SALESFORCE DEVELOPER VIRTUAL INTERNSHIP

AN INTERNSHIP REPORT

Submitted by

JANARTHANAN N

in partial fulfillment of the requirement for
the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

K.S. RANGASAMY COLLEGE OF TECHNOLOGY
(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

TIRUCHENGODE – 637 215

APRIL 2024

K.S. RANGASAMY COLLEGE OF TECHNOLOGY
TIRUCHENGODE - 637 215

BONAFIDE CERTIFICATE

Certified that this internship report titled **“Salesforce Developer Virtual Internship”** is the bonafide work of **JANARTHANAN N (73772113146)** who carried out the internship under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other internship report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. C. Rajasekaran, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor & Head
Department of ECE
K.S. Rangasamy College of Technology
Tiruchengode - 637 215

SIGNATURE
Mr. S. Saravanan, M.E.,
INTERNSHIP COORDINATOR
Assistant Professor
Department of ECE
K.S.Rangasamy College of Technology
Tiruchengode - 637 215

Submitted for an internship evaluation held on ………………

Internal Examiner 1 **Internal Examiner 2**

DECLARATION

I declare that the internship report on **“SALESFORCE DEVELOPER VIRTUAL INTERNSHIP”** is the result of original work done by me and best of my knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of Bachelor of Electronics and Communication Engineering. This internship report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor of Engineering.

Signature

____

JANARTHANAN N

Place: Tiruchengode

Date:

ACKNOWLEDGEMENT

I wish to express our sincere gratitude to our honourable Chairman**,** **Shri. R. SRINIVASAN, B.B.M., MISTE.,** for providing immense facilities at our institution.

I would like to express special thanks of gratitude to our Chief Executive Officer **Dr. AKILA MUTHURAMALINGAM, M.E., Ph.D.,** who has been the key spring of motivation to me throughout the completion of our course and internship work.

I am very proudly rendering our thanks to our Principal

Dr. R. GOPALAKRISHNAN, M.E., Ph.D., for the facilities and the encouragement given by him to the progress and completion of our internship.

I proudly render our immense gratitude to the Head of the Department **Dr.C.RAJASEKARAN, B.E., M.E., Ph.D.,** for his effective leadership, encouragement and guidance in the internship.

I am highly indebted to provide our heartfelt thanks to our internship coordinator **Mr. S. SARAVANAN, M.E.,** Assistant Professor, Department of Electronics and Communication Engineering for his valuable ideas, encouragement and supportive guidance throughout the successful conduct of the internship.

I wish to extend my sincere thanks to all members of faculty of our Electronics and Communication Engineering Department for their valuable suggestions, kind cooperation and constant encouragement for the successful completion of this internship.

Finally, I wish to acknowledge the help received from various Departments and various individuals during the preparation and editing stages of the internship report.

ABSTRACT

Salesforce developers have become essential to fostering organisational growth and maintaining operational excellence in today's fast-paced business environment. This detailed abstract seeks to give readers a thorough grasp of the many roles, abilities, and contributions that Salesforce developers play in enabling companies to prosper in the digital age. The architects responsible for customising, implementing, and optimising Salesforce solutions to meet the distinct requirements and goals of various industries are Salesforce developers. Their multidisciplinary experience in software development, data management, UX design, and integration allows them to build dependable, expandable, and user-friendly apps for the Salesforce ecosystem. The primary duties of Salesforce developers, including data model design, custom functionality development, workflow configuration, and integration with external systems to improve productivity and streamline business processes. During my internship, I had the privilege of working within the Platform Development Tools (PDT) team, a division focused on enhancing the development experience within the Salesforce ecosystem. Specifically, my project revolved around modernizing the existing apex commands by implementing a new oclif plugin framework. This initiative aimed to provide developers with a more intuitive and efficient toolset for building on top of Salesforce products and customizing their experiences.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO.
	ABSTRACT	**v**
	LIST OF TABLES	**viii**
	LIST OF FIGURES	**ix**
	LIST OF SYMBOLS AND ABBREVIATIONS	**x**
1	**INTRODUCTION**	**1**
	1.1 INTRODUCTION TO SALESFORCE	1
	1.1.1 What is salesforce	2
	1.1.2 History of Salesforce	2
	1.1.3 What Does Salesforce Do?	2
	1.2 CMR	3
	1.3 SALESFORCE PRODUCTS AND SERVICES	4
	1.3.1 Salesforce Clouds Services	4
	1.3.2 Salesforce Products	6
	1.4 ROLE OF A SALESFORCE DEVELOPER	6
2	**SALESFORCE ARCHITECTURE**	7
	2.1 INTRODUCTION	7
	2.2 SALESFORCE ARCHITECTURE	7
	2.2.1 Multi-Tenant Architecture	8
	2.2.2 Components (Objects, Fields, Records)	9
	2.3 DATA MODELING IN SALESFORCE	10
	2.3.1 Custom Objects	10
	2.4 LAYERS OF SALESFORCE	11
3	**SALESFORCE ORG**	**12**
	3.1 INTRODUCTION	12
	3.2 NAVIGATING THE SALESFORCE USER INTERFACE	12
	3.3 SALESFORCE BENDING	13
	3.4 UI CUSTOMIZATION	13
	3.5 IMPLEMENTATION	14

4 GETTING STARTED WITH APEX 15

4.1 INTRODUCTION 15

4.1.1 What is Apex 16

4.1.2 Understanding Apex Core Concepts 16

4.1.3 Syntax and Triggers 17

4.2 APEX TRIGGERS 19

4.2.1 Bulk Apex Triggers 19

4.2.2 Syntax and Triggers 19

4.3 APEX UNIT TESTS 21

4.3.1 Asynchronous Apex 22

4.3.2 Future Apex 22

4.3.3 Batch Apex 22

5 SUPER BADGES 24

5.1 INTRODUCTION 24

5.2 PROCESS AUTOMATION SPECIALIST 24

5.2.1 What You'll Be Doing to Earn This Badge 25

5.3 DATA INTEGRITY ENFORCEMENT 26

5.4 SUMMARY 27

6 CONCLUSION 28

REFERENCES 29

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Introduction to Salesforce Edition	1
1.2	Customer Relationship Management Circle	3
1.3	Types of Cloud Services	5
2.1	Architecture of salesforce developer	7
2.2	Multi – Tenant Architecture	8
2.3	Data Modelling	10
4.1	Apex Custom Page	15
4.2	Get Started with Apex	16
4.3	Version Setting User Interface	17
4.4	Batch Apex	23
5.1	Process Automation Specialist	25
5.2	Data Integrity Enforcement	27

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
API	-	Application Programming Interface
CRM	-	Customer Relationship Management
HTML	-	Hyper Text Markup Language
LWC	-	Lighting Web Components
SLDS	-	Salesforce Lighting Design Systems
UI	-	User Interface

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SALESFORCE

Salesforce, established in 1999, has emerged as a trailblazer in the realm of customer relationship management (CRM) through its revolutionary cloud-based platform. Rooted in the vision of providing businesses with a dynamic and integrated solution, Salesforce has become synonymous with innovation and agility in the digital era. The platform's multifaceted nature encompasses a suite of products and services, including Sales Cloud, Service Cloud, Marketing Cloud, and more, each tailored to address specific business needs. Salesforce operates on a robust multi-tenant architecture, fostering collaboration and scalability for organizations of all sizes is shown in the Figure 1.1. At its core, Salesforce is designed to provide a unified and holistic view of customer interactions, enabling companies to deliver personalized and seamless experiences. The platform's versatility is reflected in its various clouds, including Sales Cloud for sales automation, Service Cloud for customer service, Marketing Cloud for marketing automation, and more.

