KIET SALESFORCE APPLICATION
(Salesforce Application)

A PROJECT REPORT
for
Major Project (KCA451)
Session (2023-24)

Submitted By

Vivek Kumar
University Roll No- 2200290140184

Submitted in partial fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Dr. Akash Rajak
Professor

Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(MAY 2024)

DECLARATION

I hereby declare that the work presented in report entitled “KIET Salesforce Application” was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University of Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, that are not my original contribution. I have used quotation marks to identify verbatim sentences and give credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Vivek Kumar
Roll No.: 2200290140184

(Candidate Signature)

CERTIFICATE

Certified that **Vivek Kumar (University Roll No.- 2200290140184)** has carried out the projectwork having “**KIET Salesforce Application**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)**, Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the studenthimself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date :

Vivek Kumar (2200290140184)

This is to certify that the above statement made by the candidate is correct to the best of myknowledge.

Date:

Dr. Akash Rajak
Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Kumar Tripathi
Professor & Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

The KIET Salesforce Application is a comprehensive Customer Relationship Management (CRM) system tailored for KIET Group of Institutions. Designed to streamline operations, enhance communication, and foster a robust relationship between the institution and its stakeholders, this application leverages the power of Salesforce's cloud-based solutions.

A The primary goal of the KIET Salesforce Application is to optimize the management of student admissions, alumni relations, faculty interactions, and administrative tasks. By integrating various functionalities into a single platform, the application aims to improve efficiency, data accessibility, and decision-making processes within the institution.

In The KIET Salesforce Application represents a significant advancement in the way educational institutions manage their relationships and operations. By leveraging the robust features of Salesforce, KIET can enhance its service delivery, improve stakeholder engagement, and achieve its strategic goals more efficiently. This application serves as a model for other educational institutions looking to modernize their CRM and administrative processes.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Akash Rajak** (Professor) for their guidance, help and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions have guided me a lot in completing this project successfully.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi,** Professor and Head, Department of Computer Applications, for his insightful comments and administrative helpat various occasions. Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, completionof this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Vivek Kumar (2200290140184)

TABLE OF CONTENTS

Certificate i

Abstract . ii

Acknowledgement iii

Table of Content iv

Chapter 1 – Introduction 01 - 09

1.1 Project description 01 - 02

1.2 Literature Review 02 - 06

1.3 Hardware / Software used in Project 07

1.4 Functional Requirements 08

1.5 Non- Functional Requirements 09

Chapter 2 Feasibility Study 10 - 11

2.1 Technical feasibility 10

2.2 Operational Feasibility 10

2.3 Behavioral Feasibility 11

2.4 Operational Feasibility 11

Chapter 3 Database Design 12 - 24

3.1 Waterfall Model 12

3.2 Requirement Gathering & Analysis 13

3.3 ER Diagram 14 - 16

3.4 Use Case Diagram 17

3.5 Activity Diagram 18

3.6 Sequential Diagram 19 - 20

3.7 Collaboration Diagram 21

3.8 State Chart Diagram 22

3.9 Component Diagram 23

3.10 Deployment Diagram 34
Chapter 4 Lead Generation 35
4.1 Screenshot 36 - 47
Chapter 5 Coding(Test Cases) 48 - 52
5.1 funtional testing 53
Chapter 6 Conclusion 54
Chapter 7 Future Scope 55
Chapter 8 Bibliography 56

CHAPTER 1

INTRODUCTION

1.1 Project Description

The KIET Salesforce Application is a specialized Customer Relationship Management (CRM) system developed for the Krishna Group of Institutions. In an era where efficient management and seamless communication are crucial for institutional success, this application serves as a pivotal tool to address these needs. By harnessing the power of Salesforce's cloud-based solutions, the KIET Salesforce Application is designed to streamline various administrative and operational processes, ultimately fostering a more connected and responsive academic environment.

User-Friendly Interface

Salesforce offers a user-friendly interface designed to simplify navigation and improve user experience. The intuitive layout features a customizable dashboard, which allows users to personalize their workspace with relevant information and tools. Drag-and-drop functionality, along with point-and-click configuration, minimizes the need for extensive technical knowledge. Contextual help and guided learning paths through Salesforce Trailhead make it easy for users to quickly become proficient. The interface is consistent across different devices, ensuring a seamless experience whether accessed via desktop or mobile, facilitating higher user adoption and satisfaction among faculty, staff, and students at KIET.

Diverse Restaurant Selection

The Salesforce application facilitates a diverse course selection at KIET by offering students a comprehensive catalog spanning various disciplines. It provides detailed course information such as syllabi, prerequisites, and faculty profiles, aiding informed decisions. Students benefit from flexible enrollment options, including cross-departmental selections and customizable learning paths to align with their academic and career goals. Real-time updates ensure students have current information on course availability and schedules, enhancing their overall experience and satisfaction with the diverse range of educational opportunities available through the platform.

Real-Time Menu Updates

Real-time updates in the Salesforce application for KIET ensure timely dissemination of critical information to stakeholders. This includes instant notifications on course changes, exam schedules, and academic updates. Faculty and students benefit from current data on grades, attendance, and administrative announcements, fostering transparency and efficiency in academic operations. The system's ability to deliver immediate updates enhances communication, responsiveness, and decision-making across the institution, ensuring seamless educational management and improving overall user satisfaction with real-time access to pertinent information.

It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design to outline its architecture and functionality. Implementation involves coding and rigorous testing to ensure adherence to design specifications. Verification includes systematic testing to validate functionality and quality assurance.

It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design to outline its architecture and functionality.

Efficient Search and Filters

The Salesforce application at KIET incorporates efficient search and filtering tools to optimize data retrieval. Users can swiftly locate specific information, including student records and course details, through advanced search functionalities. Customizable filters enable refined searches by criteria such as department, instructor, and semester, enhancing accuracy and relevance of results. The interface ensures intuitive navigation, facilitating seamless access to filtered data and promoting productivity among faculty, staff, and students. With mobile accessibility, users can conveniently access information from anywhere, supporting diverse needs and improving overall operational efficiency within the institution.

Secure Payment Options

In the Salesforce application for KIET, secure data management is paramount. Data encryption techniques are employed to protect information both in transit and at rest, ensuring confidentiality and integrity. Role-based access controls (RBAC) restrict data access based on user roles, enhancing security. The system complies with stringent privacy regulations such as GDPR and FERPA, safeguarding sensitive data. Detailed audit trails track data access and modifications for accountability and transparency purposes. Regular security updates and patches are applied to mitigate risks and maintain system integrity, providing robust protection against emerging cybersecurity threats and ensuring the security and confidentiality of all institutional data.

Feedback and Ratings

The feedback and ratings feature in KIET's Salesforce application promotes continuous improvement and stakeholder engagement. Users can provide feedback on courses, instructors, and administrative processes, fostering a culture of responsiveness and transparency. A rating system enables data-driven decision-making by aggregating user ratings for courses and services. Analytics tools transform this feedback into actionable insights, guiding institutional improvements and strategic initiatives. Enhanced communication channels facilitate transparent dialogue between stakeholders, enhancing satisfaction and accountability. This iterative feedback loop ensures ongoing enhancements to educational quality and operational effectiveness, empowering KIET to meet evolving student and faculty needs effectively.

Conclusion

In conclusion, the Salesforce application represents a transformative tool for KIET, enhancing operational efficiency, data security, and stakeholder engagement. With robust features like efficient search capabilities, secure data management, and feedback mechanisms, the platform facilitates streamlined academic and administrative processes. Its user-friendly interface and mobile accessibility ensure seamless navigation and accessibility across devices. By adhering to stringent data protection standards and enabling continuous improvement through feedback and ratings, Salesforce empowers KIET to deliver enhanced educational experiences. Moving forward, leveraging Salesforce will continue to support KIET in achieving its strategic goals, fostering innovation, and meeting the evolving needs of its academic community.

It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design to outline its architecture and functionality. Implementation involves coding and rigorous testing to ensure adherence to design specifications.

1.2 Literature Review

A literature review typically involves summarizing and synthesizing existing research and scholarly articles relevant to a particular topic. Here’s a general outline for a literature review on a topic related to Salesforce applications in educational institutions.

Introduction

Salesforce applications have emerged as pivotal tools in modernizing educational management systems, offering customizable solutions to streamline administrative tasks, enhance data management, and improve stakeholder engagement. In academic settings, the adoption of Customer Relationship Management (CRM) systems like Salesforce has gained momentum, promising significant benefits such as improved operational efficiency and data-driven decision-making. This literature review explores the existing research and scholarly discourse surrounding Salesforce's role in educational institutions, highlighting its features, benefits, challenges, and implications. By synthesizing current knowledge, this review aims to provide insights into leveraging Salesforce to meet the evolving needs of educational organizations effectively.

Consumer Behavior and Adoption

Consumer behavior and adoption in educational settings, particularly regarding technologies like Salesforce, involves studying the motivations and decision-making processes of stakeholders—administrators, faculty, and students. It explores factors influencing adoption, such as perceived benefits, usability, and compatibility with existing systems. Barriers like resistance to change and organizational inertia also play critical roles. The adoption process is often analyzed using models such as the Technology Acceptance Model (TAM), which assesses factors influencing users' acceptance of new technologies. Case studies and empirical research provide insights into successful adoption strategies, guiding educational institutions in effectively implementing and integrating innovative solutions like Salesforce.

User Experience and Interface Design

User experience (UX) and interface design are critical in Salesforce applications for educational institutions. The design focuses on intuitive navigation and accessibility, ensuring usability across various devices and user proficiency levels. Integrating user feedback helps refine interfaces, enhancing functionality and user satisfaction. Visual design elements are carefully crafted to balance aesthetics with practical usability, promoting efficient interaction with educational data and management tools. By prioritizing UX and interface design, Salesforce applications in education can effectively support stakeholders—administrators, faculty, and students—in achieving their academic and administrative objectives with ease and clarity.

Application Partnerships

Application partnerships in Salesforce involve collaborating with third-party developers to integrate specialized features like LMS and SIS, providing tailored solutions for education. These partnerships ensure seamless integration, support, and innovation, enhancing Salesforce's capabilities in academic management. It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design to outline its architecture and functionality. Implementation involves coding and rigorous testing to ensure.

Technology and Innovation

Technology and innovation play a crucial role in transforming educational management systems, particularly through tools like Salesforce. These advancements empower institutions by enhancing data management, automation capabilities, and analytical insights. By integrating seamlessly with existing systems such as learning management systems (LMS) and student information systems (SIS), technology optimizes operational efficiency and supports informed decision-making. This integration and innovation not only streamline administrative processes but also improve user experiences for stakeholders like administrators, faculty, and students, ultimately helping educational institutions achieve their academic and organizational objectives more effectively.

Security and Trust

Security and trust are fundamental pillars in the implementation of Salesforce applications in educational settings. Institutions rely on robust security measures such as data encryption, access controls, and compliance with regulations like GDPR and FERPA to safeguard sensitive information. Building trust involves ensuring data integrity, confidentiality, and transparency in handling student and institutional data. By prioritizing security and trustworthiness, Salesforce enables educational institutions to manage operations confidently, protect privacy, and uphold their commitment to stakeholders' data protection and ethical standards. This approach fosters a secure environment essential for effective academic and administrative management.

Social Impact and Community Building

Salesforce applications in education promote social impact and community building by fostering collaboration and engagement among students, alumni, and stakeholders. Institutions use Salesforce to personalize communications, manage community-focused programs, and enhance diversity and inclusion efforts. These initiatives empower educational communities to measure their impact, improve accountability, and foster meaningful connections that contribute to a positive social environment and enriched student experiences. Through Salesforce, educational institutions strengthen their community ties, promote shared values, and drive collective efforts towards achieving broader social goals.

1.3 Software Used in Project

Salesforce itself serves as the central Customer Relationship Management (CRM) platform, managing data, automating processes, and providing analytics. Integration with third-party systems like Learning Management Systems (LMS) and Student Information Systems (SIS) such as Moodle, Canvas, or Ellucian enhances functionality. Analytics tools such as Tableau or Einstein Analytics offer insights, while communication tools like email platforms and collaboration software facilitate seamless interaction. Development environments like Salesforce Development Environment and Visual Studio Code support custom development, ensuring tailored solutions that optimize educational management and user experiences.

1.3.1. Integrated Development Environment (IDE)

An Integrated Development Environment (IDE) for Salesforce development, such as Salesforce Extensions for Visual Studio Code and Force.com IDE, provides tools for coding, debugging, and deploying applications. It supports efficient development of Apex, Visualforce, and Lightning components, enhancing productivity and workflow management in Salesforce projects.

Programming Languages

In Salesforce development, the primary programming languages used are:

Apex: Salesforce's proprietary language for creating business logic and backend processes.

Visualforce: Markup language for building user interfaces in Salesforce.

JavaScript: Used within Lightning components for client-side logic and UI customization.

These languages enable customization and integration within the Salesforce ecosystem, supporting a wide range of application functionalities and user interactions.

1.3.2 Database Management

Database management in Salesforce revolves around its robust data architecture:

Salesforce Database: A Built-in, scalable database handling data storage and retrieval.

Object-Relational Mapping (ORM): Defines custom objects and fields, managing relational data structure.

SOQL (Salesforce Object Query Language): Queries data in Salesforce, akin to SQL, for data manipulation and retrieval.

1.3.3. Application Programming Interfaces (APIs):

Application Programming Interfaces (APIs) in Salesforce facilitate seamless integration and data exchange:

REST API: Allows external systems to interact with Salesforce for data retrieval, manipulation, and integration.

SOAP API: Provides protocol for accessing Salesforce data and functionalities through web services.

Bulk API: Optimizes loading or deletion of large data sets, enhancing efficiency in data management tasks.

Cloud Services:

Cloud services in Salesforce provide scalable, secure, and accessible solutions for educational institutions:

Infrastructure as a Service (IaaS): Offers scalable computing resources, storage, and networking capabilities. It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design to outline its architecture.

Platform as a Service (PaaS): Facilitates development, deployment, and management of applications without infrastructure management. It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design.

Software as a Service (SaaS): Delivers applications over the internet, such as Salesforce CRM, accessible via web browsers.

These cloud services enable educational institutions to leverage Salesforce's robust platform for efficient data management, application development, and operational scalability, ensuring seamless access to educational resources and services.

Version Control:

Git: A distributed version control system used to track changes in the source code, enabling collaboration among developers and maintaining code integrity.

Version Control Systems (VCS):
Integrates with Salesforce development environments to manage changes across teams.

Branching and Merging: Allows parallel development of features and bug fixes while maintaining code integrity.

1.3. Hardware Used in Project

- Processor – Intel 3 11th Gen
- RAM – 8 GB

1.4 Functional Requirements

Functional requirements for Salesforce applications in education include student and course management, faculty administration, reporting, integration with LMS/SIS, communication tools, and robust security measures. These ensure efficient operations, data synchronization, collaboration, and secure access management, enhancing educational management and user experience.

User Registration and Authentication:
User registration and authentication in Salesforce applications for educational institutions involve:

Registration: Allows students, faculty, and administrators to create user accounts with required profile information.

Authentication: Verifies user identity through login credentials, often integrating with single sign-on (SSO) solutions for seamless access.

Password Management: Ensures secure password policies, reset options, and multi-factor authentication for enhanced security. It begins with comprehensive requirements gathering to define all aspects of the Salesforce application.
Role-based Access Control: Assigns roles and permissions based on user roles (student, faculty, admin) to manage data access and functionality.

Cost management in Salesforce for education involves overseeing expenses related to licensing, implementation, maintenance, scaling, and assessing return on investment (ROI).

Browse and Search:

Browse: Navigate through courses, student records, and institutional resources using intuitive menus and categories.

Search: Utilize search functionality to quickly locate specific information, such as courses, documents, or student profiles.

Filters: Refine search results using filters based on criteria like course subject, instructor, or student demographics.

Menu Display:

Menu display in Salesforce applications for education is crucial for intuitive navigation and accessibility. It organizes features such as student management, course administration, and reporting tools into logical categories. Menus are designed to streamline user interaction, ensuring quick access to essential functionalities like enrollment, grading, and communication tools. Customizable menus allow institutions to tailor navigation to their specific needs, enhancing user efficiency and satisfaction. Clear labeling and hierarchical organization facilitate seamless browsing and searching, empowering stakeholders—administrators, faculty, and students—to efficiently navigate and utilize Salesforce's comprehensive educational management capabilities.

