**INDUSTRIAL TRAINING REPORT**

**ON**

**CUSTOMER IDENTITY AND ACCESS MANAGEMENT**

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**

A picture containing text, font, logo

Description automatically generated

**Submitted By:**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**DAV INSTITUTE OF ENGINEERING & TECHNOLOGY**

**Jalandhar – 144008**

June 2024

**CERTIFICATE**

This is to certify that (Student of B. Tech. CSE of DAV Institute of Engineering and Technology, Jalandhar) is successfully completing her Industrial Training from “10/01/2024” to “12/07/2024”. During this training, she has worked as Support Engineer on External User Access Management in Authorization and Profiling Solutions under the guidance of. Her overall performance during the training period is Excellent so as to fulfill all the requirements for successful completion of the training.

**ABSTRACT**A Customer Identity and Access Management (CIAM) system is a sophisticated platform designed to manage and secure customer identities across a digital ecosystem that encompasses various organizations and applications. It serves as the foundational infrastructure that enables businesses to provide secure access to their services while delivering a personalized and frictionless user experience. The CIAM system is adept at handling complex hierarchies and relationships, such as those between different organizations and user groups, ensuring that each entity's data and access rights are meticulously managed. By centralizing identity management, the CIAM system streamlines authentication, authorization, and account maintenance tasks, thereby enhancing operational efficiency and bolstering security.

Within the CIAM system, **Organizations** represent distinct business entities or divisions, each with its own set of users and access control policies. **Users** are the individual customer accounts that interact with the organization's digital services. The system allows for the creation of **Organization Groups**, which are collections of users within an organization, often defined by common attributes or roles. This grouping mechanism simplifies the management of access rights and enables targeted communication and service delivery. Similarly, **Application Groups** are used to manage access to sets of applications that are grouped together based on business functions or user needs, facilitating a more organized and efficient allocation of resources.

The CIAM system's architecture is designed to be both robust and flexible, accommodating the diverse and dynamic nature of user interactions with multiple organizations and applications. It incorporates advanced features such as Single Sign-On (SSO), multi-factor authentication (MFA), and consent management, all of which are crucial for maintaining user trust and compliance with data protection regulations. The system's ability to manage complex relationships and provide granular control over access makes it an invaluable asset for businesses looking to expand their digital footprint while maintaining a secure and customer-centric approach. With its comprehensive suite of tools and modules, the CIAM system empowers organizations to navigate the challenges of digital identity management and leverage the full potential of their online presence.

**ACKNOWLEDGEMENT**

I am highly grateful to Dr. Sanjeev Naval, Principal, DAV Institute of Engineering & Technology, Jalandhar, for providing this opportunity to carry out the six months Industrial Training at STMicroelectronics.

The constant guidance and encouragement received from Mrs. Harpreet Kaur Bajaj, HOD Department of Computer Science & Engineering, DAVIET Jalandhar has been of great help in carrying out the project work and is acknowledged with reverential thanks.

I would like to express a deep sense of gratitude and thanks profusely to Jean-Marc Chery (CEO) of ST Microelectronics. Without wise counsel and able guidance, it would have been impossible to complete the report in this manner.

I express gratitude to other faculty members of CSE department of DAVIET for their intellectual support throughout the course of this work. I extend my deep gratitude to Mr. Vishav Kapoor for smoothening the whole training semester experience and being available to always resolve my problems.

Last but not least I am thankful to Almighty for giving me strength to never lose hope and try out new things and constantly learn throughout my tenure.

Manmeet Kaur

# LIST OF FIGURES

**Fig No. Figure Description Page No.**

1.1 ST Logo 1

2.5 SDLC 11

3.2 Client Server Architecture 14

3.3.1 MVC Architecture 14

3.3.2 Order Status Management 15

3.4.1 Landing Page 15

3.4.2 Details Page 16

3.4.3 Order Summary Page 16

4.1 C# Icon 20

4.2 .NET Logo 21

4.3 Microsoft Visual Studio Preview 22

4.4 SQL Server Management Studio 23

4.5 Entity Framework Core 23

4.6 HTML,CSS and JS 24

# TABLE OF CONTENTS Contents Page No.

***Certificate i***

***Abstract ii***

***Acknowledgement iii***

***List of Figures iv***

***List of Tables vi***

***Table of Contents vii***

**Chapter 1: Introduction 1**

* 1. Introduction to Organization 1
  2. Introduction to Project 2
  3. Objectives 2
  4. Project Formulation 3
  5. Identification/Reorganization of Need 4
  6. Existing System 4
  7. Proposed System 5

**Chapter 2:Requirement Analysis and System Specification 6**

2.1 Feasibility Study 6

2.2 Software Requirement Specification 7  
 2.3 Validation 8

2.4 Expected hurdles 9  
 2.5 SDLC model 10  
  
  
**Chapter 3:System Design 12**

3.1 Design Approach 12  
 3.2 Detail Design 13  
 3.3 DFD 14  
 3.4 User Interface Design 15  
 3.5 Database Design 16  
 3.6 Methodology of system 18

**Chapter 4:Implementation,Testing and Maintenance 20**

4.1 Introduction to Languages, IDE’s ,Tools and Technologies 20  
 user for Implementation  
 4.2 Version of dependencies used 25  
 4.3 Coding standards of Language used 25  
   
**Chapter 5 : Results and Discussions 28** 5.1 User Interface Representation 28  
 5.1.1 Brief Description of Various Modules of the system 28  
 5.2 Back Ends Representation 34

**Chapter 6: Conclusion and Future Scope 41**

**References 42**

**CHAPTER 1:INTRODUCTION**  
  
**1.1 Introduction to Organization**   
  
ST Microelectronics Private Limited is a global, semiconductor company that designs, develops, manufactures and markets a broad range of semiconductor integrated circuits and discrete devices. Company has many customers with whom business transactions occur across the globe, the customer list includes electronic giants like Seagate, Samsung, Apple and Ericsson. Communication with the customer is obviously the key to have relationship with them. Clear, timely, transparent communication will gain and maintain a customer’s trust and respect.