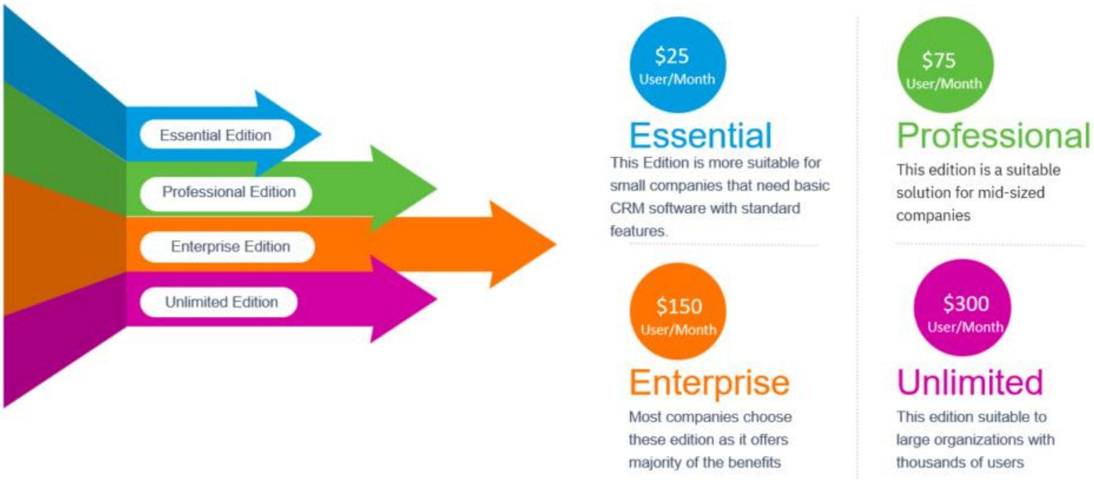

Essential Edition
Professional Edition
Enterprise Edition
Unlimited Edition
$25 User/Month
Essential
This Edition is more suitable for small companies that need basic CRM software with standard features.
$75 User/Month
Professional
This edition is a suitable solution for mid-sized companies
$150 User/Month
Enterprise
Most companies choose these edition as it offers majority of the benefits
$300 User/Month
Unlimited
This edition suitable to large organizations with thousands of users

Figure 1.1 Introduction to Salesforce Edition

Salesforce's impact extends beyond CRM, with its robust platform serving as a foundation for application development and integration. Salesforce developers leverage powerful tools like Apex, Visualforce, and Lightning Web Components to create tailored solutions that align with unique business requirements. The platform's commitment to innovation is exemplified by regular updates, introducing cutting-edge features and ensuring users stay at the forefront of technology.

1.1.1 What is Salesforce

Salesforce is a leading cloud-based Customer Relationship Management (CRM) platform. It enables businesses to manage and analyse customer interactions, streamline processes, and drive innovation. With a suite of interconnected products, including Sales Cloud and Service Cloud, Salesforce provides a unified view for personalized customer experiences. Its multi-tenant architecture ensures scalability, and developers use tools like Apex and Lightning Web Components to create tailored solutions. Salesforce is a key player in digital transformation, empowering organizations to adapt and thrive in the evolving landscape of customer engagement and business operations.

1.1.2 History of Salesforce

The company was founded in 1999 by former Oracle executive Marc Benioff, together with Parker Harris, Dave Moellenhoff, and Frank Dominguez as a software as a service (SaaS) company. Initial funding for the company came from Larry Ellison and Halsey Minor. In 2009, Salesforce passed $1 billion in annual revenue. Also in 2009, the company launched Service Cloud, an application that helps companies manage service conversations about their products and services. In 2014, the company released Trailhead, a free online learning platform. In October 2014, Salesforce announced the development of its Customer Success Platform. In September 2016, Salesforce announced the launch of Einstein, an artificial intelligence platform that supports several of Salesforce's cloud services. In February 2021, Amy Weaver, previously the chief legal officer, became CFO. Former CFO Mark Hawkins announced that he would be retiring in October. In November 2021, Bret Taylor was named vice chair and co-CEO of the company. In April 2022, Salesforce acquired Phennecs, a sandbox privacy compliance startup. Also in April 2022, Salesforce announced a five-year partnership with Formula one. In May 2022, Salesforce announced the acquisition of Slack-bot maker Troops.ai.

1.1.3 What Does Salesforce Do?

Salesforce is a cloud-based Customer Relationship Management (CRM) platform that enables businesses to manage customer data, sales operations, and marketing campaigns. It provides various features, and tools businesses can use to improve customer relationships, including customer segmentation, customer data management, and customer support. Salesforce also offers a variety of integrations with third-party applications and services, making it a powerful and versatile CRM platform.

1.2 CMR (Customer Relationship Management)

Salesforce helps businesses keep track of customer interactions and sales data. It can manage leads, contacts, opportunities, and cases. Salesforce also offers several features to help businesses automate their sales and marketing processes, such as email marketing, lead capture, and lead scoring. CRM systems like Salesforce can help businesses improve their customer service and sales operations.

SALES
SERVICE
LEAD
ORDERS
CRM
SUPPORT
ANALYSIS
STRATEGY
MARKETING
COMMUNICATION

Figure 1.2 Customer Relationship Management (CRM) Circle

Salesforce's CRM (Customer Relationship Management) is a comprehensive platform that excels in transforming sales processes and enhancing customer interactions. With robust sales management tools, it empowers teams to track leads, opportunities, and sales performance seamlessly. The platform facilitates efficient lead management, allowing businesses to capture, nurture, and convert leads into valuable customers. Customer interaction is at the forefront, with Salesforce providing a unified view of customer data for personalized engagement. Task management tools streamline workflows, ensuring that sales representatives stay organized and focused on key activities. Performance metrics, including sales forecasts and pipeline analytics, enable data-driven decision-making, enhancing overall business efficiency.

1.3 SALESFORCE PRODUCTS AND SERVICES

Salesforce offers a diverse array of products and services that collectively redefine how organizations manage their customer relationships and streamline business processes. At its core is the Customer Relationship Management (CRM) platform, comprising various clouds tailored to specific business needs. The Sales Cloud empowers sales teams with tools for lead management, opportunity tracking, and sales analytics, fostering efficient sales processes. The Service Cloud focuses on customer service, providing solutions for case management, customer support, and field service automation. Marketing Cloud is dedicated to marketing automation, enabling personalized and targeted marketing campaigns across multiple channels. Salesforce also boasts the Commerce Cloud, facilitating seamless e-commerce experiences for businesses.

1.3 SALESFORCE PRODUCTS AND SERVICES

Salesforce offers a diverse array of products and services that collectively redefine how organizations manage their customer relationships and streamline business processes. At its core is the Customer Relationship Management (CRM) platform, comprising various clouds tailored to specific business needs. The Sales Cloud empowers sales teams with tools for lead management, opportunity tracking, and sales analytics, fostering efficient sales processes. The Service Cloud focuses on customer service, providing solutions for case management, customer support, and field service automation.

Marketing Cloud is dedicated to marketing automation, enabling personalized and targeted marketing campaigns across multiple channels. Salesforce also boasts the Commerce Cloud, facilitating seamless e-commerce experiences for businesses.

1.3.1 Salesforce Clouds Services

The term “cloud” is frequently used to describe Salesforce’s products and services. Sales Cloud, Marketing Cloud, Commerce Cloud, Service Cloud, Experience Cloud, and Analytics Cloud – these are the six main clouds available on the platform is shown in the Figure 1.3. Salesforce offers a wide variety of clouds for various applications and businesses of all sizes. Healthcare, BFSI, retail, telecommunications, education, and manufacturing are among the major industries that leverage Salesforce Clouds ‘seemingly limitless capabilities.

