Software Requirement Specification

Contact List

Haraprasad Khatua

Madhusmita Maharana

Priyanka Priyadarsinee Das

Trupti Mayee Ghadei

Pragyan Paramita Rout

**VERSION RECORD**

Version record showing the amendments effected to the SRS is given below. It is to be noted that the sections having latest effective dates are the applicable ones. Section Version Details of Version Effective Date Approved by 1.0 Initial Version 9.28.2006

**TABLE OF CONTENTS**

1. Introduction...............................,.......................3-4

1.1. purpose..........,............................,........................5-6

1.2. Scope........,............................................................7-8

2. Over view...........................................................10-11

3. Requirement elication............................12-14

4. Process.............................................,.......,.............15-18

5. Issue..........................................................................19-21

6. Ambiguous........................................................22-24.

7. Contradiction..................................................25-26

8. Reference............................................................27

**1.Introduction**

**Consider The Following Scenario**

A contact list is an organized collection of information about individuals or organizations. It typically includes names, phone numbers, email addresses, and sometimes additional details such as physical addresses, job titles, and notes about the relationship or context. Here's an introduction to understanding and creating an effective contact list:

Importance of a Contact List

1. Organization: Keeps all your important contacts in one place, making it easy to find and reach out to people when needed.

2. Communication: Facilitates efficient communication by providing quick access to contact details.

3. Networking: Helps maintain professional relationships and expand your network by keeping track of connections.

4. Productivity: Saves time and effort, reducing the need to search for contact details when you need them.

Key Components of a Contact List

1. Name: Full name of the contact.

2. Phone Number: Primary contact number, which can be a mobile, home, or work number.

3. Email Address: Primary email address for correspondence.

4. Address: Physical address, if relevant.

5. Job Title and Company: Important for professional contacts.

6. Notes: Any additional information that might be useful, such as how you met the person, birthdays, or personal preferences.

Tools for Managing Contact Lists

1. Spreadsheets: Simple and flexible; tools like Microsoft Excel or Google Sheets.

2. Contact Management Software: Dedicated software like Google Contacts, Microsoft Outlook, or Apple Contacts.

3. CRM Systems: For businesses, Customer Relationship Management (CRM) systems like Salesforce or HubSpot provide advanced features for managing contacts and interactions.

4. Mobile Apps: Apps like Contacts+ or CamCard can help manage contacts on the go.

Tips for Maintaining a Contact List

1. Regular Updates: Ensure contact details are up-to-date. Regularly review and update information.

2. Categorization: Organize contacts into categories (e.g., family, friends, work, clients) for easier access.

3. Backup: Regularly back up your contact list to prevent loss of information.

4. Privacy: Be mindful of privacy concerns. Ensure sensitive information is protected and only shared with authorized individually.

**1.1.Purpose**

A contact list serves several key purposes, whether for personal, professional, or organizational use. Here are some of the main reasons why maintaining a contact list is important:

1. Efficient Communication

- Quick Access: Provides immediate access to contact information, saving time and effort when you need to reach someone.

- Multiple Channels: Includes various methods of communication (phone, email, address), allowing you to choose the most appropriate one.

2. Relationship Management

- Personal Relationships: Helps keep track of friends, family, and acquaintances, making it easier to maintain personal connections.

- Professional Networking: Essential for managing professional relationships, staying in touch with colleagues, clients, and industry contacts.

3. Business Operations

- Client Management: Central to customer relationship management (CRM), helping businesses maintain detailed records of their clients and interactions.

- Supplier and Partner Coordination: Useful for managing details about suppliers, partners, and other business contacts, facilitating smooth operations.

4. Event Planning

- Invitations and Follow-ups: Essential for organizing events, sending invitations, and following up with attendees.

- Group Communication: Simplifies the process of reaching out to multiple people for meetings, updates, or announcements.

5. Record Keeping

- Historical Reference: Acts as a historical record of interactions, useful for recalling previous communications or agreements.

- Documentation: Helps document and track changes in contact information over time.

6. Emergency Situations

- Crisis Management: Provides a reliable list of contacts to reach out to in case of emergencies, ensuring timely communication.

- Support Network: Helps identify and reach out to a support network quickly when needed.

7. Marketing and Outreach

- Targeted Campaigns: Allows businesses to segment contacts for targeted marketing campaigns, enhancing the effectiveness of outreach efforts.

