PROGRAM: 1

1.1OBJECTIVE	1.2THEORY	1.3 PROCEDURE	1.4 OUTPUT

1.1 **OBJECTIVE: Prepare a SRS document in line with the IEEE recommended standards.**

1.2 THEORY: An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

Well-designed, well-written SRS accomplishes four major goals:

1. It provides feedback to the customer.
2. It decomposes the problem into component parts.
3. It serves as an input to the design specification.
4. It serves as a product validation check.

The basic issues that the SRS shall address are the following:

1 **Functionality**. What is the software supposed to do?
2 **External interfaces**. How does the software interact with people, the system's hardware, other hardware, and other software?
3 **Performance**. What is the speed, availability, response time, recovery time of various software functions, etc.?
4 **Attributes**. What is the portability, correctness, maintainability, security, etc. considerations?

1.3 **PROCEDURE:**

IEEE Standard SRS Template

3.2 Specific requirements

3.2.1 Sequence diagrams

3.2.2 Classes for classification of specific requirements

3.3 Performance requirements

3.4 Design constraints

3.5 Software system attributes

3.5.1 Reliability

3.5.2 Availability

3.5.3 Security

3.5.4 Maintainability

3.6 Other requirements

4. Supporting information

4.1 Table of contents and index

4.2 Appendixes

Note: History of versions of this document with author/contributor info may be included before the main sections of the document.

1.4 **OUTPUT:**

SAMPLE SRS FOR CUSTOMER EXPERIENCE MANAGEMENT SYSTEM

Software Requirements Specification for

Customer Experience Management System

Table of Contents

References

Appendices

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the Customer Experience Management System (CEMS). This system will automate all customer service activities including feedback collection, support ticket management, and performance analytics. This document gives a comprehensive description of the CEMS according to the client's requirements.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document includes:

- System designers
- Administrative staff of the customer service department
- System developers
- Testers
- System analysts
- Software architects
- Maintenance engineers
- End users (customer service representatives, managers, customers)

Stakeholders directly or indirectly involved with the project should focus on different sections based on their roles:

- Executives and managers: Sections 1, 2, and 5
- Developers and technical staff: Sections 2, 3, 4, and 6
- End users: Sections 2.2, 2.3, 3.1, and 4

1.3 Product Scope

This system transforms traditional customer service management into a digital, data-driven platform. It allows organizations to collect, analyze, and respond to customer feedback efficiently while tracking performance metrics. The CEMS enables users to:

- Collect and organize customer feedback from multiple channels
- Create, assign, and track support tickets
- Generate performance analytics and reports
- Improve response time and customer satisfaction
- Identify trends and opportunities for improvement

2. Overall Description

2.1 Product Perspective

The main objective of this document is to illustrate the requirements of the Customer Experience Management System. This system replaces manual methods of customer feedback collection and support ticket management, making these tasks more efficient and data-driven. The proposed system provides a centralized platform for capturing customer interactions, managing support tickets, and analyzing performance metrics.

2.2 Product Functions

This system is primarily used by customer service representatives, managers, and customers. The following features are available:

Managers:

- View performance dashboards and analytics
- Configure feedback surveys and response templates
- Monitor team performance metrics
- Generate custom reports
- Manage user accounts and permissions