Being a support group, DTIT is responsible for maintaining the software and hardware infrastructure that is supporting the production and research divisions of the organization. DTIT offers a wide variety of tasks, whether it be development of new software (e.g. support application), maintenance (e.g. solving problems generated on a daily basis), or upgrading (improving the existing infrastructure with new designs) IOG (Infrastructure Operations group) is a subsidiary of DTIT, responsible for keeping infrastructure always on and secured, with optimized performance. IOG aims to ensure efficient Operations and support for high availability and performance of WWDC (World Wide Data center) and central services.



**Fig 1.1 ST Logo**

One of the important asset for any company (whether it’s an IT firm or a Non-IT firm) is the Authorization and Profiling Solutions Team and one of the important functions of this infrastructure is to ensure the Extended User Access Management. The Authorization and Profiling team is of paramount importance as it ensures the secure and efficient management of access to the company's systems and information. This team is responsible for implementing robust access controls, thereby safeguarding sensitive data and intellectual property from unauthorized access—a critical function in an industry driven by innovation and confidentiality. They manage user permissions, adapting access rights as employees join, move within, or leave the company, ensuring compliance with industry regulations and internal policies. The team's role extends to profiling system usage to optimize resource allocation and monitor performance, which is essential for maintaining high operational efficiency in a highly competitive technological field. By maintaining detailed audit trails, the Authorization and Profiling team also supports the company's ability to conduct thorough security audits and respond effectively to any breaches. Their cross-departmental collaboration is vital for aligning security measures with organizational changes, and they play a key role in both employee training on security best practices and disaster recovery efforts. Overall, this team's work is crucial in managing risks, protecting company assets, and ensuring business continuity, thus underpinning the trust and reliability that are the cornerstones of STMicroelectronics' reputation and success in the semiconductor industry

This report will serve as literary piece written for the purpose of acquainting the reader with the technical background required to understand the report, what the problem is, how this project has been planned to develop, how the project development is proceeding and how the results will help to eradicate the related problems.  **1.2 Introduction to Project**

### Customer Identity and Access Management (CIAM) solutions are designed to manage and secure customer identities and access rights within digital services. These solutions are crucial for businesses that operate online platforms where customers need to log in to access personalized services, content, or e-commerce functions. CIAM systems are different from traditional enterprise IAM (Identity and Access Management) in that they are outward-facing and deal with the identities of customers rather than internal employees.

### Key Features of CIAM Solutions:

* **User Registration and Profile Management**: CIAM solutions provide user-friendly interfaces for customers to create and manage their profiles. This often includes self-service capabilities for updating personal information, preferences, and security settings.
* **Authentication**: Secure and flexible authentication methods are a core feature, including password-based, multi-factor, and password less authentication to ensure that only authorized users gain access.
* **Authorization**: CIAM systems manage user permissions, ensuring that customers have access to the appropriate resources and services based on their roles or attributes.
* **Single Sign-On (SSO)**: SSO allows customers to access multiple applications or services with one set of login credentials, improving the user experience.
* **Social Login**: Integration with social media platforms enables users to register and sign in using their existing social media accounts, streamlining the login process.
* **Security and Compliance**: CIAM solutions are designed with security in mind, often including features like fraud detection and adherence to privacy regulations such as GDPR, CCPA, or HIPAA.
* **Scalability**: These systems are built to handle large volumes of users and transactions, ensuring performance during peak times and as the business grows.
* **Personalization**: By collecting and analyzing user data, CIAM solutions can help businesses personalize the user experience, marketing efforts, and service offerings.

**1.3 Objectives**

### **Enhance Security**

* **Protect Customer Data**: Safeguard personal and sensitive customer information from unauthorized access and data breaches.
* **Prevent Fraud**: Implement advanced security measures to detect and prevent fraudulent activities.
* **Ensure Compliance**: Meet legal and regulatory requirements for data protection and privacy, such as GDPR, CCPA, and others.

### **Improve User Experience**

* **Streamline User Onboarding**: Provide a smooth and efficient registration process for new users.
* **Simplify Access**: Enable easy and secure access to services through Single Sign-On (SSO) and social logins.
* **Offer Self-Service Capabilities**: Allow users to manage their own profiles, passwords, and security settings without needing to contact support.

### **Support Business Growth**

* **Scalability**: Ensure the CIAM system can handle an increasing number of users and transactions as the business grows.
* **Personalization**: Leverage user data to offer personalized experiences, recommendations, and services.
* **Enhance Customer Engagement**: Use insights from user behavior to improve engagement strategies and customer retention.

### **Operational Efficiency**

* **Automate Identity Processes**: Reduce manual work by automating identity lifecycle management, from onboarding to offboarding.
* **Minimize IT Overhead**: Lower the burden on IT staff by providing user-friendly interfaces and self-service options for customers.

### **Gain Insights**

* **Analytics**: Collect and analyze data on user activities to gain insights into customer preferences and behavior.
* **Reporting**: Generate reports to monitor the performance of digital services and compliance with policies.