Student Record:

Student records in Salesforce for education encompass vital data management, tracking personal details, enrollment statuses, academic history, grades, and communication logs. This centralized system ensures with faculty and administrators. It supports informed decision-making and proactive student support through data-driven insights. Robust security measures protect sensitive information, complying with regulations like GDPR and FERPA. By integrating these functionalities, Salesforce enhances administrative efficiency, promotes transparency, and facilitates personalized student experiences, ultimately contributing to improved educational outcomes and institutional effectiveness.

User Profile:

User profiles in Salesforce for education define roles and permissions tailored to stakeholders like students, faculty, and administrators. They control access levels and actions users can perform, ensuring efficient management of student records, course management, and administrative tasks while upholding data security and compliance with regulatory requirements like GDPR and FERPA.

Payment Integration:

Payment integration in Salesforce for educational institutions facilitates seamless financial transactions by integrating with leading payment gateways such as PayPal or Stripe. It automates billing, invoicing, and revenue tracking processes, ensuring accurate financial reporting and compliance with accounting standards. Customizable workflows accommodate diverse payment.

This integration supports transparency and reliability in financial operations, providing students, parents, and donors with secure and accessible payment options. Ultimately, it strengthens financial management capabilities within educational institutions, contributing to operational.

Review and Ratings:

Reviews and ratings in Salesforce for education facilitate valuable feedback loops among students, faculty, and administrators. This functionality allows stakeholders to provide insights on courses, instructors, and institutional services, enhancing decision-making and program improvements. Through robust analytics, institutions gain actionable data to refine educational offerings and support continuous quality enhancement.

Transparent review mechanisms build trust and accountability, fostering a collaborative environment focused on student success and institutional excellence. By leveraging reviews and ratings, Salesforce enables educational institutions to adapt and evolve, ensuring they meet the evolving needs and expectations of their academic community effectively.

Admin Dashboard:

An admin dashboard in Salesforce for education offers a centralized hub for administrators to monitor and manage institutional operations effectively. It provides intuitive data visualization, showcasing critical metrics such as enrollment trends, student performance analytics, and financial summaries. Customizable widgets and interactive charts empower administrators to track KPIs, identify trends, and make informed decisions swiftly.

The dashboard supports proactive management by highlighting areas needing attention, facilitating resource allocation, and fostering strategic planning. Ultimately, it enhances administrative efficiency, promotes data-driven decision-making, and ensures alignment with organizational goals for achieving academic excellence and operational effectiveness within educational institutions.

Feedback and Support:

Feedback and support mechanisms in Salesforce for education ensure robust stakeholder engagement:

Feedback Collection: Enables students, faculty, and administrators to provide input on courses, services, and experiences.

Ticketing System: Manages support requests and tracks resolutions for technical or administrative issues.

Analytics: Analyzes feedback data to identify trends, assess satisfaction levels, and drive continuous improvement.

Communication Tools: Facilitates responsive communication between users and support teams.

Integration with Third-Party Services:

Integration with external services for map navigation, social media, or marketing tools. Remember to include detailed descriptions, use cases, and possibly flowcharts or diagrams to illustrate how these functionalities interact. Additionally, ensure that your functional requirements align with the overall objectives and scope of the food ordering application project.

Learning Management Systems (LMS): Integrates with platforms like Moodle or Canvas to synchronize course data, grades, and student information.

Student Information Systems (SIS): Connects with systems such as PowerSchool or Banner for seamless student enrollment, attendance tracking, and academic records management.

Student Information Systems (SIS): Connects with systems such as PowerSchool or Banner for seamless student enrollment, attendance tracking, and academic records management.

Analytics and BI Tools: Integrates with tools like Tableau or Power BI for advanced data visualization and analytics on student performance and institutional metrics.

1.5 Non-Functional Requirements

Non-functional requirements in Salesforce for educational institutions focus on system attributes that describe how the system should behave, rather than specific functionalities. These include:

Performance: Ensuring fast response times and scalability to handle peak loads during registration periods or exams.

Reliability: Maintaining high availability and minimizing downtime to ensure continuous access to educational resources.

Security: Implementing robust data encryption, access controls, and compliance with data protection regulations (e.g., GDPR, FERPA).

Performance

Performance requirements in Salesforce for education ensure fast response times, scalability for peak loads, and reliable uptime, optimizing user experience accessing student records, course information, and administrative functions crucial for operational efficiency in educational institutions.

Scalability

Scalability in Salesforce for education is essential for accommodating expanding user bases, managing growing volumes of student data, and supporting increased system complexity. It involves optimizing infrastructure to handle peak loads during enrollment periods and exams, ensuring seamless performance and responsiveness.

Scalability also encompasses flexible architecture that allows integration with new modules and third-party services, adapting to evolving educational requirements. By maintaining robust scalability, Salesforce empowers educational institutions to efficiently scale operations, enhance user satisfaction, and meet the dynamic demands of educational management effectively over time.

Reliability

Reliability in Salesforce for education is critical for ensuring uninterrupted access to essential services, such as student records and administrative tools. It involves maintaining high availability, implementing fault-tolerant systems to prevent data loss, and ensuring stable performance under. By prioritizing reliability, Salesforce supports seamless operations and fosters trust among users, enabling educational institutions to deliver consistent and reliable services to students, faculty, and administrators alike.

Availability

Availability in Salesforce for education ensures that the system remains operational and accessible to users without interruptions. This involves maintaining high uptime, implementing backup systems to minimize downtime, and preparing for disaster recovery scenarios.

By prioritizing availability, Salesforce supports continuous access to essential educational resources and services, enabling institutions to deliver reliable and seamless experiences to students, faculty, and administrators. This reliability fosters trust and ensures that educational operations can proceed smoothly, even during periods of high demand or unforeseen events.

Security

Security in Salesforce for education is crucial for protecting sensitive data and ensuring compliance:

Data Protection: Implements encryption and access controls to safeguard student records, financial information, and personal data.

Compliance: Adheres to regulatory standards such as GDPR and FERPA to protect privacy rights and ensure data confidentiality.

Authentication: Utilizes secure login mechanisms and multi-factor authentication to verify user identities.

Auditing and Monitoring: Tracks system activities and audits access to detect and respond to security incidents promptly.

Data Backup and Recovery

Data backup and recovery in Salesforce for education involves implementing processes to safeguard and restore critical information. This includes regular backups of student records, course data, and administrative information to prevent data loss. Utilizing redundant storage and replication methods ensures data redundancy, minimizing the impact of hardware failures or unforeseen events.

Additionally, establishing comprehensive recovery procedures and conducting regular tests verify the integrity of backups and readiness for recovery scenarios. These practices ensure that educational institutions can maintain data integrity, comply with regulations, sustain operational continuity, and provide reliable access to essential educational resources and services. It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design to outline its architecture and functionality.

Compliance

Compliance in Salesforce for education encompasses adherence to legal regulations and organizational policies governing data privacy, security, and ethical standards. It involves ensuring that student information, financial data, and administrative records are protected according to laws like GDPR, FERPA, and CCPA.

Salesforce supports compliance through robust security measures, including encryption, access controls, and audit trails to track data access and modifications. Ongoing training and awareness programs educate staff and stakeholders on compliance requirements, reinforcing best practices for data handling. By prioritizing compliance, educational institutions maintain trust, mitigate risks of data breaches, and uphold the integrity of sensitive information.

Usability

Usability in Salesforce for education focuses on optimizing the user experience by ensuring interfaces are intuitive, easy to navigate, and efficient in facilitating tasks for administrators, faculty, and students. This includes:

Intuitive Design: Interfaces that are user-friendly and require minimal training for users to understand and navigate effectively.

Accessibility: Ensuring that the platform is accessible to all users, including those with disabilities, by adhering to accessibility standards and supporting assistive technologies.

Customization: Providing flexibility to customize layouts, workflows, and dashboards to meet specific institutional needs and user preferences.

Support and Training: Offering comprehensive resources, tutorials, and responsive support to help users maximize their use of Salesforce features.

Compatibility

Compatibility in Salesforce for education is critical for seamless integration with diverse systems and technologies. It supports interoperability with Learning Management Systems (LMS), Student Information Systems (SIS), and other educational tools through robust APIs. This ensures smooth data exchange, workflow continuity, and enhanced functionality across platforms. Salesforce's cross-platform accessibility guarantees consistent user experience across devices, browsers, and operating systems, fostering usability and accessibility.

By maintaining compatibility with various software versions and technological infrastructures, Salesforce enables educational institutions to optimize data management, streamline operations, and improve overall efficiency in managing educational processes and resources.

Maintainability

Maintainability in Salesforce for education focuses on the system's ability to be easily updated, managed, and sustained over time. This includes structuring the system in a modular way that allows for individual components to be updated without affecting the entire system. Rigorous testing protocols ensure that updates and modifications are thoroughly validated to maintain system reliability and performance standards. By prioritizing maintainability, Salesforce enables educational institutions to adapt to evolving needs and technologies while minimizing disruptions and ensuring long-term operational efficiency.

Monitoring and Logging

Implement monitoring tools to track system performance, user activity, and error rates.
Comprehensive logging for debugging and auditing purposes.

Monitoring: Constantly observes system health, performance metrics, and user interactions in real-time to swiftly identify and address issues.

Logging: Records detailed logs of user activities, system events, and errors for auditing, troubleshooting, and regulatory compliance.

Alerting: Establishes alerts for critical events or performance thresholds, enabling proactive response to maintain system reliability.

Analysis: Utilizes logged data for thorough analysis to optimize system efficiency, detect trends, and implement improvements over time.

Load Testing

Load testing in Salesforce for education verifies system performance under varying user loads and conditions. It assesses response times, throughput, and resource usage to identify potential bottlenecks and optimize scalability. By simulating peak usage scenarios, it ensures system reliability and responsiveness during critical periods like registration and exams.

User Support

Provide a responsive customer support system with defined response times.
Clear communication channels for user inquiries and issue resolution.

Integration with External Systems

Ensure seamless integration with third-party services, such as payment gateways and mapping APIs. Compatibility with various POS (Point of Sale) systems.

Cost:

Cost management in Salesforce for education involves overseeing expenses related to licensing, implementation, maintenance, scaling, and assessing return on investment (ROI).
This includes budgeting for licenses needed by administrators, faculty, and staff, as well as initial setup, customization, and integration costs. Ongoing expenses cover updates, support, and training to ensure system performance, while scalability planning anticipates costs associated with accommodating growth.
Salesforce for education involves overseeing expenses related to licensing, implementation, maintenance, scaling, and assessing return on investment (ROI).
This includes budgeting for licenses needed by administrators, faculty, and staff, as well as initial setup, customization, and integration costs. Ongoing expenses cover updates, support, and training.

CHAPTER 2

FEASIBILITY STUDY

A feasibility study of the KIET Salesforce application in education would examine its potential across several dimensions. This includes evaluating its ability to meet functional requirements such as student record management and course administration, assessing technical feasibility in terms of integration capabilities with existing systems, determining operational impacts on administrative efficiency and data security, and analyzing economic factors like initial costs versus long-term benefits and return on investment (ROI).

Such a comprehensive study ensures that adopting Salesforce at KIET aligns with strategic goals, enhances operational efficiency, and maximizes educational management effectiveness.:

2.1 Technical Feasibility:

The technical feasibility of the KIET Salesforce application involves evaluating its ability to integrate smoothly with existing systems such as Learning Management Systems (LMS) and Student Information Systems (SIS). It also includes assessing whether the current infrastructure can handle the anticipated growth in user numbers, data volumes, and transactional requirements.

Furthermore, customization options are examined to ensure Salesforce can be adapted to meet specific institutional workflows and operational needs effectively. By conducting a thorough technical feasibility assessment, KIET can determine if implementing Salesforce aligns with its technological capabilities and supports enhanced educational management and administrative efficiency.

2.2. Economic Feasibility:

The economic feasibility of implementing the KIET Salesforce application involves assessing its financial viability and benefits comprehensively. This includes evaluating initial costs such as licensing, customization, and integration with existing systems like LMS and SIS.

Ongoing operational expenses for maintenance, support, and training are also considered. The analysis focuses on potential returns on investment (ROI), including savings from improved administrative efficiency, better data management, and enhanced user experience.

A cost-benefit analysis determines whether the anticipated benefits outweigh the initial and ongoing costs, ensuring that the investment aligns with KIET's financial capabilities and strategic educational objectives.

2.3. Legal Feasibility:

Legal feasibility of the KIET Salesforce application entails ensuring compliance with relevant laws and regulations governing data privacy, security, and educational operations. This includes adherence to standards like GDPR, FERPA, and local privacy laws to protect student information. Legal reviews, policies, and measures are implemented to safeguard data, uphold contractual obligations, and mitigate legal risks, ensuring that the application aligns with regulatory requirements and supports KIET's educational mission effectively.

2.4. Operational Feasibility:

User Adoption: Assess whether users are likely to adopt the application based on market research and user feedback.

User Training: Evaluate the ease of use and potential training needs for users, restaurant owners, and delivery personnel.

Scalability: Determine if the application can scale to accommodate growth in user numbers and transactions.

2.5. Scheduling Feasibility:

Project Timeline: Develop a realistic timeline for the development, testing, and deployment.

Dependencies: Identify any dependencies on external factors, such as third-party APIs or regulatory approvals. Milestones: Set clear project milestones to track progress.

2.6. Risk Analysis:

Identify Risks: Identify potential risks, such as technical challenges, market competition, or changes in regulations.

Risk Mitigation Strategies: Develop strategies to mitigate identified risks and minimize their impact.

2.6.7. Market Analysis:

Target Audience: Define the target audience for the application and assess the demand for food ordering services in the target market.

Competitor Analysis: Analyze existing food ordering apps, identify their strengths and weaknesses, and determine how your app can differentiate itself.

8. Environmental Feasibility:

Sustainability: Consider the environmental impact of the application and evaluate whether sustainable practices are feasible and desirable.

Green Technologies: Explore the use of environmentally friendly technologies in the development and operation of the app.

2.9. Conclusion and Recommendations:

Feasibility Assessment: Summarize the findings of the feasibility study, indicating whether the project is feasible.

Recommendations: Provide recommendations for moving forward with the project, modifying the scope, or reconsidering based on the feasibility analysis.

Remember to involve key stakeholders, including developers, business analysts, and potential users, in the feasibility study process to gather diverse perspectives and insights.

Recommendations for moving forward with the project, modifying the scope, or reconsidering based on the feasibility analysis.

Remember to involve key stakeholders, including developers, business analysts, and potential users, in the feasibility study process to gather diverse perspectives and insights.

CHAPTER 3

DATABASE DESIGN

3.1 Waterfall Model

For the KIET Salesforce project, adopting the Waterfall Model entails a structured approach across distinct phases. It begins with comprehensive requirements gathering to define all aspects of the Salesforce application, followed by meticulous system design to outline its architecture and functionality. Implementation involves coding and rigorous testing to ensure adherence to design specifications. Verification includes systematic testing to validate functionality and quality assurance. Deployment and maintenance follow, ensuring ongoing support and updates. While offering clarity and structure, the Waterfall Model may pose challenges in accommodating evolving requirements and stakeholder feedback, necessitating careful planning to mitigate risks and ensure project success.

