Software Requirements Specification

Document(SRS)

1.Introduction:

The Software Requirements Specifications (SRS) document serves as the foundational guide for the developing stage of the "Rakatdaan" Android app project, adhering to the outlined reference principles. In accordance with the SRS directive, this introduction offers a concise overview of the entire document, delineating the project's scope and objectives without delving into design intricacies. The primary purpose is to articulate what the system, in this case, the Rakatdaan Android app, must achieve, providing a comprehensive understanding for the development team. As per the reference guidelines, the focus remains on elucidating functionalities rather than design specifics. The Rakatdaan project, at its core, aims to facilitate efficient blood management by creating an Android app that connects donors, recipients, and healthcare professionals seamlessly. By emphasizing clarity on what the system must do, this introduction lays the groundwork for subsequent sections of the SRS to delve into specific functionalities, ensuring a coherent and purposeful development process for the Rakatdaan Android app.

1.1 Purpose

The purpose of the Software Requirements Specifications (SRS) for the "Rakatdaan" Android app project is to provide a comprehensive blueprint for the development team, elucidating the essential functionalities and requirements that the application must fulfill. This SRS serves as a guiding document, outlining the project's scope, objectives, and constraints related to blood management within the Android app. The intended audience comprises developers, designers, and stakeholders involved in the Rakatdaan project. By clearly defining the purpose of the SRS, this document aims to establish a shared understanding among team members, ensuring that the resulting Android app aligns seamlessly with the specific goals of facilitating blood donation, enhancing user experience, and contributing to the overall efficiency of blood management processes.

1.2 Scope

1. Software Products:

The Rakatdaan Android app is the primary software product to be developed. It is aimed at providing a comprehensive platform for blood management, enabling efficient communication between blood donors, recipients, and healthcare professionals.

2. Functionality:

The Rakatdaan app will facilitate the following key functionalities:

- User Registration and Profiles: Allow users to create profiles with relevant information.

- Blood Donation Coordination: Enable donors to find nearby blood donation events and coordinate donations.

- Blood Request System: Allow healthcare providers to request and manage blood donations.

- Real-time Communication: Provide a messaging system for users to communicate and coordinate blood-related activities.

- Donor History Record: Implement a system for that keeps donors blood donation records based on their contributions.

3. Application:

The application of the Rakatdaan app extends to:

- Bridging the Gap: Facilitate seamless interaction between blood donors, recipients, and healthcare professionals.

- Emergency Response: Enhance the speed and efficiency of blood donation responses during emergencies.

- Community Engagement: Foster a sense of community and altruism among users by encouraging regular blood donations.

4. Benefits, Objectives, and Goals:

Benefits: The Rakatdaan app aims to streamline blood management processes, ensuring a stable and accessible blood supply network.

Objectives:

1. Improve the availability of blood for medical emergencies, strengthen community engagement in blood donation, and enhance the overall efficiency of blood management.

2. To let donors to schedule appointments for blood donations, reducing wait times and streamlining the donation process.

3. To provide a feature for hospitals and individuals to send out emergency blood requests, allowing users to respond quickly.

4. Ensuring screening of user's eligibility to donate blood based on factors like recent donations, medical history, and travel & recent activities.

5. To enable donors to track their donation history, including the date, location, and quantity donated

Goals: Achieve a user-friendly interface, increase the number of registered donors, and establish a robust communication network for prompt blood-related coordination.

5. Consistency:

This scope aligns with higher-level specifications by emphasizing the app's role in blood management, maintaining consistency with the overarching project objectives.

1.3 Definitions, Acronyms, and Abbreviations

1. Rakatdaan Android App : Refers to the mobile application in development for blood management, facilitating blood donation, and enhancing communication between donors, recipients, and healthcare professionals.

2. SRS: Software Requirements Specifications, the comprehensive document outlining the functional and non-functional requirements of the Rakatdaan Android app.

3. API: Application Programming Interface, a set of rules that allows different software applications to communicate with each other, facilitating data exchange for the Rakatdaan app.

4. UI: User Interface, the visual elements and design components of the Rakatdaan app that users interact with.

5. UX: User Experience, encompassing the overall experience and satisfaction of users interacting with the Rakatdaan app.

6. DBMS: Database Management System, the software used to manage and organize the database that stores information for the Rakatdaan app.

7. CMS: Content Management System, if applicable, for handling and updating content within the Rakatdaan app.

8. OCR: Optical Character Recognition, if relevant, for extracting text information from images or documents within the Rakatdaan app.

9. SSL: Secure Sockets Layer, a security protocol ensuring encrypted communication between users and the Rakatdaan app.

1.4 References

[1] S P Kurlekar, K Pranali, R Komal M, Shrutika M, Aishwarya Kurlekar, S P, Pranali K, Shrutika M., & Aishwarya M. “Android app for Quick Access of Blood Bank” (2017) Journal of Embedded Systems and Processing, 2(1).

[2] S Periyanayagi, A Manikandan, M Muthukrishnan and M Ramakrishnan, Professor and students of Ramco Institute of Technology, Rajapalayam, India. “BDoor App-Blood Donation Application using Android Studio “(2021) Journal of Physics Conference Series.

[3] M. R. A. Hamlin, J. A. Mayan “Blood donation and life saver-blood donation app” (2016) Published in International Conference on Control, Instrumentation, Communication and Computational Technologies.

[4] Brislin M.R.A et al, ‘Blood Donation and Life Saver App’, 2nd Int. (2017) Conf. on Communication and Electronics Systems (ICCES), DOI:10.1109/CESYS.2017.8321318, pp 446-451.

[5] Clementeena A, Sankar K and Kannan S, ‘A Study on Blood Bank Management System’,(2014),Middle East Journal of Scientific Research, Vol. 19, No. 8, pp 1123-1126.

[6] Fahim M, Cebe H.I, Rasheed J and Kiani F, ‘Mhealth: Blood Donation Application Using Android Smartphone’, 6st Int. (2016) Conf. on Digital Information and Communication Technology and its Applications (DICTAP), Konya, Turkey, 2016.

[7] Meiappane A, et al. ‘DWORLD: Blood Donation App Using Android’, (2019), IEEE Int. Conf. on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019,

[8] Priya P, et al. ‘The Optimization of Blood Donor Information and Management System by

Technopedia’, Int. (2014) Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, pp 1-5.

[9] Vikas Kulshreshtha, and Sharad Maheshwari, ‘Blood Bank Management Information System in India’, Int. (2012) Journal of Engineering Research and Applications (IJERA), Vol .1, No. 2, pp 260-263

[10] Vikas Kulshreshtha, and Sharad Maheshwari, ‘Benefits of Management Information System in Blood Bank’, Int. (2013) Journal of Engineering and Science, Vol. 1, No. 12, pp

5 -7.

Web Reference

- https://nevonprojects.com/online-blood-bank-project
- https://www.phptpoint.com/projects/blood-bank-management-system
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5682362
- https://dribbble.com/tags/blood_donation_app

1.5 Overview

Overview of the Rakatdaan Android App Project:

1. Contents of the SRS:

The Software Requirements Specifications (SRS) for the Rakatdaan Android app project comprehensively outlines the functional and non-functional requirements essential for the app's development. It encapsulates detailed information about user interactions, system functionalities, and the overall architecture of the application.

