

| **TITLE:** Requirement **S**pecification Document |
| --- |

**AIM:** To learn and understand the way of analysing the gathered information in the previous phase for the development process and prepare requirement specification document. A concept of software engineering. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Course outcome of Experiment:**

Process of gathering requirements and converting them into specifications.

Document created will be used by both, the customer and the developer, to understand WHAT is going to be developed.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Roger Pressman, Software Engineering: A practitioners Approach, McGraw Hill, 2010 ,6th edition

2. Ian Somerville, Software Engineering , Addison Wesley,2011,9th edition

1. http://en.wikipedia.org/wiki/Software\_requirements\_specification

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

**Requirements analysis** in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. It is an early stage in the more general activity of requirements engineering which encompasses all activities concerned with eliciting, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Conceptually, requirements analysis includes three types of activities:

* **Eliciting requirements**: the task of identifying the various types of requirements from various sources including project documentation, (e.g. the project charter or definition), business process documentation, and stakeholder interviews. This is sometimes also called requirements gathering.
* **Analysing requirements**: determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts.
* Recording requirements: Requirements may be documented in various forms, usually including a summary list and may include natural-language documents, use cases or process specifications.

New systems change the environment and relationships between people, so it is important to identify all the stakeholders, taken into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. These may include the development of scenarios, the identification of use cases, the use of workplace observation or ethnography, holding interviews, or focus groups (more aptly named in this context as requirements workshops, or requirements review sessions) and creating requirements lists. Prototyping may be used to develop an example system that can be demonstrated to stakeholders. Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced

Different types of Requirements

* Functional requirements
* Usability requirements
* Reliability requirements
* Performance requirements
* Security requirements

A typical SRS document template is shared subsequently. This document acts as a reference and will be used by both, the customer (for whom the software system is to be developed), and the organization which develops the solution. Typically, prepared by the development organization at the early stage of development by the professionals after interacting with the customer.

**Software Requirements Specification for:**

**ManSick:Mental Health App**

**Version 1.0**

**Prepared by Kashish Mamania**

**KJSCE**

**18/07/2024**

**Table of Contents**

**Table of Contents ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Product Scope 1

1.3 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Introduction**

**Translating mental health research into actionable community initiatives faces hurdles like information accessibility, communication gaps, and stigma, necessitating innovative solutions.**

**Purpose**

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

**The objective of our mental health website project is to create a supportive online platform that promotes mental wellness by offering a safe space for sharing experiences, raising awareness about mental health conditions, and providing access to valuable resources. The platform aims to empower users with tools and coping strategies, foster a sense of community through discussions and support groups, and facilitate access to professional help. Ultimately, we seek to help individuals take control of their mental health, find hope, and build resilience to navigate life's challenges effectively.**

**Product Scope**

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>*

**The scope of our mental health website project includes providing a supportive community for users to share experiences, offering a curated repository of resources on mental health topics, and implementing search and filtering capabilities. The platform will offer expert advice from mental health professionals, facilitate community support through discussions and peer groups, and provide educational content on conditions and treatment options. Accessibility and robust privacy measures will be prioritized, ensuring a secure and inclusive experience. Additionally, feedback will be collected for continuous improvement, and collaborations with mental health organizations will enhance the platform's credibility and accuracy.**

**References**

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

**[1] S. Agarwal, S. Gupta, and V. Goel, "Development of College Management System using ASP.NET and SQL Server," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 11, pp. 563-567, 2013.**

**[2] A. Kulkarni and P. Rao, "Design and Implementation of College Management System using PHP and MySQL," International Journal of Computer Applications, vol. 128, no. 7, pp. 19-25, 2015.**