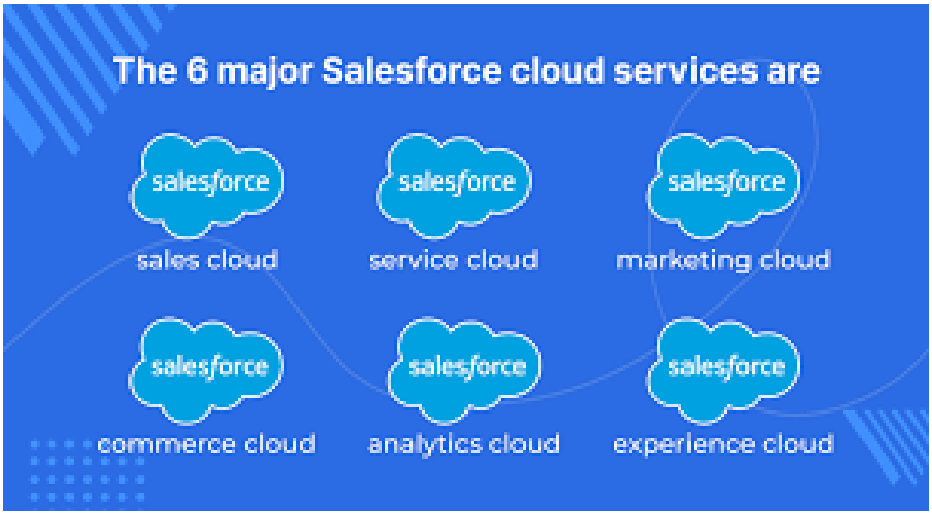


Figure 1.3 Types of Cloud Services

Sales Cloud: The Salesforce Sales Cloud is an application designed to help organizations sell smarter and faster by centralizing customer information, logging interactions, allowing collaboration across teams, and automating various tasks. It puts all your customer and prospect information in one place and makes sales teams more productive.

Service Cloud: Salesforce Service Cloud is among the most significant Salesforce Clouds and provides all the necessary functionalities that enhance customer experience and engagement. Its intelligent workflow automates business processes and facilitates customer journeys. It also obtains real-time customer information and uses Artificial Intelligence to increase sales and revenue.

Marketing Cloud: The Salesforce Marketing Cloud connects companies with every customer interaction and experience across multiple different channels. It is easy to control messaging flows, react to customer actions, and discover useful insights to grow your business.

Commerce Cloud: The name says it all – a cloud-based eCommerce platform from the commerce cloud market where we can create eCommerce websites, manage, and sell products. It provides businesses with a complete infrastructure and tools to set up an eCommerce store effortlessly.

Experience Cloud: The Salesforce Experience Cloud platform is integrated with Salesforce and enables businesses to engage with customers, partners, and employees. It consists of online portals or websites hosted by Salesforce named Experience Sites. These are exquisitely branded digital experiences that are easy to create and maintain.

Analytics Cloud: Analytics Cloud is a robust cloud-based analytical solution that accelerates exploring and analysing huge volumes of data. The Salesforce Analytics Cloud is powered by Einstein Analytics and Tableau tools. Analytics Cloud users can identify trends, improve efficiency and comprehend the logic behind the data.

1.3.2 Salesforce Products

Salesforce offers a diverse range of products, revolutionizing customer relationship management and business operations. At its core is the Salesforce Customer 360 platform, comprising essential products like Sales Cloud for sales automation, Service Cloud for customer service, and Marketing Cloud for marketing automation. Commerce Cloud elevates e-commerce experiences, while the Platform and App Clouds empower developers to create customized applications. Industry-specific solutions such as Financial Services Cloud and Health Cloud cater to specialized needs. Innovations like Einstein Analytics bring artificial intelligence into play, while MuleSoft ensures seamless integration. Salesforce's product suite reflects its commitment to providing comprehensive solutions for organizations seeking digital transformation and enhanced customer engagement.

1.4 ROLE OF A SALESFORCE DEVELOPER

A Salesforce developer plays a pivotal role in leveraging the Salesforce platform to craft tailored solutions for businesses. Proficient in programming languages such as Apex and Lightning Web Components, they design and implement custom applications, ensuring alignment with organizational objectives. Responsible for data modelling, integration, and creating dynamic user interfaces, Salesforce developers bridge the gap between business requirements and technical capabilities. Their expertise extends to optimizing system performance, coding triggers and classes, and contributing to the overall efficiency of Salesforce implementations. As key contributors to digital transformation, Salesforce developers are essential in building scalable, innovative solutions that enhance customer engagement and drive business success.

CHAPTER 2

SALESFORCE ARCHITECTURE

2.1 INTRODUCTION

Salesforce's architecture provides the structural foundation for its cloud-based customer relationship management (CRM) product suite. It represents a dynamic ecosystem created to give companies the security, scalability, and flexibility they need to manage their operations and data. The Salesforce architecture is essentially made up of multiple layers, each of which has a specific function in providing a wide range of CRM functionalities. The business logic layer is in charge of automating procedures and enforcing rules, while the presentation layer concentrates on user interface design. The data layer, on the other hand, stores structured data in a scalable, multi-tenant environment.

Furthermore, Salesforce has a strong security layer that guarantees the privacy and protection of sensitive data, as well as an integration layer that makes it easy to communicate with external systems. Building customised CRM solutions that meet unique business needs and promote organisational growth requires a foundational understanding of the complexities of Salesforce architecture, which is essential for developers, administrators, and stakeholders.

2.2 ARCHITECTURE OF SALESFORCE DEVELOPER

Salesforce Architecture is the multilayer architecture; it contains a series of layers situated on the top of each other. The architectural view of the salesforce are shown in the Figure 2.1.

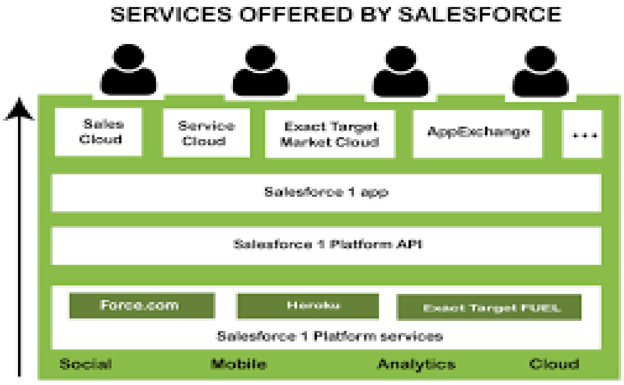


Figure 2.1 Architecture of Salesforce Developer

Salesforce offers a comprehensive suite of platform services designed to empower businesses with powerful tools for customer relationship management and business automation. These services include Sales Cloud, which provides capabilities for managing leads, opportunities, and sales pipelines; Service Cloud, for delivering exceptional customer service and support across various channels; Marketing Cloud, enabling personalized marketing campaigns and customer engagement; Commerce Cloud, for building scalable ecommerce solutions; Community Cloud, facilitating collaboration and engagement with customers, partners, and employees; Analytics Cloud, offering advanced analytics and data visualization capabilities; Salesforce Einstein, providing AI-driven insights and automation; and Salesforce Lightning Platform, a powerful development platform for building custom applications and extending Salesforce functionality. Together, these platform services form a robust ecosystem that enables organizations to streamline processes, drive innovation, and deliver exceptional experiences to customers and stakeholders.

2.2.1 Multi – Tenant Architecture

The multi-tenancy concept is that numerous users share computational, networking, and storage resources without seeing each other’s data. A multi-tenant application is customized for every group of users (so-called tenants) while the entire architecture shown in the Figure 2.2, and core functionality remain the same. Multi-tenancy is a typical approach for Software-as-a-Service (SaaS) vendors.