**1.4 Problem Formulation**

Problem formulation in the context of Customer Identity and Access Management (CIAM) involves identifying and defining the challenges that a CIAM solution aims to address within an organization. This process is critical for developing a CIAM strategy that meets the specific needs of the business and its customers. Here's how you might formulate the problem:

### **Identifying the Challenges**

1. **Security Risks**: How can the organization protect against unauthorized access, data breaches, and identity theft?
2. **Regulatory Compliance**: What are the requirements for compliance with data protection and privacy laws (e.g., GDPR, CCPA)?
3. **User Experience**: How can the registration, authentication, and account management processes be made more user-friendly?
4. **Scalability**: Can the current system scale to accommodate growth in the number of users and transactions?
5. **Integration Complexity**: How can the CIAM system be integrated with existing applications and infrastructure?
6. **Fraud Detection**: What measures are needed to detect and prevent fraudulent activities effectively?
7. **Data Management**: How can the organization manage and leverage customer data while ensuring privacy and consent?

### **Defining the Problem Statement**

After identifying the challenges, the next step is to articulate a clear problem statement.

### **Setting Objectives**

With the problem statement in place, the organization can set specific objectives for the CIAM solution:

* Implement multi-factor authentication to enhance security.
* Develop a consent management framework to comply with privacy laws.
* Create a seamless onboarding process to improve user experience.
* Ensure the CIAM system can handle an anticipated increase in user numbers.
* Integrate the CIAM solution with existing CRM and marketing tools.
* Utilize machine learning for real-time fraud detection.
* Establish a centralized data repository for customer profiles that supports analytics and personalization.

### **Developing Requirements**

The final step in problem formulation is to translate the objectives into detailed requirements that the CIAM solution must fulfill. These requirements will guide the selection, customization, or development of the CIAM system.

By systematically formulating the problem, organizations can ensure that their CIAM strategy is well-aligned with their business goals and customer needs, setting the stage for a successful implementation.

**1.5 Identification/Reorganization of Need**    
  
The identification and reorganization of needs for a Customer Identity and Access Management (CIAM) solution involves a thorough analysis of the current state of identity management within an organization and the desired future state. This process helps to prioritize requirements and align them with business objectives. Here's how you might approach it:

### **Step 1: Current State Assessment**

* **Audit Existing Systems**: Review current identity management practices, including authentication, authorization, user data storage, and access controls.
* **Identify Pain Points**: Gather feedback from customers, IT staff, and other stakeholders to understand where the current system is lacking.
* **Assess Security Posture**: Evaluate the current security measures and identify any vulnerabilities or compliance gaps.
* **Analyze User Behavior**: Look at how users interact with the current system to identify areas for improvement in the user experience.

### **Step 2: Future State Definition**

* **Define Business Goals**: Align the CIAM strategy with the organization's overall business objectives, such as improving customer satisfaction or expanding into new markets.
* **Establish Security Requirements**: Determine the level of security needed based on the sensitivity of the data and the regulatory environment.
* **Consider User Expectations**: Understand what users expect in terms of ease of use, personalization, and self-service capabilities.
* **Plan for Scalability**: Anticipate future growth and ensure that the CIAM solution can scale accordingly.

### **Step 3: Gap Analysis**

* **Compare Current vs. Desired States**: Identify the discrepancies between the current CIAM capabilities and what is needed to achieve the desired outcomes.
* **Prioritize Gaps**: Determine which gaps are most critical to address based on their impact on security, user experience, and business objectives.

### **Step 4: Needs Reorganization**

* **Categorize Needs**: Group needs into categories such as must-have, should-have, could-have, and will-not-have (the MoSCoW method).
* **Align with Objectives**: Ensure that each need is directly tied to a specific business objective or user requirement.
* **Stakeholder Buy-In**: Engage with stakeholders to validate the reorganized needs and gain their support.

### **Step 5: Requirement Specification**

* **Develop Detailed Requirements**: Translate the reorganized needs into specific, measurable, and testable requirements for the CIAM solution.
* **Create a Roadmap**: Outline a phased approach to implementing the CIAM solution, starting with the most critical needs.

### **Step 6: Continuous Improvement**

* **Implement Feedback Loops**: Establish mechanisms for ongoing feedback from users and stakeholders to continuously refine the CIAM strategy.
* **Monitor Trends**: Stay informed about new technologies, security threats, and regulatory changes that may affect CIAM needs.

By following these steps, an organization can ensure that its CIAM solution is well-tailored to its specific needs, providing a secure, compliant, and user-friendly system that supports the organization's strategic goals.

Single Sign On

User Profile Management

Progressive profiling

User Onboarding

Authentication

Consent and Privacy Management

Customer Identity and Access Management

**1.5 Key Feature of CIAM**

**CHAPTER 2. REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION**

**2.1 Feasibility study**A feasibility study for a Customer Identity and Access Management (CIAM) solution, which you've referred to as ACAM, is a critical assessment that evaluates the practicality and potential benefits of implementing the proposed system within an organization. The study aims to determine whether the project is viable, what resources are required, and what challenges may be encountered. Here's how you might conduct a feasibility study for a CIAM solution:

### **1. Technical Feasibility**

* **Assess Compatibility**: Determine if the proposed CIAM solution is compatible with existing IT infrastructure.
* **Evaluate Scalability**: Ensure the solution can handle the expected growth in user numbers and transaction volumes.
* **Review Security**: Check that the solution meets the required security standards and can integrate with current security protocols.
* **Test Performance**: Analyze whether the proposed system can deliver the required performance levels without significant latency or downtime.

### **2. Economic Feasibility**

* **Cost Analysis**: Calculate the total cost of ownership, including initial setup costs, licensing fees, maintenance, and potential savings from improved efficiency.
* **Return on Investment (ROI)**: Estimate the financial benefits of the proposed system in terms of increased revenue, reduced fraud, and improved customer retention.
* **Budget Constraints**: Consider the organization's budgetary limitations and how the project fits within financial plans.

