University of Dhaka

Department of Computer Science and Engineering

CSE-3112: Software Engineering Lab

Meramot : Comprehensive Tech Services and Supports

Software Requirements Analysis Document (RAD)

Submitted By:

Abdullah Ibne Hanif Arean
Roll: 12

Mahmudul Hasan
Roll: 20

Submitted To :

Dr. Saifuddin Md. Tareeq
Professor And Chairman
Department of Computer Science and Engineering
University of Dhaka
Redwan Ahmed Rizvee
Lecturer
Department of Computer Science and Engineering
University of Dhaka
Submitted On :

February 21, 2023

Contents

1 Introduction

1.1 Purpose of the System

"Meramot: Comprehensive Tech Services and Supports" aims at creating a locally operated, web-based platform that aims to provide a comprehensive and convenient solution for technology-related issues in Bangladesh. It offers an interactive interface for posting and resolving issues, live chat support, on-demand doorstep paid servicing with live tracking and a fully functioning question-answering web app. The system seeks to fill the gap in the market by catering to the needs of both technical and non-technical individuals in Bangladesh and improving the accessibility of Bangla resources. The system addresses the lack of locally operated tech support platforms and the heavy dependence on repair shops in Bangladesh that often have unqualified technicians and charge unnecessary fees to non-technical users. The project's purpose is to simplify and streamline the tech support experience, making it more convenient, reliable, and efficient for all users.

1.2 Scope of the System

The software requirements specified in this document are for the Meramot platform, version 1.0. The scope of this RAD [1] includes the entire Meramot system, which encompasses all subsystems and functionality. This document describes the entire scope of the Meramot platform and includes all subsystems necessary to support its functionality. The Meramot platform is intended to be an all-encompassing solution that replaces traditional applications with additional, useful features such as collaboration, advanced search and categorization capabilities, authorization controls, and more.

1.3 Objective and Success Criteria of the Project

- Increase the accessibility of Bangla resources for non-technical individuals, through the provision of the platform in both Bangla and English.
- Increase the uptake of the platform, with an increase in the number of users posting problems and engaging with the question-answering web app.
- Reduce the turnaround time for resolving technology-related issues reported on the platform.
- More users are taking paid live chat support and on-demand doorstep paid-to-service.
- Simplify and streamline the tech support experience, making it more convenient, reliable, and efficient.

1.4 Definitions, Acronyms and Abbreviations

- RAD = Requirement Analysis Document
- NPM = Node Package Manager
- UML = Unified Modeling Language, It is a standardized language used for modeling software systems and designing software architectures. UML provides a common set of notations, diagrams, and methods for describing and visualizing software systems and their components, helping to improve communication and understanding among development teams, stakeholders, and users.

1.5 References

[1] A. H. D. Bernd Brugge, *Object Oriented Software Engineering Using UML, Patterns, And Java Third Edition*. Prentice-Hall, Inc., 2010.

[2] R. S. Pressman, *Software Enginnering A Practitioner's Approach Sixth Edition*. Mc Graw Hill Higher Education, 2005.

[3] Freecodecamp.org, *Uml for begineers*, https://www.youtube.com/watch?v=WnMQ8HlmeXc&ab_channel=freeCodeCamp.org, [Online; accessed 21-February-2023], 2022.

[4] L. Chart, *Uml with lucid chart*, https://lucid.app/, [Online; accessed 21-February-2023], 2023.

1.6 Overview

Meramot is a locally operated, comprehensive web-based platform that offers one-stop solutions for all software and hardware-related issues in Bangladesh, with interactive interfaces for posting and resolving issues, and on-demand doorstep servicing with live tracking, and live chat.

2 Overall Description

2.1 Product Perspective

"Meramot: Comprehensive Tech Services and Supports" is a web-based platform that aims to provide users with a collaborative problem-solving environment. This product can be thought of as the replacement for certain applications that provide a lot of similar functionalities. However, it has additional useful functionalities compared to the existing alternatives such as collaboration, advanced search and categorization features, authorization, etc. In essence, Meramot provides a one-stop solution for technical problem-solving by connecting users with technical experts and service providers in real-time. The platform is designed to be easy-to-use, efficient, and provide a personalized experience to all its users.

2.2 Product Function Detail

- The system will provide a locally operated, web-based platform for comprehensive tech services and support in Bangladesh, addressing the current lack of such platforms and providing a solution to common technology-related issues.
- The platform will be available in both Bangla and English, ensuring accessibility for non-technical individuals and increasing the availability of Bangla resources for those who may not be fluent in English.
- The system will offer an interactive interface for posting and resolving issues.
- On-demand doorstep/shop paid servicing with live tracking, and live chat support to provide users with comprehensive, convenient, and efficient tech support.
- The system will allow users to post technology-related questions, issues, or problems they are facing in an interactive and user-friendly interface.