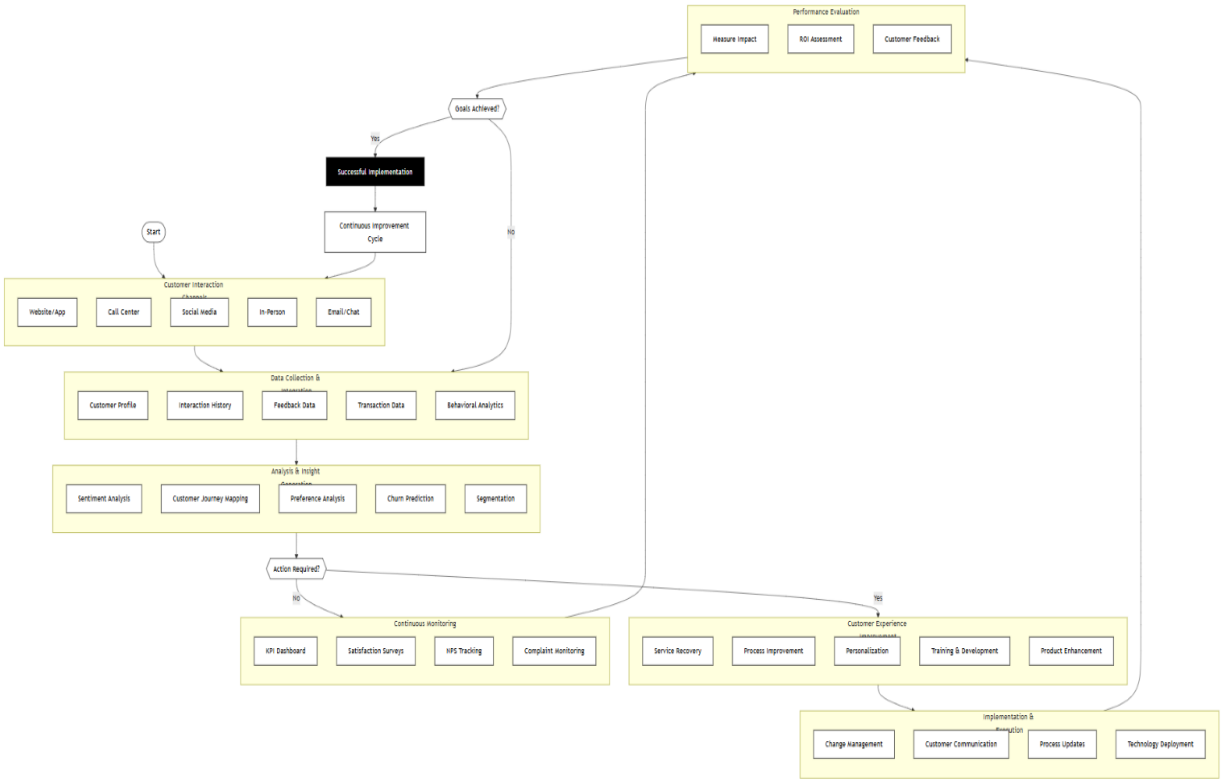
Customer Service Representatives:

- Respond to customer feedback
- Create and update support tickets

- Track ticket status and history
- Collaborate with team members
- Access knowledge base resources

Customers:

- Submit feedback through multiple channels
- Create support tickets
- View ticket status and history
- Receive automated notifications on ticket updates



2.3 User Classes and Characteristics

There are three main user types for this product:

1. Managers (Administrators)
    a. Need comprehensive access to all system features
    b. Require analytics and reporting capabilities
    c. Manage team performance and system configurations
    d. Need ability to configure surveys and templates

    e. Access to user management functions
2. Customer Service Representatives
    a. Need access to customer information and history
    b. Create and manage support tickets
    c. Respond to customer feedback
    d. Require collaboration tools
    e. Need access to knowledge base and templates
3. Customers
    a. Submit feedback and create tickets
    b. Track ticket status
    c. Update personal information
    d. Communicate with representatives
    e. Access self-service resources

2.4 Operating Environment

The product will be operated on the following platforms:

Server Environment:

- Cloud-based deployment (AWS or Azure)
- Linux-based operating system
- Containerized deployment using Docker
- Load balancing for high availability

Client Environment:

- Web browsers: Chrome, Firefox, Safari, Edge (latest two versions)
- Mobile applications: iOS 14+ and Android 10+
- Desktop applications: Windows 10+, macOS 11+

Hardware requirements for the server:

- CPU: 8+ cores
- RAM: 16+ GB
- Storage: 500+ GB SSD
- Network: High-speed internet connection (100+ Mbps)

2.5 Design and Implementation Constraints

The system is developed with the following constraints:

- Frontend: React.js for web application, React Native for mobile applications
- Backend: Node.js with Express framework
- Database: MongoDB for customer data, Redis for caching
- Authentication: OAuth 2.0 and JWT for secure access
- API Design: RESTful architecture with JSON payloads
- Security: HTTPS/TLS 1.3 for all communications
- Compliance: GDPR and CCPA requirements for data privacy
- Integration: Must support third-party CRM systems via API
- Accessibility: Must comply with WCAG 2.1 Level AA standards

2.6 User Documentation

The following user documentation will be provided:

- Online Help System: Context-sensitive help accessible within the application
- User Manuals: Role-specific guides for administrators, representatives, and customers
- Video Tutorials: Step-by-step demonstrations of key features
- Quick Start Guides: Onboarding materials for new users
- API Documentation: Comprehensive guide for developers integrating with the system
- Knowledge Base: Searchable repository of common questions and solutions

2.7 Assumptions and Dependencies

Assumptions:

- Users have basic computer literacy
- The system will be accessible 24/7 with 99.9% uptime
- Reliable internet connectivity is available for all users
- Mobile users have smartphones with up-to-date operating systems
- Data migration from legacy systems will be performed before system launch
- Initial training will be provided to all administrative users

Dependencies:

- Third-party email service for notifications
- SMS gateway for mobile alerts
- Cloud infrastructure provider (AWS/Azure)
- Payment processing service for subscription management
- Analytics libraries for data visualization
- External authentication providers (Google, Microsoft, etc.)
- CDN services for media content delivery

3. External Interface Requirements

3.1 User Interfaces

The system provides a modern, responsive interface for users across devices:

Dashboard Interface:

- Customizable widgets for key metrics
- Role-based views with relevant information
- Responsive design that adapts to different screen sizes
- Dark/light mode toggle for user preference
- Accessibility features (screen reader compatibility, keyboard navigation)

Ticket Management Interface:

- List view with filtering and sorting options
- Detailed ticket view with communication history
- Status indicators using color coding
- Search functionality with advanced filters
- Drag-and-drop interface for ticket assignment

Customer Portal:

- Simple, intuitive design for non-technical users
- Guided ticket creation process
- Feedback submission forms with rating scales
- Status tracking with visual indicators

- Account management section for personal information

Mobile Interface:

- Native applications for iOS and Android
- Push notification support
- Offline capability for basic functions
- Touch-optimized controls
- Biometric authentication option

3.2 Hardware Interfaces

Server Side:

- Cloud infrastructure with auto-scaling capabilities
- Load balancers for high availability
- Database servers with replication
- Backup systems with automated scheduling
- Monitoring systems for performance metrics

Client Side:

- Standard web browsers on desktop and mobile devices
- Mobile devices running iOS 14+ or Android 10+
- Minimum 4GB RAM recommended for optimal performance
- Camera and microphone access for media attachments (optional)
- Printer interface for report generation

3.3 Software Interfaces

The system interfaces with the following software components:

- Email Servers: SMTP/IMAP protocol for sending notifications and receiving emails
- SMS Gateway: API integration for text message notifications
- CRM Systems: REST API integration with popular CRM platforms
- Payment Processors: Secure API integration for subscription management
- Analytics Platforms: Data export to business intelligence tools
- Social Media Platforms: API integration for feedback collection
- Knowledge Base Systems: Integration with existing documentation

- Authentication Services: OAuth 2.0 support for single sign-on

3.4 Communications Interfaces

The system utilizes the following communication protocols:

- HTTPS: Secure communication for all web traffic
- WebSockets: Real-time updates for dashboards and notifications
- REST API: Integration with external systems
- SMTP/IMAP: Email communication
- Push Notification Services: Mobile alerts
- WebRTC: Optional video chat support for customer interactions
- Webhooks: Event-driven integration with third-party systems

4. System Features

4.1 System Login

4.1.1 Description and Priority

The system provides secure authentication for all users with role-based access control. Priority: High

4.1.2 Stimulus/Response Sequences

1. User navigates to the login page
2. System displays login form with username/email and password fields
3. User enters credentials and submits the form
4. System validates credentials against the database
5. If valid, system redirects to appropriate dashboard based on user role
6. If invalid, system displays error message and prompts for retry
7. System provides "Forgot Password" option for recovery

4.1.3 Functional Requirements

- The system shall support multiple authentication methods (username/password, SSO, multi-factor)
- The system shall enforce password complexity requirements

- The system shall lock accounts after multiple failed attempts
- The system shall log all authentication attempts
- The system shall support session timeout after a configurable period of inactivity
- The system shall allow users to reset passwords via email verification

4.2 Manage Customer Feedback

4.2.1 Description and Priority

This feature enables collection, categorization, and response to customer feedback across multiple channels. Priority: High

4.2.2 Stimulus/Response Sequences

1. Feedback is received through web form, email, or integrated channels
2. System automatically categorizes feedback using AI analysis
3. System assigns priority based on sentiment and keywords
4. System routes feedback to appropriate representative
5. Representative reviews and responds to feedback
6. System tracks response time and resolution
7. System sends satisfaction survey after resolution

4.2.3 Functional Requirements

- The system shall collect feedback from multiple channels (web, email, social media)
- The system shall categorize feedback using configurable rules and AI
- The system shall assign priority levels to feedback items
- The system shall route feedback to appropriate teams/individuals
- The system shall track response times and resolution status
- The system shall provide templates for common responses
- The system shall support bulk operations for similar feedback items

