**Experiment 7**

**PART A**

|  |  |
| --- | --- |
| Roll.No. : A016, A018, A022 | Name: Varun, Simran, Kartik |
| Sem/Year : Sem-07/ 4th year | Batch: 1 |
| Date of Experiment : 14-10-2022 | Date of Submission: 19-10-2022 |

**A.1 Aim**

To identify a project such as website or mobile app to redesign through the design life cycle (Discovery, define,Requirements analysis).

**A.2 Prerequisite**

Understanding of design life cycle

**A.3 Outcome**

After experimentation, students will be able:

* To understand the design life cycle.

**A.4 Theory**

Cyclic model that incorporates three phases (Bias and Mayhew, 2005):

1. **Requirements analysis**—In this step, you establish your user characteristics, what tasks the product requires for operation so you can determine what the users need to do, set your goals for the usability study, and determine the usability study design guidelines.
2. **Design, testing, and development**—In this step, you create a structured, top-down approach to designing the product, be it a user interface, Web site, documentation, or a combination of the three. This is the step that requires the most feedback from your project team.
3. **Installation**—In this step, you gather feedback from users during and after the development process and share this feedback with the project team to determine if you need to make any product changes.

**A.5 Tasks to perform**

* 1. Identify a project and get approved by the faculty.
  2. Specify the problem statement and Requirement gathering of project.

**(PART - B)**

(TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical)

|  |  |
| --- | --- |
| Roll.No. : A016, A018, A022 | Name:Varun, Simran, Kartik |
| Sem/Year : Sem-07/ 4th year | Batch: 1 |
| Date of Experiment : 14-10-2022 | Date of Submission: 19-10-2022 |
| Grade -- |  |

**B.1: Task assigned:**

Identify a project and get it approved.

SRS Document attached below.

**B.2: Observations and Learnings:**

Write the documents containing problem definition, requirement gathering.

**B.3: Conclusion:**

(Students must write the problem statement, requirement and complete the redesign a project)

Therefore, problem statement and requirement analysis was performed

**Software Requirements Specification**

**for**

**ThoughtSpot**

**Version 1.0 approved**

**Prepared by Varun Khadayate A016**

**Simran Kumari A018**

**Kartik Padave A022**

**SVKMs NMIMS Navi Mumbai**

**October 14, 2022**

**Table of Contents**

**Table of Contents** [**ii**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.ihv636)

**Revision History** [**ii**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.30j0zll)

**1. Introduction** [**1**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.32hioqz)

1.1 Purpose [1](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.1hmsyys)

1.2 Document Conventions [1](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.41mghml)

1.3 Intended Audience and Reading Suggestions [1](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2grqrue)

1.4 Product Scope [1](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.vx1227)

1.5 References [1](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.3fwokq0)

**2. Overall Description** [**2**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.1v1yuxt)

2.1 Product Perspective [2](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.4f1mdlm)

2.2 Product Functions [2](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2u6wntf)

2.3 User Classes and Characteristics [2](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.19c6y18)

2.4 Operating Environment [2](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.3tbugp1)

2.5 Design and Implementation Constraints [2](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.28h4qwu)

2.6 User Documentation [2](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.nmf14n)

2.7 Assumptions and Dependencies [3](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.37m2jsg)

**3. External Interface Requirements** [**3**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.1mrcu09)

3.1 User Interfaces [3](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2jxsxqh)

3.2 Hardware Interfaces [3](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.46r0co2)

3.3 Software Interfaces [3](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2lwamvv)

3.4 Communications Interfaces [3](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.111kx3o)

**4. System Features** [**4**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.3l18frh)

4.1 System Feature 1 [4](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.206ipza)

4.2 System Feature 2 (and so on) [4](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.4k668n3)

**5. Other Nonfunctional Requirements** [**5**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2zbgiuw)

5.1 Performance Requirements [5](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2bn6wsx)

5.2 Safety Requirements [5](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.1egqt2p)

5.3 Security Requirements [5](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.3ygebqi)

5.4 Software Quality Attributes [5](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2dlolyb)

5.5 Business Rules [5](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.sqyw64)

**6. Other Requirements** [**5**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.3cqmetx)

**Appendix A: Glossary** [**5**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.1rvwp1q)

**Appendix B: Analysis Models** [**5**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.4bvk7pj)

**Appendix C: To Be Determined List** [**6**](https://docs.google.com/document/d/1PnvrE679xKZX76dszOMcz89X7vtp8yAD/edit#heading=h.2r0uhxc)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Date** | **Name of the Requirement** | **Version** |
|  | 14-10-2022 | Introduction, Description, Requirements | V.01 |
|  | 17-10-2022 | System features, Other nonfunctional requirements | V.02 |
|  |  |  |  |

# Introduction

## **Purpose**

This document outlines the usage and implementation of the ThoughtSpot app. It is aimed at website and app developers, marketing staff, end users, project managers, and testers. Through this document, one may comprehend the project's problem statement and the functioning of each module, which will aid everyone in gaining a thorough grasp of this entire project.