- Customer Engagement: Facilitates regular engagement with customers through newsletters, promotions, or updates.

8. Collaboration

- Team Coordination: Enables team members to easily communicate and collaborate, improving project management and teamwork.

- Resource Sharing: Simplifies the sharing of resources and information among a group or organization.

Maintaining a well-organized contact list is fundamental for effective communication and relationship management across various contexts, making it a vital tool for individuals and organizations alike

**1.2.Scope**

The scope of a contact list refers to the range and extent of information it encompasses, its intended use, and the way it is managed and maintained. Here's a breakdown of the different aspects that define the scope of a contact list:

1. Types of Contacts

- Personal Contacts: Friends, family, acquaintances.

- Professional Contacts: Colleagues, clients, business partners, industry peers.

- Organizational Contacts: Employees, departments, stakeholders.

- Service Providers: Vendors, suppliers, contractors.

- Emergency Contacts: Medical professionals, emergency services, key contacts for crises.

2. Information Included

- Basic Information: Names, phone numbers, email addresses.

- Additional Details: Physical addresses, job titles, company names, social media profiles.

- Custom Fields: Notes, birthdays, relationship context, preferences.

3. Purpose and Use

- Personal Use: Keeping track of friends and family, social events, personal networking.

- Professional Use: Managing business relationships, client management, networking, sales and marketing efforts.

- Organizational Use: Internal communication, project management, stakeholder engagement.

- Event Planning: Organizing attendees, sending invitations, managing RSVPs.

4. Management and Maintenance

- Regular Updates: Ensuring information is current and accurate.

- Categorization: Grouping contacts into relevant categories for easier access and management.

- Data Security: Protecting sensitive information and ensuring privacy.

5. Tools and Platforms

- Digital Tools: Spreadsheets (Excel, Google Sheets), contact management apps (Google Contacts, Apple Contacts).

- CRM Systems: For businesses, using Customer Relationship Management systems (Salesforce, HubSpot) to manage extensive contact information and interactions.

- Mobile Apps: Apps for managing contacts on-the-go (Contacts+, CamCard).

6. Integration and Accessibility

- Cross-Platform Integration: Syncing contacts across multiple devices and platforms.

- Accessibility: Ensuring contact lists are easily accessible when needed, whether through mobile devices, desktop applications, or cloud-based services.

7. Legal and Ethical Considerations

- Data Privacy: Complying with data protection regulations (e.g., GDPR, CCPA).

- Consent: Ensuring consent is obtained for storing and using personal information.

- Usage Policies: Establishing clear policies on how contact information is used and shared within an organization.

The scope of a contact list can vary greatly depending on its intended use and the context in which it is maintained. Whether for personal,

**2.Over view**

A contact list is a collection of data about individuals or organizations, typically used for communication purposes. Here’s an overview of what it generally includes and its uses:

Components of a Contact List:

1. Basic Information:

- Names: Full names of the individuals or organizations.

- Phone Numbers: Mobile, landline, or office numbers.

- Email Addresses: For electronic communication.

2. Additional Details:

- Physical Addresses: Residential or business addresses.

- Job Titles and Companies: Professional information for business contacts.

- Social Media Handles: Links to profiles on platforms like LinkedIn, Twitter, etc.

3. Categorization:

- Groups or Categories: Contacts can be grouped (e.g., family, friends, colleagues).

- Tags or Labels: Additional tags to facilitate search and organization.

Uses of a Contact List:

1. Personal Use:

- Staying in Touch: Easily reach friends and family.

- Event Planning: Send invitations and manage RSVPs.

2. Professional Use:

- Networking: Maintain connections with colleagues, clients, and partners.

- Customer Management: Keep track of client information and communication history.

- Marketing: Targeted outreach for promotions and information dissemination.

3. Emergency Situations:

- Emergency Contacts: Quick access to important numbers in case of an emergency.

Tools for Managing Contact Lists:

1. Digital Address Books: Integrated in smartphones, email clients (e.g., Google Contacts, Outlook).

2. Customer Relationship Management (CRM) Systems: Advanced tools for managing business contacts (e.g., Salesforce, HubSpot).