**[3] Stansfield, J. (2023) Developing Knowledge Translation Mechanisms to Advance Public Mental Health as a**

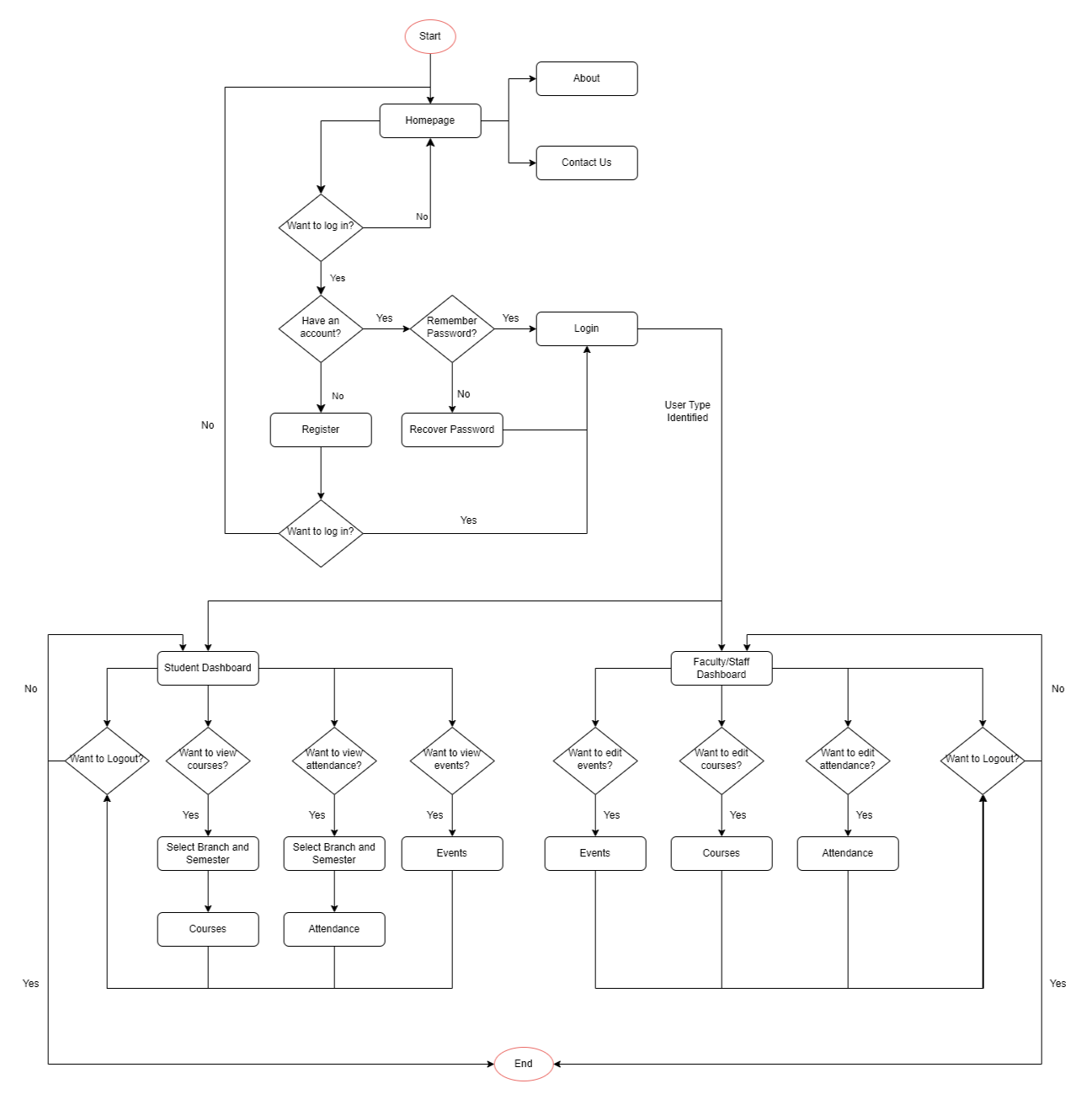
**Complex Public Health Priority [Online]. Leeds Beckett University.**

**Overall Description**

**Product Perspective**

*<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>*

Module Wise Flow Diagram:



**Product Functions**

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

**The design of the mental health website focuses on creating a user-friendly and supportive environment with a calming UI, intuitive navigation, and accessibility considerations. The homepage highlights key features like the user sharing platform, resource repository, and community support options. Users can create personalized profiles with privacy settings and engage in anonymous sharing, discussions, and peer support. The platform also provides access to curated mental health resources and expert advice from professionals. Optimized for mobile responsiveness, the site ensures seamless access across devices. Regular feedback and updates ensure the platform remains relevant and effective.**

**Operating Environment**

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

**Software used:**

**Application Development Tools:**

**1. IDE: Visual Studio Code**

**2. DB: xampp (Apache, MySQL), phpmyadmin**

**3. Testing: Internet Browser**

**Programming Softwares:**

**1. Web page Semantics: HTML**

**2. Web page Design: CSS**

**3. Web page Design: Bootstrap CSS**

**4. Web page Functionalities: JavaScript**

**5. Backend: PHP**

**6. Database: MySQL**

**Hardware used:**

**1. Device with internet connection**

**2. 2 GB RAM**

**3. Web Browser (Google Chrome, Opera, Firefox)**

**Design and Implementation Constraints**

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

* **Regulatory and Corporate Policies: Compliance with data privacy laws (e.g., HIPAA, GDPR) and corporate standards.**
* **Hardware Limitations: Constraints on server capacity and memory affecting performance.**
* **Interfaces to Other Applications: Challenges with integration and compatibility with third-party APIs.**
* **Specific Technologies and Tools: Limitations due to the choice of technologies (e.g., Flutter, Firebase).**
* **Parallel Operations: Need for compatibility with existing systems and data synchronization.**
* **Language Requirements: Support for multiple languages and localization efforts.**
* **Communications Protocols: Security protocols for data transmission, like HTTPS.**
* **Security Considerations: Robust security measures, including authentication and encryption.**
* **Design Conventions and Programming Standards: Adherence to established design and coding standards.**