2. Organization of the SRS:

The SRS is structured to cater to the distinct needs and concerns of various stakeholders:

- Introduction: Provides an executive-level overview, highlighting the purpose, definitions, acronyms, and the scope of the Rakatdaan project.

- Overall Description: Offers a high-level description of the Rakatdaan app, outlining the product perspective, functionality, user characteristics, and constraints. This section is particularly relevant to customers and potential users seeking an understanding of the app's purpose and capabilities.

- Specific Requirements: Forms the core of the document and is primarily for developers. It delves into detailed technical specifications, including functional requirements, performance requirements, and design constraints. Developers and technical teams will find crucial information here for the actual implementation of the Rakatdaan Android app.

- Other Sections: Appendices and additional sections contain supplementary information such as prototypes, data models, and any other details necessary for a comprehensive understanding of the Rakatdaan project.

2. The Overall Description

Overall Description of the Rakatdaan Android App:

The Rakatdaan Android app represents a cutting-edge solution at the intersection of healthcare and technology, aiming to streamline and optimize blood management processes. This section provides a comprehensive overview of the factors influencing the development of the Rakatdaan app, offering insights into its purpose, functionality, and potential impact.

1. Product Perspective: The Rakatdaan app exists as a standalone product designed to operate on Android devices. It interfaces with external systems for location services, messaging, and potentially, a central blood bank database. As a user-centric application, it serves as a bridge between blood donors, recipients, and healthcare professionals, enhancing communication and coordination.

2. Functionality: The primary functionalities of the Rakatdaan app include user registration, blood donation event coordination, blood request management, real-time communication, and donor recognition. The app prioritizes user experience, ensuring a seamless interface for intuitive interactions.

3. User Characteristics: The target users encompass a diverse range, including potential blood donors, recipients in need of blood, and healthcare providers managing blood-related activities. The app accommodates users with varying levels of technological proficiency, promoting inclusivity and accessibility.

4. Constraints: The development of the Rakatdaan app is subject to constraints such as technological limitations, compliance with data protection regulations, and potential dependencies on external services. These constraints are carefully considered to ensure the app's functionality aligns with ethical and legal standards.

5. Assumptions and Dependencies: Certain assumptions, such as the availability of reliable internet connectivity and GPS services, are integral to the Rakatdaan app's functionality. Dependencies on external services, like messaging platforms, are also acknowledged and managed to ensure seamless integration.

6. Goals and Objectives: The overarching goal of the Rakatdaan app is to contribute to efficient blood management by fostering a community of engaged donors, facilitating timely donations, and enhancing communication within the blood donation ecosystem. The objectives include improving the availability of blood during emergencies, increasing donor participation, and creating a user-friendly and impactful platform.

By providing this overall description, the Rakatdaan Android app project sets the stage for a more detailed and technical exploration of requirements in Section 3, facilitating a comprehensive understanding for both customers and developers.

2.1 Product Perspective

The Rakatdaan Android app is designed as an independent and self-contained product focused on revolutionizing blood management processes. It operates as a standalone system with no direct dependencies on external software, ensuring its autonomy and ease of integration into diverse environments. As a user-centric application, Rakatdaan serves as a comprehensive solution, connecting blood donors, recipients, and healthcare professionals within a singular platform.

Interaction with External Systems:

- Location Services: The Rakatdaan app interacts with the device's location services to provide real-time information about nearby blood donation events and facilitate efficient coordination.

- Messaging Platforms: For seamless communication, the app may integrate with external messaging services to enable real-time interaction between users.

Comparisons to Other Systems:

- Marketplace Position: Rakatdaan stands out in the marketplace as a purpose-built Android app solely dedicated to blood management. While other health-related applications may exist, Rakatdaan uniquely focuses on fostering a community of blood donors and recipients.

- Differentiators: Unlike generic health apps, Rakatdaan's primary differentiators lie in its specialization in blood donation coordination, real-time communication, and donor recognition.

Block Diagram:

- User Interface: The app presents a user-friendly interface allowing donors to find events, healthcare providers to manage blood requests, and recipients to communicate with potential donors.

- Database Management: A secure database system manages user profiles, events, and communication logs, ensuring data integrity and privacy.

- External Interfaces: Interfaces with the device's GPS for location-based services and potentially integrates with external messaging platforms for efficient communication.

This block diagram depicts the major components of Rakatdaan's larger system, emphasizing its independence while showcasing essential interactions. The app is presented as a black box, highlighting its role in facilitating blood management without delving into internal architecture details. The design document will further elaborate on the app's internals and technical specifications.

2.1.1 System Interfaces

System Interfaces for Rakatdaan Android App:

1. Location Services:

- Functionality: The Rakatdaan app interacts with the device's location services to provide users with information about nearby blood donation events and relevant healthcare facilities.

- Interface Description: Utilizing Android Location APIs, the app accesses the device's GPS functionality to retrieve real-time location data. This interface ensures accurate event recommendations based on the user's geographical position.

2. Messaging Platforms:

- Functionality: To facilitate real-time communication between blood donors, recipients, and healthcare professionals, the Rakatdaan app may integrate with external messaging platforms.

- Interface Description: The app would utilize APIs or SDKs from messaging services (e.g., Firebase Cloud Messaging) to send and receive messages. This integration ensures efficient and immediate communication within the Rakatdaan community.

3. External Database Management System:

- Functionality: Rakatdaan manages user profiles, blood donation events, and communication logs through a secure database system.

- Interface Description: The app interfaces with an external Database Management System (DBMS), possibly utilizing APIs like SQLite for local storage on the device or a cloud-based solution like Firebase Realtime Database for synchronized data across users.

4. Event Management System:

- Functionality: To coordinate blood donation events, Rakatdaan may interact with external systems managing event logistics and schedules.

- Interface Description: The app could use APIs or web services to connect with an Event Management System, allowing for the retrieval and updating of event details, ensuring accurate and up-to-date information for users.

These system interfaces are crucial for Rakatdaan's seamless operation, ensuring accurate location-based services, real-time communication, secure data management, and synchronized event coordination. The specifications of these interfaces guide the development team in integrating external systems with the Rakatdaan Android app effectively.

2.1.2 Interfaces

Interfaces for Rakatdaan Android App Development:

1. Graphical User Interface (GUI):

- Logical Characteristics: Rakatdaan employs an intuitive and user-friendly GUI, featuring a clean design with easily navigable menus. It incorporates visual elements such as buttons, icons, and interactive screens to facilitate seamless user interactions.

- Optimization Aspects: The GUI is optimized for user accessibility and engagement, ensuring a straightforward and efficient experience. Consideration is given to font size, color contrast, and interactive elements to enhance usability. ADA compliance is a priority, ensuring inclusivity for users with disabilities through features such as screen reader compatibility and voice command options.

2. Event Coordination Interface:

- Logical Characteristics: Users can coordinate and participate in blood donation events through a dedicated interface. This includes features for event discovery, registration, and real-time updates.