### 3. **Operational Feasibility**

* **Impact on Operations**: Assess how the implementation of the CIAM solution will affect current business operations.
* **Staffing Requirements**: Determine if the organization has the necessary staff with the right skills or if additional training or hiring is needed.
* **Change Management**: Evaluate the organization's readiness for change and the impact on company culture.

**2.2 Software Requirement Specification**

**Functional Requirements:**

### **1. User Registration and Onboarding**

* **FR1.1**: The system shall allow new users to register using their email address, phone number, or social media accounts.
* **FR1.2**: The system shall provide an option for users to verify their identity through email or SMS verification.
* **FR1.3**: The system shall capture and store user consent for data processing in compliance with relevant regulations.

### **2. Authentication**

* **FR2.1**: The system shall support multiple authentication methods, including password-based, social logins, and biometrics.
* **FR2.2**: The system shall implement multi-factor authentication (MFA) for enhanced security.
* **FR2.3**: The system shall provide a mechanism for account lockout after a configurable number of failed login attempts.

### **3. Account Management**

* **FR3.1**: The system shall allow users to update their profile information, such as name, contact details, and preferences.
* **FR3.2**: The system shall enable users to change their password securely.
* **FR3.3**: The system shall support account recovery options for users who have forgotten their login credentials.

### **4. Authorization and Access Control**

* **FR4.1**: The system shall manage user roles and permissions to control access to resources and services.
* **FR4.2**: The system shall enforce attribute-based access controls (ABAC) based on user attributes and context.
* **FR4.3**: The system shall provide administrators with tools to define and manage access policies.

### **5. Single Sign-On (SSO)**

* **FR5.1**: The system shall support SSO to allow users to access multiple applications with a single set of credentials.
* **FR5.2**: The system shall maintain a secure session for users across different services.

### **6. Privacy and Compliance**

* **FR6.1**: The system shall provide features for users to manage their consent and privacy settings.
* **FR6.2**: The system shall enable users to download their data and request data deletion in compliance with regulations like GDPR.
* **FR6.3**: The system shall log and audit all consent changes and data access for compliance reporting.

## **Non-Functional Requirements:**

### **Performance Requirements**

* The system shall handle at least 10,000 concurrent users without performance degradation.

### **Security Requirements**

* All customer data shall be encrypted at rest and in transit.
* The system shall comply with the Open Web Application Security Project (OWASP) Top Ten.

**2.3 Validation**In a Customer Identity and Access Management (CIAM) system, validations are crucial to ensure the integrity, accuracy, and security of data, as well as to enforce business rules and compliance with regulations. Here are some common types of validations that might be implemented in a CIAM system:

### **1. Data Input Validation**

* **Field Format Validation**: Ensuring that data entered into forms meets specific formatting requirements, such as email addresses, phone numbers, and postal codes.
* **Data Type Checking**: Verifying that the data entered is of the correct type (e.g., strings, integers, dates).
* **Range Checking**: Confirming that numerical values fall within acceptable ranges (e.g., age restrictions).
* **Mandatory Field Checking**: Ensuring that all required fields are filled in before submission.

### **2. Authentication Validation**

* **Credential Verification**: Checking that usernames and passwords match the stored credentials.
* **MFA Checks**: Validating multiple factors of authentication, such as OTPs, biometric data, or security tokens.
* **Account Status**: Confirming that the user's account is active, not locked, or disabled.

### **3. Authorization and Access Control Validation**

* **Role Verification**: Ensuring that a user's role grants them permission to access the requested resource or perform a specific action.
* **Attribute Checks**: Applying attribute-based access controls to validate that user attributes meet policy requirements for accessing resources.
* **Contextual Validation**: Assessing the context of access requests, such as time of day, location, or device, to determine if access should be granted.

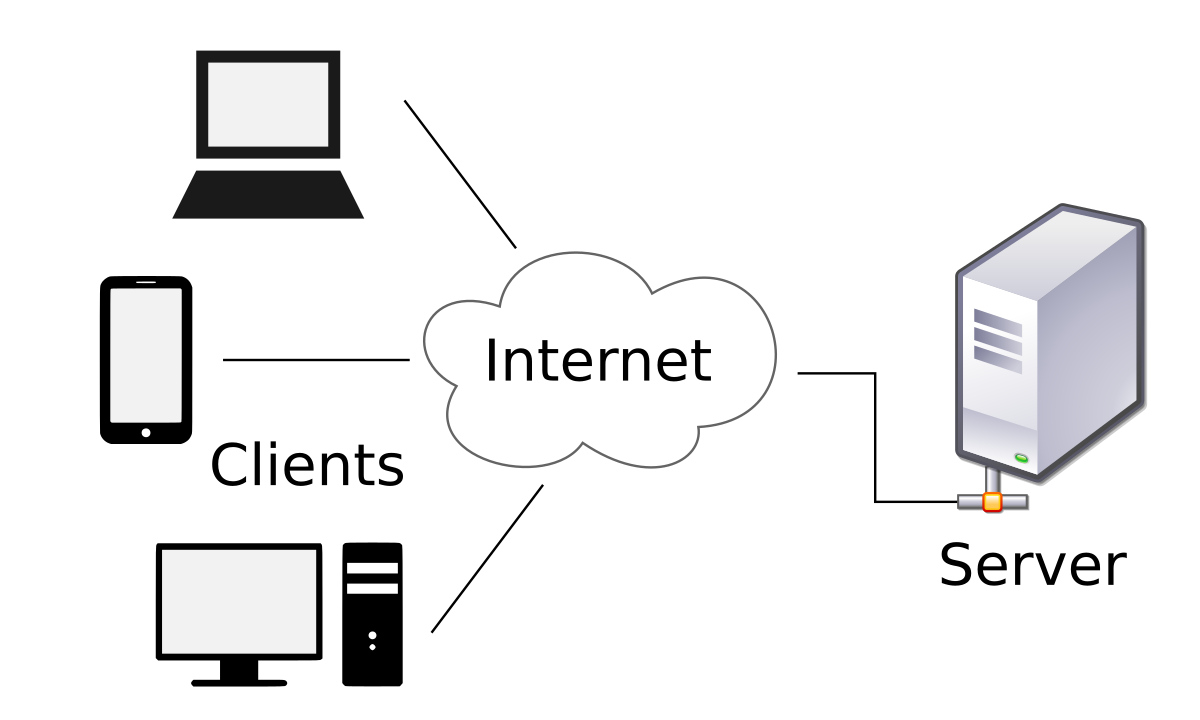
### **4. User Registration and Profile Update Validation**