**User Documentation**

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

* **User Manual:**

**Comprehensive guide with detailed instructions and screenshots.**

**Covers all website features and functionalities.**

**Provided in PDF format.**

* **Online Help:**

**Integrated help system with context-sensitive topics.**

**Searchable knowledge base for easy access to information.**

* **FAQs:**

**Frequently asked questions section addressing common queries.**

**Organized by categories for quick reference.**

* **In-App Tips and Notifications:**

**On-screen tips and notifications for new features and updates.**

**Tooltips and pop-up messages for contextual help.**

**Assumptions and Dependencies**

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

**Assumptions:**

* **Users will have reliable internet connectivity for accessing the website.**
* **Third-party APIs and services will remain stable and functional.**
* **Legal and regulatory standards (e.g., GDPR, HIPAA) will not change significantly.**

**Dependencies:**

* **Integration with third-party APIs for authentication, payments, and content.**
* **Reliance on chosen database (e.g., Firebase) for data management.**
* **Use of tools for accessibility and data privacy compliance.**
* **Dependency on content management systems for resource delivery.**
* **Dependence on hosting providers for performance and scalability.**

**External Interface Requirements**

**User Interfaces**

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

**Logical Characteristics:**

* **Visuals will be designed to ensure clarity and ease of use, following standard user interface design principles.**
* **Adherence to established GUI standards and product family style guides to ensure consistency and usability.**
* **Consistent layout with clear navigation elements and section headings.**
* **Includes common buttons like "Help," "Submit," and "Cancel" on every screen.**
* **Common shortcuts for navigation and actions (e.g., Ctrl+S for Save).**
* **Clear, user-friendly error messages displayed with suggestions for resolution.**

**Software Components:**

* **Interfaces for creating and managing user profiles.**
* **Navigation and search functions for accessing articles, videos, and other resources.**
* **Interfaces for forums, chat rooms, and peer support groups.**
* **Scheduling and messaging features for consultations with mental health professionals.**

**Hardware Interfaces**

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

**Supported Devices:**

**Desktop computers, laptops, smartphones, and tablets.**

**Compatible with standard input devices (keyboard, mouse) and touchscreens.**

**Data and Control Interactions:**

**Data exchanged between the website and hardware includes user input, display output, and media content.**

**Uses standard web protocols (HTTP/HTTPS) for data transmission between hardware and the web server.**

**Software Interfaces**

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

**Databases: Firebase (for data storage and user management).**

**Operating Systems: Compatible with major operating systems (Windows, macOS, iOS, Android).**

**Tools and Libraries: Flutter (for frontend development), and other relevant libraries for added functionality.**

**Integrated Components: Third-party APIs for authentication, payment processing, and content integration.**

**Incoming Data: User input, resource requests, and interaction data.**

**Outgoing Data: Response data, notifications, and content delivery.**

**Services Needed: Data synchronization, authentication, and real-time communication.**

**Data Sharing: Shared data includes user profiles, resource usage statistics, and community interactions.**

**Communications Interfaces**

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

**Protocols: Utilizes HTTP/HTTPS for secure web communications.**

**Message Formatting: Standard web formats (JSON, XML) for data exchange.**

**Security and Encryption: Implement SSL/TLS for secure data transfer.**

**Data Transfer Rates: Optimized for typical web speeds, with considerations for mobile users.**

**Synchronization: Real-time updates for interactive features and content delivery.**

**System Features**

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

**System Feature 1**

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

4.1.1 Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

4.1.2 Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:

REQ-2:

1. **User Authentication and Profile Management**

**Description: Secure user accounts with authentication and profile creation featuring personalization options.**

**Stimulus/Response Sequences:**

* **Stimulus: User navigates to the login page and enters credentials.**
  + **Response: System authenticates credentials and grants access to the user’s profile.**
* **Stimulus: User accesses the profile creation page.**
  + **Response: System allows the user to enter and save personal information and preferences.**

**Functional Requirements**

**REQ-1: The system must authenticate user credentials securely using encrypted protocols.**

**REQ-2: The system must allow users to create and update profiles with personal information and customization options.**