- The interface will include fields for tags, titles, descriptions, and screenshots if necessary, providing users with an easy way to accurately describe their issue and provide necessary details.
- A search bar will be available on the interface, enabling users to look for previously posted questions and their answers, potentially providing immediate solutions without needing to repost a question.
- Other users and experts on the platform can collaborate to solve the issue by responding with potential solutions or asking for further clarification, providing a community approach to resolving tech problems.
- Once the issue is resolved, the person who posted the question can mark it as resolved or choose the best answer, which provides a clear indication to others who may have similar issues.
- The intuitive interface of the platform will enable users to navigate through the system with ease and speed, allowing them to find the information they need quickly and efficiently.

2.3 User Profiles

2.3.1 Guest

- Characteristics: Unregistered user with no authorization to access advanced features of the platform.
- Frequency of use: As often as needed to browse or search for problem solutions.
- Subset of product functions used: Can view landing page, search and browse existing posts, comments, and solutions, but cannot post anything or provide feedback on existing content without registering for an account.

2.3.2 User

- Characteristics: This user is the main user of the system, and will use the system to post problems and get solutions. They may have varying levels of technical expertise but should be able to use the system without extensive technical knowledge.
- Frequency of use: Regular, as the system will be used to solve everyday problems.
- Subset of product functions used: The user will use most of the core functionalities of the product, including posting problems, searching for solutions, collaborating with others, and marking problems as resolved.

2.3.3 Technical Experts

- Characteristics: This user provides technical expertise in a specific area, and will use the system to provide solutions to problems related to their area of expertise. They have a high level of technical knowledge and experience.
- Frequency of use: Varies depending on the amount of problems posted by users related to their area of expertise.
- Subset of product functions used: The technical expert will use the system to search for problems related to their area of expertise, view details of the problems, and post solutions to the problems.

2.4 Constraints

- **Hardware Limitations:** Meramot is currently an Webapp-only application and requires a modern Web Browser to use it. Users will need to have access to a device with a supported web browser to use the platform.
- **Online Connection:** Meramot is an online platform and an internet connection is required to sign up and use the platform. It is also required for syncing data to an online database, which is essential for data management.
- **Time Constraint:** The Meramot project has a fixed deadline, which makes effective project management crucial to ensure timely completion. The development team will need to work efficiently to meet the deadline and ensure that the project is delivered on time.
- **Budget Constraint:** The Meramot project has a fixed budget, and it is important to manage costs effectively to ensure that the project stays within budget. The development team will need to be mindful of expenses and allocate resources efficiently to avoid overspending.
- **Technical Constraint:** The Meramot project will require advanced technical skills and knowledge, and it is important to ensure that the development team has the necessary technical capabilities. The team will need to have expertise in the technologies being used to ensure that the platform is developed effectively.
- **Data Privacy and Security Constraints:** The Meramot project will handle a lot of user data, and it is important to ensure that the platform is secure and that user data is protected by implementing standard security protocols and guidelines. It is essential to maintain the privacy and security of user data to build trust with the user base.
- **Scalability Constraints:** The Meramot project aims to cater to a large number of users, and it is important to ensure that the platform can handle a large number of users and traffic spikes. The platform should be scalable enough to handle any future growth in user traffic.
- **Language Constraint:** The Meramot platform will be available in both Bangla and English, and it is important to ensure that the platform is accessible to users who are not fluent in either language. This will require effective translation and localization of the platform to make it easy to use for all users.
- **Legal Constraints:** The Meramot project will be subject to local laws and regulations, and it is important to ensure that the platform complies with all relevant laws and regulations. The development team will need to ensure that the platform is legally compliant to avoid any legal issues in the future.
- **Technical Debt:** The Meramot project will use multiple technologies, and it is important to ensure that the platform is maintainable and that the development team is aware of the trade-offs of using different technologies. The team will need to ensure that the platform is easy to maintain and update to avoid any technical debt.

2.5 Assumption and Dependencies

2.6 Assumptions

- The device that will run the program must fulfill the minimal hardware requirements.
- To use the program without interruptions, the user must have access to a reliable internet connection.
- There will be no unexpected difficulties with the application's backend, which might cause substantial delays or even failure in development.
- There will be no schedule conflicts or time limits during the development process, which might adversely impact the project's timeframe.
- The user will have basic computer literacy and be able to navigate and use the platform's features effectively.
- Users will be able to accurately and clearly articulate their questions and problems, which will allow other users to provide helpful and relevant answers.

2.7 Dependencies

- React JS is used for building the user interface of the application, providing a modern and efficient front-end framework.
- SpringBoot Java Framework is utilized for building the Rest API and back-end, ensuring a robust and reliable system.
- PostMan API is employed for testing and debugging the Rest API.
- Maven and NPM are utilized to efficiently and flexibly build the application, providing a streamlined and manageable process.