* **Uniqueness Checks**: Verifying that identifiers like usernames or email addresses are not already in use.
* **Password Strength Validation**: Enforcing password complexity requirements to enhance security.
* **Consent Recording**: Validating that user consent is properly captured, recorded, and can be audited.

**Chapter 3. SYSTEM DESIGN**

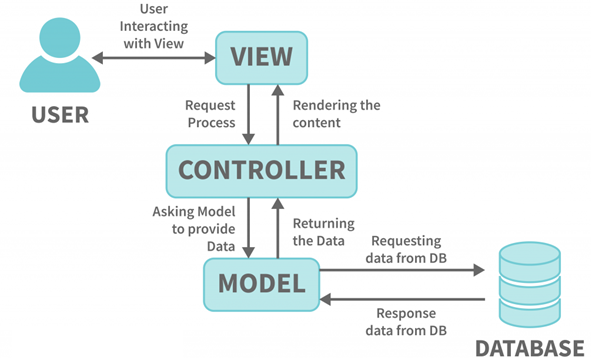
**3.1 Design Approach**The design approach for a Customer Identity and Access Management (CIAM) system involves a series of strategic and technical considerations to ensure that the system meets the needs of both the business and its customers. The design should be user-centric, secure, scalable, and compliant with relevant regulations.

**3.2 Detail Design**A Customer Identity and Access Management (CIAM) system,typically follows a client-server architecture. In this setup, the CIAM system acts as the server that provides identity management services, and the clients are the applications or services that rely on the CIAM system to authenticate and authorize user access.



**Fig. 3.2 Client-Server Architecture**

**3.3 DFD**



**Fig 3.3.1 MVC Architecture**

**3.6 Methodology of the system**

The adopted Methodology is **Agile/SCRUM** with its best practices.

Such a Methodology relies on the following main roles, each with the respective responsibilities but all interacting in a synergic work:

* The **Scrum Team**, composed by technicians working together to deliver the requested and committed product improvements. They are responsible for the timing and quality of the delivery.
* The **Scrum Master**, coordinator of the Scrum Team of which is a member.
* The **Scrum Product Owner**, who combines the classic product and project managers’ responsibilities covering the expertise of end users and other stakeholders, being responsible of the correctness and effectiveness of the delivered products.

The accomplishments are the following:

* The **Scrum Product Backlog** is the overall list of fulfilments, including features, bug fixes, non-functional requirements, and whatever useful to deliver a viable product. It is in charge of the Scrum Product Owner that collects the requirements cooperating with other Stakeholders, with the possible help of the Scrum Master.
* The **Sprint** is a fixed period (1-2 weeks) that starts with a **Sprint Planning Meeting** session, where the Scrum Team and Product Owner examine the Product Backlog items from the top to the bottom, optionally dividing complex items in smaller and better-defined tasks. This produces the **Sprint Backlog** as a result; it contains the list of activities that the Scrum Team will perform. Then the Scrum Team executes the planned tasks, updating the status of the items and clarifying any doubt through short daily Stand-up Meetings (**Daily Scrum Meetings**). A **Sprint Review Meeting** allows, at the end of each Sprint, the Scrum Product Owner to check completeness and correctness of the committed tasks. When necessary, a Sprint Retrospective Meeting is also provided, involving all the actors, to check and improve the processes, substantially discussing about “What was good during the Sprint”, “what should continue as it is” and “what should be improved”.

The first part is organized in standard Sprints iterations, spanning through the task planned time-line till all the requirements are covered. During each single Sprint’s time-frame, a set of one or more self-consistent functions from the Functional Analysis document is selected by the Scrum Master with the functional Analysts, who will review it and translate those functions’ descriptions into technical specifications, in the form of Features and single Backlog Items that will feed the Scrum Product Backlog; each item is reduced to one or more single tasks, and assigned to the development resources. At the start of each Sprint, the development resources in the Scrum Team take charge of each task assigned to them and process them in the allotted time. In specifics, they implement the code needed to produce the requirements described in the technical specifications generated during the previous Sprint while the technical specifications produced during the current Sprint will be developed and tested during the following one. With this pattern, a stream of technical specifications is always available to the development resources. Every two Sprint the appointed test team will perform the Integration Test of each newly released software against the current version of the system already released.

The second part is organized in Daily Sprint iterations. It is executed when all the in-scope functionalities are successfully implemented. Firstly, the software is built and deployed to the UAT environment to be tested as a whole, following the instructions of the Product Owner. During each Sprint one or more functionality is tested and each eventual bug or anomaly is reported. The functional Analyst will evaluate any issue reported during the previous Sprint and plan any necessary fixing activity.