- Optimization Aspects: The event coordination interface is optimized for quick event discovery, with features like filters based on location and event type. ADA compliance ensures that users with varying abilities can seamlessly navigate and participate in events.

3. Communication Interface:

- Logical Characteristics: Rakatdaan incorporates a messaging interface for real-time communication between donors, recipients, and healthcare professionals. This includes individual and group messaging features.

- Optimization Aspects: The communication interface is optimized for simplicity and responsiveness, with features like push notifications for immediate updates. ADA compliance is considered, allowing users with disabilities to effectively use and engage with the messaging system.

4. User Registration and Profile Interface:

- Logical Characteristics: Users interact with an interface for creating profiles, managing personal information, and tracking their blood donation history.

- Optimization Aspects: The registration and profile interface is optimized for ease of use during onboarding, featuring clear instructions and accessible form fields. ADA compliance ensures that users with disabilities can complete the registration process without barriers.

5. Accessibility Features:

- Logical Characteristics: Rakatdaan incorporates accessibility features to ensure an inclusive experience for all users.

- Optimization Aspects: This includes adjustable font sizes, high color contrast, and voice command options. The app is designed to comply with ADA standards, catering to users with different abilities and providing a universally accessible interface.

2.1.3 Hardware Interfaces

Hardware Interfaces for Rakatdaan Android App Development:

The Rakatdaan Android app primarily relies on the hardware components inherent to Android devices, and it does not have direct control or interaction with external hardware devices. As such, there are no specific hardware interfaces required for the functioning of the Rakatdaan app. The app operates within the standard configurations and capabilities of Android smartphones and tablets.

Device Compatibility:

- Logical Characteristics: The app is designed to be compatible with a range of Android devices, ensuring flexibility and accessibility for a broad user base.

- Configuration Characteristics: Rakatdaan is optimized to function seamlessly across various screen sizes, resolutions, and hardware specifications commonly found in Android devices.

Connectivity Protocols:

- Logical Characteristics: The app relies on standard connectivity protocols inherent to Android devices, such as Wi-Fi, mobile data, and Bluetooth for specific functionalities.

- Configuration Characteristics: Rakatdaan is configured to adapt to different network environments, ensuring efficient communication and data exchange without imposing specific connectivity requirements.

Hardware Features:

- Logical Characteristics: Rakatdaan utilizes certain hardware features, such as GPS, for location-based services to enhance the user experience.

- Configuration Characteristics: The app is configured to access and utilize the device's GPS functionality to provide accurate location information for event recommendations.

Protocols for Data Transfer:

- Logical Characteristics: Data transfer within the app relies on standard Android protocols for secure communication.

- Configuration Characteristics: Rakatdaan ensures the use of secure protocols for data transmission, prioritizing user privacy and the integrity of information shared within the app.

2.1.4 Software Interfaces

Software Interfaces for Rakatdaan Android App Development:

The Rakatdaan Android app interfaces with several software components to enhance its functionality and provide a seamless user experience.

1. **MongoDB**

    (1) **Name:** MongoDB

    (2) **Mnemonic:** N/A

    (3) **Specification Number:** N/A

    (4) **Version Number:** Latest stable version (e.g., MongoDB 5.0)

    (5) **Source:** Official MongoDB website (*https://www.mongodb.com/*)

Interfaces

1. **Interface with MongoDB for Data Storage**

    (1) **Purpose:** MongoDB serves as the primary database system for storing and managing data related to donor registrations, donation history, and blood requests within the Blood Donation App.

    (2) **Message Content and Format:** Communication with MongoDB occurs using MongoDB's native query language and data format, primarily JSON (JavaScript Object Notation). The interface involves CRUD (Create, Read, Update, Delete) operations for managing donor records, donation information, and blood request details.

2. **Google Maps API Integration**

    (1) **Purpose:** Integration with Google Maps API enables location-based services within the Blood Donation App, allowing users to search for nearby blood donation centers, hospitals, or donors based on geographical information.

    (2) **Message Content and Format:** The interface with Google Maps API involves sending requests for geolocation data and receiving responses containing map data, coordinates, and relevant location information. The format may include HTTP requests and JSON/XML responses to display maps and location-based data to app users.

2.1.5 Communications Interfaces

Communications Interfaces for Rakatdaan Android App Development:

In the development of the Rakatdaan Android app, various communication interfaces are integral for enabling smooth interactions and data exchange. These interfaces encompass both local network protocols and standard communication methods.

1. HTTP/HTTPS Protocol:

- Description: Rakatdaan utilizes the HTTP/HTTPS protocols for communication with external servers and services. HTTP is employed for general data exchange, while HTTPS ensures secure and encrypted communication, safeguarding user data and information during transactions.

2. Firebase Cloud Messaging (FCM):

- Description: FCM serves as a crucial communication interface for real-time messaging between Rakatdaan users. It enables the delivery of push notifications, ensuring prompt communication regarding events, messages, and updates.

3. Location-Based Services (GPS):

- Description: The app leverages the device's GPS capabilities for location-based services. This communication interface allows Rakatdaan to access accurate geographic coordinates, facilitating event recommendations and user positioning.

4. Firebase Realtime Database Communication:

- Description: Rakatdaan communicates with Firebase Realtime Database for cloud-based storage and synchronized data access. This interface involves sending and receiving JSON-formatted data to maintain consistency across users and devices.

5. Local Database Interaction (SQLite):

- Description: The app communicates with the local SQLite database for efficient storage and retrieval of user profiles, event data, and communication logs. SQL queries are used for interacting with the database, supporting offline functionality.

6. Device-to-Device Communication (Bluetooth - Optional):

- Description: As an optional feature, Rakatdaan may incorporate Bluetooth for direct device-to-device communication. This interface allows users in close proximity to exchange information without relying on network connectivity.

These communication interfaces play a vital role in the functionality of Rakatdaan, facilitating data exchange, real-time messaging, and location-based services. By leveraging these protocols, the app ensures a responsive and interactive user experience while maintaining the security and integrity of transmitted information.

2.1.6 Memory Constraints

Memory Constraints for Rakatdaan Android App Development:

In the development of the Rakatdaan Android app, specific memory constraints are crucial for optimizing performance and ensuring compatibility across a diverse range of Android devices.

1. Primary Memory (RAM):

- Constraint: The Rakatdaan app is designed to operate efficiently within the memory constraints typical of Android devices. The target design footprint for RAM utilization should not exceed 128MB.

- Rationale: This constraint is based on market research indicating that a significant portion of the target user base has devices with RAM capacities ranging between 128MB to 512MB. By setting a maximum limit of 128MB, the app ensures smooth operation even on devices with more modest hardware configurations.

2. Secondary Memory (Storage):

- Constraint: Rakatdaan aims to maintain a lightweight storage footprint, considering the limited storage capacities of certain Android devices. The target maximum size for the app installation should not exceed 50MB.

- Rationale: Many Android devices, especially entry-level and mid-range models, come with limited internal storage. Setting a constraint of 50MB ensures that the app remains accessible to a broader audience without imposing significant storage burdens.