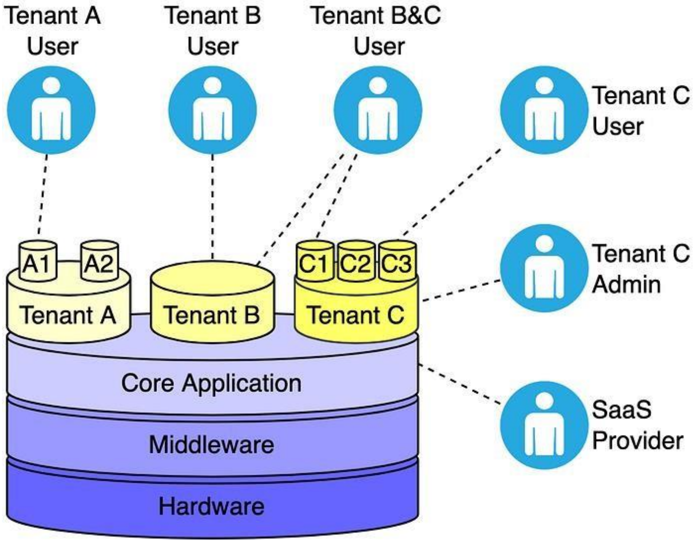


Figure 2.2 Multi – Tenant Architecture

In a multi-tenant system, the application's infrastructure, database, and user interface are typically designed to support isolation between tenants while maximizing resource utilization. This often involves techniques such as database schema separation, where each tenant's data is stored in a dedicated schema or partition, and configurable user interfaces that allow tenants to customize their experience within the shared application environment.

In the realm of cloud computing, SaaS (Software as a Service), AaaS (Applications as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service) have become established models for delivering software, applications, platforms, and infrastructure over the internet. However, for scenarios where multiple clients require access to the same stack of algorithms with either uniform or modular functionality adaptable to individual client needs, a slightly different approach emerges.This scenario is reminiscent of a tailored service model, where the core algorithms or functionalities remain consistent across clients but are customizable to accommodate specific requirements. Here, the focus shifts from providing complete software applications or platforms to delivering specialized algorithmic stacks or functional modules. This model can be aptly termed "Algorithm as a Service" (AaaS).

2.2.2 Components (Objects, Fields, Records)

In databases and data management systems, components refer to fundamental elements that structure and organize data. These components include objects, fields, and records, each serving a specific role in organizing and representing data. Objects represent the entities or categories of data within a database. They can include tables, views, or other data structures that encapsulate related information. Objects define the overall structure of the database and serve as containers for storing and managing data. Fields, also known as columns or attributes, define the individual characteristics or properties of objects. Each field represents a specific piece of data associated with an object. For example, in a database table representing employees, fields might include attributes such as employee ID, name, department, and salary.

Records, also referred to as rows or tuples, represent individual instances or entries within objects. Each record corresponds to a complete set of data values that describe a particular entity or item. Continuing with the example of an employee database table, each record would represent a specific employee, with values assigned to each field such as employee ID, name, department, and salary. Together, these components form the basic building blocks of a database, enabling the storage, retrieval, and manipulation of data in a structured and organized manner.

2.3 DATA MODELING IN SALESFORCE

Salesforce Data Model is essentially the manner in which tables of data are represented within your Salesforce database to make them understandable to anyone who views them are elaborated in the Figure 2.3.

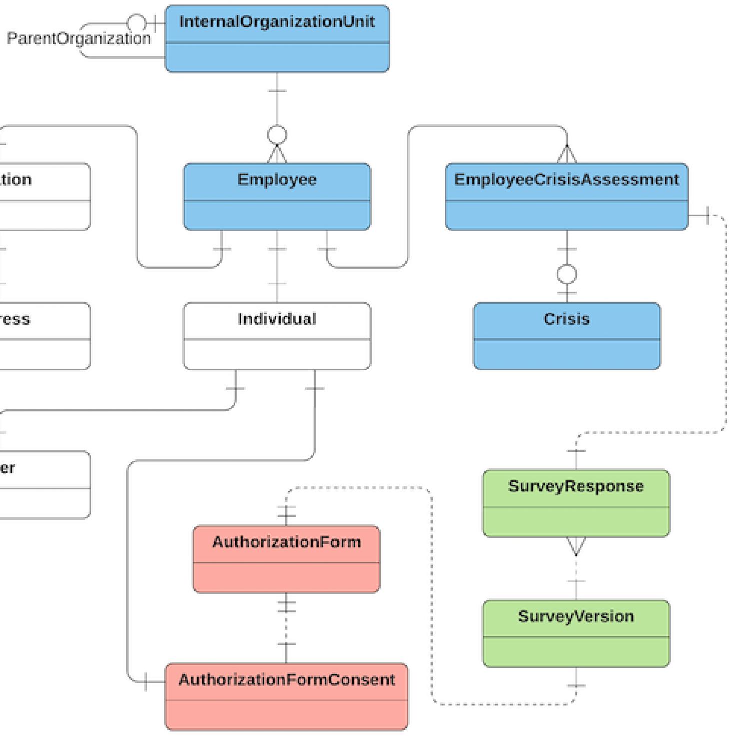


Figure 2.3 Data Modelling

Data modeling helps you make sense of the data residing within your system. It helps you establish specific relationships between Salesforce objects and fields, necessary to perform distinct CRM activities. SFDC data model is an aggregation of Salesforce objects and fields, which makes it important for Salesforce users to be well-versed with these terms to have a better understanding of data modelling.

2.3.1 Custom Objects

Custom Salesforce Objects are the objects created by users after implementing Salesforce if the pre-existing Standard Objects are not enough to comply with specific business requirements. These objects are unique to an organization and are created to serve their specific needs. The record data present in these objects is used for creating reports and dashboards. There are two main types of objects in Salesforce. Standard objects are the objects that are included in Salesforce by default. These include accounts, cases, contacts, and opportunities—the kind of information that almost every organization is keeping tabs on in Salesforce. Custom objects go beyond what comes out of the box. They're objects you create to handle the specific needs of your organization or industry.

2.4 LAYERS OF SALESFORCE

The foundation of Salesforce's CRM product suite is its architecture, which gives companies the tools they need to manage their data and operations efficiently. Salesforce provides a comprehensive platform that is scalable, flexible, and secure, accommodating a wide range of CRM functionalities through its multi-layered approach that encompasses business logic, presentation, data, security, and integration layers. For developers, administrators, and stakeholders who want to create specialised CRM solutions that meet their specific business needs, it is essential to comprehend the nuances of Salesforce architecture.

Organisations are able to improve user experiences, automate procedures, and preserve data integrity by utilising the different layers of Salesforce architecture. But in order for developers, administrators, and stakeholders to fully realise Salesforce's potential and create tailored CRM solutions that spur organisational growth, they must have a thorough understanding of its architecture.

CHAPTER 3

SALESFORCE ORG

3.1 INTRODUCTION

A Salesforce org is an entity which consists of the users, data, and automation corresponding to an individual organization. An organization could be a virtual space given to a person client of Salesforce. Your organization incorporates all of your information and applications, and is isolated from all other organizations. Salesforce is a leading cloud-based customer relationship management (CRM) platform that offers a plethora of features and functionalities to businesses for managing their sales, marketing, customer service, and other related operations and their types. Within the Salesforce environment, an "Org" refers to the specific instance or environment of Salesforce that a company or organization utilizes to manage its data and processes.