3 Proposal System

3.1 Overview

Meramot is a locally operated, comprehensive web-based platform that offers one-stop solutions for all software and hardware-related issues in Bangladesh, with interactive interfaces for posting and resolving issues, and on-demand doorstep servicing with live tracking, and live chat.

3.2 Functional Requirement

3.2.1 Requirement 1 (User Visits as Guest)

- ID: Meramot001
- Name: User Visits as Guest

- Description: The system shall allow non-registered and unauthorized users to access the landing page (home page) of the platform. These users will be able to view all posts and comments on the platform, search for expected problems and solutions, and access the Sign-Up option for registration. However, non-registered and unauthorized users will not be able to post anything, comment on existing posts, or upvote any solution.
- Priority: HIGH
- Reference: This requirement supports the system's functionality of allowing users to explore the platform's content without registration, providing a transparent and user-friendly interface, and promoting user engagement through the Sign-Up option.

3.2.2 Requirement 2 (User Registration and Authentication)

- ID: Meramot002
- Name: User Registration and Authentication
- Description: The system must provide secure user authentication, allowing registered users to log in to their accounts using their email address and password. Upon successful login, users will be granted access to their personalized dashboard and all features associated with their user role.
- Priority: HIGH
- Reference: This requirement is critical for ensuring user account security, maintaining user privacy, and delivering personalized user experiences.

3.2.3 Requirement 3 (User Post Functionality)

- ID: Meramot003
- Name: User Post Functionality
- Description: The system shall allow registered users to post their technical problems on the platform's home page. The post can include a description and images of the problem. The post will be displayed on the timeline of other registered users who are viewing the home page. The system shall provide instant suggestions for the problem posted. The user who posted the problem can mark it as resolved if they can solve it.
- Priority: HIGH
- Reference: This requirement facilitates collaborative problem-solving by enabling registered users to share their technical issues with others and get instant suggestions.

3.2.4 Requirement 4 (Commenting on Posts)

- ID: Meramot004
- Name: Commenting on Posts

- Description: The system should enable users, service providers, technical experts, and administrators to comment on user posts, offering potential solutions and providing assistance. The commenting feature should be user-friendly and support rich-text formatting, including the ability to add images and videos. Comment threads should be well-organized, facilitating clear and concise communication.
- Priority: MEDIUM
- Reference: This requirement is crucial for fostering collaborative problem-solving and knowledge-sharing among the platform's user base.

3.2.5 Requirement 5 (Upvoting and Downvoting Solutions)

- ID: Meramot005
- Name: Upvoting and Downvoting Solutions
- Description: The system should allow users to easily upvote or downvote suggested solutions based on their usefulness. The upvoting and downvoting functionality should be intuitive and clearly displayed on each solution post. Solutions with more upvotes should be displayed prominently to help users quickly identify the best solutions.
- Priority: LOW
- Reference: This requirement ensures that the most effective solutions are given priority, enhancing the overall efficiency of the problem-solving experience for all users.

3.2.6 Requirement 6 (Live Chat Support)

- ID: Meramot006
- Name: Live Chat Support
- Description: The system must offer live chat support to users, enabling them to address their tech-related problems in real-time. Users should be able to initiate a live chat session with available service providers and technical experts. The chat interface should facilitate the exchange of text, images, and files, providing a better understanding of the problem at hand.
- Priority: HIGH
- Reference: This requirement provides users with fast and efficient real-time support, enhancing the overall user experience.

3.2.7 Requirement 7 (Requesting Service Provider)

- ID: Meramot007
- Name: Requesting Service Provider

- Description: The system should enable users to search and request service providers for technical support. Users should be able to search for service providers based on their location, services offered, and ratings. Once a service provider is selected, users can request their services and arrange a mutually convenient time and date for the service to be performed.

- Priority: MEDIUM

- Reference: This requirement supports the system's core functionality of connecting users with technical experts and service providers to resolve their technical issues.

3.3 Non Functional Requirement

3.3.1 Usability

- The platform must have an intuitive and user-friendly interface.

- The platform should be accessible to users with varying degrees of technical expertise.

- The platform should support the use of Bangla language for better understanding by non-technical users.

- The platform should have easy-to-follow instructions and prompts to help users navigate and perform tasks.

3.3.2 Reliability

- The platform should be highly reliable, with minimal downtime or outages.

- The platform should have a system for regular backups of data to ensure no data is lost.

- The platform should be designed with a fail-safe mechanism to prevent the loss of user data.

3.3.3 Performance

- The platform must be highly responsive and have fast load times.

- The platform must be able to handle a large volume of users simultaneously.

- The responsiveness of Mermot will not be affected by online connectivity but fetching details for the links will require a stable internet connection.

- Fetching details should ideally be done in less than 3 seconds on a 4G connection.

- Apart from the user's connection downtimes, Meramot's user-specific information should be synchronized to the cloud 99 percent of the time.