3. Optimization for Low-RAM Devices:

- Constraint: Rakatdaan is optimized to run smoothly on devices with as low as 1GB of RAM. While the app can operate on devices with lower RAM capacities, the user experience may be affected, and the app will prioritize essential functionalities to accommodate such constraints.

- Rationale: Recognizing the diverse range of Android devices in the market, including those with lower RAM specifications, the app is developed with optimization strategies to ensure usability on a broad spectrum of hardware.

2.1.7 Operations

Operations for Rakatdaan Android App Development:

In the development of the Rakatdaan Android app, various operations are essential to ensure smooth user interactions, data processing, and system maintenance. These operations encompass both normal user activities and special functions vital for the application's functionality.

1. User Operations:

- Modes of Operations:

  - Donor Mode: Allows users to register as blood donors, update their profiles, and view upcoming blood donation events.

  - Recipient Mode: Enables users to create blood requests, search for donors, and communicate with potential donors.

  - Healthcare Professional Mode: Supports healthcare professionals in managing blood donation events, tracking donor participation, and coordinating with recipients.

- Interactive and Unattended Operations: Users can perform interactive operations during real-time activities such as event registration, messaging, and profile updates. Unattended operations include background processes like event notifications and data synchronization.

2. Data Processing Support Functions:

- Profile Management: Users can update personal information, including contact details and blood type, ensuring accurate donor and recipient profiles.

- Event Coordination: The app processes event-related data, including event creation, participant registration, and real-time updates.

- Communication Handling: Rakatdaan facilitates real-time messaging, supporting functions such as sending and receiving messages, notifications, and alerts.

3. Backup and Recovery Operations:

- Scheduled Backups: The app may implement scheduled backups of user data and system configurations to prevent data loss.

- Recovery Mechanisms: In the event of a system failure or data corruption, Rakatdaan includes recovery mechanisms to restore the app's state to a previously stable version.

4. System Maintenance Operations:

- Software Updates: Rakatdaan allows users to receive and install software updates, ensuring access to the latest features, security patches, and improvements.

- Server Communication: The app communicates with external servers for data synchronization, ensuring consistency across users and devices.

These operations not only define the user's engagement with the app but also encompass critical data processing and system maintenance functions. By addressing these aspects, Rakatdaan is designed to offer a comprehensive and user-friendly experience while incorporating robust mechanisms for data integrity, backup, and recovery.

2.1.8 Site Adaptation Requirements

Site Adaptation Requirements for Rakatdaan Android App Development:

In the development of the Rakatdaan Android app, certain site adaptation requirements may be necessary to ensure optimal functionality and compatibility with specific installations. These requirements encompass both data and initialization sequences as well as features that may need modification based on the site or mission.

1. Initialization Sequences:

- Requirement: Prior to system activation, a one-time initialization sequence is required to set up essential configurations. This includes the initial setup of the Firebase Realtime Database, creation of necessary tables, and configuration of messaging services.

- Rationale: The initialization process ensures that the app is correctly configured to interact with external services, facilitating seamless data synchronization and messaging functionality.

2. Data Table Configuration:

- Requirement: New data tables specific to the Rakatdaan system must be created on the organization's existing Database Management System (DBMS). These tables should be populated with initial data to support app functionalities.

- Rationale: This requirement ensures that the app has access to the necessary data structures, allowing for efficient storage and retrieval of user profiles, event information, and communication logs.

3. Server Communication Setup:

- Requirement: The organization's servers should be configured to allow communication with the Rakatdaan app. This involves setting up secure communication protocols, firewall configurations, and ensuring compatibility with the app's data exchange format.

- Rationale: Proper server communication setup is crucial for real-time data synchronization, event coordination, and messaging functionalities.

4. User Site Equipment:

- Requirement: Users should have access to Android devices with a minimum recommended RAM capacity of 1GB for an optimal experience. While the app is designed to be lightweight, ensuring sufficient device resources enhances overall performance.

- Rationale: By specifying minimum hardware requirements, Rakatdaan aims to provide users with a consistent and reliable experience, minimizing potential issues related to device performance.

These site adaptation requirements are essential for ensuring a successful deployment of the Rakatdaan Android app. They address both software-specific configurations and hardware considerations, providing guidelines for system administrators and users to prepare the environment for optimal app performance.

2.2 Product Functions

Product Functions for Rakatdaan Android App Development:

The Rakatdaan Android app is designed to provide a comprehensive set of functions to fulfill the needs of blood donors, recipients, and healthcare professionals. These functions are organized for clarity and presented in a way that aligns with the logical flow of user interactions.

1. User Registration and Profiles:

- Function: Users can register on the Rakatdaan platform, creating profiles with essential information, including name, contact details, blood type, and location.

- Purpose: Facilitates the creation of a user database, allowing for personalized experiences, targeted event recommendations, and effective donor-recipient matching.

2. Event Creation and Coordination:

- Function: Donors and healthcare professionals can create and coordinate blood donation events, specifying details such as location, date, and required blood types.

- Purpose: Enables the organization and management of blood donation events, ensuring efficient coordination and participation.

3. Blood Requests and Matching:

- Function: Recipients can create blood requests, specifying their blood type and urgency. The app matches recipients with potential donors based on compatibility.

- Purpose: Facilitates timely and accurate blood matching, connecting recipients with donors who meet their specific requirements.

4. Real-Time Messaging:

- Function: Users can communicate in real-time through messaging features, allowing donors, recipients, and healthcare professionals to coordinate and exchange information.

- Purpose: Supports seamless communication for event coordination, blood requests, and general inquiries.

5. Location-Based Services:

- Function: Utilizes GPS functionality to provide location-based services, including event recommendations, donor-recipient proximity matching, and navigation to event locations.

- Purpose: Enhances user experience by offering relevant information and services based on geographical location.

6. User Notifications and Alerts:

- Function: Sends push notifications and alerts to users for important events, messages, and updates, ensuring timely communication and participation.

- Purpose: Keeps users informed and engaged, enhancing their overall experience and encouraging active participation.

7. Profile Updates and Privacy Settings:

- Function: Allows users to update their profiles, manage privacy settings, and control the visibility of their information on the platform.

- Purpose: Empowers users to maintain accurate and up-to-date profiles while ensuring control over the visibility of personal details.

These product functions form the core capabilities of the Rakatdaan Android app, addressing the specific needs of blood donors, recipients, and healthcare professionals. The logical organization of functions ensures a user-friendly and intuitive experience, fostering effective communication and coordination within the blood donation community.

2.3 User Characteristics

User Characteristics for Rakatdaan Android App Development:

Understanding the characteristics of the intended users is crucial for tailoring the design of the Rakatdaan Android app to meet their needs effectively. The following general characteristics provide insights into the user base, influencing both user interface (UI) design and internal system design considerations.

1. Diverse Educational Backgrounds:

- Characteristics: The user base is expected to have a diverse educational background, ranging from individuals with basic education to those with advanced degrees.