3.2 NAVIGATING THE SALESFORCE USER INTERFACE

Navigating the Salesforce user interface is an intuitive experience designed to streamline workflows and maximize productivity. Upon logging in, users are greeted with a customizable homepage, known as the Salesforce Lightning Experience, where they can access key features and tools tailored to their roles. The navigation bar at the top provides quick access to essential modules such as Home, Leads, Opportunities, Accounts, Contacts, and more. From here, users can easily navigate to different records, create new entries, and perform actions like logging calls or sending emails. The global search bar allows for effortless exploration of records across the entire Salesforce Org. Additionally, Salesforce employs a responsive design, making it accessible on various devices, including desktops, tablets, and smartphones, ensuring users can stay connected and productive regardless of their location. With its user-friendly interface and comprehensive navigation options, Salesforce empowers users to efficiently manage their CRM activities and drive business success. The navigation bar in Lightning Experience provides an efficient and consistent interface to navigate through your organization’s various apps and items. Similar to Salesforce Classic, apps in Lightning Experience give your users access to sets of objects, tabs, and other items all in one convenient bundle in the navigation bar. However, apps in Lightning Experience take things to another level beyond apps in Salesforce Classic by letting you brand your apps with a custom colour and logo. In Lightning Experience you can even include Lightning page tabs and a utility bar that allows instant access to productivity tools, like integrated voice, in the footer of Lightning Experience.

3.3 SALESFORCE BENDING

Salesforce branding in Salesforce Developer refers to the visual elements, design standards, and guidelines provided by Salesforce for developers to create applications and customize user interfaces within the Salesforce platform while maintaining consistency with the Salesforce brand. This includes adhering to Salesforce's design principles, typography, color palette, and iconography to ensure a cohesive and seamless user experience across all Salesforce applications and customizations. Developers can leverage Salesforce's branding guidelines to create custom themes, layouts, and stylesheets for their applications, incorporating Salesforce's logo, color, and visual elements to align with the overall look and feel of the Salesforce ecosystem. By following these branding guidelines, developers can build applications that integrate seamlessly with Salesforce's native functionalities and provide users with a familiar and intuitive experience. Additionally, Salesforce provides resources such as the Salesforce Lightning Design System (SLDS), which offers a library of pre-designed components and patterns that developers can use to build responsive and visually appealing user interfaces that adhere to Salesforce's branding standards. Overall, Salesforce branding in Salesforce Developer ensures that applications and customizations developed on the Salesforce platform maintain a consistent visual identity and user experience, reinforcing the strength and integrity of the Salesforce brand while delivering innovative solutions to users.

3.4 UI CUSTOMIZATION

User interface (UI) customization in Salesforce empowers organizations to tailor the look, feel, and functionality of their Salesforce instance to meet their specific business needs and branding requirements. With a robust set of tools and capabilities, Salesforce offers extensive UI customization options that enable organizations to create intuitive, userfriendly interfaces that drive productivity and user adoption. One of the key features of UI customization in Salesforce is the Lightning App Builder, a powerful drag-and-drop tool that allows administrators and developers to design custom interfaces without writing code. With the Lightning App Builder, users can create custom pages, record pages, and home pages by selecting and arranging components from a vast library of pre-built and custom components. These components include standard Salesforce components like lists, forms, charts, and related lists, as well as custom components developed using the Lightning Component Framework. By arranging these components on the canvas and configuring their properties, users can create highly tailored interfaces that meet their unique requirements. UI customization in Salesforce empowers organizations to create tailored and visually appealing user interfaces that align with their business needs and branding requirements.

3.5 IMPLEMENTATION

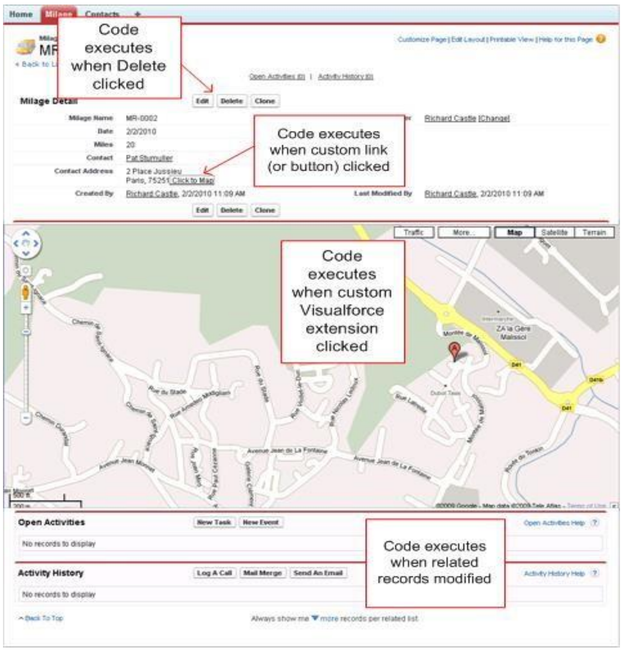
A Salesforce org depends on various factors, including the specific goals and objectives of the organization, the effectiveness of Salesforce implementation, and the ongoing support and maintenance of the platform. Generally, a well-implemented Salesforce org can lead to improved sales productivity, better customer relationship management, enhanced data management, and streamlined business processes. It can also provide valuable insights through analytics and reporting, helping the organization make informed decisions. However, the success of a Salesforce org is not guaranteed and requires continuous monitoring, optimization, and adaptation to evolving business needs and technological advancements. Regular training and updates are essential to ensure that users are effectively utilizing the platform and taking advantage of its capabilities. The effectiveness and success of a Salesforce org depend on how well it aligns with the organization's objectives, how effectively it is implemented and maintained, and how well it is utilized by its users. With proper attention and investment, a Salesforce org can be a valuable asset for driving growth and innovation within an organization.

CHAPTER 4

GETTING STARTED WITH APEX

4.1 INTRODUCTION TO APEX

Salesforce, as a leading cloud-based customer relationship management (CRM) platform, provides a robust and powerful programming language known as Apex. Apex is specifically designed for building scalable and secure applications on the Salesforce platform. This section introduces the key aspects of Apex and highlights its significance in Salesforce development. Apex is based on familiar Java idioms, such as variable and expression syntax, block and conditional statement syntax, loop syntax, object and array notation. Where Apex introduces new elements, it uses syntax and semantics that are easy to understand and encourage efficient use of the Lightning Platform is shown in the Figure 4.1. Therefore, Apex produces code that is both succinct and easy to write.



Milage Detail

Milage Name MR-0002

Date 2/2/2010

Miles 20

Contact Pat Stumuller

Contact Address 2 Place Jussieu
Paris, 75251 Click to Map

Created By Richard Castle, 2/2/2010 11:09 AM

Last Modified By Richard Castle, 2/2/2010 11:09 AM

Richard Castle [Change]

Open Activities

No records to display

Activity History

No records to display

Always show me ▼ more records per related list

Home
Milage
Contacts
+
Code executes when Delete clicked
Customize Page | Edit Layout | Printable View | Help for this Page
Open Activities (0) | Activity History (0)
Edit
Delete
Clone
Code executes when custom link (or button) clicked
Traffic
More...
Map
Satellite
Terrain
Code executes when custom Visualforce extension clicked
New Task
New Event
Open Activities Help
Code executes when related records modified
Log A Call
Mail Merge
Send An Email
Activity History Help
Back To Top

Figure 4.1 Apex Custom Page

Apex provides built-in support for unit test creation and execution. It includes test results that indicate how much code is covered, and which parts of your code could be more efficient. Salesforce ensures that all custom Apex code works as expected by executing all unit tests prior to any platform upgrades.

4.1.1 What is Apex

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on Salesforce servers in conjunction with calls to the API. Using syntax that looks like Java and acts like database stored procedures is shown in the Figure 4.2, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.