3. Spreadsheets: Simple method for organizing and maintaining a contact list manually.

Best Practices

1. Regular Updates: Ensure contact information is current.

2. Privacy: Protect sensitive information and comply with data protection regulations.

3. Backup: Regularly back up contact lists to prevent data loss.

3.Requirements Elicitation

1.1 Identify Stakeholders

Determine who will use the application (e.g., individual users, professionals) and who will be involved in its development and maintenance.

1.2 Gather Requirements

Conduct interviews and surveys with potential users to understand their needs.

Hold workshops and brainstorming sessions with stakeholders.

Analyze existing systems and competitor applications to identify best practices and features to include.

1.3 Document Requirements

Create initial drafts of requirements based on gathered information.

Categorize requirements into functional and non-functional categories.

2. Requirements Analysis

2.1 Validate Requirements

Review requirements with stakeholders to ensure they accurately reflect their needs.

Prioritize requirements based on their importance and feasibility.

2.2 Resolve Conflicts

Address any conflicting requirements or constraints.

Reach a consensus among stakeholders on critical features and functions.

2.3 Create Use Cases

Develop use cases or user stories to describe how users will interact with the system.

Identify and document all possible scenarios and edge cases.

3. Requirements Specification

3.1 Write the SRS Document

Introduction: Define the purpose, scope, and context of the application.

Overall Description: Provide a high-level overview of the system, including product perspective, functions, user characteristics, constraints, and assumptions.

Specific Requirements: Detail functional and non-functional requirements.

Functional Requirements: Describe each feature in detail, including adding, editing, deleting, viewing, and searching contacts, as well as categorizing and importing/exporting contacts.

Non-Functional Requirements: Specify performance, usability, reliability, security, and design constraints.

3.2 Review and Revise

Circulate the draft SRS among stakeholders for feedback.

Revise the document based on feedback to ensure accuracy and completeness.

4. Requirements Validation

4.1 Conduct Review Meetings

Hold formal review sessions with stakeholders to validate the requirements.

Ensure all requirements are testable and feasible within the project constraints.

4.2 Finalize the SRS

Obtain formal approval of the SRS from all stakeholders.

Baseline the SRS for use in subsequent project phases.

5. Management of Requirements

5.1 Requirements Traceability

Create a traceability matrix to map requirements to design, implementation, and testing artifacts.

Ensure each requirement is tracked through the project lifecycle.

5.2 Change Management

Establish a process for handling changes to requirements.

Maintain version control of the SRS document to track changes and updates.

5.3 Communication

Ensure ongoing communication with stakeholders to manage expectations and provide updates on requirement changes.

6. Development and Testing

6.1 Design and Implementation

Use the SRS as a reference to guide system design and development.

Develop the application iteratively, continuously referring to the SRS to ensure compliance with requirements.

6.2 Testing

Create test plans and cases based on the SRS.

Perform unit, integration, and system testing to verify that the application meets the specified requirements.

6.3 User Acceptance Testing

Conduct user acceptance testing with stakeholders to ensure the system meets their needs.

Address any issues or gaps identified during testing.

7. Deployment and Maintenance

7.1 Deployment

Deploy the application to the production environment.

Ensure all deployment activities align with the requirements specified in the SRS.

7.2 Maintenance

Provide ongoing support and maintenance for the application.

Manage any new requirements or changes through the established change management process.

7.3 Feedback Loop

Continuously gather user feedback to identify areas for improvement.

Update the SRS as needed to reflect changes in user needs and system functionality.

4.Process

1. Add Contact

Description: Users should be able to add new contacts to the list.

Inputs: Name, phone number, email address, physical address, company name, job title, birthday, notes.

Outputs: Confirmation of successful addition, error messages for invalid input.

Processing: Validate input fields for correctness (e.g., proper email format, phone number format). Save the contact information to the database.

2. Edit Contact

Description: Users should be able to modify existing contact details.

Inputs: Modified fields such as name, phone number, email, etc.

Outputs: Confirmation of successful update, error messages for invalid input.

Processing: Retrieve the contact details, apply changes, and update the database.

3. Delete Contact

Description: Users should be able to remove contacts from the list.

Inputs: Contact identifier (e.g., ID, name).

Outputs: Confirmation of deletion, error message if contact not found.

Processing: Prompt user for confirmation before deletion. Remove contact details from the database.

4. View Contact

Description: Users should be able to view detailed information of a selected contact.

Inputs: Contact identifier (e.g., ID, name).

Outputs: Display contact details (name, phone number, email, etc.).