- Impact on Design: The app's UI should be intuitive and user-friendly, accommodating users with varying levels of education. Clear and straightforward navigation will be essential to ensure accessibility for all.

2. Varied Technological Experience:

- Characteristics: Users may have diverse levels of experience with technology, from tech-savvy individuals to those with limited exposure to mobile applications.

- Impact on Design: The UI should strike a balance between simplicity for users with limited technological experience and advanced features for tech-savvy users. Onboarding processes should be designed to cater to users with varying levels of technological proficiency.

3. Multilingual User Base:

- Characteristics: The potential user base may encompass individuals with different primary languages and language proficiencies.

- Impact on Design: The app should support multiple languages, and UI elements should be designed to accommodate users with varying language preferences. Language selection and localization features will be essential for a global user base.

4. Age Diversity:

- Characteristics: Users may span various age groups, including young donors, middle-aged recipients, and healthcare professionals of different ages.

- Impact on Design: The UI should be designed with considerations for age-related usability. Font sizes, color contrasts, and interactive elements should be optimized for readability and usability across different age demographics.

5. Accessibility Considerations:

- Characteristics: Users with diverse abilities and disabilities may access the app, requiring considerations for accessibility.

- Impact on Design: The UI and interactions should adhere to accessibility standards, including features such as voice-over support, adjustable font sizes, and color contrasts for users with visual impairments.

6. Geographical and Cultural Diversity:

- Characteristics: Users may belong to different geographical regions and cultural backgrounds.

- Impact on Design: The app's UI should be culturally sensitive and inclusive. Visual elements, symbols, and terminology should be chosen with consideration for diverse cultural interpretations.

Understanding these user characteristics is fundamental to creating an inclusive and user-centric design for the Rakatdaan Android app. The design considerations aim to ensure that the app is accessible, intuitive, and accommodating to a diverse user base with varying educational backgrounds, technological experiences, and cultural contexts.

2.4 Constraints

Constraints for Rakatdaan Android App Development:

In the development of the Rakatdaan Android app, various constraints exist that will shape the design and functionality of the application. These constraints encompass regulatory, hardware, interface, operational, and security considerations, influencing the development process.

1. Regulatory Policies:

- Constraint: The app must adhere to regional and international regulations governing health data privacy, user consent, and blood donation protocols.

- Impact: Regulatory compliance will guide the design of data security features, consent mechanisms, and user information handling to ensure alignment with legal requirements.

2. Hardware Limitations:

- Constraint: The app should operate efficiently on Android devices with a minimum recommended RAM capacity of 1GB.

- Impact: Hardware limitations will influence the optimization of resource usage, ensuring the app's compatibility with a wide range of Android devices.

3. Interface to Other Applications:

- Constraint: Rakatdaan must seamlessly interface with external applications and services for functionalities such as messaging and real-time data synchronization.

- Impact: The app's design will need to include standardized interfaces and protocols to ensure smooth communication and integration with external systems.

4. Reliability Requirements:

- Constraint: The app must maintain high reliability, with minimal downtime and a low error rate, to ensure a positive user experience.

- Impact: Development efforts will focus on robust error handling, data consistency, and failover mechanisms to meet reliability requirements.

5. Safety and Security Considerations:

- Constraint: Security measures must be implemented to protect user data, ensuring confidentiality, integrity, and availability.

- Impact: The design will incorporate encryption, secure communication protocols, and access control mechanisms to address safety and security concerns.

These constraints guide the development process, ensuring that Rakatdaan aligns with legal, operational, and security considerations. By addressing these constraints, the app aims to provide a reliable, secure, and user-friendly platform for blood donation coordination.

2.5 Assumptions and Dependencies

Assumptions and Dependencies for Rakatdaan Android App Development:

In the development of the Rakatdaan Android app, several assumptions and dependencies shape the requirements and design considerations. These factors, while not design constraints, are critical to the successful implementation of the project.

1. Operating System Availability:

- Assumption: The assumption is made that the designated Android devices will have the required operating system versions compatible with the Rakatdaan app.

- Dependency: If the operating system availability changes, the app's requirements and compatibility may need to be adjusted accordingly.

2. Network Connectivity:

- Assumption: It is assumed that users will have access to a stable internet connection for real-time data synchronization and messaging features.

- Dependency: Changes in network connectivity may impact the effectiveness of real-time functionalities, necessitating adjustments in design and data handling.

3. External API Stability:

- Assumption: The stability and availability of external APIs, used for messaging and coordination, are assumed to remain consistent.

- Dependency: Any changes or unavailability in external APIs may require modifications in the app's communication mechanisms and integration points.

4. User Device Compatibility:

- Assumption: Users are expected to have Android devices that meet the minimum hardware requirements for optimal app performance.

- Dependency: Changes in the user device landscape may necessitate adjustments to resource optimization and hardware-specific functionalities.

5. Database Management System Stability:

- Assumption: The stability and availability of the Database Management System (DBMS) hosting Rakatdaan's data are assumed to be maintained.

- Dependency: Changes in the DBMS or its availability may require adaptations in data handling, storage, and retrieval processes.

6. Legal and Regulatory Environment:

- Assumption: The legal and regulatory environment governing health data, privacy, and blood donation practices will remain consistent.

- Dependency: Changes in regulations may impact data handling, consent mechanisms, and overall compliance, necessitating adjustments in the app's features.

7. User Adoption and Participation:

- Assumption: The assumption is made that users will adopt and actively participate in the Rakatdaan community.

- Dependency: Changes in user adoption rates or participation levels may require adjustments in user engagement features and incentive mechanisms.

8. External Collaboration:

- Assumption: Collaborations with external organizations, such as blood banks and healthcare institutions, are assumed to continue as planned.

- Dependency: Changes in external collaborations may impact event coordination, data sharing, and integration points, requiring adjustments in the app's functionalities.

9. Language and Cultural Considerations:

- Assumption: The assumption is made that language and cultural considerations identified in the user characteristics section will remain consistent.

- Dependency: Changes in user demographics or cultural preferences may necessitate adjustments in language localization and cultural sensitivity features.

These assumptions and dependencies serve as considerations for the Rakatdaan Android app development, providing a framework for anticipating potential changes and ensuring adaptability to evolving circumstances. Regular assessments of these factors will guide adjustments to the app's design and functionalities as needed.

2.6 Apportioning of Requirements.

Apportioning of Requirements for Rakatdaan Android App Development:

In the development of the Rakatdaan Android app, there may be requirements that can be apportioned based on priority, feasibility, and resource constraints. This section aims to identify requirements that may be delayed until future versions of the system, allowing for a phased and strategic approach to development and delivery.

1. Core Functionalities (Must-Have):

- Description: This category includes the essential features necessary for the initial release of the Rakatdaan app, such as user registration, blood donation event coordination, and real-time messaging.

- Rationale: Core functionalities are crucial for providing users with a functional and valuable experience from the initial release.

2. Enhanced User Engagement (Priority Level 1):

- Description: Features aimed at enhancing user engagement, such as personalized recommendations, gamification elements, and social sharing capabilities.

- Rationale: While not essential for the basic functionality, these features contribute to a more interactive and engaging user experience.