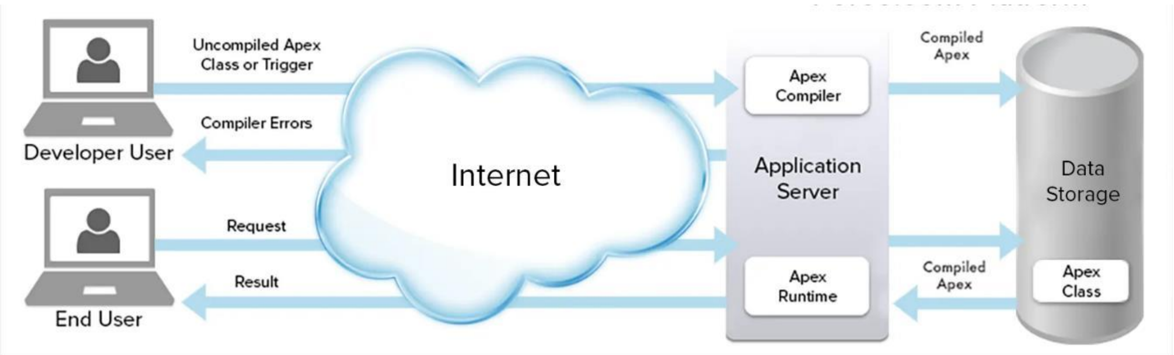
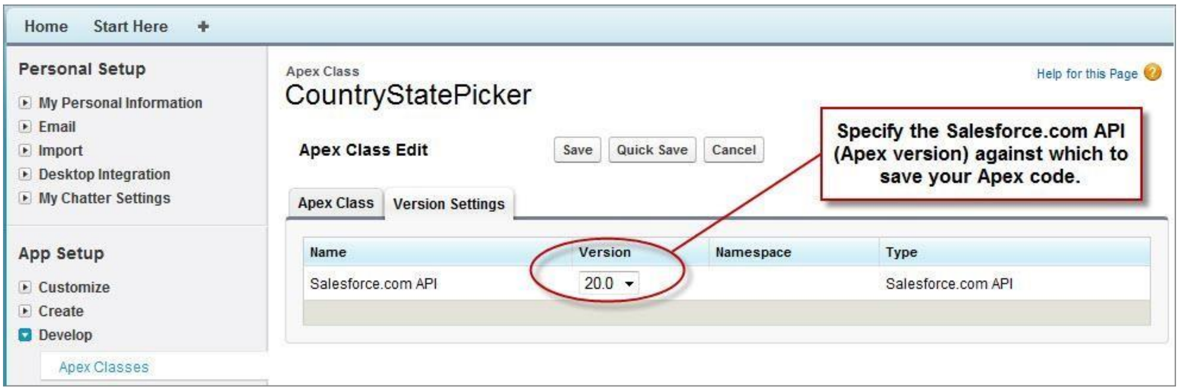


Figure 4.2 Get Started with Apex

Apex is saved, compiled, and executed on the server as the Lightning Platform. Object oriented-Apex supports classes, interfaces, and inheritance. Apex validates references to objects at compile time. Apex runs in a multitenant platform, it guards closely against runaway code by enforcing limits, which prevent code from monopolizing shared resources. Integrated with the database. It is straightforward to access and manipulate records. Apex provides direct access to records and their fields, and provides statements and query languages to manipulate those records. Apex provides transactional access to the database, allowing you to roll back operations. Apex is based on familiar Java idioms. Apex provides built-in support for unit test creation, execution, and code coverage. Salesforce ensures that all custom Apex code works as expected by executing all unit tests prior to any platform upgrades. Custom Apex code can be saved against different versions of the API.

4.1.2 Understanding Apex Core Concepts

In the Salesforce user interface, you can specify a version of the Salesforce API against which to save your Apex class or trigger. This setting indicates not only the version of SOAP API to use, but which version of Apex as well. You can change the version after saving. Every class or trigger name must be unique. You cannot save the same class or trigger against different versions is shown in the Figure 4.3.



Specify the Salesforce.com API (Apex version) against which to save your Apex code.

Name	Version	Namespace	Type
Salesforce.com API	20.0		Salesforce.com API

Figure 4.3 Version Setting User Interface

It can also use version settings to associate a class or trigger with a particular version of a managed package that is installed in your organization from AppExchange. This version of the managed package will continue to be used by the class or trigger if later versions of the managed package are installed, unless you manually update the version setting. To add an installed managed package to the settings list, select a package from the list of available packages. The list is only displayed if you have an installed managed package that is not already associated with the class or trigger.

4.1.3 Syntax and Triggers

Apex is a powerful and versatile programming language designed specifically for building applications on the Salesforce platform. This section provides an in-depth coverage of Apex, delving into its syntax and triggers.

1. Syntax

Apex syntax shares similarities with Java, making it accessible for developers familiar with object-oriented programming languages. Key components of Apex syntax include Variables and Data Types**,** Control Structures and Classes and Objects.

Variables and Data Types: Declaring variables and specifying data types.

```
String myString = 'Hello, Apex!';

Integer myNumber = 42;
```

Control Structures: Using if-else statements and loops for flow control.

```
if (condition) {
   // Code block executed if condition is true
} else {
   // Code block executed if condition is false
} for (Integer i = 0; i < 5; i++)
{
   // Code block executed in each iteration
}
```

Classes and Objects: Creating classes and instances.

```
public class MyClass {
public String myMethod () {
return 'Apex is powerful!';
   }
}
MyClass myInstance = new MyClass ();
String result = myInstance.myMethod();
```

2. Apex Triggers

Apex triggers are event handlers that execute code before or after records are inserted, updated, or deleted in Salesforce. Triggers play a crucial role in automating business processes and enforcing data integrity. Key aspects of Apex triggers include Trigger Context Variables

Trigger Context Variables: Accessing information about the records being processed.

```
trigger MyTrigger on MyObject__c (before insert) {
for (MyObject__c record: Trigger.new) {
      // Access and manipulate record fields
  }
```

4.2 APEX TRIGGERS

Apex triggers enable you to perform custom actions before or after events to records in Salesforce, such as insertions, updates, or deletions. Just like database systems support triggers, Apex provides trigger support for managing records. Typically, you use triggers to perform operations based on specific conditions, to modify related records or restrict certain operations from happening. You can use triggers to do anything you can do in Apex, including executing SOQL and DML or calling custom Apex methods. The figure 4.2 shows the prerequisites of Apex Specialist.

4.2.1 Bulk Apex Triggers

Apex triggers are optimized to operate in bulk. We recommend using bulk design patterns for processing records in triggers. When you use bulk design patterns, your triggers have better performance, consume less server resources, and are less likely to exceed platform limits. The benefit of bulkifying your code is that bulkified code can process large numbers of records efficiently and run within governor limits on the Lightning Platform. These governor limits are in place to ensure that runaway code doesn't monopolize resources on the multitenant platform.

4.2.2 Syntax and Triggers

Apex is a powerful and versatile programming language designed specifically for building applications on the Salesforce platform. This section provides an in-depth coverage of Apex, delving into its syntax and triggers.

1. **Syntax**

Apex syntax shares similarities with Java, making it accessible for developers familiar with object-oriented programming languages. Key components of Apex syntax include Variables and Data Types, Control Structures and Classes and Objects.