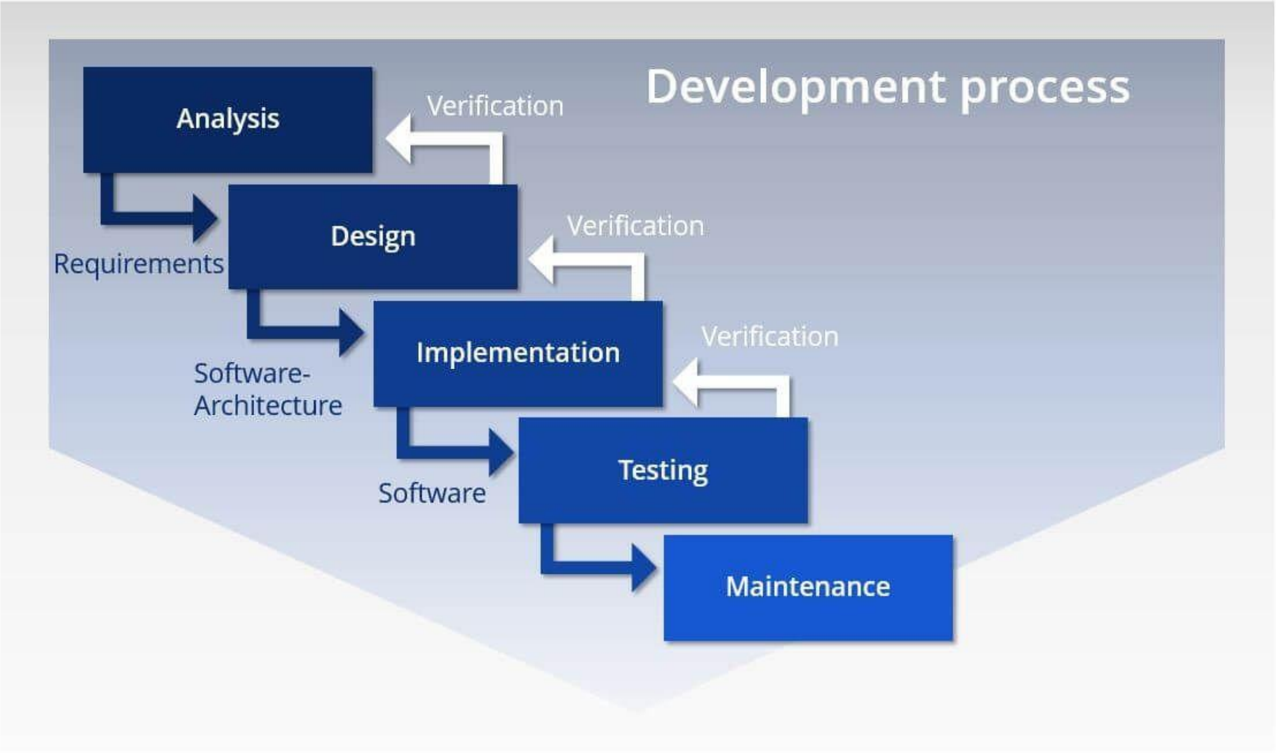


Figure 3.1. Waterfall Model

3.2 Requirements Gathering & Analysis

Requirements gathering and analysis for the KIET Salesforce project involve several key activities. Firstly, engaging stakeholders—such as administrators, faculty, and staff—to understand their needs and expectations from the Salesforce application. Secondly, documenting both functional and non-functional requirements, including user stories, use cases, and system specifications. Thirdly, analyzing these requirements meticulously to ensure clarity, feasibility, and alignment with institutional goals.

Fourthly, validating requirements with stakeholders to confirm understanding and consensus on priorities and scope. Finally, documenting finalized requirements in detail to guide subsequent phases of design, development, and implementation effectively. These steps are crucial to ensuring the Salesforce solution meets KIET's operational needs effectively.

Conduct Stakeholder Interviews:

Conducting stakeholder interviews for the KIET Salesforce project is a structured process critical to understanding and documenting user requirements. It involves identifying key stakeholders—such as administrators, faculty, staff, and possibly students—relevant to the Salesforce application.

Preparation includes developing thoughtful interview questions aimed at uncovering their needs, challenges, and expectations. Interviews are scheduled and conducted with careful attention to gathering detailed insights and clarifying responses. Analyzing the gathered feedback helps identify common priorities and potential conflicts among stakeholders. Reporting findings comprehensively guides the requirements gathering phase, ensuring the Salesforce solution meets KIET's operational and strategic objectives effectively.

Document Functional Requirements:

Student Record Management
Course Administration
Communication Tools
Integration with Existing Systems
Payment processing integration
Reviews and ratings
Notifications

Document Non-Functional Requirements:

Performance metrics
Security measures
Usability and accessibility standards
Compatibility with Android devices

Prioritize Requirements:

Student Record Management
User Registration and Authentication
Course Administration -
Security
Integration with Existing Systems

Prototyping:

Prototyping is an essential phase in the development of the KIET Salesforce project. It involves creating a preliminary version of the application to visualize and test key functionalities before full-scale development. Here’s a detailed approach to prototyping for this project:

Identify Core Features:

Focus on the high-priority functional requirements identified during the requirements gathering phase.

Core features to prototype might include Student Record Management, Course Administration, User Registration and Authentication, and Reporting and Analytics**.**

Define Objectives:

Clearly outline the goals of the prototype. For instance, it could be to validate user interface design, gather user feedback, or test technical feasibility.

Create Wireframes:

Develop wireframes for the main user interfaces. These should be simple, focusing on layout and navigation rather than detailed design.

Requirements Validation:

Share requirements and prototypes for validation.

Review and Approval:

Conduct formal reviews and obtain stakeholder approval.

Establish a process for handling changes to requirements.

Documentation:

Documentation for the KIET Salesforce project plays a crucial role in ensuring clarity, consistency, and effective management throughout its lifecycle. This includes documenting detailed functional and non-functional requirements gathered from stakeholders, architectural and technical designs for system implementation, and user guides/manuals for end-users.

Additionally, comprehensive testing plans and results, along with change management procedures, are documented to maintain system integrity and manage updates. Training materials are also prepared to facilitate stakeholder understanding and adoption of the Salesforce application.

Well-documented information supports smooth implementation, efficient operations, and ongoing support of the Salesforce solution tailored to KIET's educational management needs.

3.2 ER Diagram

Creating an Entity-Relationship (ER) diagram for the KIET Salesforce project involves identifying and describing the primary entities, their attributes, and the relationships between them. Here’s how the ER diagram might be conceptualized in words:

A) Entities:

Student:

Attributes: StudentID (Primary Key), Name, Address, Email, Phone, DateOfBirth, EnrollmentDate.

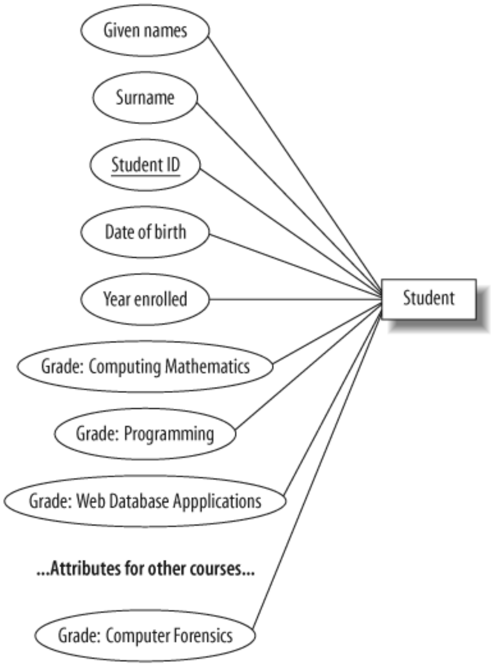
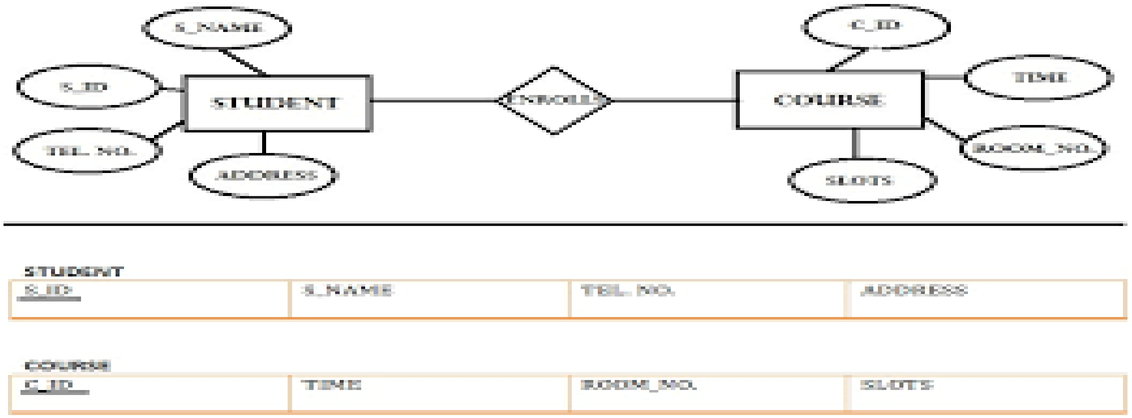

Given names
Surname
Student ID
Date of birth
Year enrolled
Grade: Computing Mathematics
Grade: Programming
Grade: Web Database Appplications
...Attributes for other courses...
Grade: Computer Forensics
Student

Figure A: **Student Entity**

Course:

Attributes: CourseID (Primary Key), CourseName, Description, Credits, Department, StartDate, EndDate.



STUDENT

S_ID	S_NAME	TEL. NO.	ADDRESS

COURSE

C_ID	TIME	ROOM_NO.	SLOTS

Figure B: Course Entity

Faculty:
Attributes: FacultyID (Primary Key), Name, Department, Email, Phone, OfficeLocation.

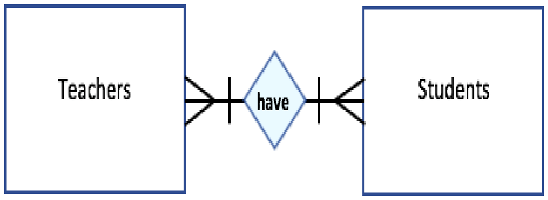


Figure C: Faculty Entity

Enrollment
Attributes: EnrollmentID (Primary Key), StudentID (Foreign Key), CourseID (Foreign Key), EnrollmentDate, Grade.

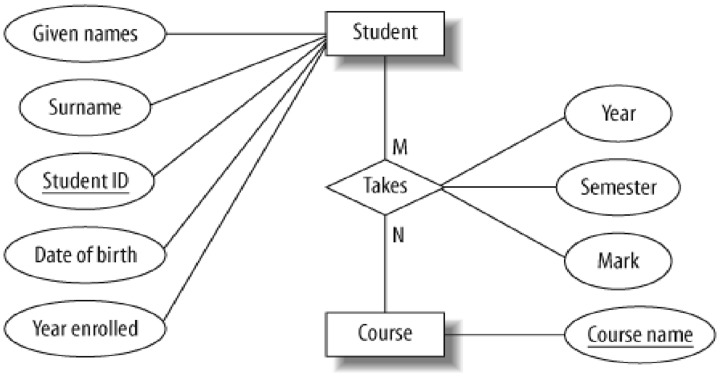


Figure C: Enrollment Entity

B). Student enrolls in Course:

Relationship Type: Many-to-Many
Description: A student can enroll in multiple courses, and a course can have multiple students enrolled. This relationship is facilitated by the Enrollment entity, which links Students and Courses

Faculty teaches Course:

Relationship Type: Many-to-Many
Description: A faculty member can teach multiple courses, and a course can be taught by multiple faculty members. This relationship is often facilitated by an intermediary entity, but for simplicity, it can be assumed directly.

Cardinality:

Student - Enrollment - Course:

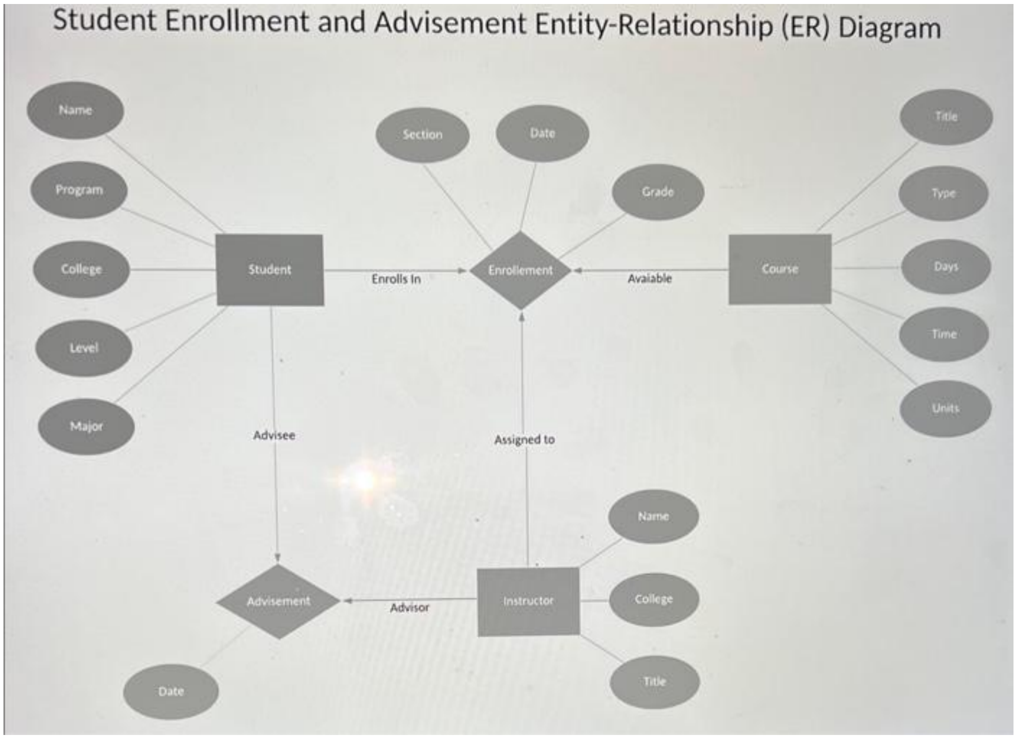


Figure D: **Student - Enrolment - Course**

Student to Enrollment: One-to-Many (A student can have multiple enrollments).

Course to Enrollment: One-to-Many (A course can have multiple enrollments).

Faculty - Course:
Faculty to Course: Many-to-Many (Each faculty member can teach multiple courses, and each course can be taught by multiple faculty members).

Example Relationships:
A **Student** (identified by StudentID) can enroll in multiple Courses (each identified by CourseID). Each enrollment instance is tracked in the Enrollment entity with additional details like EnrollmentDate and Grade.

A **Faculty** member (identified by FacultyID) can be associated with multiple Courses. Conversely, each Course can be taught by multiple faculty members, ensuring flexibility in course management.

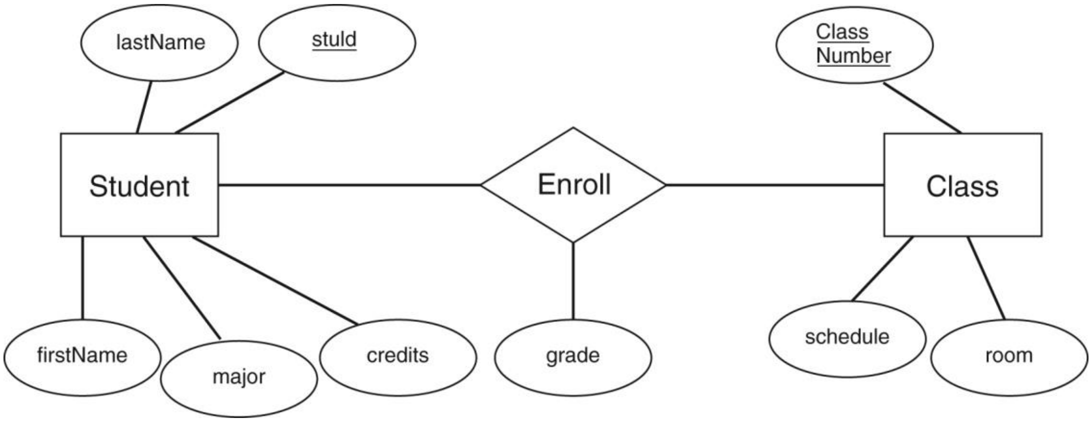


Figure 3.2 ER Diagram of KIET Salesforce Application

3.3 Use Case Diagram

A Use Case Diagram is a visual representation used in software engineering to depict the functional requirements of a system. It illustrates how different users (actors) interact with the system to achieve specific goals (use cases). Here’s an overview of the theory behind Use Case Diagrams:

A) Actors:

Represent entities that interact with the system. Actors can be users, other systems, or hardware devices.
Actors are typically depicted as stick figures.
Example: Students, Faculty, Administrators, External Systems.

B) Use Cases:
Represent functions or services provided by the system to the actors.
Use cases are depicted as ovals.
Example: Enroll in Course, Grade Assignment, Generate Reports, Manage Student Records.