3. Localization and Multilingual Support (Priority Level 2):

- Description: Implementing language localization and multilingual support to cater to a diverse user base.

- Rationale: Important for expanding the app's reach to users from different linguistic backgrounds.

4. Advanced Security Features (Priority Level 3):

- Description: Advanced security measures, including biometric authentication and additional encryption layers.

- Rationale: While basic security is a priority, more advanced features can be considered in subsequent releases to further enhance user data protection.

5. Integration with Wearable Devices (Priority Level 4):

- Description: Integration with wearable devices to track health metrics and provide personalized health insights.

- Rationale: This feature, while innovative, can be apportioned for future releases to focus on core functionalities first.

6. AI-Powered Blood Matching Algorithms (Priority Level 5):

- Description: Implementation of advanced AI algorithms for more precise blood matching and recommendation.

- Rationale: While this feature can significantly enhance blood matching accuracy, it can be apportioned to future releases to prioritize the core functionality.

7. Community and Donor Recognition (Priority Level 6):

- Description: Features recognizing active donors and fostering a sense of community, such as donor leaderboards and achievements.

- Rationale: While valuable for community building, these features can be considered for future releases to streamline the initial development.

8. Integration with External Health Systems (Priority Level 7):

- Description: Integration with external health systems for seamless sharing of health records and information.

- Rationale: This complex integration can be apportioned for future releases, focusing on core functionalities initially.

9. Advanced Analytics and Reporting (Priority Level 8):

- Description: Advanced analytics features for tracking blood donation trends, user behaviors, and system performance.

- Rationale: Valuable for long-term insights, these features can be deferred to subsequent releases.

10. Augmented Reality (AR) Features (Priority Level 9):

- Description: Implementation of AR features for interactive blood donation events and campaigns.

- Rationale: While innovative, AR features can be apportioned to future versions, allowing the app to evolve gradually.

By apportioning requirements based on priority levels, Rakatdaan can deliver a robust initial release while strategically planning for future enhancements and innovations. Regular collaboration with the customer is essential to prioritize and adjust the apportioned requirements based on evolving needs and project constraints.

3.Specific Requirements

3.1 External Interfaces

→Registration Page

Name of Item: Registration Form

Description: Capturing user details for account creation

Source of Input: User

Valid Range: Text fields for name, address, blood type, contact details

Units of Measure: N/A

Timing: Upon user submission

Relationships: Links to user database

Screen Format: User-friendly form layout

→Request Page

Name of Item: Donation Request Form

Description: Form for blood donation requests

Source of Input: Recipient

Valid Range: Type of blood needed, urgency level, location

Units of Measure: N/A

Timing: Upon request submission

Relationships: Links to recipient database

Screen Format: Form-based layout for request submission

→Donate Page

Name of Item: Donor Registration Form

Description: Form for donor registration

Source of Input: Donor

Valid Range: Personal details, blood type, availability

Units of Measure: N/A

Timing: Upon registration submission

Relationships: Links to donor database

Screen Format: User-friendly layout for donor information

→Search Donor Page

Name of Item: Donor Search Interface

Description: Interface to search for available donors

Source of Input: Recipient

Valid Range: Blood type, location, availability

Units of Measure: N/A

Timing: Upon search query

Relationships: Queries donor database

Screen Format: Search-based layout for results presentation.

3.2 Functions

- Login:

The system provides security features through username-password matching where only authorized user can access the system with different authorization level. Admin Input :-Username, Password Output: - Invalid or Update Blood Details, logout

-User Registration:

Given that user has accessed application, then the user should be able to register through the application. The donor user must provide first name, gender, blood group, location, contact, email, username and password.

-Search result in a list view:

Search result can be viewed in a list. Each element in the list represents a specific donor. Each element should include first name, gender, blood group, location, contact according to the user position.

-Request Blood:

User should be able to request for blood at emergency situation, user need to define blood group, location, required date, units, contact. The order requested will be sent to blood bank and then to the Inventory to check the availability. If available, the requested blood will be sent to the requested donor.

-Donate history:

The records of all donors/recipient and their history are kept in one centralized database and thus reducing duplicate data in the database. User could able to track his donate history. The record of donation is maintained by the system.

3.3 Performance Requirements

Static Numerical Requirements

1. **Number of Terminals and Simultaneous Users**

    (1) The app should support a minimum of 500 terminals (mobile devices or browsers) concurrently.

    (2) Simultaneously, the app should accommodate at least 250 users interacting with the system.

2. **Amount and Type of Information Handling**

    (1) The app should handle user profiles with a minimum of 10 fields for registration, including name, address, contact information, and blood type.

Dynamic Numerical Requirements

1. **Transaction Processing Time**

    (1) 95% of user transactions, including registration, donation requests, donor registrations, and search operations, shall be processed in less than 2 seconds under normal workload conditions.

    (2) During peak workload conditions, 90% of transactions should still be processed within 3 seconds.

2. **Data Processing**

    (1) The system should handle a minimum of 500 donation requests per hour without performance degradation.

    (2) The search functionality should process and display results for a query involving 5000 donor profiles within 5 seconds under normal load.

3.4 Logical Database Requirements

Types of Information Used by Various Functions

1. **Login Page**

    (1) Username, Email/Phone, Password

    (2) Authentication Tokens

    (3) Last Login Timestamp

2. **Donor Registration Page**

    (1) Donor Information: Name, Address, Contact Details, Blood Type, Health Information

    (2) Donation History: Date, Blood Type Donated, Recipient Details

3. **Blood Request Page**

    (1) Recipient Information: Name, Hospital/Organization, Blood Type Needed, Urgency Level, Location

    (2) Request History: Date, Request Status, Donor Matched (if applicable)

4. **Donate History Page**

    (1) Donor Information: Name, Blood Type, Donation Date, Recipient Details

Frequency of Use

1. **Login Page**: Accessed each time a user logs in, frequency based on user activity.

2. **Donor Registration Page**: Accessed during initial registration, updated on donor profile changes.
3. **Blood Request Page**: Accessed when a recipient submits a request, updated upon request status changes.
4. **Donate History Page**: Accessed by donors to view their donation history, updated after each donation.

Accessing Capabilities

1. **User Authentication**: Read and write access for user login and profile data.
2. **Donor Registration**: Write access to donor details, read access for viewing own profile.
3. **Blood Request Submission**: Write access for recipients, read access to view submitted requests.
4. **Donate History**: Read access for donors to view their donation history.

Data Entities and Their Relationships

1. **Users Table**: Stores login credentials and user profiles.
2. **Donors Table**: Contains donor-specific information linked to the Users table.
3. **Blood Requests Table**: Stores information related to blood donation requests linked to user profiles.
4. **Donation History Table**: Records details of donations made by donors.