Processing: Retrieve contact details from the database and display them.

5.Search Contact

Description: Users should be able to search for contacts by various criteria.

Inputs: Search criteria (name, phone number, email, etc.).

Outputs: List of contacts matching the search criteria.

Processing: Query the database with search criteria and display matching contacts.

6.Categorize Contact

Description: Users should be able to assign contacts to categories.

Inputs: Contact identifier, category (e.g., Family, Friends, Work).

Outputs: Confirmation of successful categorization.

Processing: Update the contact's category in the database.

7. Import Contacts

Description: Users should be able to import contacts from external files.

Inputs: File upload (CSV, vCard).

Outputs: Confirmation of successful import, error messages for invalid file formats.

Processing: Parse the file, validate data, and insert contacts into the database.

8. Export Contacts

Description: Users should be able to export contacts to external files.

Inputs: Export format selection (CSV, vCard).

Outputs: Downloadable file with contact data.

Processing: Retrieve contacts from the database and generate the file in the selected format.

2. Non-Functional Requirements

2.1 Performance

Description: The system should perform efficiently under various loads.

Requirements:

System should respond to user actions within 2 seconds.

System should handle up to 10,000 contacts without performance degradation.

2.2 Usability

Description: The system should be easy to use for all users.

Requirements:

Interface should be intuitive and user-friendly.

System should provide clear instructions and feedback to users.

System should support accessibility features for users with disabilities.

2.3 Reliability

Description: The system should be reliable and available.

Requirements:

System should have an uptime of 99.9%.

System should ensure data integrity and prevent data loss.

System should handle errors gracefully and provide meaningful error messages.

2.4 Security

Description: The system should protect user data and ensure privacy.

Requirements:

System should implement authentication mechanisms to restrict access to authorized users.

System should encrypt sensitive data to protect user privacy.

System should comply with relevant data privacy laws and regulations.

2.5 Maintainability

Description: The system should be maintainable and extensible.

Requirements:

Code should be modular and well-documented to facilitate maintenance.

System should be designed to accommodate future enhancements and scaling.

System should support logging and monitoring for troubleshooting and performance tuning.

2.6 Compatibility

Description: The system should be compatible with various devices and platforms.

Requirements:

System should be accessible on both mobile and desktop platforms.

System should support major web browsers (Chrome, Firefox, Safari, Edge).

System should be compatible with different operating systems (Windows, macOS, Linux, iOS, Android).

3. Design Constraints

3.1 Technology Stack

Description: The technologies used should support the system requirements and future growth.

Requirements:

System should be developed using a technology stack that supports cross-platform compatibility (e.g., React Native, Flutter).

Database should be chosen based on scalability and performance needs (e.g., PostgreSQL, MongoDB).

System architecture should support future scalability and maintainability.

3.2 Compliance

Description: The system should comply with relevant standards and regulations.

Requirements:

System should adhere to industry best practices for security and data protection.

System should comply with data privacy regulations (e.g., GDPR, CCPA).

**5.Issue**

Technical Issues

Data Synchronization

Issue: Difficulty in keeping contact data synchronized across multiple devices.

Mitigation: Implement real-time data synchronization using cloud-based solutions (e.g., Firebase, AWS Amplify). Ensure data consistency with conflict resolution strategies.

Performance Degradation

Issue: Performance might degrade as the number of contacts increases.

Mitigation: Optimize database queries, use indexing, and implement efficient data retrieval algorithms. Conduct performance testing and tuning regularly.

Cross-Platform Compatibility

Issue: Ensuring the application works seamlessly on different operating systems and devices.

Mitigation: Use cross-platform development frameworks (e.g., React Native, Flutter). Perform extensive testing on various devices and operating systems.

Usability Issues

User Interface Complexity

Issue: The interface might become complex and difficult to use, especially for non-technical users.

Mitigation: Conduct user experience (UX) research and usability testing. Simplify the interface and provide clear instructions and help features.

Accessibility

Issue: The application may not be accessible to users with disabilities.

Mitigation: Follow accessibility guidelines (e.g., WCAG). Implement features like screen reader support, high-contrast modes, and keyboard navigation.

Data Security and Privacy Issues

Data Breaches

Issue: Unauthorized access to user data.

Mitigation: Implement strong authentication and authorization mechanisms. Encrypt sensitive data both at rest and in transit. Regularly update and patch the application to fix vulnerabilities.