Variables and Data Types: Declaring variables and specifying data types.

```
String myString = 'Hello, Apex!';

Integer myNumber = 42;
```

Control Structures: Using if-else statements and loops for flow control.

```
 if (condition) {
 // Code block executed if condition is true
 } else {
// Code block executed if condition is false
 } for (Integer i = 0; i < 5; i++) {
// Code block executed in each iteration
}
```

Classes and Objects: Creating classes and instances.

```
 public class MyClass {
public String myMethod () {
 return 'Apex is powerful!';
}
 } MyClass myInstance = new MyClass ();
String result = myInstance.myMethod();
```

2. Triggers

Apex triggers are event handlers that execute code before or after records are inserted, updated, or deleted in Salesforce. Triggers play a crucial role in automating business processes and enforcing data integrity. Key aspects of Apex triggers include Trigger Context Variables

Trigger Context Variables: Accessing information about the records being processed.

```
trigger MyTrigger on MyObject c (before insert) {
```

```
for (MyObject c record: Trigger.new) {
// Access and manipulate record fields
 }
```

4.3 APEX UNIT TESTS

The apex testing framework enables you to write and execute tests for your apex classes and triggers on the lightning platform. Apex unit tests ensure high quality for your apex code and let you meet requirements for deploying apex. testing is the key to successful long-term development and is a critical component of the development process. The apex testing framework makes it easy to test your apex code. Apex code can only be written in a sandbox environment or a developer org, not in production. Apex code can be deployed to a production org from a sandbox also, app developers can distribute apex code to customers from their developer orgs by uploading packages to the lightning platform app exchange. In addition to being critical for quality assurance, apex unit tests are also requirements for deploying and distributing apex. The following are the benefits of apex unit tests.

- Ensuring that your apex classes and triggers work as expected

- Having a suite of regression tests that can be rerun every time classes and triggers are updated to ensure that future updates you make to your app don’t break existing functionality Meeting the code coverage requirements for deploying Apex to production or distributing Apex to customers via packages

- High-quality apps delivered to the production org, which makes production users more productive

- High-quality apps delivered to package subscribers, which increase your customers trust.

4.3.1 Asynchronous Apex

An asynchronous process is a process or function that executes a task "in the background" without the user having to wait for the task to finish. Here’s a real-world example. Let’s say you have a list of things to accomplish before your weekly Dance Revolution practice. Your car is making a funny noise, you need a different colour hair gel and you have to pick up your uniform from your mom’s house. You could take your car to the mechanic and wait until it is fixed before completing the rest of your list (synchronous processing), or you could leave it there and get your other things done, and have the shop call you when it’s fixed (asynchronous processing). If you want to be home in time to iron your spandex before practice, asynchronous processing allows you to get more stuff done in the same amount of time without the needless waiting.

4.3.2 Future Apex

Future Apex is used to run processes in a separate thread, at a later time when system resources become available. Note: Technically, you use the @future annotation to identify methods that run asynchronously. However, because "methods identified with the @future annotation" is laborious, they are commonly referred to as 21 "future methods" and that’s how we’ll reference them for the remainder of this module. When using synchronous processing, all method calls are made from the same thread that is executing the Apex code, and no additional processing can occur until the process is complete. You can use future methods for any operation you’d like to run asynchronously in its own thread. This provides the benefits of not blocking the user from performing other operations and providing higher governor and execution limits for the process. Everyone’s a winner with asynchronous processing.

4.3.3 Batch Apex

Batch Apex is used to run large jobs (think thousands or millions of records!) that would exceed normal processing limits. Using Batch Apex, you can process records asynchronously in batches (hence the name, “Batch Apex”) to stay within platform limits. If you have a lot of records to process, for example, data cleansing or archiving, Batch Apex is probably your best solution. Here’s how Batch Apex works under the hood. Let’s say you want to process 1 million records using Batch Apex. The execution logic of the batch class is called once for each batch of records you are processing. Each time you invoke a batch class, the job is placed on the Apex job queue and is executed as a discrete transaction as shown in figure 4.4.

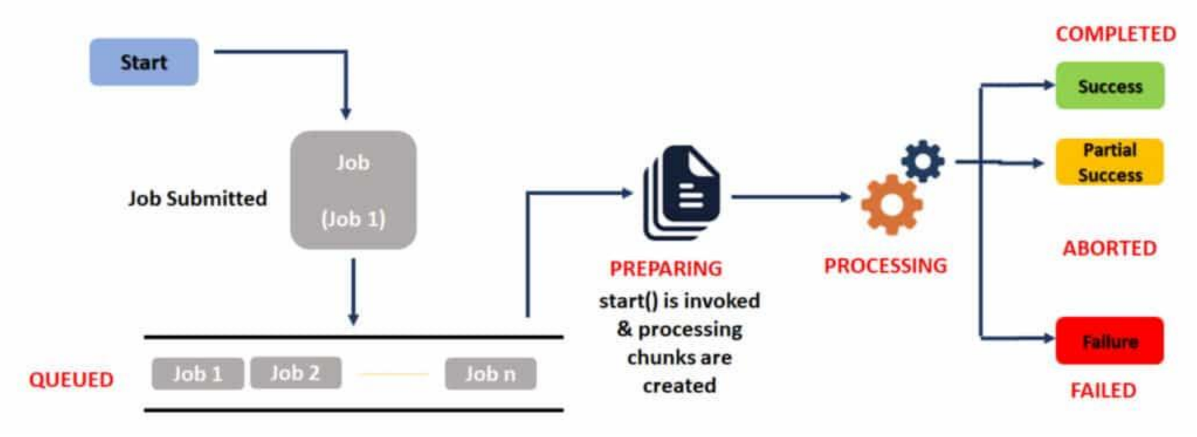


Figure 4.4 Batch Apex

CHAPTER 5

SUPER BADGES

5.1 INTRODUCTION

Aspiring developers follow a systematic learning path that ends with the unlocking of Super Badges following the successful completion of all Trailhead training modules and live sessions. For students who are committed to the developer route, achieving particular Super Badges that are meant to improve real-world skills and demonstrate mastery in critical areas is how expertise is confirmed. Super Badges function as an all-encompassing assessment tool, necessitating that students exhibit a profound comprehension of developer centric ideas and utilize their acquired knowledge in practical situations. Super Badges offer an interactive learning experience by pushing developers-in-training to tackle hard issues, construct strong solutions, and apply industry best practices.

As students acquire each Super Badge, they not only demonstrate their mastery of developer skills but also obtain concrete evidence of their ability, laying the groundwork for a lucrative and fulfilling career in the rapidly changing fields of software development and technology. Developers are certain to leave the learning process with a comprehensive skill set and the self-assurance to take on real-world difficulties in their chosen area thanks to this well-planned development.

5.2 PROCESS AUTOMATION SPECIALIST

The role of a Process Automation Specialist involves a multifaceted skill set and a deep understanding of streamlining and optimizing business processes through automation technologies. These experts play a crucial role in locating automation opportunities inside a company, evaluating current workflows, and putting customized solutions into place to boost production and efficiency. They are adept at using a range of automation platforms and tools, including workflow automation, business process management (BPM) systems, and robotic process automation (RPA). Process automation specialists work in conjunction with various stakeholders, such as IT teams and business analysts, to design, develop, and implement automation solutions that are consistent with the objectives of the organization are shown in the Figure 5.1. Their duties also include the ongoing observation and improvement of automated processes, making sure that the systems run well and adjust to changing business requirements.

Prerequisites
Formulas and Validations
Approve Records with Approval Processes
Leads & Opportunities for Lightning Experience
Process Automation Specialist

Figure 5.1 Process Automation Specialist

Process Automation Specialists are essential in spearheading digital transformation projects in contemporary firms because of their emphasis on lowering manual intervention, decreasing errors, and enhancing overall operational effectiveness. They are vital contributors to the development of effective and flexible corporate operations because of their knowledge in the nexus of technology, business process optimization, and strategic decision-making.