- The platform should provide real-time updates on the status of user requests.

3.3.4 Supportability

- The platform should provide multiple channels for user support, including email, live chat, and phone support.
- The platform should provide a comprehensive and easily accessible user manual or help section.
- The platform should be designed to handle user requests in a timely and efficient manner.

3.3.5 Implementation

- The platform should be implemented using modern web technologies and frameworks.
- The platform should be scalable and able to handle future growth and expansion.
- The platform should be easy to maintain and update as needed.

3.3.6 Scalability

- The platform should be designed to scale easily to accommodate growing numbers of users and service providers.
- The platform should be able to handle a large volume of user requests without impacting performance.

3.3.7 Security

- The platform must be highly secure and protect user data and personal information from unauthorized access and breaches.
- The platform should be designed with secure communication protocols to protect user data in transit.
- The platform should provide multi-factor authentication and other security features to prevent unauthorized access to user accounts.

3.3.8 Testability

1. The platform should have a robust testing and quality assurance process to ensure that it is functioning properly.
2. The platform should be tested thoroughly before deployment to ensure that it meets all functional and non-functional requirements.
3. The platform should be tested regularly to identify and address any issues or bugs that arise.

3.3.9 Maintainability

- The platform should be designed to be easily maintained and updated as needed.
- The platform should have a clear and well-documented codebase.
- The platform should have a system for tracking and addressing bugs and issues in a timely manner.
- The code's cyclomatic complexity should not exceed 7. No method in any object may include more than 500 lines of code.

3.4 System Models0

[2]

3.4.1 Scenarios

3.4.1.1 User Visits as Guest

- A user who isn't registered/authorized will be able to see the landing page(home page).
- Can see posts and comments.
- Can search the page for expected problems and solutions.
- Can't post anything.
- Can't comment or upvote to any solution.
- If the user wishes can register from the Sign-Up option.

3.4.1.2 Registers/Signs Up

- When the user hits the Register button, the user will be routed to the registration page.
- On the registration page, the user will enter his name, email address, password, and profile picture.
- After requesting registration with the necessary information, the user will get a response from our system.
- If the system responds with a successful message the user will be redirected to the profile page.
- If the system responds with a message other than successful, the warning/error message will be shown to the user.

3.4.1.3 Logs In

- If the user isn't logged in, by clicking the log-in button, the user will be routed to the login page.
- User will provide email and password to log in.
- System will authenticate and authorize the user.
- If the user is authenticated, the system will respond with the necessary data and the user will be redirected to the home page.
- If any error occurs, the system will show an error/warning message.

3.4.1.4 Edits Profile

- In the edit page, the user will be able to change the name, password, and profile picture by providing the password.
- If the user provides valid info and the password, he can update his info.

3.4.1.5 User Posts

- A registered user can post his problem from the home page.
- The post may contain pictures and a description of the problem.
- The post will appear in the home page's timeline of viewing users.
- User will get an instant suggestion regarding the problem in the post.
- User can mark the post as resolved if he/she can solve his problem.

3.4.1.6 Comments in Posts

- Any registered user or technical experts can comment on the posts.
- Any comment can be upvoted/downvoted depending on the quality of the comment.
- Comments will be sorted depending on the upvote count.

3.4.1.7 User Consults with Technical Experts

- User can consult with technical experts through messages.
- When a user gives a message for consultation with our technical panel, an expert will be assigned to the user to help.

3.4.2 Use Cases

3.4.2.1 User Login

Actor:

- User
- Technical Expert

Flow of Events:

1. Clicks to the Login button.
2. Enters Email Address and Password.
3. Submits for authentication and authorization.
4. Redirects to the Landing page.

Entry Condition: Already having registered.
Exit Condition: Successfully logged in.

3.4.2.2 Registration

Actor: User
Flow of Events:

1. Clicks to the Register button.
2. Enters name, email, password, description.
3. Submits for authentication and authorization.
4. Redirects to the Landing page.

Entry Condition: Internet connection with a web browser.
Exit Condition: Successful Registration message or Account already exists message.

3.4.2.3 Edit Profile

Actor: User
Flow of Events:

1. Clicks to the Profile button.
2. In the Profile Section, click to Edit Profile
3. Edits required information.
4. Provides password to update info.
5. Submits for updating information.

Entry Condition: Logged-in to the system.
Exit Condition: Successfully Updates profile or leaves without changing.

3.4.2.4 User Posts

Actor: User
Flow of Events:

1. Goes to the Home page/
2. Clicks to the Create Post button.
3. Enters description of the problem.
4. Uploads necessary photos.
5. Chooses some categories to the relevant problem
6. Makes post by clicking Post button

Entry Condition: User Logged-In to the System.
Exit Condition: Makes a successful post.