**REQ-3: The system must handle invalid login attempts by displaying an error message and providing options to reset the password.**

1. **Mood Tracking**

**Description: Feature for daily mood tracking and journaling to monitor emotional well-being.**

**Stimulus/Response Sequences**

* **Stimulus: User logs mood for the day via the mood tracking interface.**
  + **Response: System records the mood entry and updates the user's mood history.**
* **Stimulus: User writes a journal entry.**
  + **Response: System saves the journal entry and displays a confirmation message.**

**Functional Requirements**

**REQ-4: The system must provide an interface for users to log their mood daily.**

**REQ-5: The system must allow users to write and save journal entries.**

**REQ-6: The system must handle input errors by validating data and providing user feedback.**

1. **Mindfulness and Relaxation Techniques**

**Description: Provides guided meditation sessions, breathing exercises, and relaxing sounds or music playlists.**

**Stimulus/Response Sequences**

* **Stimulus: User selects a guided meditation session.**
  + **Response: System starts the session and provides audio or video playback.**
* **Stimulus: User initiates a breathing exercise.**
  + **Response: System provides instructions and guides the user through the exercise.**

**Functional Requirements**

**REQ-7: The system must offer various guided meditation sessions with audio or video content.**

**REQ-8: The system must provide interactive breathing exercises with visual or auditory guidance.**

**REQ-9: The system must allow users to access and play relaxing sounds or music playlists.**

1. **Sleep Tracking**

**Description: Integration with sleep tracking features and tips for improving sleep hygiene.**

**Stimulus/Response Sequences**

* **Stimulus: User connects a sleep tracking device.**
  + **Response: System syncs data and displays sleep metrics.**
* **Stimulus: User accesses sleep hygiene tips.**
  + **Response: System provides a list of tips and recommendations.**

**Functional Requirements**

**REQ-10: The system must integrate with external sleep tracking devices and sync sleep data.**

**REQ-11: The system must provide actionable tips for improving sleep hygiene based on user data.**

1. **Crisis Support**

**Description: Provides emergency helplines, crisis chat or text support, and integration with local emergency services.**

**Stimulus/Response Sequences**

* **Stimulus: User selects a crisis helpline.**
  + **Response: System displays contact information or connects the user directly.**
* **Stimulus: User initiates a crisis chat or text.**
  + **Response: System provides a real-time chat or text interface with support services.**

**Functional Requirements**

**REQ-12: The system must provide up-to-date emergency helplines and contact information.**

**REQ-13: The system must support real-time chat or text communication for crisis support.**

**REQ-14: The system must integrate with local emergency services where available.**

1. **Therapist Directory**

**Description: Searchable database of mental health professionals with reviews, ratings, and appointment scheduling options.**

**Functional Requirements**

**REQ-15: The system must provide a searchable directory of mental health professionals with filters.**

**REQ-16: The system must allow users to read reviews, view ratings, and schedule appointments.**

**REQ-17: The system must handle appointment scheduling and send confirmation notifications.**

**System Feature 2**

**Other Nonfunctional Requirements**

**Performance Requirements**

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

**Interactive tools should respond within 2 seconds of user input to maintain a smooth user experience and handle up to 10,000 simultaneous users without performance issues. Real-time data synchronization must occur within 5 seconds to ensure accurate reflection of user progress.**

**Safety Requirements**

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

**To safeguard users, interactive tools must include encryption and secure storage to protect user data, content warnings for sensitive topics, and comply with safety regulations and data protection standards such as GDPR.**

**Security Requirements**

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

**The system must implement multi-factor authentication for accessing sensitive features and ensure data encryption both in transit and at rest. Compliance with security standards like ISO/IEC 27001 and local data protection laws is mandatory.**

**Software Quality Attributes**

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

**The interactive tools should be user-friendly, with a focus on ease of use over ease of learning, and ensure 99.9% system uptime. The code should be modular, well-documented, and adaptable to future changes, with a high standard of maintainability and reliability.**

**Business Rules**

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*

**Access to interactive tools should be role-based, with administrators having the ability to configure and update the tools, while content should be moderated to ensure quality and safety.**

**Other Requirements**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**The project requires a scalable database to manage dynamic content and user interactions, support for multiple languages and regional formats for internationalization, and adherence to legal requirements including data protection and accessibility laws.**

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Post Laboratory Activity:**

1. You are required to prepare an SRS document for any project. ( It could be the mini project you have completed in semester IV
2. Prepare a questionnaire for the allotted project considering your lab instructor is the client for requirement gathering.
3. Consider the following scenario: An institute is interested in developing a Library Information System (LIS) for the benefit of students and employees of the institute. LIS will enable the members to borrow a book (or return it) with ease while sitting at his desk/chamber. The system also enables a member to extend the date of his borrowing if no other booking for that particular book has been made. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books online. However, issuing or returning books is restricted to valid users (members) of LIS only.