4.3 Create and Track Support Tickets

4.3.1 Description and Priority

This feature enables creation, assignment, and lifecycle management of customer support tickets. Priority: Critical

4.3.2 Stimulus/Response Sequences

1. User (customer or representative) creates a new ticket
2. System assigns a unique ID and timestamps the creation
3. System categorizes and prioritizes the ticket
4. System assigns ticket to appropriate representative or queue
5. Representative updates ticket status and adds communication
6. System notifies customer of updates
7. Representative resolves ticket and system requests confirmation
8. Customer confirms resolution or reopens ticket

4.3.3 Functional Requirements

- The system shall generate unique identifiers for each ticket
- The system shall support ticket creation by customers and representatives
- The system shall provide configurable categories and priorities
- The system shall implement customizable workflows for different ticket types
- The system shall track ticket history and all communications
- The system shall provide escalation paths for unresolved tickets
- The system shall support SLA tracking with alerts for approaching deadlines
- The system shall provide knowledge base integration for solution suggestions

4.4 Generate Analytics Reports

4.4.1 Description and Priority

This feature provides comprehensive analytics and reporting on customer experience metrics. Priority: Medium

4.4.2 Stimulus/Response Sequences

1. User navigates to analytics section
2. System displays dashboard with key performance indicators
3. User selects report type and parameters (date range, filters)
4. System generates report with visualizations and data tables
5. User can export report in various formats (PDF, Excel, CSV)
6. User can schedule recurring reports
7. System delivers scheduled reports via email

4.4.3 Functional Requirements

- The system shall provide real-time dashboards with key metrics
- The system shall support custom report creation with various parameters
- The system shall generate visualizations (charts, graphs) for data analysis
- The system shall support export in multiple formats (PDF, Excel, CSV)
- The system shall allow scheduling of recurring reports
- The system shall track trends and highlight anomalies
- The system shall provide team and individual performance metrics
- The system shall support drill-down capabilities for detailed analysis

5. Other Non-functional Requirements

5.1 Performance Requirements

- The system shall support a minimum of 1000 concurrent users
- Web page load time shall not exceed 2 seconds under normal conditions
- API response time shall not exceed 200ms for 95% of requests
- Database queries shall complete within 500ms for 99% of operations
- The system shall handle at least 10,000 tickets per day
- Real-time dashboards shall update within 5 seconds of data changes
- Report generation shall complete within 30 seconds for standard reports
- Mobile app response time shall not exceed 3 seconds on 4G connections

5.2 Safety Requirements

- The system shall implement automatic backup of all data

- Critical operations shall require confirmation before execution
- The system shall maintain audit logs for all significant actions
- Recovery point objective (RPO) shall be no more than 1 hour
- Recovery time objective (RTO) shall be no more than 4 hours
- The system shall prevent concurrent editing of the same ticket
- The system shall implement rate limiting to prevent abuse

5.3 Security Requirements

- All data transmission shall be encrypted using TLS 1.3
- Passwords shall be stored using industry-standard hashing algorithms
- The system shall implement role-based access control
- Authentication shall support multi-factor options
- Session management shall include automatic timeout
- The system shall maintain detailed access logs
- Sensitive customer data shall be encrypted at rest
- The system shall comply with GDPR and CCPA requirements
- Security audits shall be conducted quarterly

5.4 Software Quality Attributes

Reliability:

- The system shall achieve 99.9% uptime
- The system shall gracefully handle unexpected inputs
- Automated tests shall cover at least 85% of code

Maintainability:

- The system shall use modular architecture
- Code shall follow consistent style guidelines
- Documentation shall be maintained for all interfaces
- The system shall support feature toggles for phased rollouts

Performance:

- Database indexes shall be optimized for common queries
- Caching shall be implemented for frequently accessed data

- Background processes shall not impact user experience

Security:

- Regular security assessments shall be conducted
- The system shall be resistant to common attack vectors (XSS, CSRF, SQL injection)
- Security patches shall be applied within 48 hours of release

Usability:

- The UI shall be consistent across all system components
- All features shall be accessible via keyboard navigation
- The system shall support internationalization

5.5 Business Rules

- Support tickets must be acknowledged within 1 hour during business hours
- Critical tickets must be escalated if not resolved within 4 hours
- Customer feedback must be categorized within 24 hours of receipt
- Negative feedback with sentiment score below threshold must trigger manager notification
- Customer satisfaction surveys must be sent within 24 hours of ticket resolution
- Subscription renewal notifications must be sent 30, 15, and 5 days before expiration
- Reports for managers must be generated automatically at month-end
- User passwords must be changed every 90 days