## **Document Conventions**

This document was created based on the IEEE template for System Requirement Specification Documents. Main headings are numbered in whole numbers with a bold, Times New Roman font. Subheadings are numbered as a number following the decimal point that is 1.1, 1.2, and so on.

## **Intended Audience and Reading Suggestions**

The target audience would include people all age groups, genders, and segments of the population. Individuals seeking to cope with daily stressors and find relaxation, novice meditators, students, working professionals, etc.

## **Product Scope**

The growing pressures of daily life are driving more people to use mental health apps when seeking psychological support. This trend has accelerated significantly in the past year due to social distancing demands that have been imposed as a result of the COVID-19 pandemic. While digital mental health services are increasingly being endorsed by governments and health professionals as a low-cost alternative to therapy, there are potential downsides. The aim is to design an effective mental health solution that will address the numerous concerns with the current apps and will serve users of all age groups, and offer cost-effective and easily accessible support services. The app will enhance user experience, expand accessibility, provide customized settings, support groups facilities, and one-on-one therapy sessions and a lot more. It will also tackle problems of data breaches by providing better security so that users don’t have to worry about privacy risks.

## **References**

[User Research Document](https://drive.google.com/file/d/1bBIvR-LiZ3ZyVpS66Rzk5Xreix3zoxtl/view?usp=sharing) where the research was conducted using interviews to form a general opinion about needs of users and their expectations from the solution being developed, followed by a thorough comparative analysis of existing solutions in the market.

[User Persona Document](https://drive.google.com/file/d/1RpX4ogBQV-m0H36-eOvIIGGuVQIT8GYt/view?usp=sharing) which helped in thoroughly understanding the users pain points and their requirements.

# Overall Description

## **Product Perspective**

ThoughtSpot is a mobile app developed keeping the drawbacks of the recent apps in mind. It seeks to close the gaps left by current solutions and offer a better and more efficient way to monitor and manage your mental health. The app offers a wide range of functionalities, and serves users of all age groups. The features include short meditations, thought journals, support groups, one-on-one therapy sessions, various personalised recommendations like movies, podcasts, books, and self-monitoring options. In addition, it also provides platform to learn new skills from sources like udemy, coursera, skillshare, etc. Most importantly, it tackles problems of data breaches by providing better security so that users don’t have to worry about privacy risks.

Information Architecture: It organizes, structures, and labels contents in an effective and suitable way in order to help users find information and complete tasks.

## **Product Functions**

**Home Page**

The homepage has a distinctive vibe and aesthetic that makes the user feel welcome and engaged. It includes information like a crisis helpline, notifications, and quick links to other pages like podcasts, blogs, and course suggestions. It also has a goals page for users to track their own emotions, view previous records, journal, set daily tasks, and gain insights.

**Experts**

This page allows users who need professional help, to connect and book a session with a certified expert with only a click. It offers crucial details about the therapist as well as details on their methodological approaches. The sessions are offered via voice, video, and text messages.

**Chat Window**

In order to facilitate efficient communication, it gives users a quick way to interact with members of their community group and their therapists.

**Community Page**

It helps users to connect to various community groups available for them to join in and discuss their problems. There’s also a feature where the users can post and comment anonymously.

**Recommendation Page**

This pages consists of various suggestions for users based on what they are looking for. It includes recommendations for podcasts, articles and blogs, and also suggests users to enroll for various courses based on their interests.

**Profile Page**

It contains information like account settings, notifications, provides users with option of anonymous mode, track their goals, about us, and user feedback options. It offers users a convenient way to edit their profile, check important information, and offer feedback.

## **User Classes and Characteristics**

## **Operating Environment**

ThoughtSpot is mobile application which is compatible with both the smartphone platform i.e. Apple or Android .

1. **User Documentation**

Users will have an 24/7 support System and Helping Guide to get started. Users can also access FAQs for Frequently faced issues.

## **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).>

# 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.>

## **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.>

## **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.>

## **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.>

# 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:

## **System Feature 2** (and so on)

# 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.>

## **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.>

## **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.>

## **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.>

## **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.>

# 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.>

**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.>

**Appendix C: To Be Determined List**

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>