C). Use Cases:
Browse Restaurants:
The "Browse Courses" use case allows students to view the available courses in the KIET Salesforce application. Students can search for courses by various criteria such as department, instructor, or schedule, and access detailed information about each course, including descriptions, prerequisites, and enrollment options.

The system allows for filtering and searching based on various criteria such as department, course level, instructor, schedule, and availability. Each course listing includes detailed information such as course descriptions, prerequisites, credit hours, and instructor profiles.

Additionally, students can view course ratings and reviews from their peers, helping them make informed decisions about which courses to enroll in. The browsing experience is designed to be user-friendly, ensuring that students can quickly and efficiently find the courses that best fit their academic plans

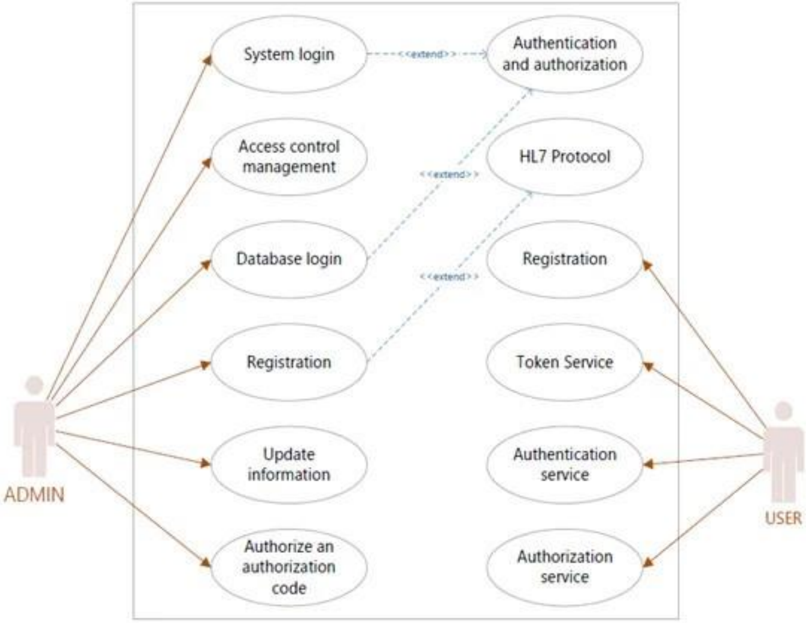


Figure 3.3 Use CASE Diagram

3.5 Activity Diagram

An activity diagram for a food ordering system provides a visual representation of the workflow and interactions between different activities or actions within the system. Below is a simplified example of an activity diagram for a food ordering system:
Explanation:

A). Start:
The starting point of the activity diagram.

B). Select Restaurant:
Users select a restaurant from the available options.

C). Browse Menu:
Users browse the menu of the selected restaurant.

D). Add Items to Cart:
Users add food items to their shopping cart.

E). View Cart:
Users review the items in their cart.

F). Place Order:
Users initiate the order placement.

G). Confirm Order:
Confirmation step to verify the order details.

Payment Process:
Users go through the payment process for the order.
Notify Restaurant:

The system notifies the selected restaurant about the new order.

H). Track Order:
Users can track the status of their order.

End:
The end point of the activity diagram.
This activity diagram provides an overview of the major steps involved in the food ordering process. Depending on the complexity of your system, you may need to include additional details, decision points, or parallel activities in the diagram. The goal is to represent the flow of activities in a clear and understandable manner.

3.5.1 State Diagram

A State Diagram visually represents the states and transitions of an object within a system. For the KIET Salesforce application, it could illustrate a student's course registration process: starting from "Logged Out," transitioning to "Logged In," then "Browsing Courses," moving to "Registering for Courses," and finally to "Registration Complete." Each transition is triggered by user actions or system events:

A). Start State:

Description: Initial state when the application is launched.
Transitions: Move to the "Login" state if the user needs to log in or to the "Menu" state if the user is already logged in.
Login State:

B). Description: User is required to log in.

Transitions:
Successful login transitions to the "Menu" state.
Unsuccessful login may return to the "Start" state or remain in the "Login" state.
Menu State:

C). Transitions:

Selecting a restaurant transition to the "Restaurant Details" state.
Choosing a food item transitions to the "Order Details" state.
Going to the "Cart" state to review and modify the order.
Logout transitions to the "Start" state.
Restaurant Details State:

D). Description: Display details about the selected restaurant.

Transitions:
Going back to the "Menu" state.
Selecting a food item transitions to the "Course Details" state.
Course Details State:

Description: Display details about the selected Course item.
Transitions:
Adding the Course to the cart transitions to the "Cart" state.
Going back to the "Menu" state.

E). Description: Finalize the order and provide payment details.

Transitions:
Successful payment transitions to the "Course Confirmation" state.
Cancelling the Course returns to the "Cart" state.
Order Confirmation State:

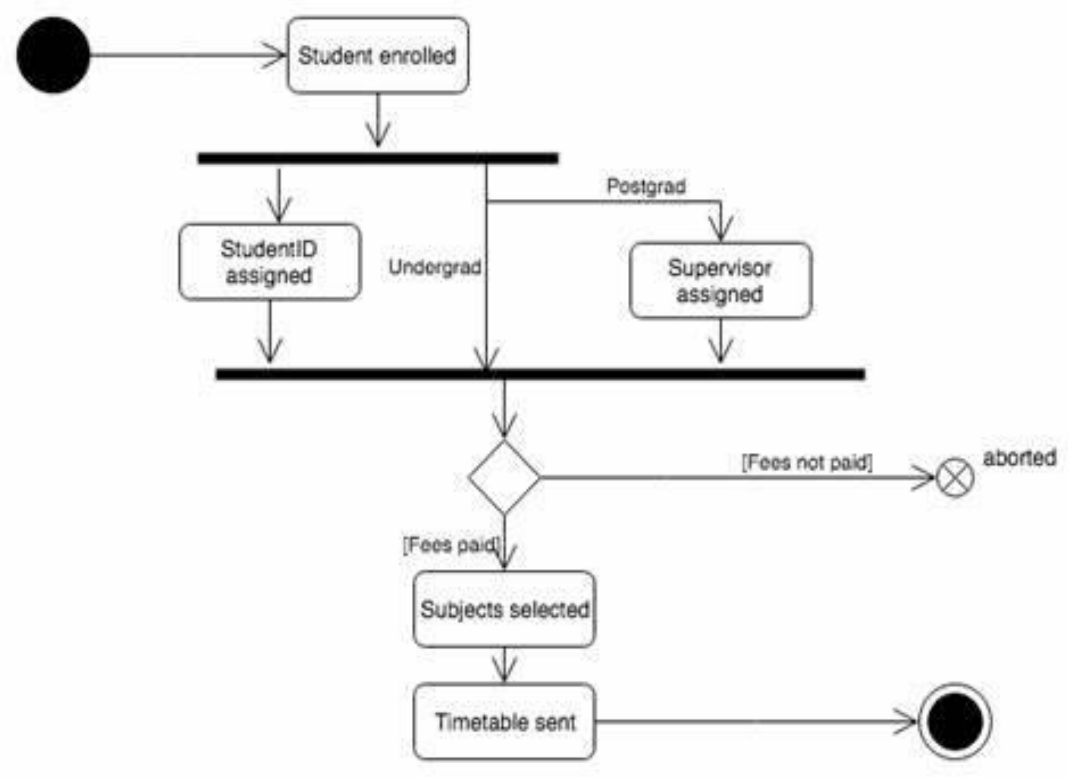


Figure 3.4 State Diagram

3.6 Component Diagram

A Component Diagram is a type of UML diagram that depicts how components within a system are organized and how they interact with each other. It provides a high-level view of the system's architecture, focusing on the components and their interfaces.

In the KIET Salesforce application, a Component Diagram might include components such as the User Interface (UI), Business Logic Layer, Data Access Layer, Authentication Service, and Notification Service. The UI component interacts with users, allowing them to browse courses, register, and submit assignments. The Business Logic Layer handles the core application logic, processing requests from the UI and interacting with the Data Access Layer to retrieve or update information in the database. The Authentication Service ensures secure login and access control, while the Notification Service manages alerts and communications to users.

Each component is represented as a rectangular box, with interfaces shown as small circles or lines connecting the components, indicating the flow of data and control between them. This diagram helps developers and stakeholders understand the system's structure and how different parts of the application work together.

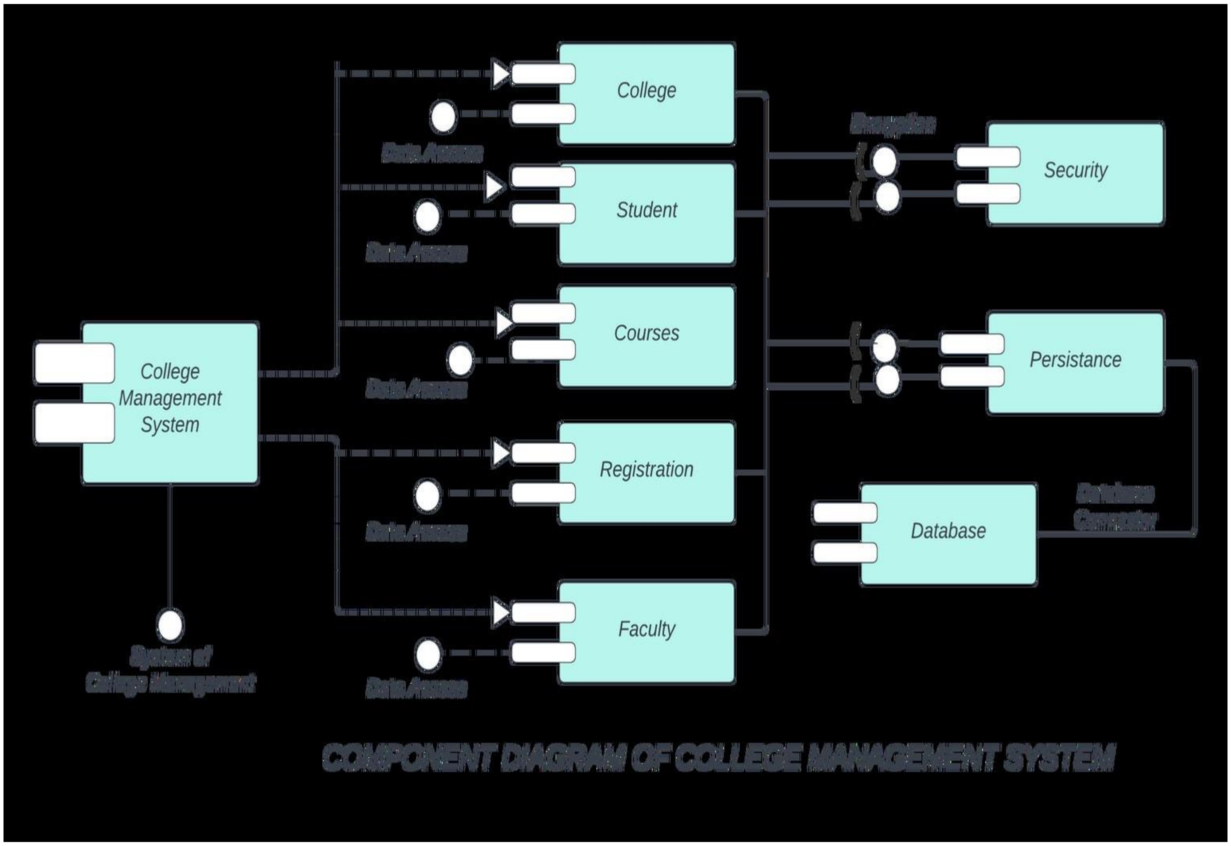
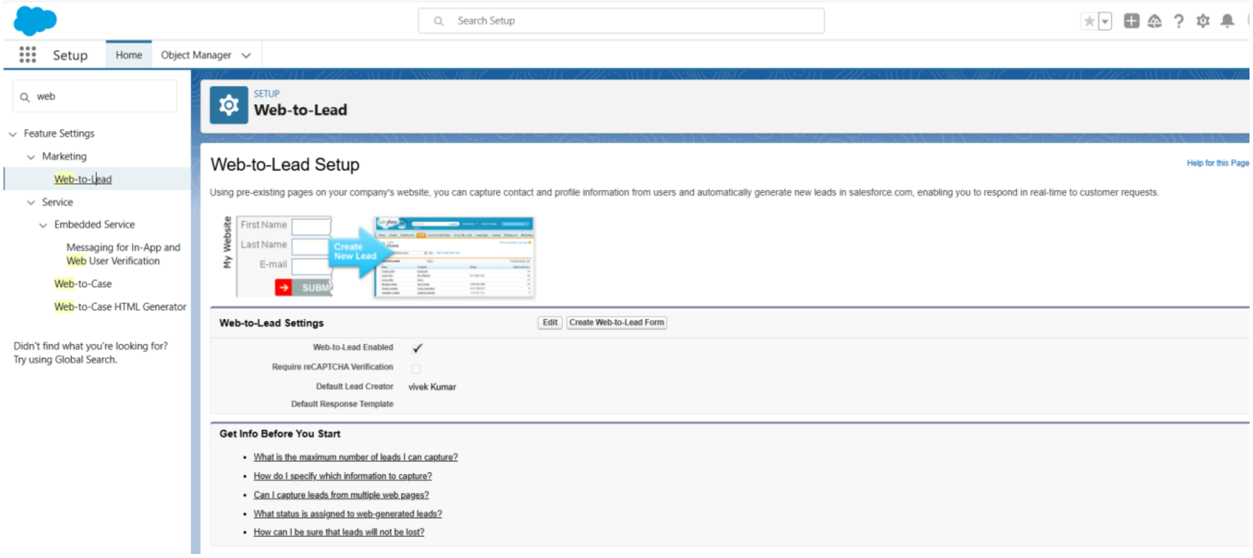


Figure 3.5 Component diagram for Firebase backend

LEAD GENERATION



Web-to-Lead Setup

Using pre-existing pages on your company's website, you can capture contact and profile information from users and automatically generate new leads in salesforce.com, enabling you to respond in real-time to customer requests.

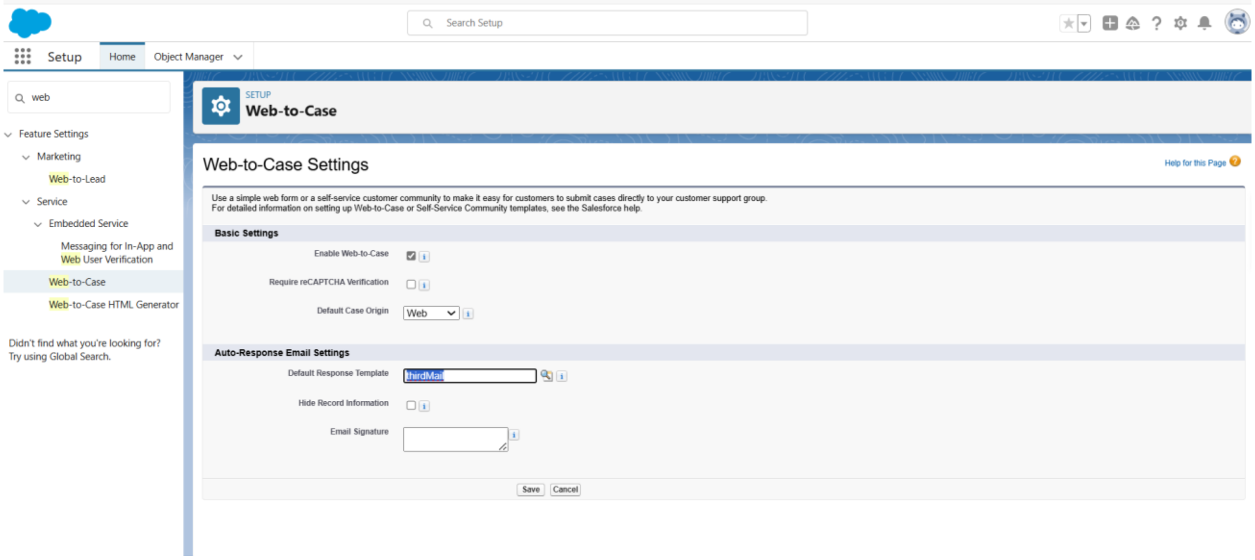
Web-to-Lead Settings

Web-to-Lead Enabled	✓
Require reCAPTCHA Verification	□
Default Lead Creator	vivek Kumar
Default Response Template	

Get Info Before You Start

- What is the maximum number of leads I can capture?
- How do I specify which information to capture?
- Can I capture leads from multiple web pages?
- What status is assigned to web-generated leads?
- How can I be sure that leads will not be lost?