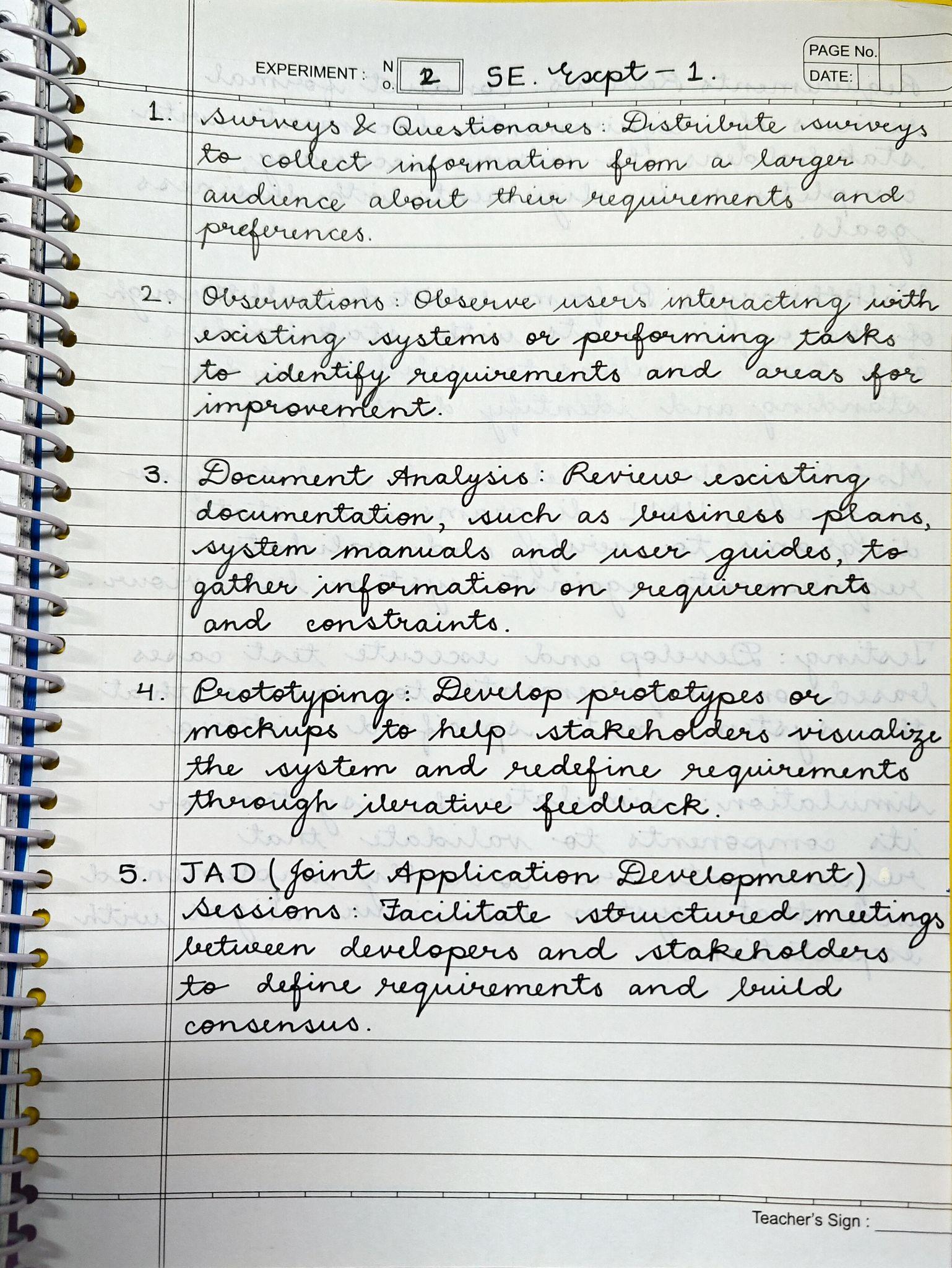
The final deliverable would be a web application (using the recent HTML 5), which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (e.g. passwords) is stored in plain text.

Prepare an SRS document for the same in the format discussed in the write-up.

**Post Lab Descriptive Questions answers must be handwritten and to be submitted BEFORE the next term**.

* 1. What are different techniques to gather information for software development?

**Ans:**



* 1. List verification and validation techniques for requirements.

**Ans:**