Compliance with Privacy Regulations

Issue: Ensuring the application complies with data privacy laws and regulations (e.g., GDPR, CCPA).

Mitigation: Regularly review and update privacy policies. Implement features that allow users to manage their data (e.g., data export, deletion). Conduct privacy impact assessments.

4.4 Functional Issues

Data Loss

Issue: Accidental deletion or corruption of contact data.

Mitigation: Implement data backup and recovery mechanisms. Provide users with options to restore deleted contacts within a certain period.

Integration with Other Services

Issue: Difficulty in integrating with external services (e.g., email clients, social media).

Mitigation: Use standardized APIs and protocols for integration. Ensure comprehensive testing of integration points.

Project Management Issues

Scope Creep

Issue: Uncontrolled changes or additions to the project scope.

Mitigation: Implement a change management process. Clearly define project scope and requirements upfront. Regularly review project progress and make necessary adjustments.

Resource Constraints

Issue: Limited resources (time, budget, personnel) may affect project delivery.

Mitigation: Conduct thorough project planning and resource allocation. Prioritize critical features and tasks. Monitor project progress and adjust resource allocation as needed.

User Acceptance Issues

User Resistance

Issue: Users may resist adopting the new application.

Mitigation: Engage users early in the development process. Provide training and support. Highlight the benefits and improvements the application

Inadequate Testing

Issue: Insufficient testing may lead to undetected issues.

Mitigation: Implement a comprehensive testing strategy, including unit, integration, system, and user acceptance testing. Use automated testing tools where appropriate.

**6.Ambiguous**

Ambiguous Requirements and Clarifications

Ambiguity in "Add Contact"

Ambiguity: What specific details are required for each contact? Are all fields mandatory?

Clarification:

Mandatory fields: Name, phone number.

Optional fields: Email address, physical address, company name, job title, birthday, notes.

Validation rules: Specify format requirements for each field (e.g., email must follow standard email format).

Ambiguity in "Edit Contact"

Ambiguity: Are there any restrictions on which fields can be edited? How should the system handle invalid updates?

Clarification:

Editable fields: All fields (name, phone number, email, etc.) can be edited.

Handling invalid updates: Provide real-time validation and error messages for incorrect formats or missing mandatory information.

Ambiguity in "Delete Contact"

Ambiguity: What happens to linked or dependent data when a contact is deleted?

Clarification:

Linked data: Ensure no dependent data (e.g., linked notes, categories) is orphaned or left in an inconsistent state.

Confirmation: Provide a confirmation dialog before deletion, explaining what data will be affected.

Ambiguity in "Search Contact"

Ambiguity: What fields can be searched? Is the search case-sensitive?

Clarification:

Searchable fields: Name, phone number, email, company name, job title.

Search behavior: Case-insensitive, partial matches allowed (e.g., searching for "John" should return "Johnathan").

Ambiguity in "Categorize Contact"

Ambiguity: How many categories can a contact belong to? Are categories predefined or user-defined?

Clarification:

Category assignment: A contact can belong to multiple categories.

Category types: Allow both predefined (e.g., Family, Friends, Work) and user-defined categories.

Ambiguity in "Import Contacts"

Ambiguity: What happens if there are conflicts or duplicate contacts during import? What file formats are supported?

Clarification:

Conflict handling: Provide options to merge, overwrite, or skip duplicates.

Supported formats: CSV, vCard.

Validation: Validate file content before import and provide error reports for invalid entries.

Ambiguity in "Export Contacts"

Ambiguity: Are all fields included in the export? How is the data formatted?

Clarification:

Included fields: All contact details (name, phone number, email, etc.) are included.

Data formatting: Follow standard CSV and vCard formats. Provide options for users to select specific fields if needed.

Ambiguity in Performance Requirements

Ambiguity: What specific actions should meet the 2-second response time requirement? Under what conditions?

Clarification:

Actions: Adding, editing, deleting, viewing, and searching contacts.

Conditions: Response time should be met under normal operating conditions with up to 10,000 contacts in the database.

Ambiguity in Usability Requirements

Ambiguity: What constitutes a "user-friendly" interface? How is usability measured?

Clarification:

User-friendly: The interface should be intuitive, with a clear layout, easy navigation, and consistent design.