Fig : Web To Lead



Web-to-Case Settings

Use a simple web form or a self-service customer community to make it easy for customers to submit cases directly to your customer support group.
For detailed information on setting up Web-to-Case or Self-Service Community templates, see the Salesforce help.

Basic Settings

Enable Web-to-Case	☑
Require reCAPTCHA Verification	□
Default Case Origin	Web

Auto-Response Email Settings

Default Response Template	thirdMail
Hide Record Information	□
Email Signature	

Fig : Web To Case

SCREEN SHOTS

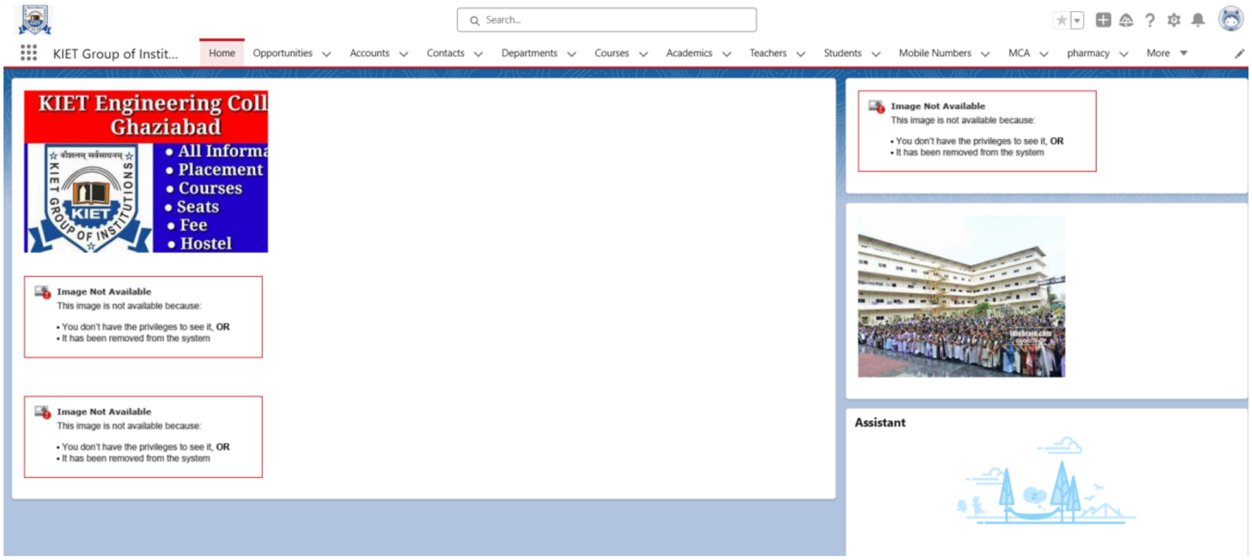


Image Not Available
This image is not available because:
• You don't have the privileges to see it, OR
• It has been removed from the system

Image Not Available
This image is not available because:
• You don't have the privileges to see it, OR
• It has been removed from the system

Image Not Available
This image is not available because:
• You don't have the privileges to see it, OR
• It has been removed from the system

Figure 1: Home Page

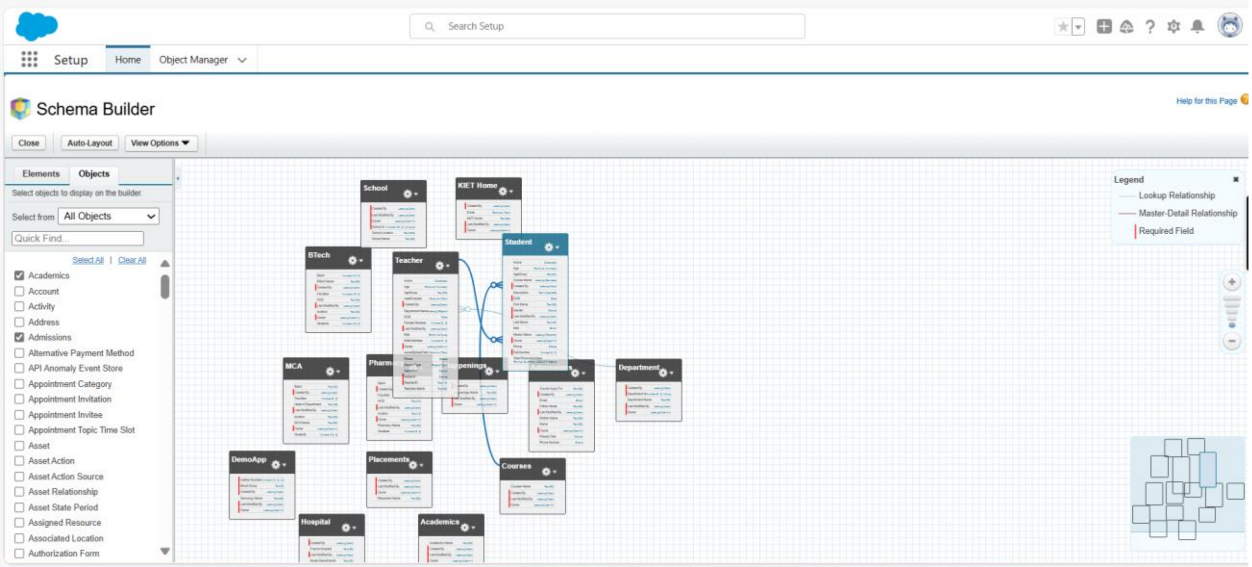
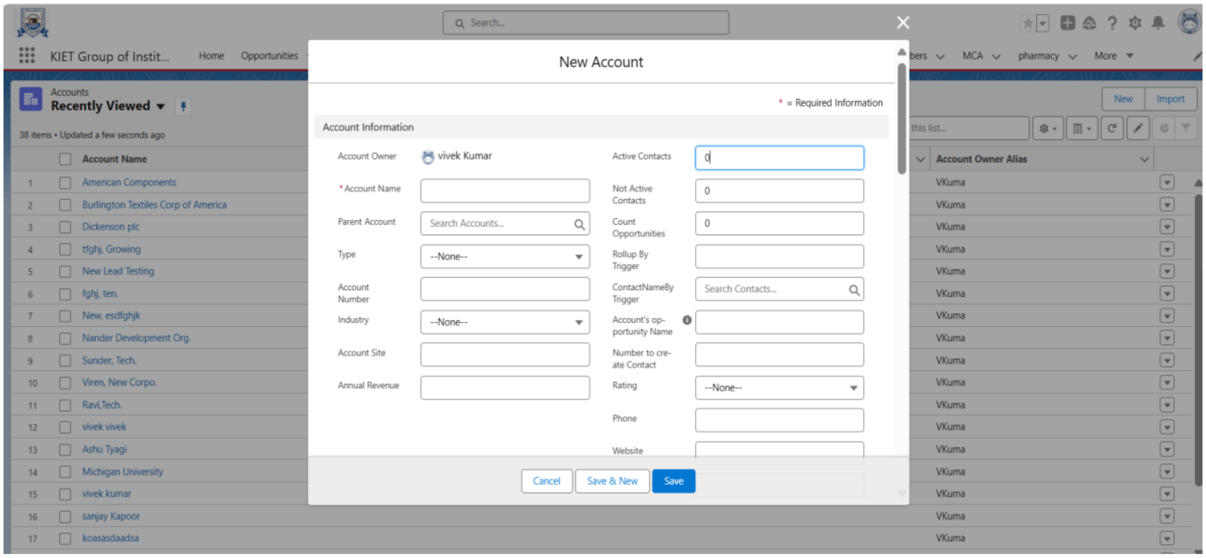


Fig 1b : Schema Builder



Accounts
Recently Viewed
38 items • Updated a few seconds ago

	Account Name	Account Owner Alias
1	American Components	VKuma
2	Burlington Textiles Corp of America	VKuma
3	Dickenson plc	VKuma
4	tfghj, Growing	VKuma
5	New Lead Testing	VKuma
6	fghj, ten.	VKuma
7	New, esdfghjk	VKuma
8	Nander Development Org.	VKuma
9	Sunder, Tech.	VKuma
10	Viren, New Corpo.	VKuma
11	Ravi,Tech.	VKuma
12	vivek vivek	VKuma
13	Ashu Tyagi	VKuma
14	Michigan University	VKuma
15	vivek kumar	VKuma
16	sanjay Kapoor	VKuma
17	koasasdaadsa	VKuma

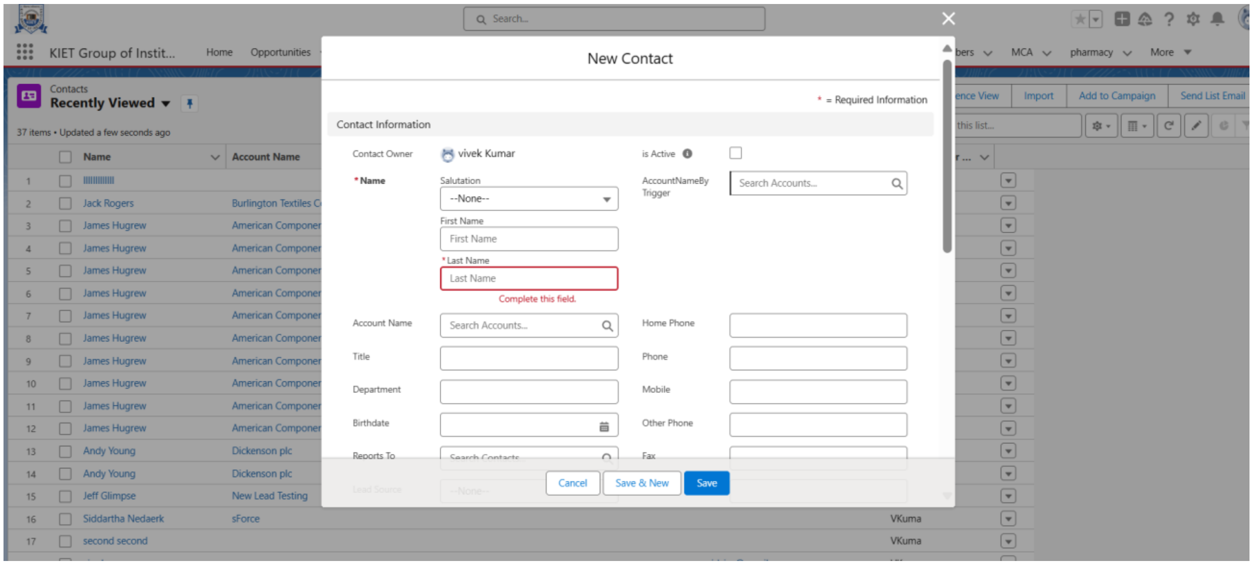
New Account

* = Required Information

Account Information

Account Owner	vivek Kumar	Active Contacts	0
*Account Name		Not Active Contacts	0
Parent Account	Search Accounts...	Count Opportunities	0
Type	--None--	Rollup By Trigger	
Account Number		ContactNameBy Trigger	Search Contacts...
Industry	--None--	Account's opportunity Name	
Account Site		Number to create Contact	
Annual Revenue		Rating	--None--
		Phone	
		Website	

Fig 2: Account Page



Contacts
Recently Viewed

37 items • Updated a few seconds ago

	Name	Account Name	r ...
1	IIIIIIIIIII		
2	Jack Rogers	Burlington Textiles C	
3	James Hugrew	American Componer	
4	James Hugrew	American Componer	
5	James Hugrew	American Componer	
6	James Hugrew	American Componer	
7	James Hugrew	American Componer	
8	James Hugrew	American Componer	
9	James Hugrew	American Componer	
10	James Hugrew	American Componer	
11	James Hugrew	American Componer	
12	James Hugrew	American Componer	
13	Andy Young	Dickenson plc	
14	Andy Young	Dickenson plc	
15	Jeff Glimpse	New Lead Testing	
16	Siddartha Nedaerk	sForce	VKuma
17	second second		VKuma

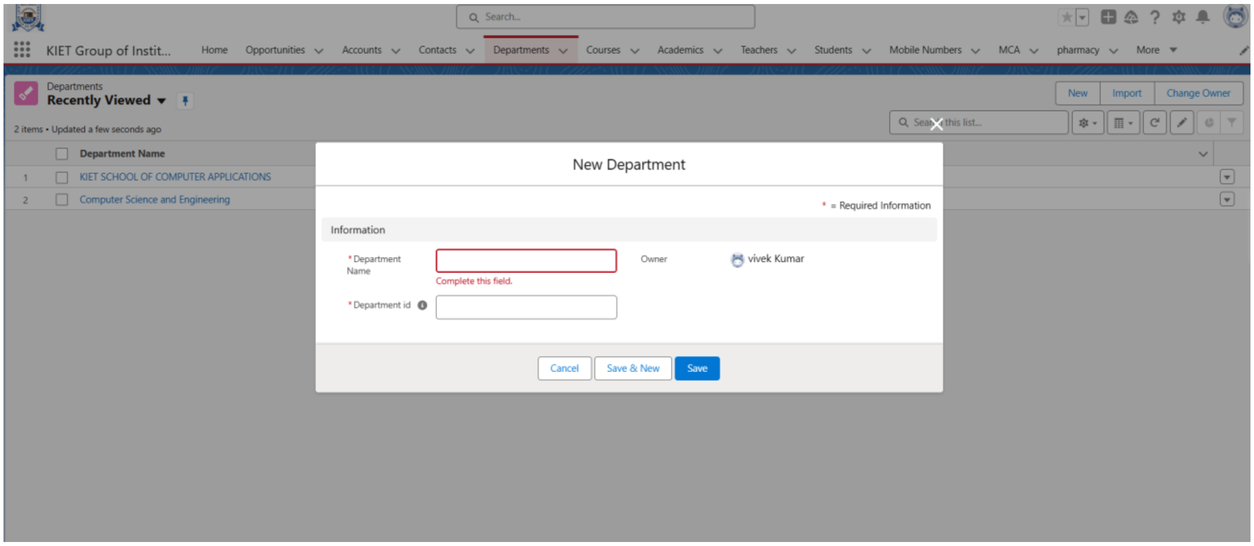
New Contact

* = Required Information

Contact Information

Contact Owner	vivek Kumar	is Active	□
* Name	Salutation: --None-- First Name: First Name * Last Name: Last Name Complete this field.	AccountNameBy Trigger	Search Accounts...
Account Name	Search Accounts...	Home Phone	
Title		Phone	
Department		Mobile	
Birthdate		Other Phone	
Reports To	Search Contacts...	Fax	
Lead Source	--None--		

Figure 3: Contacts page



Departments
Recently Viewed

2 items • Updated a few seconds ago

	Department Name
1	KIET SCHOOL OF COMPUTER APPLICATIONS
2	Computer Science and Engineering

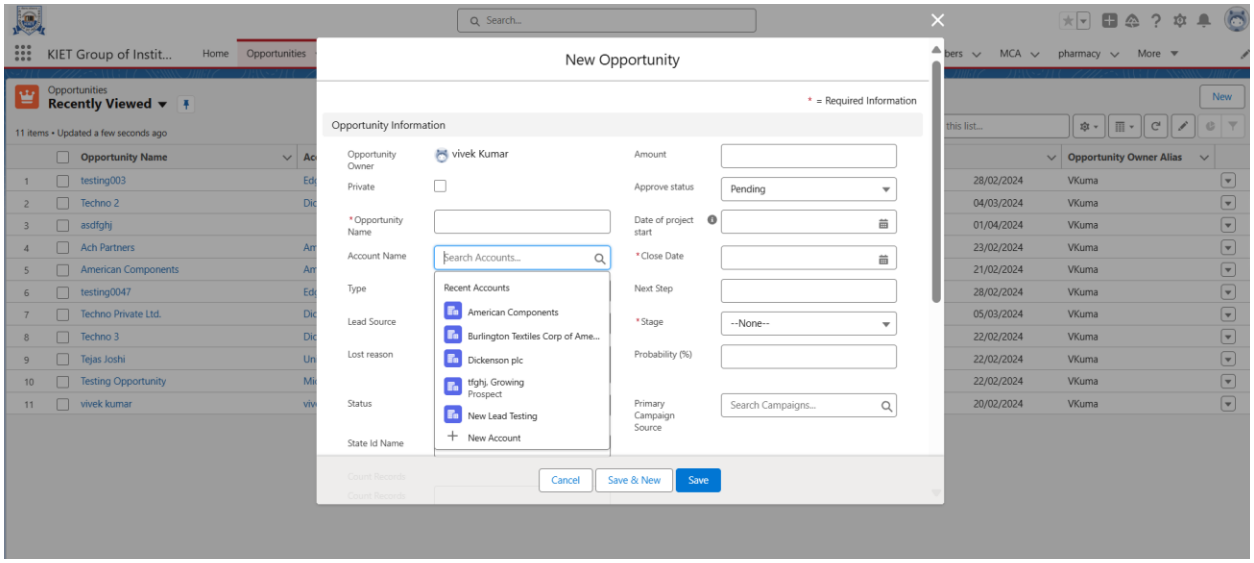
New Department

* = Required Information

Information

* Department Name	Complete this field.	Owner	vivek Kumar
* Department id			

Figure 4: Department page



KIET Group of Instit...