When all tasks are successfully completed the package will be built, delivered, and deployed to the staging and live environments, ready to be commercialized. The build process is automatically performed by the adopted build system (DevOps) under the Technical Manager supervision, and consists in creating the *build definition*, compiling the source code, packaging the solution, and copying it to the project’s drop folder*.*

Following, the principal practices applied:

* **Continuous integration**: practice of merging all developer-working copies to a shared mainline several times a day, to prevent integration problems.
* **Code Refactoring:** process of restructuring existing code, changing the factoring, without changing its external behavior. Refactoring improves non-functional attributes of the software, such as code readability and complexity. This enhances the source-code maintainability and creates either a more expressive internal architecture or object model to improve extensibility.

**CHAPTER 4. IMPLEMENTATION, TESTING AND MAINTENANCE**

**4.1 Introduction to Languages, IDE’s, Tools and Technologies used for**

**Implementation**

**1. C#**

C# (pronounced "C Sharp") is a modern, object-oriented, and type-safe programming language developed by Microsoft as part of its .NET initiative. It was created by Anders Hejlsberg and his team and was first released in 2000. C# has its roots in the C family of languages and is similar in syntax to C and C++, but it also incorporates many features of other languages like Java and Delphi.

* **Key Features of C#:**

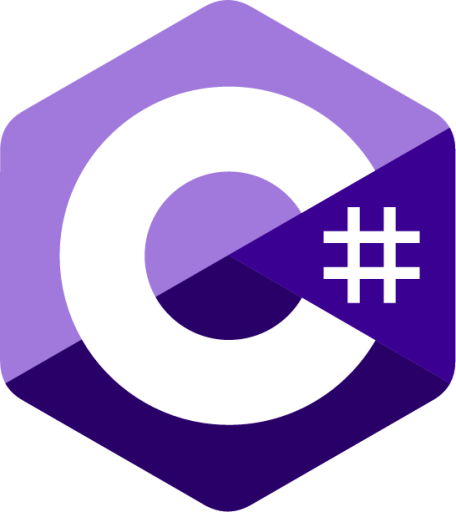
Object-Oriented: C# is fundamentally object-oriented, meaning it supports the concepts of encapsulation, inheritance, and polymorphism.

Type-Safe: It enforces strict type-checking at compile-time, preventing type errors that could lead to runtime exceptions.

Memory Management: C# uses automatic memory management and garbage collection, freeing developers from manual memory management tasks.

Cross-Platform: With the introduction of .NET Core, C# code can run on multiple platforms, including Windows, macOS, and Linux.

Rich Library Support: C# comes with a comprehensive standard library, the .NET Framework, which provides a broad range of functionalities, from file I/O to network communications.



**Fig 4.1 C# Icon**

**2. .NET**

. NET is a free, cross-platform, open-source developer platform created by Microsoft for building many different types of applications. With .NET, you can use multiple languages, editors, and libraries to build for web, mobile, desktop, games, IoT, and more.

* Key Components of .NET:

.NET Framework: The original implementation of .NET, primarily for Windows applications. It includes a large class library called the Framework Class Library (FCL) and provides language interoperability across several programming languages.

.NET Core: A cross-platform, open-source rewrite of .NET that is designed to be agile and performant. It's used to create server applications that run on Windows, Linux, and macOS. .NET Core was rebranded as .NET 5 (and later versions) as it unified the .NET platform.

ASP.NET: A subset of .NET designed for building web applications. ASP.NET Core is the cross-platform version that can run on .NET Core.

Mono: An open-source implementation of the .NET Framework that allows applications to run on Linux, macOS, and other platforms.

* **Features of .NET:**

Language Support: .NET supports multiple programming languages, but C#, F#, and Visual Basic are the most common.

Base Class Library (BCL): Provides a set of standard classes that serve as the collective API for .NET applications.

Common Language Runtime (CLR): The execution engine that handles running applications, including services like memory management, type safety, exception handling, garbage collection, and more.

Interoperability: .NET can interoperate with other languages and systems, making it versatile for integrating with existing infrastructure.

Security: .NET has a robust security mechanism, including code access security and validation and verification.

NuGet: A package manager for .NET that allows developers to share and use code from other developers and libraries.



**Fig 4.2 .Net Logo**

**3. Microsoft Visual Studio Preview**

Microsoft Visual Studio Preview refers to the pre-release versions of the Visual Studio integrated development environment (IDE). These preview versions are made available to developers before the official release of the next version of Visual Studio. The purpose of the preview releases is to allow developers to test new features, provide feedback, and adapt to changes ahead of the general availability.

* Features of Visual Studio Preview:

Early Access to Features: Developers get to try out the latest features and improvements that are being considered for the next version of Visual Studio.

Feedback Opportunity: Users can provide feedback directly to Microsoft about bugs, usability issues, or feature suggestions, which can influence the final product.

Side-by-Side Installation: Visual Studio Preview can be installed alongside the stable release of Visual Studio, allowing developers to use both versions for different projects without interference.

Updated Regularly: Preview versions are updated more frequently than the stable release, often with fixes and enhancements based on user feedback.

  
  
**Fig 4.3 Microsoft Visual Studio Preview**

**4. SQL Sever Management Studio**  
SQL Server Management Studio (SSMS) is an integrated environment used to manage the infrastructure of SQL Server, Azure SQL Database, and Azure Synapse Analytics. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases. It allows users to deploy, monitor, and upgrade the data-tier components used by applications, as well as build queries and scripts.

* **Key Features of SSMS:**

Graphical Interface: Offers a user-friendly graphical interface to manage the components of SQL Server.