6. Other Requirements

Data and Category Requirements

The system handles various types of data including:

- Customer profiles (personal information, preferences, interaction history)
- Support tickets (categories, priorities, status, communications)
- Feedback (source, sentiment, category, response)
- Performance metrics (response times, resolution rates, satisfaction scores)

Data retention policies must comply with relevant regulations:

- Active customer data: retained while account is active
- Resolved tickets: retained for 3 years
- Anonymized analytics: retained indefinitely
- Audit logs: retained for 7 years

Ethical Requirements

The system must operate ethically with respect to:

- Customer privacy: data collection limited to necessary information
- Data usage: transparent policies on how data is used
- Bias prevention: algorithms regularly audited for bias
- Accessibility: equal access for users with disabilities
- Cultural sensitivity: support for multiple languages and customs

References

Websites:

- www.wikipedia.org
- https://www.geeksforgeeks.org/use-case-diagram-for-customer-experience-management-system/
- https://www.researchgate.net/publication/customer-management-systems
- https://www.researchgate.net/publication/customer-management-systems

Books:

- Software Engineering, Seventh Edition Ian Sommerville
- IEEE Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Customer Experience Management: A Revolutionary Approach to Connecting with Your Customers by Bernd H. Schmitt

Appendix A: Glossary

- **CEMS**: Customer Experience Management System

- **SLA**: Service Level Agreement - defines the level of service expected from a vendor
- **Ticket**: A record of a customer issue or request
- **Dashboard**: Visual display of key performance indicators
- **KPI**: Key Performance Indicator - a measurable value that demonstrates how effectively objectives are being achieved
- **API**: Application Programming Interface - allows different software systems to communicate
- **GDPR**: General Data Protection Regulation - EU regulation on data protection and privacy
- **CCPA**: California Consumer Privacy Act - regulation protecting consumer privacy rights
- **SSO**: Single Sign-On - authentication process that allows users to access multiple applications with one set of credentials
- **Webhook**: Automated message sent from one application to another triggered by specific events
- **RPO**: Recovery Point Objective - maximum targeted period of data loss acceptable in disaster recovery
- **RTO**: Recovery Time Objective - targeted duration of time within which a business process must be restored

PROGRAM: 2

2.1OBJECTIVE	2.2THEORY	2.3 PROCEDURE	2.4 OUTPUT

2.1 OBJECTIVE: **Draw the use case diagram and specify the role of each of the actors. Also state the precondition, post condition and function of each use case.**

2.2 THEORY:

Creating Use Case Diagrams

We start by identifying as many actors as possible. You should ask how the actors interact with the system to identify an initial set of use cases. Then, on the diagram, you connect the actors with the use cases with which they are involved. If actor supplies information, initiates the use case, or receives any information as a result of the use case, then there should be an association between them.

The following elements are available in a usecase diagram.

- Actor
- UseCase
- Association
- Directed Association
- Generalization

Actor

An actor defines a coherent set of roles that users of an entity can play when interacting with the entity. An actor may be considered to play a separate role with regard to each use case with which it communicates

Use Case

The use case construct is used to define the behavior of a system or other semantic entity without revealing the entity's internal structure. Each use case specifies a sequence of actions, including variants, that the entity can perform, interacting with actors of the entity.

2.3 PROCEDURE:

Procedure for creating Actor

In order to create Actor, click **[Toolbox] -> [UseCase] -> [Actor]** button and click the position where to place Actor. Actor is shown in the form of stick man or rectangle with icon that is decoration view.

To display actor in decoration view, select **[Format] -> [Stereotype Display] -> [Decoration]** menu item or select **[Decoration]** item in [combo button on toolbar.

Procedure for creating multiple Use Cases used by Actor at once

In order to create multiple Use Cases related to Actor at once, use shortcut creation syntax of Actor.

1. At the Actor's quick dialog, enter Use Case's name after "-()" string. To create multiple Use Cases, enter same but separate Use Case's name by "," character.
2. And press **[Enter]** key. Several Use Cases associated with the Actor are created and arranged vertically.