Opportunities
Recently Viewed

11 items • Updated a few seconds ago

	Opportunity Name	Ac		Opportunity Owner Alias
1	testing003	Ed	28/02/2024	VKuma
2	Techno 2	Dic	04/03/2024	VKuma
3	asdfghj		01/04/2024	VKuma
4	Ach Partners	Am	23/02/2024	VKuma
5	American Components	Am	21/02/2024	VKuma
6	testing0047	Ed	28/02/2024	VKuma
7	Techno Private Ltd.	Dic	05/03/2024	VKuma
8	Techno 3	Dic	22/02/2024	VKuma
9	Tejas Joshi	Un	22/02/2024	VKuma
10	Testing Opportunity	Mi	22/02/2024	VKuma
11	vivek kumar	viv	20/02/2024	VKuma

New Opportunity

* = Required Information

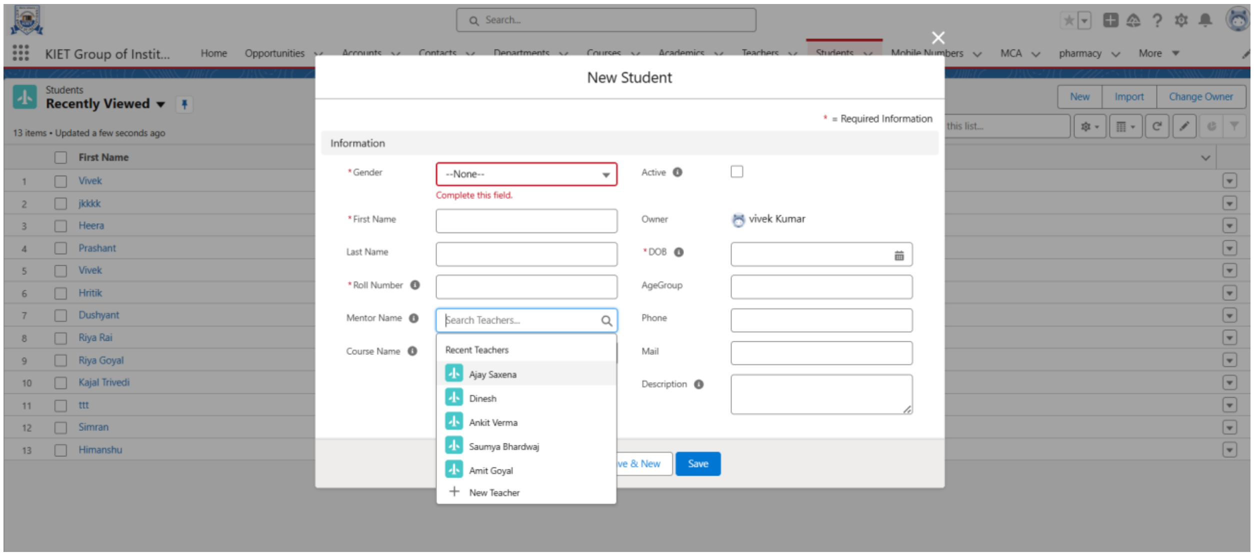
Opportunity Information

Opportunity Owner: vivek Kumar
Private: □
*Opportunity Name:
Account Name: Search Accounts...
Type:
Lead Source:
Lost reason:
Status:
State Id Name:

Recent Accounts
- American Components
- Burlington Textiles Corp of Ame...
- Dickenson plc
- tfghj, Growing Prospect
- New Lead Testing
- New Account

Amount:
Approve status: Pending
Date of project start:
*Close Date:
Next Step:
*Stage: --None--
Probability (%):
Primary Campaign Source: Search Campaigns...

Figure 5: Opportunity page



KIET Group of Instit...

Students
Recently Viewed

13 items • Updated a few seconds ago

	First Name
1	Vivek
2	jkkkk
3	Heera
4	Prashant
5	Vivek
6	Hritik
7	Dushyant
8	Riya Rai
9	Riya Goyal
10	Kajal Trivedi
11	ttt
12	Simran
13	Himanshu

New Student

* = Required Information

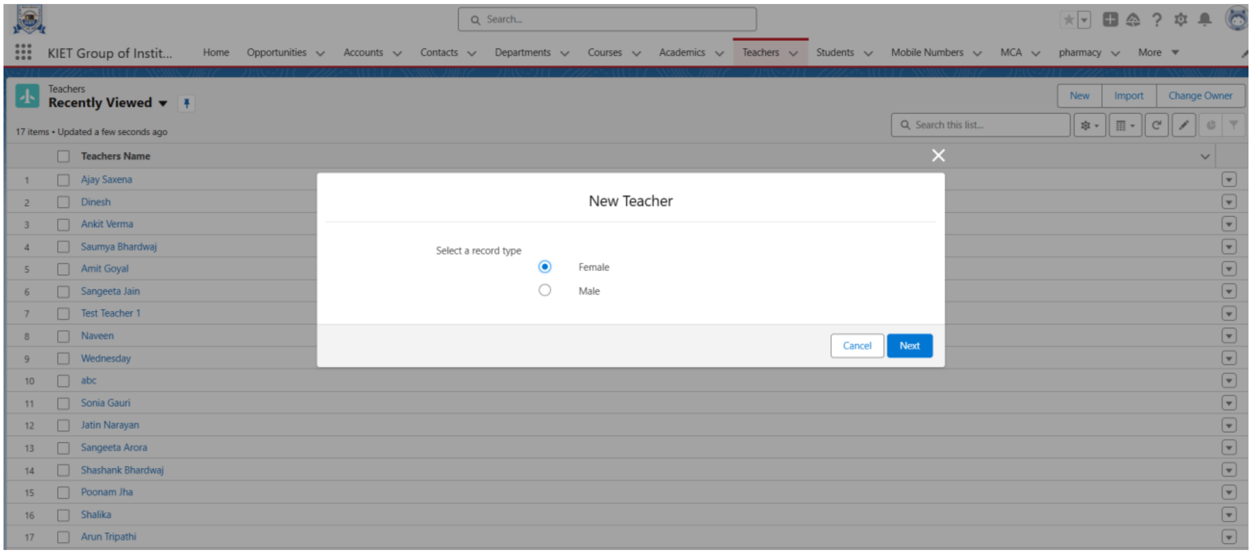
Information

*Gender: --None--
Complete this field.
*First Name:
Last Name:
*Roll Number:
Mentor Name: Search Teachers...
Course Name:

Recent Teachers
- Ajay Saxena
- Dinesh
- Ankit Verma
- Saumya Bhardwaj
- Amit Goyal
- New Teacher

Active: □
Owner: vivek Kumar
*DOB:
AgeGroup:
Phone:
Mail:
Description:

Figure 6: Student page



Teachers
Recently Viewed

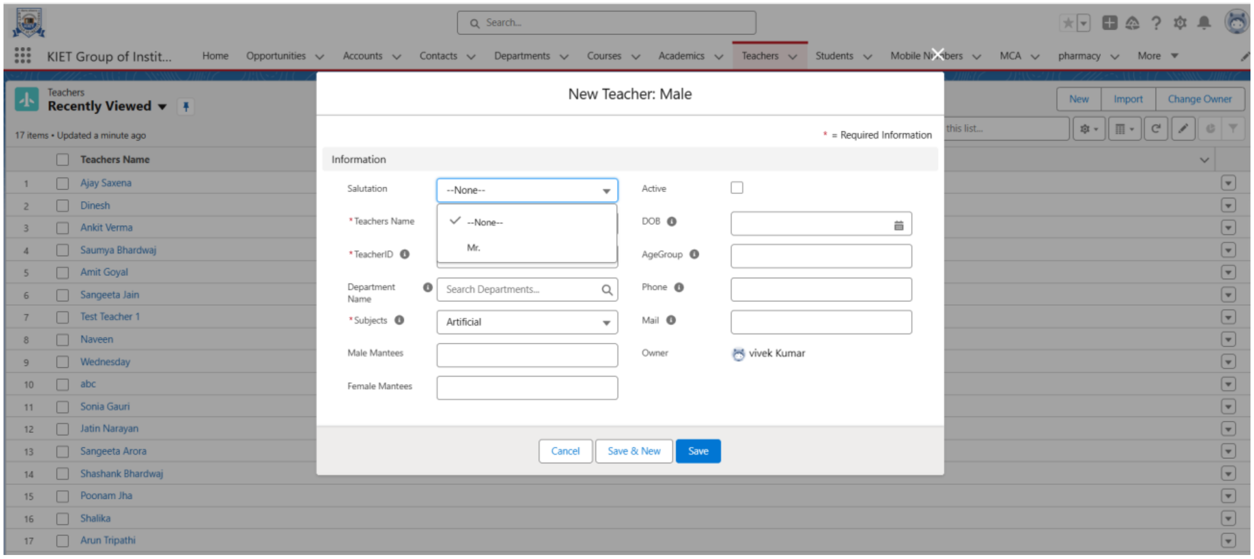
17 items • Updated a few seconds ago

	Teachers Name
1	Ajay Saxena
2	Dinesh
3	Ankit Verma
4	Saumya Bhardwaj
5	Amit Goyal
6	Sangeeta Jain
7	Test Teacher 1
8	Naveen
9	Wednesday
10	abc
11	Sonia Gauri
12	Jatin Narayan
13	Sangeeta Arora
14	Shashank Bhardwaj
15	Poonam Jha
16	Shalika
17	Arun Tripathi

New Teacher

Select a record type
● Female
○ Male

Figure 7a: New Teacher Record Type



Teachers
Recently Viewed

17 items • Updated a minute ago

	Teachers Name
1	Ajay Saxena
2	Dinesh
3	Ankit Verma
4	Saumya Bhardwaj
5	Amit Goyal
6	Sangeeta Jain
7	Test Teacher 1
8	Naveen
9	Wednesday
10	abc
11	Sonia Gauri
12	Jatin Narayan
13	Sangeeta Arora
14	Shashank Bhardwaj
15	Poonam Jha
16	Shalika
17	Arun Tripathi

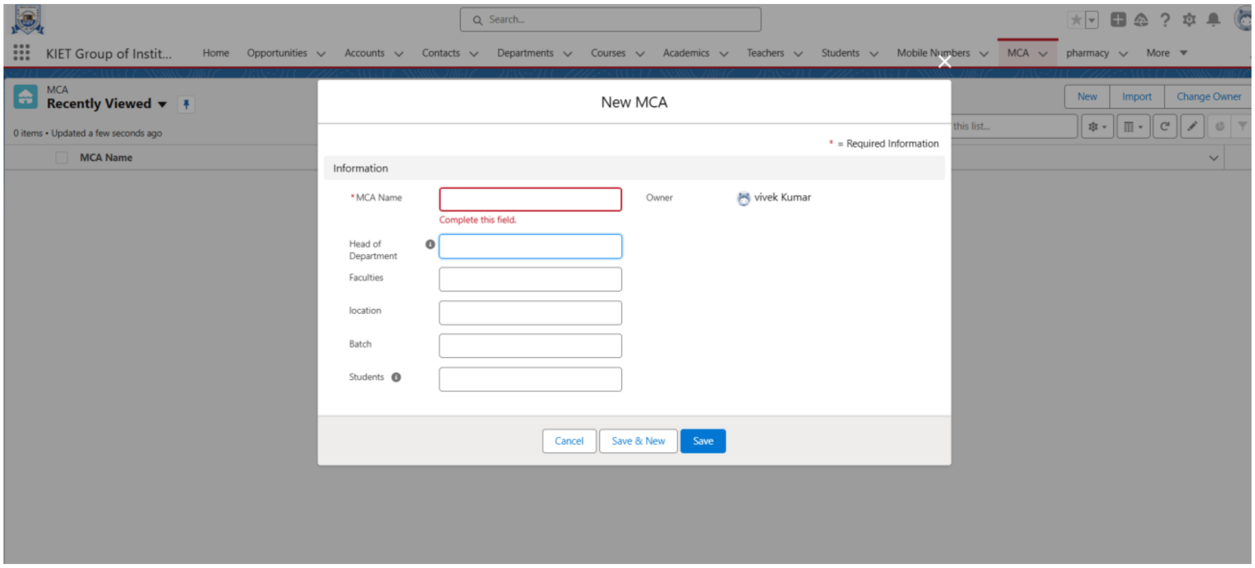
New Teacher: Male

* = Required Information

Information

Salutation	--None-- ✓ --None-- Mr.	Active	□
*Teachers Name		DOB	
*TeacherID		AgeGroup	
Department Name	Search Departments...	Phone	
*Subjects	Artificial	Mail	
Male Mantees		Owner	vivek Kumar
Female Mantees			

Fig 7b : New Teacher



New MCA

* = Required Information

Information

*MCA Name:

Complete this field.