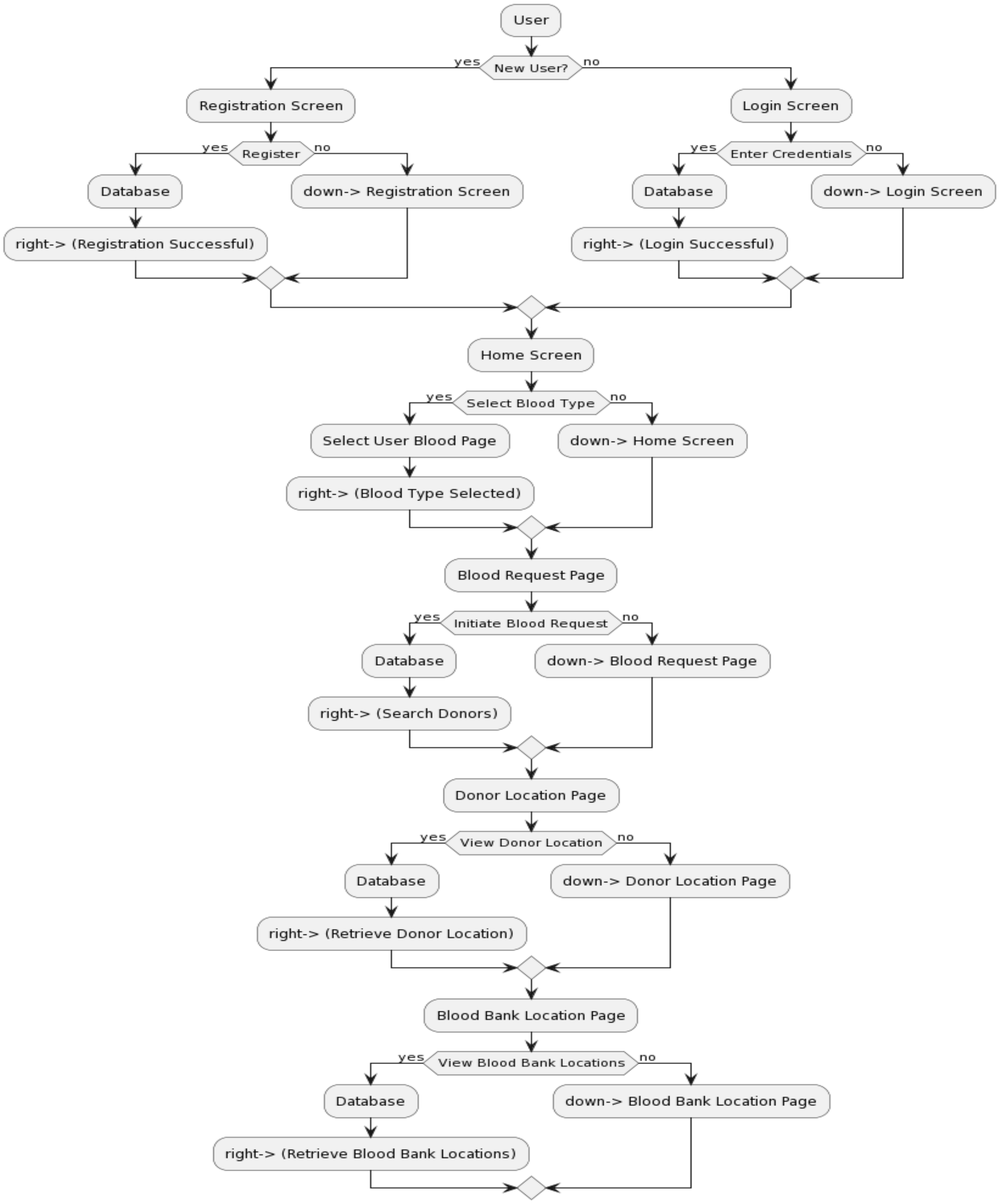
Integrity Constraints

1. **Unique User Identification**: Each user has a unique username/email for login.
2. **Referential Integrity**: Donor and Blood Request records linked to corresponding user profiles.

Data Retention Requirements

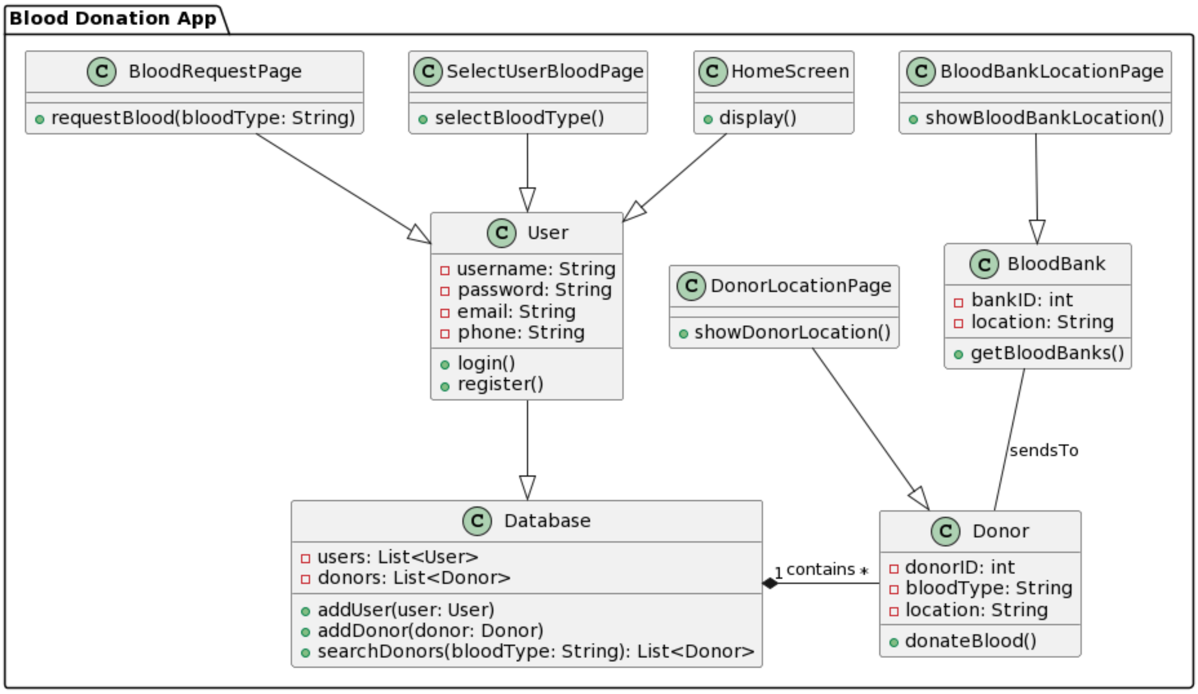
1. **Users and Donors Information**: Retained indefinitely unless account deletion requested.
2. **Blood Requests and Donation History**: Retained for audit and historical purposes, archived after a certain period.

Data Flow Diagram



This Data Flow Diagram demonstrates how data flows through different modules of application.

Class Diagram



3.5 Design Constraints

3.5.1 Standards Compliance

1. **Data Protection Regulations**

   The database design must comply with data protection laws such as GDPR, HIPAA, or any other relevant local regulations regarding the storage and handling of sensitive user health information.

2. **Database Security Standards**

   (1) Implementation of encryption methods for sensitive data at rest and during transmission.