Usability measurement: Conduct usability testing with target users, gather feedback, and make iterative improvements.

Ambiguity in Security Requirements

Ambiguity: What specific security measures are required? How is sensitive data defined?

Clarification:

Security measures: Implement authentication (e.g., username/password, two-factor authentication), encryption (SSL/TLS for data in transit, AES for data at rest), and regular security audits.

Sensitive data: Define sensitive data as any personal information that could identify an individual, including names, phone numbers, and email addresses.

Ambiguity in Compatibility Requirements

Ambiguity: Which versions of operating systems and browsers are supported?

Clarification:

Supported platforms:

Mobile: Android (version 8.0 and above), iOS (version 12 and above).

Desktop: Windows 10, macOS 10.14 and above, major Linux distributions.

Browsers: Latest versions of Chrome, Firefox, Safari, and Edge.

**7.Contradiction**

Potential Contradictions and Resolutions

Contradiction in "Add Contact" vs. "Edit Contact"

Contradiction: The "Add Contact" requirement specifies that some fields are mandatory (e.g., name, phone number), but the "Edit Contact" requirement does not specify if these fields remain mandatory during editing.

Resolution:

Clarify that during the "Edit Contact" process, the mandatory fields (name, phone number) must still be validated to ensure they are not left empty or invalid.

Update the requirement to state: "All fields specified as mandatory in the 'Add Contact' feature must remain valid and non-empty during editing."

Contradiction in "Delete Contact" vs. Data Integrity

Contradiction: The "Delete Contact" requirement allows for complete deletion of contacts, but this could contradict the need to maintain data integrity, especially if there are related records (e.g., communication logs, linked notes).

Resolution:

Implement a soft delete mechanism where contacts are marked as deleted but not physically removed from the database. This maintains data integrity and allows recovery if needed.

Clarify the requirement: "When deleting a contact, implement a soft delete where the contact is flagged as deleted but remains in the database to maintain data integrity."

Contradiction in "Import Contacts" vs. Duplicate Handling

Contradiction: The "Import Contacts" requirement mentions options to merge, overwrite, or skip duplicates, but this could conflict with performance requirements, especially under high loads.

Resolution:

Specify a clear strategy for handling duplicates that balances performance and usability. For example, perform duplicate checks in batches and allow users to configure their preferred method of handling duplicates.

Update the requirement: "During import, handle duplicates by allowing users to configure their preferred method (merge, overwrite, skip). Perform duplicate checks in batches to maintain performance."

Contradiction in "Search Contact" vs. Privacy Requirements

Contradiction: The "Search Contact" requirement allows searching by various criteria, but this could conflict with privacy requirements if sensitive information is exposed unintentionally.

Resolution:

Limit search results to non-sensitive fields by default and provide options to search sensitive fields only with appropriate permissions.

Clarify the requirement: "Search functionality should prioritize non-sensitive fields. Searching sensitive fields (e.g., email, phone number) should be restricted and require appropriate user permissions."

Contradiction in "Categorize Contact" vs. Usability

Contradiction: Allowing multiple categories per contact could lead to complexity in the user interface, conflicting with usability requirements.

Resolution:

Implement an intuitive categorization system with clear instructions and easy-to-use controls for managing multiple categories.

Update the requirement: "Allow contacts to belong to multiple categories while ensuring the interface remains user-friendly and intuitive, with clear instructions for managing categories."

Contradiction in Performance Requirements vs. Compatibility

Contradiction: Ensuring a 2-second response time for all actions might conflict with compatibility requirements, especially on older devices or slower networks.

Resolution:

Define performance benchmarks separately for different device classes and network conditions, and provide optimized experiences accordingly.

Clarify the requirement: "Ensure a 2-second response time for core actions on modern devices and typical network conditions. For older devices or slower networks, optimize the experience to maintain reasonable performance."

Contradiction in Security Requirements vs. Usability

Contradiction: Strong authentication mechanisms (e.g., two-factor authentication) may conflict with usability requirements by adding complexity for the user.

Resolution:

Provide a balance by offering strong security options as configurable settings, allowing users to choose their preferred level of security.

Update the requirement: "Implement strong authentication mechanisms with options for users to configure their preferred security level, balancing security and usability."

8.Reference

IEEE Standard for Software Requirements Specifications (IEEE Std 830-1998)

Relevant user interface design guidelines

Data privacy regulations and guidelines