3.4.2.5 Edit Posts

Actor: User
Flow of Events:

1. Clicks to the post.
2. Makes necessary edit.
3. Cliks on update button.

Entry Condition: Already having a posted.
Exit Condition: Successfully done posting.

3.4.2.6 Comment on Post

Actor:

- User
- Technical Expert

Flow of Events:

1. Clicks to the comment button.
2. Clicks in comment edit text field.
3. Writes comment that is relevant to the post.
4. Submits the comment

Entry Condition: Logged-In to the system.
Exit Condition: Makes a successful comment.

3.4.2.7 Search Posts

Actor:

- User
- Technical Expert
- Guest

Flow of Events:

1. Clicks on search bar.
2. Write searching keywords.
3. May choose tags.
4. Clicks search icon.
5. Shows the expected posts.

Entry Condition: User Logged-In to the System.
Exit Condition: Makes a successful post.

3.4.2.8 View Posts
Actor:

- User
- Technical Expert
- Guest

Flow of Events:

1. Searches the posts
2. Filters the posts
3. Clicks on posts to see problems
4. Clicks on comments to view problem suggestions

Entry Condition: Having internet connection.
Exit Condition: Done seeing expected problem and solutions.

3.4.2.9 Upvote/Downvote Suggested Solution
Actor: User
Flow of Events:

1. Go to the comment section.
2. Justify the solution.
3. Click upvote/downvote depending on your justification.

Entry Condition: Logged-In to the system.
Exit Condition: Makes a upvote or downvote.

3.4.2.10 Live Chat Support
Actor:

- User
- Technical Expert

Flow of Events:

1. Clicks to the Live Chat button.
2. Describe the problem to get help in message.
3. Sends the message.
4. Technical Experts receives the message.
5. Consults with the User about the solution.

Entry Condition: Logged-In to the system.
Exit Condition: Interactive discussion takes place.

3.4.2.11 Authenticate
Actor:

- Invoked by User
- Invoked by Technical Expert

Flow of Events:

1. Checks the email address and password.
2. If correct returns successful.
3. If error happens, sends error message.

Entry Condition: Already having registered.
Exit Condition: Having valid credentiality.

3.4.2.12 Authorization
Actor:

- Invoked by User
- Invoked by Technical Expert

Flow of Events:

1. Check the role.
2. Sends data and authority according to role

Entry Condition: Already having registered and correct credentials.
Exit Condition: Having information about the user.

3.4.2.13 Handles Errors
Actor:

- User
- Technical Expert
- Guest

Flow of Events:

1. Checks any special scenario occurs such as connection down.
2. If any error occurs, shows the error/warnings.

Entry Condition: Any web browser and having access to the meramot website.
Exit Condition: Getting proper message from system.

3.4.3 Use Case Model

[3]