Procedure for creating Use Case

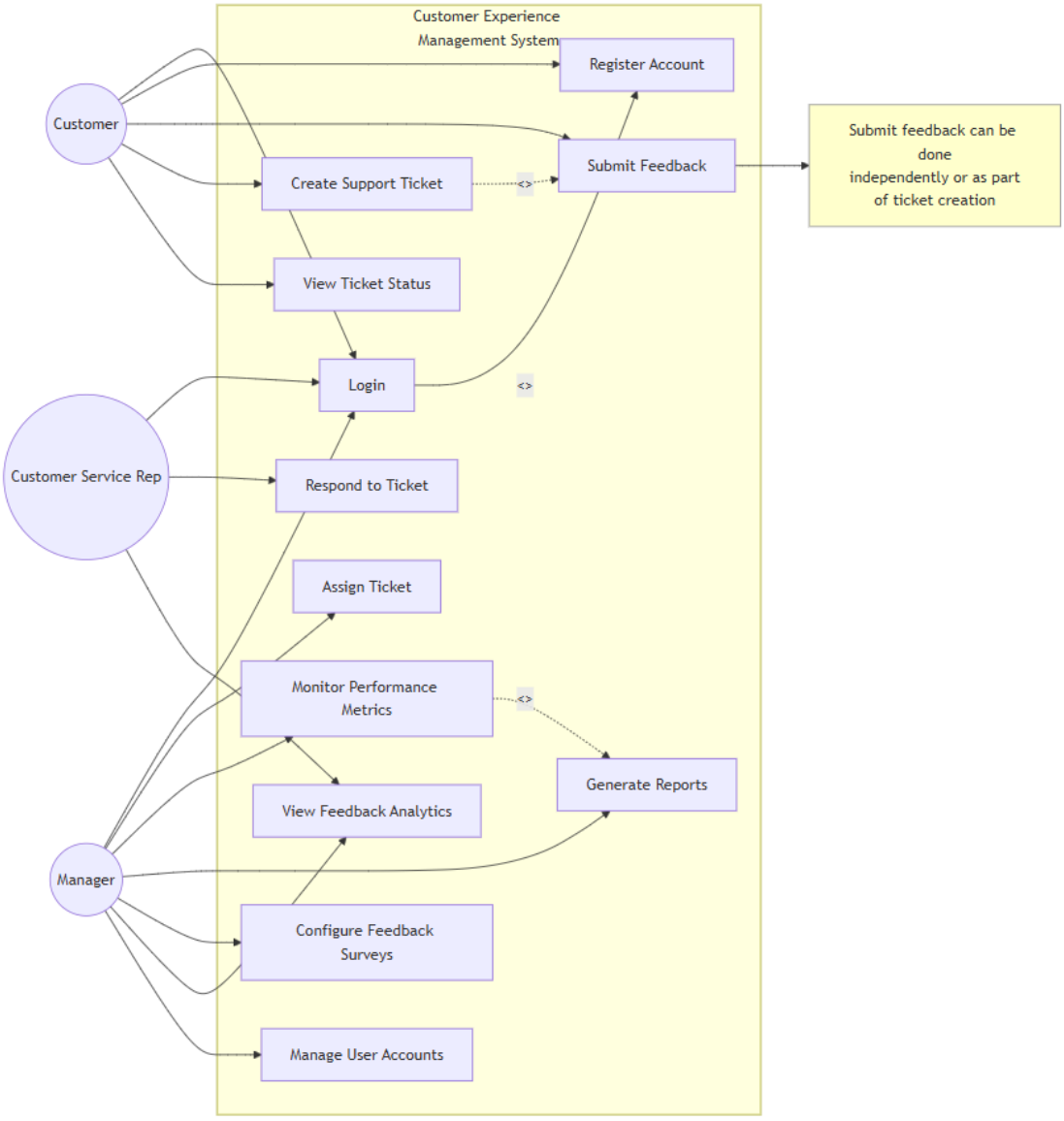
In order to create Use Case, click **[Toolbox]** -> **[Use Case]** button and click the position where to place UseCase on the **[main window]**.

UseCase is expressed in the forms of textual, decoration, iconic. To change UseCase's view style, select menu item under **[Format] -> [Stereotype Display]** .

Procedure for adding Extension

An extension point references one or a collection of locations in a use case where the use case may be extended.
To edit Extension Points of UseCase, click UseCase's **[Collection Editor...]** popup menu or click button of **[Extension Points]** collection property.

2.4 OUTPUT:


Customer Experience Management System
Register Account
Customer
Submit Feedback
Submit feedback can be done independently or as part of ticket creation
Create Support Ticket
<<
View Ticket Status
Login
<<
Customer Service Rep
Respond to Ticket
Assign Ticket
Monitor Performance Metrics
<<
Generate Reports
View Feedback Analytics
Manager
Configure Feedback Surveys
Manage User Accounts