5.2.1 What You'll Be Doing to Earn This Super badge

In order to obtain the Process Automation Specialist Super Badge, candidates must complete a set of activities covering a wide range of competencies linked to Salesforce process automation. Here is a summary of the particular tasks and outcomes that must be completed: In the realm of Salesforce customization and optimization, several key strategies can be employed to enhance efficiency and accuracy within your organization's workflow processes.

Firstly, automating lead ownership through assignment rules ensures a seamless distribution of leads based on predetermined criteria, facilitating more effective follow-up and conversion efforts. This involves configuring rules within Salesforce to automatically assign leads to specific users or teams, streamlining the lead management process. Secondly, enforcing data integrity is crucial for maintaining the quality and reliability of your Salesforce database. By creating and implementing formula fields, you can dynamically compute values according to predefined standards, ensuring consistency and accuracy across your records. Additionally, the implementation of validation rules serves to prevent inaccurate or incomplete data entry, thereby maintaining the integrity of your database. Thirdly, establishing custom objects in a master-detail relationship with standard objects helps to enhance data organization and relationship management within Salesforce. By designing and creating custom objects that are linked to standard objects through a masterdetail relationship, you can effectively manage and track related information in a structured manner.

Moreover, defining an opportunity sales process using stages, record types, and validation rules allows for better monitoring and management of opportunities within Salesforce. By creating a well-defined sales process with distinct phases and utilizing record types to group opportunities according to specific criteria, you can streamline the sales pipeline and ensure consistency in data management. Additionally, the implementation of validation rules helps to maintain data accuracy and consistency within opportunity records. Furthermore, automating business processes such as sending emails, creating related records, and submitting opportunities for approval enhances workflow efficiency and productivity. By creating automated procedures and workflows within Salesforce, you can trigger actions in response to preset events or requirements, reducing manual effort and improving data completeness.

Additionally, leveraging flows in Salesforce to display dynamic information on Lightning record pages enhances user experience and productivity. By creating and implementing flows that display contextual data based on user interactions, you can provide users with relevant information in real-time, enhancing their ability to make informed decisions and take action within the Salesforce environment. Lastly, creating processes to evaluate and update records in accordance with specified criteria helps to ensure data accuracy and consistency over time. By establishing procedures within Salesforce to assess record criteria and automatically update records based on evaluation results, you can maintain the integrity of your database and ensure that information remains up-to-date and relevant.

5.3 DATA INTEGRITY ENFORCEMENT

Data integrity enforcement in Salesforce is a critical aspect of maintaining the accuracy, consistency, and reliability of the information stored within the platform. Salesforce provides various mechanisms and features to ensure data integrity, primarily through validation rules, workflows, and automation processes shown in Figure 5.2.

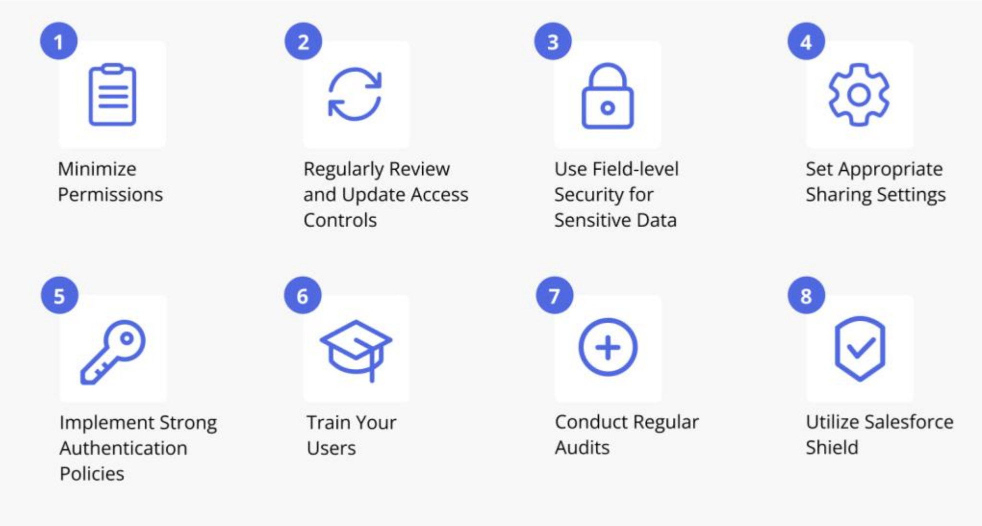

1
Minimize Permissions
2
Regularly Review and Update Access Controls
3
Use Field-level Security for Sensitive Data
4
Set Appropriate Sharing Settings
5
Implement Strong Authentication Policies
6
Train Your Users
7
Conduct Regular Audits
8
Utilize Salesforce Shield

Figure 5.2 Data Integrity Enforcement

Validation rules allow administrators to define specific criteria that data must meet before it can be saved in Salesforce. These rules can range from simple checks, such as ensuring that certain fields are not left blank, to more complex validations involving multiple fields and conditions. By enforcing these rules, Salesforce helps prevent the entry of incorrect or incomplete data, thus maintaining the integrity of the database. In addition to validation rules, Salesforce offers workflows and automation tools that enable businesses to automate processes and enforce data integrity guidelines. Workflows can trigger actions based on predefined criteria, such as sending alerts or updating records, ensuring that datarelated tasks are completed consistently and accurately.

5.4 SUMMARY

To maintaining the data integrity in Salesforce is crucial for ensuring the accuracy, consistency, and reliability of information stored within the platform. Salesforce provides a range of mechanisms and features, including validation rules, workflows, and automation tools, to enforce data integrity guidelines effectively. Validation rules allow administrators to define specific criteria that data must meet before being saved, preventing the entry of incorrect or incomplete data. Workflows and automation tools further enhance data integrity by automating processes based on predefined criteria, reducing the risk of human error and ensuring consistency in data-related tasks. By leveraging these features, organizations can uphold data integrity standards throughout their Salesforce system, ultimately enhancing the overall reliability of their database and facilitating informed decision-making.

CHAPTER 6

CONCLUSION

The record-level access controls at the heart of the Lightning Platform are extremely flexible and powerful, and serve the collaboration and security needs of all customers—from those working in small sales teams to those working in very large enterprises. With the knowledge and features described in this paper, Salesforce developers and administrators can optimize system performance while continuing to deliver the flexibility their companies require in access control. Embarking on an internship as a Salesforce developer presents an exciting opportunity to delve into the dynamic realm of Customer Relationship Management (CRM) and Salesforce, a pioneering platform in this domain. Salesforce, a cloud-based software, has evolved exponentially over the years, becoming a cornerstone for businesses in managing their customer interactions and data. Understanding the multi-tenant architecture is fundamental. This architecture allows multiple users to share the same infrastructure while maintaining data isolation, ensuring scalability and efficiency. Key components such as Objects, Fields, and Records form the building blocks of Salesforce architecture, providing a robust foundation for data modelling. In the realm of data modelling, Salesforce offers the flexibility of creating Custom Objects and establishing Relationships, be it Lookup or Master-Detail. Navigating the Salesforce User Interface (UI) is crucial, and the platform's UI customization features empower developers to tailor the interface according to specific business requirements, ensuring an intuitive user experience. Apex, Salesforce's proprietary programming language, is at the heart of development. Mastering Apex involves understanding its core concepts, syntax, and triggers. Additionally, Visualforce Pages and Lightning Components enhance the user interface, offering customization through custom controllers and extensions. Embarking on an internship as a Salesforce developer is an enriching journey through the intricacies of CRM, Salesforce architecture, data modelling, and development tools.

REFERENCES

1. https://www.salesforce.com/events/
2. https://success.salesforce.com/
3. https://developer.salesforce.com/docs/
4. https://trailhead.salesforce.com/