Report**: Gamers, Subscriptions, and Cases combined.
- Opened **Subscription Dashboard** (active vs expired subscriptions + new subs in last 30 days).
- Opened **Support Dashboard** (open vs closed cases + average resolution time).

5. Security Review

- Demonstrated **Sharing Settings**: Contacts and Subscriptions set to Private, Campaigns set to Public Read-Only.
- Showed **Field-Level Security**: Sensitive fields hidden from Standard Users.
- Highlighted **Login IP Ranges** and **Audit Trail** as additional protections.

Result: The walkthrough demonstrated how agents can serve gamers, how managers can track business metrics, and how security ensures compliance.