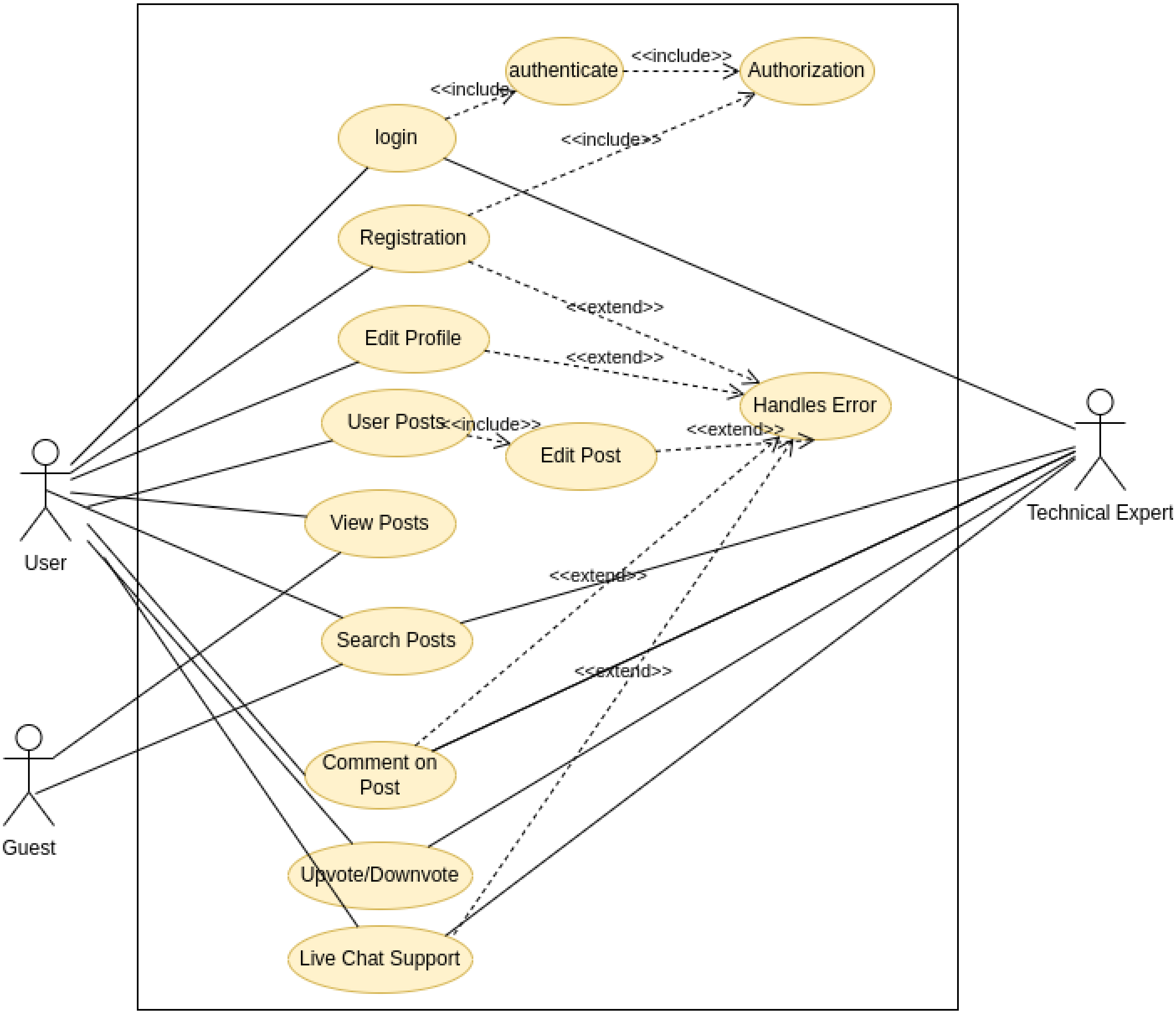

Figure 1: Use Case Diagram

3.4.4 Dynamic Model

3.4.4.1 Sequence Diagram

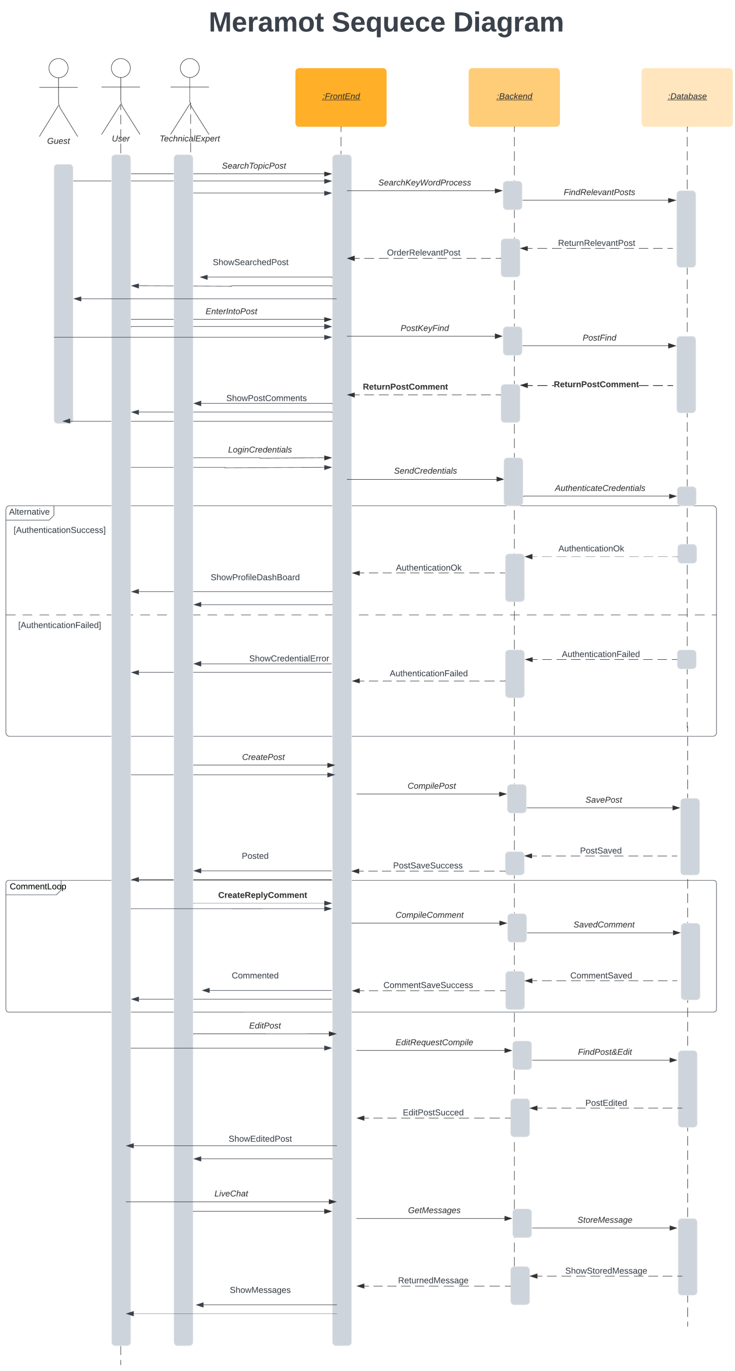

Figure 2: Meramot's Sequence Diagram

3.4.4.2 Activity Diagram

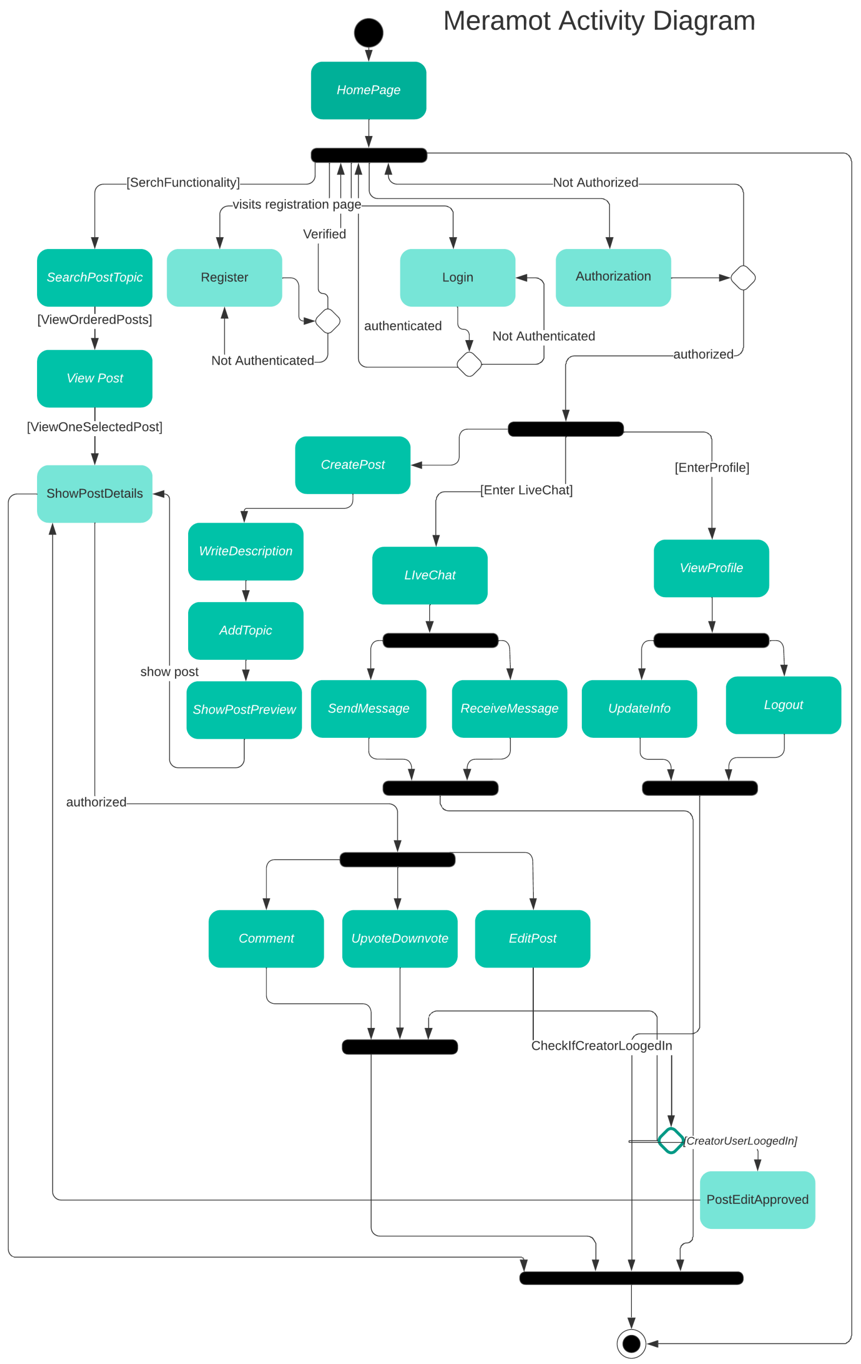

Figure 3: Activity Diagram

3.4.4.3 State Diagram

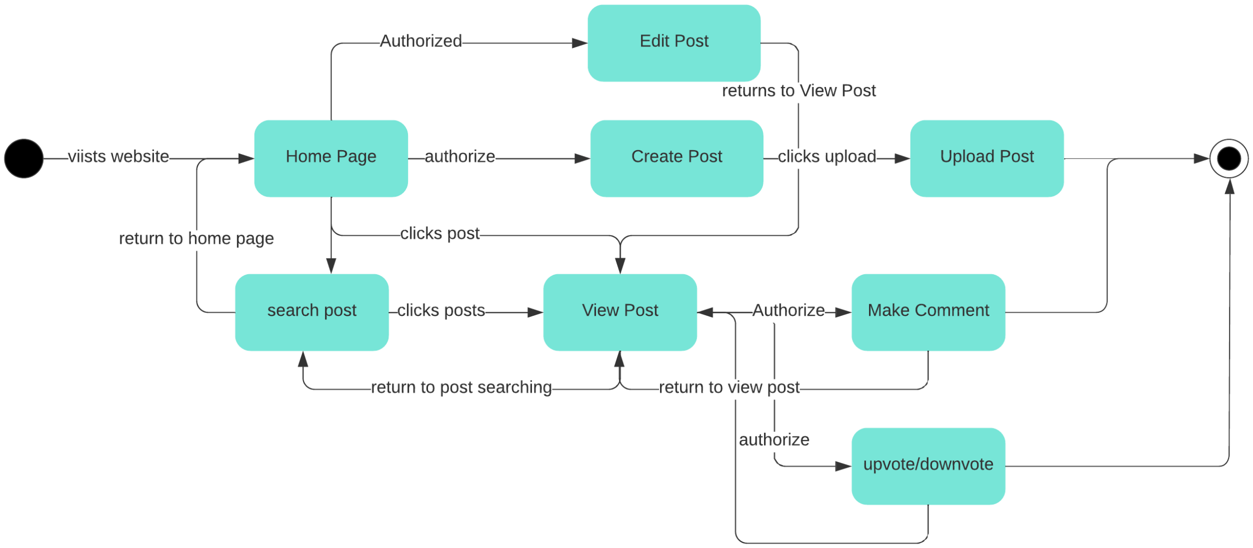

Figure 4: User Activity State Diagram

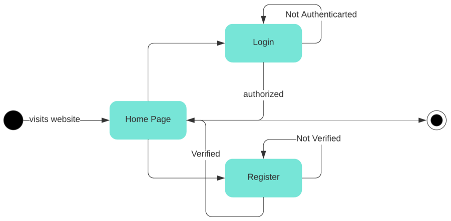

Figure 5: Login-Registration State Diagram

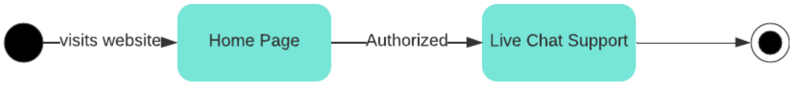

Figure 6: Live Chat Support State Diagram

3.4.5 User Interface

3.4.5.1 User Interface The User Interface of Meramot is designed according to conventional design standards with thoughtful implementations of different components. The User Interface of Meramot can be estimated to have the following screens to the best of our knowledge, however this is subject to change based on changing requirements -

1. Login Page
    - (a) email field
    - (b) password field
    - (c) login button
    - (d) registration page route
2. Registration Page