   (2) Access controls and user authentication mechanisms to safeguard donor and recipient information.

3.**Storage Capacity**

   (1) The database design should be scalable to accommodate increasing user data without compromising performance.
   (2) Constraints on the maximum size of the database due to hardware limitations should be considered.

3.6 Software System Attributes

3.6.1 Reliability

Factors Required for Reliability:

- **Mean Time Between Failures (MTBF)**: The database system should demonstrate an MTBF of at least 10,000 hours.

- **Error Handling**: Robust error handling mechanisms to prevent data corruption and maintain data integrity.
- **Backup and Recovery**: Regular automated backups of the database to ensure quick recovery in case of failures.
- **Consistency**: Ensuring data consistency during simultaneous access or transactions.

3.6.2 Availability

Factors Required for Availability:

- **System Uptime**: The database should maintain an uptime of at least 99.9% over a defined period.
- **Fault Tolerance**: Implementing fault-tolerant mechanisms to ensure continuous operation despite hardware or software failures.
- **Checkpoint and Recovery**: Regular checkpoints to facilitate quick recovery in case of system failure.
- **Minimal Data Loss**: In the event of a failure, the system should recover with minimal data loss (e.g., no more than five seconds' worth of data loss).

3.6.3 Security

Factors Required for Security:

- **Encryption**: All sensitive data should be encrypted both at rest and during transmission.
- **Access Controls**: Implementing role-based access controls (RBAC) to restrict unauthorized access to sensitive data.
- **Audit Trails**: Maintaining logs of user activities and access for auditing purposes.
- **Data Integrity**: Implementing mechanisms to ensure the integrity of critical data variables and preventing unauthorized modifications.

3.6.4 Maintainability

Factors Required for Maintainability:

- **Modularity**: Designing the database in a modular fashion to ease future updates or modifications.
- **Documentation**: Comprehensive documentation of the database schema, relationships, and data flow to aid future maintenance.
- **Standardization**: Adherence to coding standards and database design best practices for easy comprehension by other maintainers.
- **Clear Interfaces**: Well-defined interfaces and separation of concerns to facilitate future modifications without affecting other parts of the system.

3.6.5 Portability

Factors Required for Portability:

- **Platform Independence**: Reducing host-dependent code to less than 20% of the entire database structure.
- **Use of Portable Languages**: Utilizing languages or frameworks known for their portability across different operating systems.
- **Database System Agnosticism**: Designing the database to be compatible with multiple database management systems for potential migration or cross-platform support.
- **Standard Compliance**: Ensuring compliance with industry standards to enhance the ease of porting the database to other systems or platforms.

3.7 Organizing the Specific Requirements

3.7.1 System Mode

Operational Modes:

1. **Normal Mode**:

    (1) Regular user activities including login, donor registration, donation history, and blood request.

2. **Admin Mode**:

    (1) Specific administrative privileges allowing access to manage user profiles, donation requests, and database maintenance.

3.7.2 User Class

User Categories:

1. **Donors**:

    (1) Requirements specific to donor registration, donation history maintenance, and login functionalities.

2. **Recipients**:

    (1) Requirements for blood request submissions, search for donors, and login functionalities.

3.7.3 Objects

Database Objects and Associated Services:

1. **User Object**:

    (1) Attributes: Username, Email/Phone, Password

    (2) Services: User authentication, profile management

2. **Donor Object**:

    (1) Attributes: Donor Information, Donation History

    (2) Services: Donor registration, donation history updates

3. **Blood Request Object**:

(1) Attributes: Recipient Information, Request History

(2) Services: Blood request submission, request status updates

4. **Search Donor Object**:

   (1) Attributes: Search criteria (blood type, location)

   (2) Services: Search functionality to find available donors

3.7.4 Feature

Raktdaan Features:

1. **Login Feature**:

   (1) Stimulus: User login request

   (2) Response: Authentication, access to user-specific functionalities

2. **Donor Registration Feature**:

   (1) Stimulus: New donor registration request

   (2) Response: Collection and storage of donor information in the database

3. **Blood Request Feature**:

   (1) Stimulus: Recipient's request submission

   (2) Response: Recording the request and updating the request status

4. **Blood Search Feature**:

   (1) Stimulus: Recipient's search query for available donors

   (2) Response: Displaying a list of matching donors

5. **Donate History Feature**:

   (1) Stimulus: Donor's request to view donation history

   (2) Response: Retrieval and display of donation history data

3.7.5 Stimulus

System Functions Driven by Stimuli:

1. **User Interaction**:

   (1) Login attempts, registration submissions, donation history queries

2. **Request Submission**:

   (1) Blood donation requests, donor registrations

3. **Search Query**:

   (1) Recipient's search for available donors based on specific criteria

3.7.6 Response

System Functions Generating Responses:

1. **Authentication Response**:
   (1) Successful login, failed login attempts
2. **Data Entry Response**:
   (1) Successful donor registration, blood request submission
3. **Search Result Response**:
   (1) Displaying search results for available donors

3.7.7 Functional Hierarchy

Functions Organized by Common Inputs/Outputs:

1. **Input-Oriented Functions**:
   (1) Processes related to user input (login credentials, donor details, request information)
2. **Output-Oriented Functions**:
   (1) Processes generating responses or displaying output (authentication results, search results, history display)

3.8 Additional Comments

The Logical Database Requirements for the Raktdaan can be organized through various perspectives. One approach involves categorizing requirements based on different operational modes. This includes outlining functionalities for both Normal Mode, encompassing user activities like login, donor registration, and blood request, and Admin Mode, focusing on database management tasks for administrators.

Another organizational method revolves around specific objects within the database. For instance, requirements pertaining to the User Object encompass user authentication and profile management, while the Donor Object involves donor registration and donation history maintenance. The Blood Request Object handles the submission and status tracking of blood requests, and the Search Donor Object deals with query execution and donor information retrieval.

A third approach involves structuring requirements around distinct features offered by the app. These features encompass login functionality, donor registration, blood request submission, blood search capabilities, and donation history viewing. Each feature is presented as a stimulus eliciting a specific system response.

Lastly, functional hierarchy-based requirements categorize functions into input-oriented and output-oriented processes. Input-oriented functions handle user input related to login details, donor information, and request submission. Meanwhile, output-oriented functions generate system responses or display outputs such as authentication results, search findings, and donation history display.

This multifaceted organization ensures a comprehensive understanding of the Raktdaan's database requirements, catering to different operational modes, database objects, featured functionalities, and the overall functional hierarchy of the system.

4. Change Management Process

To manage changes in the Logical Database Requirements for the Raktdaan, a systematic change management process is in place. Any proposed changes are logged and evaluated for their impact on the database design and project scope. Changes are formally submitted using a predefined request form or template, preferably in writing via email or a dedicated change management system. The team responsible for requirements management assesses the feasibility and alignment of each change with project goals. Significant changes may require consensus among stakeholders or the project team for approval. Upon approval, updates are made to the System Requirement Specification (SRS) document, maintaining proper documentation and version control. This structured process ensures controlled modification of requirements while aligning with project objectives

********************************Thank You **********************************