Head of Department:

Faculties:

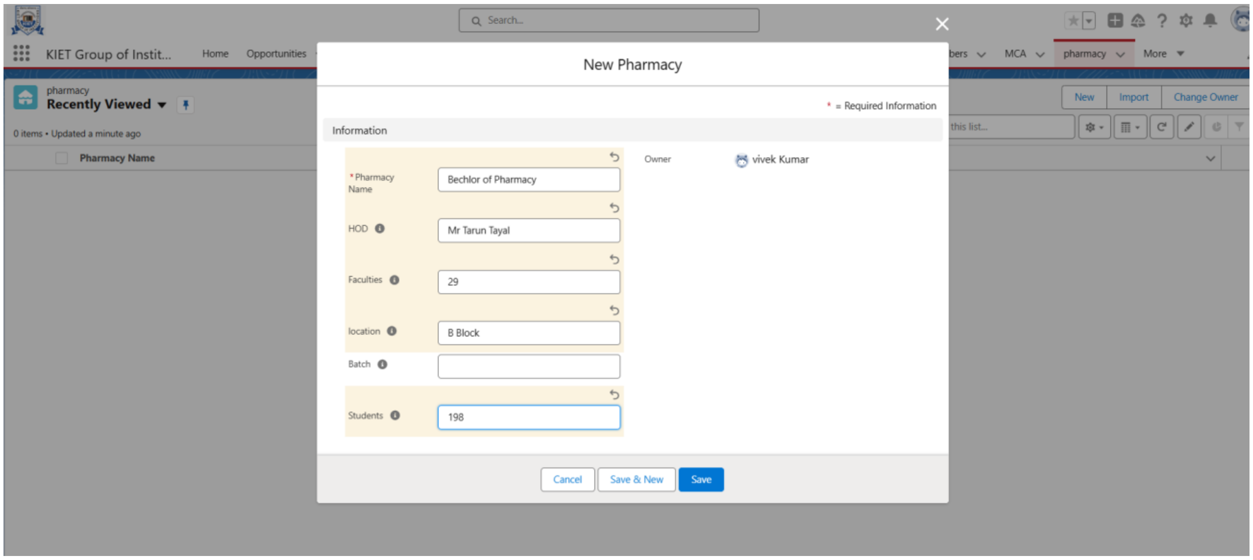
location:

Batch:

Students:

Owner: vivek Kumar

Fig 8: MCA Department



New Pharmacy

* = Required Information

Information

*Pharmacy Name: Bechlor of Pharmacy

HOD: Mr Tarun Tayal

Faculties: 29

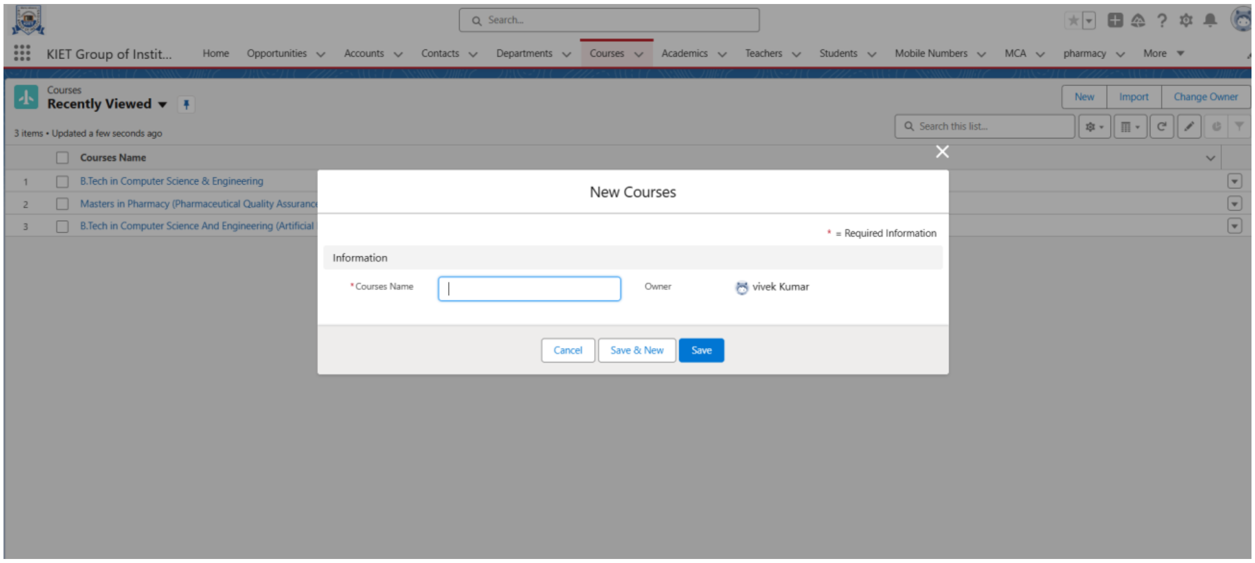
location: B Block

Batch:

Students: 198

Owner: vivek Kumar

Fig 9: Pharmacy Department



Courses

Recently Viewed

3 items • Updated a few seconds ago

	Courses Name
1	B.Tech in Computer Science & Engineering
2	Masters in Pharmacy (Pharmaceutical Quality Assuranc
3	B.Tech in Computer Science And Engineering (Artificial

New Courses

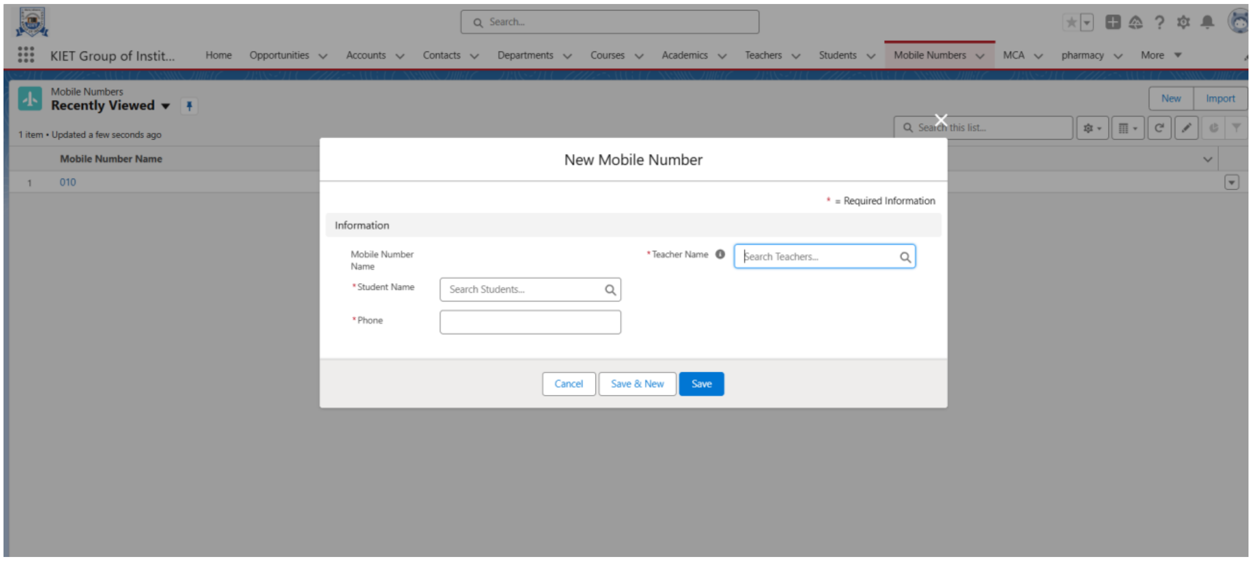
* = Required Information

Information

*Courses Name

Owner: vivek Kumar

Figure 10: Add New Course



Mobile Numbers

Recently Viewed

1 item • Updated a few seconds ago

	Mobile Number Name
1	010

New Mobile Number

* = Required Information

Information

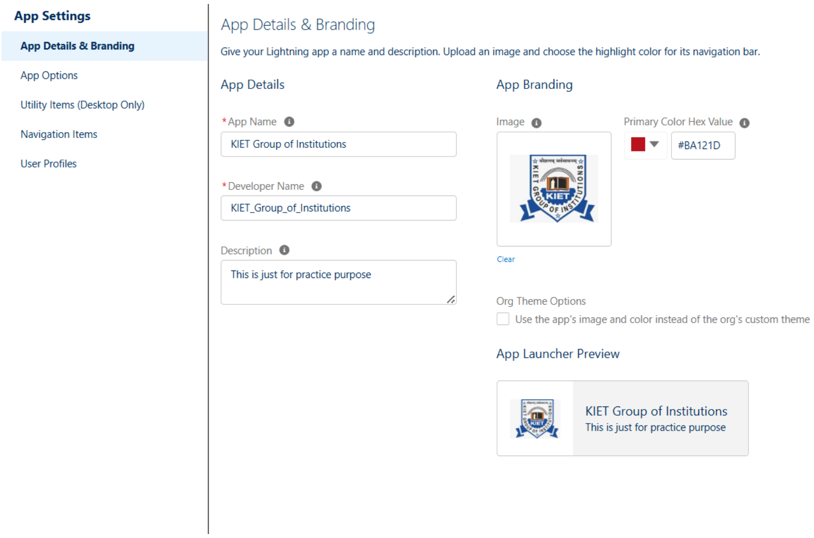
Mobile Number Name

*Teacher Name: Search Teachers...

*Student Name: Search Students...

*Phone

Figure 11: Junction Object(Many to Many Relation)



App Settings

- App Details & Branding
- App Options
- Utility Items (Desktop Only)
- Navigation Items
- User Profiles

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

*App Name: KIET Group of Institutions

*Developer Name: KIET_Group_of_Institutions

Description: This is just for practice purpose

App Branding

Image

Clear

Primary Color Hex Value: #BA121D

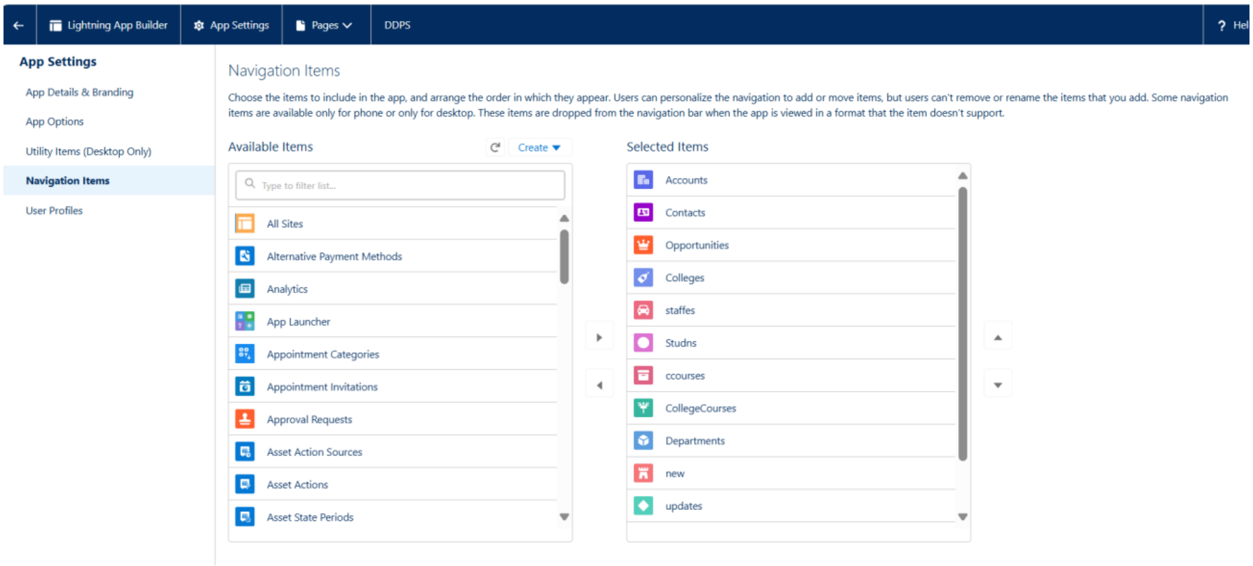
Org Theme Options

□ Use the app's image and color instead of the org's custom theme

App Launcher Preview

KIET Group of Institutions
This is just for practice purpose

Figure 12: Application Details



App Settings

- App Details & Branding
- App Options
- Utility Items (Desktop Only)
- Navigation Items
- User Profiles

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

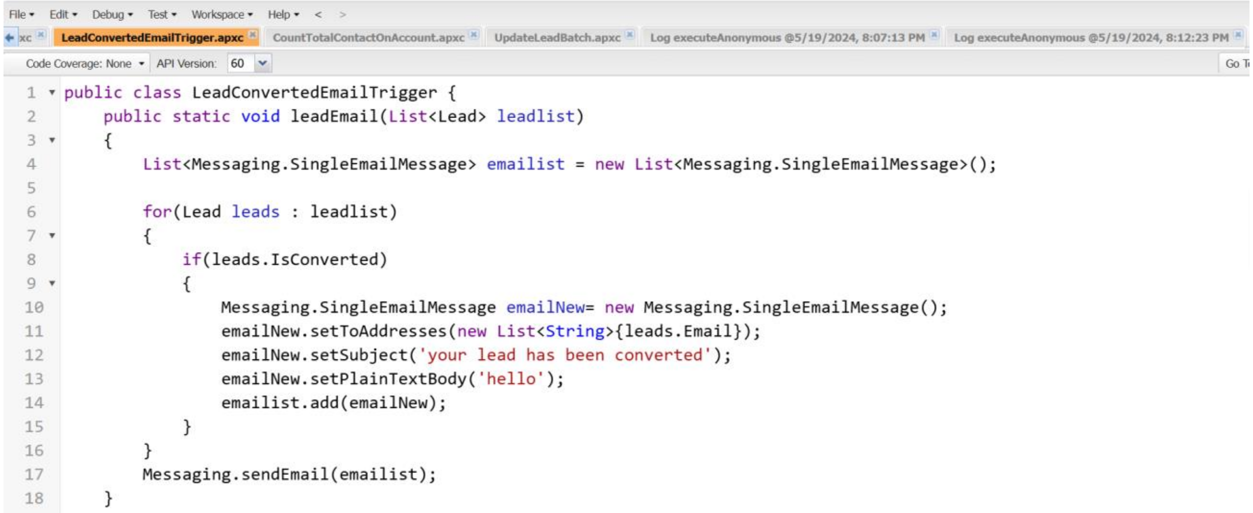
Available Items

- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests
- Asset Action Sources
- Asset Actions
- Asset State Periods

Selected Items

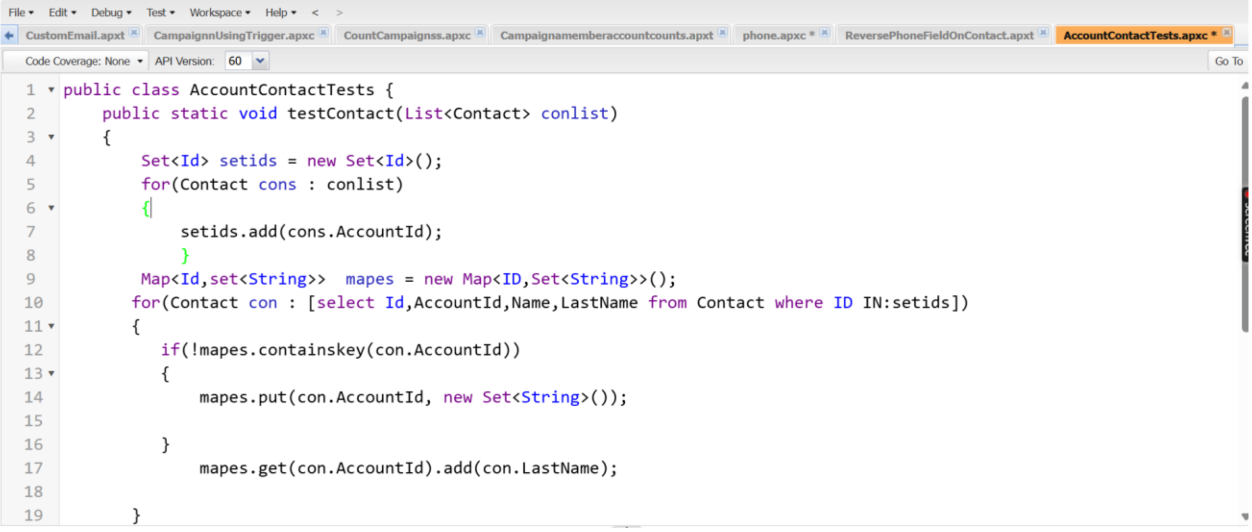
- Accounts
- Contacts
- Opportunities
- Colleges
- staffes
- Studns
- ccourses
- CollegeCourses
- Departments
- new
- updates

Fig 13: Navigation Items


File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
xc
LeadConvertedEmailTrigger.apxc
CountTotalContactOnAccount.apxc
UpdateLeadBatch.apxc
Log executeAnonymous @5/19/2024, 8:07:13 PM
Log executeAnonymous @5/19/2024, 8:12:23 PM
Code Coverage: None ▾
API Version: 60
Go To