Object Explorer: Allows users to browse, select, and act upon any of the objects within the server.

Query Editor: Provides a powerful editor for writing and executing T-SQL queries.

Database Design: Tools to create and manage databases, tables, indexes, keys, constraints, and stored procedures.

Security Management: Manage server and database security, including logins, roles, and user permissions.

Backup and Restore: Facilitate the process of backing up databases and restoring them.

Import/Export Wizard: Assists in transferring data to and from SQL Server..

Integration with other Tools: Works with other Microsoft tools and services, such as Visual Studio and Azure services.   
  
  
**Fig 4.4 SQL Server Management Studio**

5. **Entity Framework Core**   
  
Entity Framework Core (EF Core) is a lightweight, extensible, open-source, and cross-platform version of Entity Framework, which is Microsoft's object-relational mapper (ORM) for .NET. EF Core serves as a bridge between your application's domain or entity classes and the database, allowing you to write code using your .NET objects and have EF Core translate your operations into SQL queries and commands.

* **Key Features of Entity Framework Core:**

LINQ Queries: EF Core allows you to use LINQ (Language Integrated Query) to write queries against your DbContext, and it translates them into SQL queries.

Change Tracking: EF Core keeps track of changes made to the instances of your entity classes and applies these changes to the database when you call SaveChanges().

Migrations: EF Core provides a set of tools to automatically update the database schema to match your entity classes through a feature called migrations.

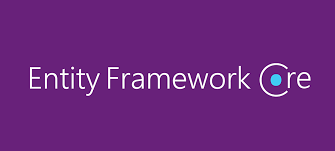
Database Providers: EF Core supports multiple database providers, allowing you to target SQL Server, SQLite, PostgreSQL, MySQL, and others.

Concurrency Control: It includes features for handling concurrency conflicts when multiple users are updating the same data.

Transactions: EF Core supports transactions, which are essential for ensuring data integrity.

Caching: It includes a first-level cache out of the box, where it stores the results of queries for faster access.

Asynchronous Programming: EF Core supports asynchronous operations, which can improve the scalability of your application by freeing up threads for other work while database I/O operations are in progress.

  
 **Fig 4.5 Entity Framework Core**

6.**HTML**

HTML is the language for describing the structure of Web pages. HTML gives authors the means to Publish online documents with headings, text, tables, lists, photos, etc. Retrieve online information via hypertext links, at the click of a button. Web browsers receive HTML

documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

7. **CSS**

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.CSS is among the core languages of the open web and is standardized across Web

browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations.

8.**JavaScript**

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.  
  


**Fig 4.8 HTML.CSS ,Javascript**

**9. TinyMCE**TinyMCE is a platform-independent web-based JavaScript WYSIWYG (What You See Is What You Get) editor control, released as open-source software under the LGPL. It enables users to edit and format content in a more user-friendly way that resembles how the content would appear in a published format, or in other words, as it would look on a webpage or document. TinyMCE is designed to integrate into form fields or other HTML elements on a site and provides a rich set of features like text formatting, tables, lists, image insertion, and more, without requiring the user to write HTML or CSS. It's widely used in content management systems, email marketing software, and other applications where ease of content creation is essential.

**10. Sweet Alerts**

SweetAlert is a customizable, promise-based JavaScript library used to replace the native alert, prompt, and confirm dialog boxes provided by browsers with more aesthetically pleasing versions that are also more versatile and functional. It offers developers the ability to create beautifully designed pop-up messages that can display a wide range of content, such as text, images, and HTML, and can include features like custom icons, animations, buttons with custom actions, and various forms of user input. SweetAlert's modals are not only visually appealing but also responsive and accessible, enhancing the user interface and interaction in web applications. **11. DataTables API**The DataTables API is a powerful interface for developers working with the DataTables plugin for jQuery. DataTables is a highly flexible tool that enhances the functionality of HTML tables and allows for advanced interaction with the table data. The API provides a wide array of programmatically accessible methods and properties that enable developers to manipulate the table data and presentation dynamically. This includes tasks such as adding, removing, or updating rows and columns, filtering and sorting data, retrieving information about the table state, and integrating with other data sources like server-side processing for handling large datasets. The API is designed to be intuitive and easy to use, yet robust enough to allow for complex operations on the table data, making it a popular choice for web developers who need to display and manage tabular data in a user-friendly manner.

**4.2 Version requirements**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **NAME** | **VERSION** |
| 1. | .Net Framework | 4.7.2 |
| 2. | SQL Server Management Studio | 2018 |
| 3. | SQL Express | 2019 |
| 4. | ASP.NET | 4.8(under IIS) |
| 5. | PowerShell | 5.x |

**4.3 Coding standards of Language used**Coding standards are a set of guidelines that recommend programming styles, practices, and methods for each aspect of a program written in a given language. They are intended to improve the readability of code and make it consistent across the vast spectrum of software development projects. For C# and .NET MVC (Model-View-Controller) applications, coding standards help maintain a high level of code quality and ensure that code is easy to understand, maintain, and extend.

**General C# Coding Standards:**

* Naming Conventions:

Use PascalCase for class names and method names.

Use camelCase for local variables and method arguments.

Use \_camelCase with an underscore prefix for private fields.

Use ALL\_CAPS for constants.

Avoid using abbreviations and use descriptive names.

* Code Layout:

Use spaces instead of tabs with a common size (e.g., 4 spaces per indent).

Keep lines of code reasonably short (e.g., no longer than 120 characters).

Use braces for all control structures and put the opening brace on the same line as the control statement.

* Commenting and Documentation:

Use XML comments for documentation of public APIs.

Write meaningful comments where necessary, but prefer self-explanatory code.