    - (a) email field
    - (b) password field
    - (c) login button
    - (d) registration page route
3. Home Page
    - (a) User Post
    - (b) Search bar
    - (c) Post Creation option for registered user
    - (d) Live Chat Option for registered user
4. Live Chat Page
    - (a) Chat Inbox
5. User Profile Page
    - (a) User Info
    - (b) User Info Update Option
6. Create Post Page
    - (a) Picture and Text input options
    - (b) Post Upload Button
7. User Post
    - (a) Problem Description
    - (b) Comment Section
    - (c) Upvote and downvote options

3.4.5.2 Software Interface

Browser:

1. Chrome Dev 112 (112.0.5594.1) or Higher
2. Firefox 111.0 or higher
3. Safari 13.1.2 or higher

Operating System:

1. Windows 7 or higher
2. Any version of MacOS
3. Any Version of Linux Kernerl Based OS

3.4.5.3 Hardware Interface

Screen size and resolution: As **Meramot** is a responsive website, it is readable and navigable in various devices of various screen sizes and resolution.
Input methods: The application is designed to accommodate different input methods, such as touchscreens, keyboards, and mice, to ensure that it can be easily used on a variety of devices.
Performance requirements: We considered the hardware capabilities of the devices that users will use to access the app, and ensured that the app's performance requirements are appropriate for a range of devices. The app works on any devices having at least 2GB of RAM.
Compatibility: We tested the app's compatibility on different hardware configurations and operating systems to ensure that it can be accessed by as many users as possible. The app is compatible in any OS e.g. MacOS, Ububtu, POPOS, Windows etc.
Accessibility: We designed the app to be accessible to users with different needs, such as users with visual impairments or mobility impairments, by incorporating special hardware, such as screen readers or alternative input devices.

4 Supporting Information

This document does not need any supporting information from other documents. All the diagrams presented in this document were created with LucidChart [4] Online Diagram maker.