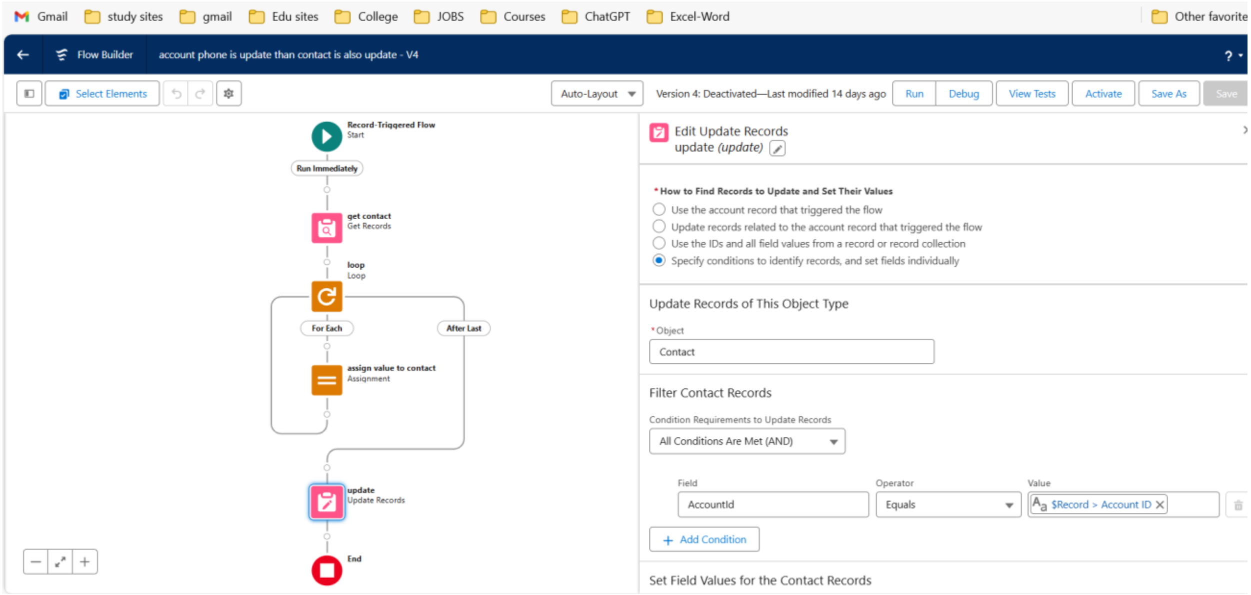
```
public class LeadConvertedEmailTrigger {
    public static void leadEmail(List<Lead> leadlist)
    {
        List<Messaging.SingleEmailMessage> emailist = new List<Messaging.SingleEmailMessage>();

        for(Lead leads : leadlist)
        {
            if(leads.IsConverted)
            {
                Messaging.SingleEmailMessage emailNew= new Messaging.SingleEmailMessage();
                emailNew.setToAddresses(new List<String>{leads.Email});
                emailNew.setSubject('your lead has been converted');
                emailNew.setPlainTextBody('hello');
                emailist.add(emailNew);
            }
        }
        Messaging.sendEmail(emailist);
    }
```

Figure 14: Apex Code to send Mail when lead converted


File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
CustomEmail.apxt
CampaignnUsingTrigger.apxc
CountCampaignss.apxc
Campaignamemberaccountcounts.apxt
phone.apxc *
ReversePhoneFieldOnContact.apxt
AccountContactTests.apxc *
Code Coverage: None ▾
API Version: 60
Go To

```
public class AccountContactTests {
    public static void testContact(List<Contact> conlist)
    {
        Set<Id> setids = new Set<Id>();
        for(Contact cons : conlist)
        {
            setids.add(cons.AccountId);
            }
        Map<Id,set<String>>  mapes = new Map<ID,Set<String>>();
       for(Contact con : [select Id,AccountId,Name,LastName from Contact where ID IN:setids])
       {
          if(!mapes.containskey(con.AccountId))
          {
              mapes.put(con.AccountId, new Set<String>());

          }
              mapes.get(con.AccountId).add(con.LastName);

       }
```

Fig. 15 : Validate a user to add Duplicate contact



Edit Update Records

update *(update)*

* How to Find Records to Update and Set Their Values

○ Use the account record that triggered the flow

○ Update records related to the account record that triggered the flow

○ Use the IDs and all field values from a record or record collection

● Specify conditions to identify records, and set fields individually

Update Records of This Object Type

* Object: Contact

Filter Contact Records

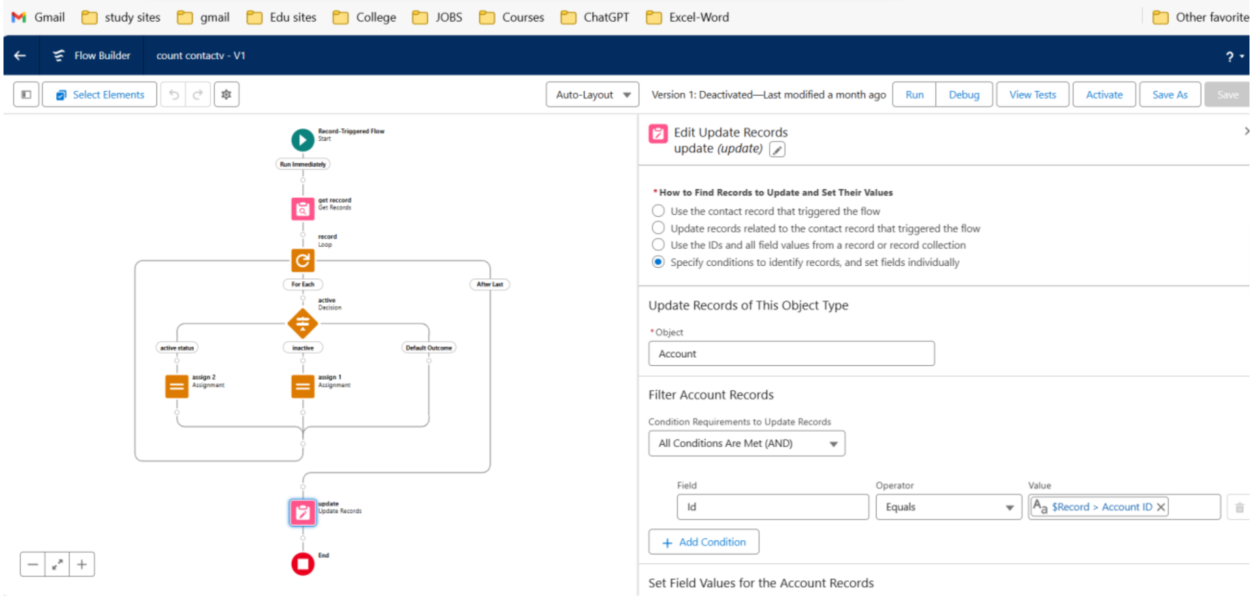
Condition Requirements to Update Records: All Conditions Are Met (AND)

Field	Operator	Value
AccountId	Equals	$Record > Account ID

+ Add Condition

Set Field Values for the Contact Records

Figure 16: A flow to update contact's Phone when Account Phone is updated



Edit Update Records

update *(update)*

* How to Find Records to Update and Set Their Values

○ Use the contact record that triggered the flow

○ Update records related to the contact record that triggered the flow

○ Use the IDs and all field values from a record or record collection

● Specify conditions to identify records, and set fields individually

Update Records of This Object Type

* Object: Account

Filter Account Records

Condition Requirements to Update Records: All Conditions Are Met (AND)

Field	Operator	Value
Id	Equals	$Record > Account ID

+ Add Condition

Set Field Values for the Account Records

Figure 17: Count the Active and Inactive Courses in the college.

TEST CASES

- In the context of Salesforce applications, test cases refer to specific scenarios designed to validate the functionality, performance, and reliability of the application. These test cases are written to ensure that the application behaves as expected under various conditions. They help identify bugs or issues before the application is deployed to production.

  Here are some key aspects of test cases in Salesforce:

- **Unit Tests**: These are written to test individual components or pieces of code, such as Apex classes and triggers. The Salesforce requires that at least 75% of the Apex code is covered by unit tests before it can be deployed to production.

- **Integration Tests**: These tests validate the interactions between different components of the Salesforce application or between Salesforce and other systems.

- **Functional Tests**: These tests ensure that the application's functionality works as intended. They validate the application against the business requirements and use cases.

- **Regression Tests**: These tests are run to ensure that new changes or enhancements do not negatively impact existing functionality.

- **User Acceptance Testing (UAT)**: These tests are conducted by end-users to validate that the application meets their requirements and is ready for deployment.

- **Performance Tests**: These tests assess how the application performs under various conditions, such as high load or stress.

  In Salesforce, test cases can be written using Apex test classes and methods. The **@isTest** annotation is used to define test classes and methods. Here is a simple example of an Apex test class:

1). Example of Test Class:

```
@isTest
public class MyApexClassTest {
   @isTest
   static void testMyMethod() {
      // Set up test data
      Account testAccount = new Account(Name = 'Test Account');
      insert testAccount;

      // Call the method to be tested
      MyApexClass.myMethod(testAccount.Id);

      // Verify the results
      Account result = [SELECT Id, Name FROM Account WHERE Id = :testAccount.Id];
      System.assertEquals('Updated Account', result.Name);
   }
}
```

This example demonstrates setting up test data, calling the method under test, and verifying the results using assertions. Properly written test cases ensure the robustness and reliability of Salesforce applications.

2). Second Example of Test Class:

Case Scenario:

(i) An apex class that updates portal user details.

(ii) Guest users are never able to access this page.

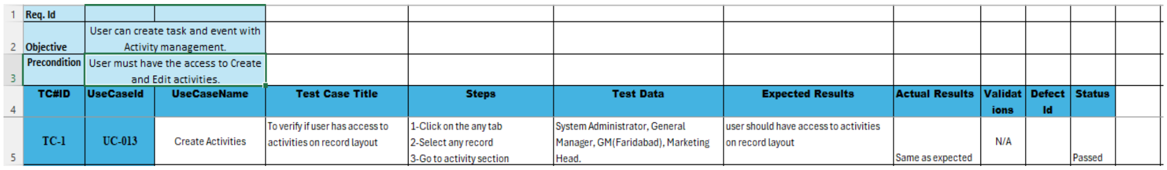
Creating the Class as:

```
/**
    * An apex class that updates portal user details.
  Guest users are never able to access this page.
 */
public with sharing class MyProfilePageController {

   private User user;
   private boolean isEdit = false;

   public User getUser() {
      return user;
  }

  public MyProfilePageController() {
    user = [SELECT id, email, username, usertype, communitynickname, timezonesidkey,
languagelocalekey, firstname, lastname, phone, title,
        street, city, country, postalcode, state, localesidkey, mobilephone, extension, fax,
contact.email
        FROM User
        WHERE id = :UserInfo.getUserId()];
    // guest users should never be able to access this page
    if (user.usertype == 'GUEST') {
      throw new NoAccessException();
    }
  }

  public Boolean getIsEdit() {
    return isEdit;
  }

  public void edit() {
    isEdit=true;
  }

  public void save() {
    try {
      update user;
      isEdit=false;
    } catch(DmlException e) {
      ApexPages.addMessages(e);
    }
  }

  public void cancel() {
    isEdit=false;
    user = [SELECT id, email, username, communitynickname, timezonesidkey,
languagelocalekey, firstname, lastname, phone, title,
        street, city, country, postalcode, state, localesidkey, mobilephone, extension, fax,
contact.email
        FROM User
        WHERE id = :UserInfo.getUserId()];
  }
}
```

Writing its Test Class as :

```
/**
 * An apex class that updates details of a portal user.
   Guest users are never able to access this page.
 */
@IsTest public with sharing class MyProfilePageControllerTest {

    @IsTest(SeeAllData=true) static void testSave() {
        // Modify the test to query for a portal user that exists in your org
        List<User> existingPortalUsers = [SELECT id, profileId, userRoleId FROM User WHERE
UserRoleId <> null AND UserType='CustomerSuccess'];

        if (existingPortalUsers.isEmpty()) {
            User currentUser = [select id, title, firstname, lastname, email, phone, mobilephone, fax,
street, city, state, postalcode, country
                                 FROM User WHERE id =: UserInfo.getUserId()];
            MyProfilePageController controller = new MyProfilePageController();
            System.assertEquals(currentUser.Id, controller.getUser().Id, 'Did not successfully load
the current user');
            System.assert(controller.getIsEdit() == false, 'isEdit should default to false');
            controller.edit();
            System.assert(controller.getIsEdit() == true);
            controller.cancel();
            System.assert(controller.getIsEdit() == false);


System.assert(Page.ChangePassword.getUrl().equals(controller.changePassword().getUrl()));

            String randFax = Math.rint(Math.random() * 1000) + '5551234';
            controller.getUser().Fax = randFax;
            controller.save();
            System.assert(controller.getIsEdit() == false);

            currentUser = [Select id, fax from User where id =: currentUser.Id];
            System.assert(currentUser.fax == randFax);
        } else {
            User existingPortalUser = existingPortalUsers[0];
            String randFax = Math.rint(Math.random() * 1000) + '5551234';

            System.runAs(existingPortalUser) {
                MyProfilePageController controller = new MyProfilePageController();
                System.assertEquals(existingPortalUser.Id, controller.getUser().Id, 'Did not
successfully load the current user');
                System.assert(controller.getIsEdit() == false, 'isEdit should default to false');
                controller.edit();
                System.assert(controller.getIsEdit() == true);
                controller.cancel();
                System.assert(controller.getIsEdit() == false);

                controller.getUser().Fax = randFax;
                controller.save();
                System.assert(controller.getIsEdit() == false);
            }

            // verify that the user was updated
            existingPortalUser = [Select id, fax from User where id =: existingPortalUser.Id];
            System.assert(existingPortalUser.fax == randFax);
        }
    }
}
```

Functional Testing



Req. Id										
Objective	User can create task and event with Activity management.									
Precondition	User must have the access to Create and Edit activities.									
TC#ID	UseCaseId	UseCaseName	Test Case Title	Steps	Test Data	Expected Results	Actual Results	Validations	Defect Id	Status
TC-1	UC-013	Create Activities	To verify if user has access to activities on record layout	1-Click on the any tab 2-Select any record 3-Go to activity section	System Administrator, General Manager, GM(Faridabad), Marketing Head.	user should have access to activities on record layout	Same as expected	N/A		Passed

Here are number of test cases which needed to be fulfil and to be passed by the developer for assuring the proper functioning of the code.

Functional testing in Salesforce involves verifying that the application behaves according to the specified functional requirements and use cases. It ensures that each feature of the application works correctly and that the end-to-end business processes are functioning as expected. Functional testing typically includes testing the user interfaces, APIs, databases, and integration points.

Key Aspects of Functional Testing in Salesforce

1. **Requirement Analysis**: Understanding and documenting the functional requirements of the Salesforce application.

2. **Test Planning**: Defining the scope, approach, resources, and schedule for the testing activities.

3. **Test Case Design**: Creating detailed test cases that cover all functional aspects of the application. This includes defining the input data, execution steps, and expected outcomes.

4. **Test Execution**: Running the test cases on the Salesforce application to verify that the

functionality works as expected.

5. **Defect Reporting**: Logging any issues or defects found during testing and tracking them until resolution.

6. **Regression Testing**: Re-running functional tests after changes are made to the application to ensure that existing functionality is not broken.

CONCLUSION

1. Creating a Salesforce application for a college entails integrating various functionalities tailored to enhance administrative efficiency, student engagement, and academic management. By leveraging Salesforce's robust platform, colleges can streamline admissions, manage student records, facilitate course enrollment, and empower faculty with tools for grading and communication.
2. A well-designed Salesforce application offers benefits such as centralized data management, real-time analytics for informed decision-making, and enhanced communication channels between students, faculty, and administration. It supports personalized student experiences through features like academic advising, event management, and alumni relations.
3. Moreover, integrating Salesforce with existing systems ensures scalability and adaptability to future technological advancements. Security measures safeguard sensitive data, maintaining compliance with industry standards and regulations.
4. In conclusion, a Salesforce application for a college not only optimizes operational processes but also fosters a collaborative learning environment, driving institutional success and student achievement. Its versatility empowers stakeholders to navigate complex educational landscapes effectively, contributing to long-term institutional growth and excellence.

FUTURE SCOPE OF PROJECT

- The future scope of a Salesforce project for a college holds promise for continual enhancement and adaptation to evolving educational needs and technological advancements. Here are key aspects to consider:

- Expansion of Functionalities: Enhance the application with additional modules for alumni relations, career services, fundraising, and research management to support the holistic needs of the college community.

- Integration with Emerging Technologies: Incorporate artificial intelligence (AI) and machine learning (ML) capabilities to personalize student experiences, predict academic outcomes, and optimize administrative processes.

- Mobile Optimization: Develop mobile applications to provide students, faculty, and administrators with seamless access to critical functions like course registration, grade updates, and campus communications.

- Enhanced Analytics and Reporting: Implement advanced analytics tools to derive.

BIBLIOGRAPHY

[1] Salesforce Platform Developer I Certification Handbook" by Siddhesh Kabe

[2] "Salesforce CRM: The Definitive Admin Handbook" by Paul Goodey: A practical guide for Salesforce administrators covering configuration, customization, and best practices for managing Salesforce CRM.

[3] https://www.udemy.com

[4] https://www.pluralsight.com

[5] https://www.salesforce.stackexchange.com

[6] https://www.youtube.com

[7] https://www.google.com

[8] https://www.wikepedia.com

[9] https://www.trailhead.salesforce.com

[10] https://www.developer.salesforce.com