Avoid comments that simply restate the code.

* Best Practices:

Follow SOLID principles for object-oriented design.

Prefer readonly to mutable fields where possible.

Use var for local variable declarations when the type is apparent from the right side of the assignment.

Handle exceptions using try-catch blocks and throw meaningful exceptions.

* Testing:

Write unit tests for critical parts of the code.

Name test methods clearly, often based on the method being tested and the expected outcome.

* .NET MVC Specific Standards:

Models:

Keep your models lightweight and use them to represent the data and business rules of your application. Use Data Annotations for model validation.

Views:

Keep logic out of views; they should only contain markup and minimal presentation logic.

Use Razor syntax effectively and consistently.

Use partial views for reusable components.

Controllers:

Keep controllers concise and focused on handling user input and returning responses.

Use action filters for cross-cutting concerns like logging and authorization.

Follow the "fat models, skinny controllers" principle.

* Routing:

Use attribute routing for clarity and control over URL patterns.

Name routes clearly and consistently.

* Security:

Use anti-forgery tokens to prevent Cross-Site Request Forgery (CSRF) attacks.

Always validate user input to prevent SQL injection and other attacks.

**CHAPTER 5. RESULTS AND DISCUSSIONS**

**5.1 Back Ends Representation**The decision to use SQL Server Management Studio (SSMS) for a project is typically based on several factors that align with the project's requirements and the features that SSMS offers. Here are some reasons why a project might choose to use SSMS:

Comprehensive Tool: SSMS is a comprehensive tool for managing SQL Server instances, including databases, security, server settings, and more.

User-Friendly Interface: It provides a graphical user interface that simplifies complex database operations, which can be more accessible for users who are not comfortable with command-line tools.

Integrated Development Environment: SSMS serves as an IDE for SQL development, allowing for the creation, testing, and optimization of SQL scripts and stored procedures.

Intellisense: SSMS includes Intellisense for SQL, which speeds up the development process by providing auto-completion, syntax highlighting, and quick info about database objects.

Query Optimization: SSMS offers tools like the Query Analyzer and Execution Plan viewer to help developers write efficient SQL queries and optimize database performance.

Monitoring Tools: Built-in monitoring tools such as Activity Monitor and SQL Server Profiler help in diagnosing performance issues and monitoring server activity.

Security Management: SSMS provides a straightforward interface for managing database security, including user roles, permissions, and other security features.

Audit and Compliance: It can assist in setting up and managing audits, which are essential for compliance with various regulatory standards.

Backup and Restore: SSMS simplifies the process of backing up and restoring databases, which is crucial for data recovery and business continuity.

Maintenance Plans: The ability to create and schedule maintenance plans helps ensure that databases are regularly optimized, backed up, and checked for integrity.

SQL Server Integration: SSMS is designed to work seamlessly with SQL Server, providing a consistent management experience across all SQL Server services.

Support for SQL Server Features: It supports the latest SQL Server features and is regularly updated to align with new SQL Server releases.

Free Tool: SSMS is available for free, which can be a significant consideration for projects with budget constraints.

Widely Used: As a widely adopted tool, it's easier to find resources, community support, and experienced professionals familiar with SSMS.

Project Files: SSMS allows developers to save scripts and project files, making it easier to share and collaborate on database development tasks.

Microsoft Support: Being a Microsoft product, SSMS has strong support and regular updates, ensuring long-term viability for managing SQL Server databases.

In summary, a project might use SSMS because it is a powerful, user-friendly, and well-supported tool that can handle a wide range of database management tasks efficiently. It's particularly well-suited for projects that rely on SQL Server as their database system.

**CHAPTER 6. CONCLUSION**

In conclusion, a Customer Identity and Access Management (CIAM) system serves as the cornerstone of modern digital business operations, providing a secure and efficient framework for managing customer identities and access privileges. The strategic implementation of a CIAM system not only fortifies an organization's cybersecurity posture but also enhances the overall customer experience. By offering streamlined registration processes, robust authentication protocols, and seamless access across various platforms, CIAM systems play a pivotal role in fostering customer trust and loyalty. Furthermore, they enable businesses to gather valuable insights into customer behavior, which can inform personalized marketing strategies and drive business growth.

Moreover, a CIAM system is instrumental in ensuring that an organization remains compliant with ever-evolving data protection regulations, thereby safeguarding against legal and financial repercussions. The agility and scalability provided by a CIAM system are essential for businesses looking to adapt to the dynamic digital landscape and accommodate growth. As digital interactions continue to dominate the customer-business relationship, investing in a comprehensive CIAM solution is not just a tactical move but a strategic imperative that can yield long-term competitive advantages.   
  
**Future Scope**

**REFERENCES**

* Microsoft. ASP.NET MVC. Retrieved from https://dotnet.microsoft.com/apps/aspnet/mvc
* Microsoft Entity Framework. Retrieved from https://docs.microsoft.com/en-us/ef/
* W3Schools. HTML Tutorial. Retrieved from https://www.w3schools.com/html/
* CSS-Tricks. CSS Basics. Retrieved from https://css-tricks.com/guides/basics/
* Microsoft. C# Programming Guide. Retrieved from <https://docs.microsoft.com/en-us/dotnet/csharp/>
* Microsoft. .NET Documentation. Retrieved from https://docs.microsoft.com/en-us/dotnet/
* Stack Overflow. Community-driven question and answer site for programmers. Retrieved from https://stackoverflow.com/
* Gupta, R. (2019). Learning ASP.NET MVC Programming. Packt Publishing.
* Freeman, A., & Sanderson, S. (2014). Pro ASP.NET MVC 5. Apress.
* Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.