Department of Computer Science

University of Gujrat

Voice-Based Blogging Website For Blind People

Session : BSCS-2020

Project Advisor: Mr. Muhammad Abrar

Submitted By

AMINA FAYYAZ
20011519-057

Department of Computer Science
University of Gujrat

STATEMENT OF SUBMISSION

This is certify that **Amina Fayyaz** Roll No. 20011519-057 has successfully completed the final year project named as **Voice-Based Blogging Website for Blind People** at the Department of Computer Science, University of Gujrat, to fulfill the requirement of the degree of **Bachelor's in Computer Science.**

_____

Mr. Muhammad Abrar
Project Supervisor

_____

Project Coordination Office
Faculty of CS&IT-UOG

_____

Mr. Zaheer
Head of the Department

ACKNOWLEDGEMENT

I truly acknowledge the cooperation and help provided by **Mr. Zaheer Ahmed**, Head of Computer Science Department, University of Gujrat. He has been a constant source of guidance throughout the course of this project. I also like to thank **Mr. Muhammad Abrar** for his help and guidance throughout this project. I am also thankful to our friends and families whose silent support led me to complete my project.

Amina Fayyaz

Date: 22nd July, 2024

Abstract

This project introduces a pioneering project focused on the development of a voice-based blogging website tailored specifically for blind individuals, aimed at fostering inclusive online communication and expression. In response to the digital accessibility challenges faced by visually impaired users, the project integrates state-of-the-art voice recognition technology with intuitive user interfaces to facilitate seamless content creation and interaction. Through a user-centered design approach, the platform offers an accessible and intuitive environment where blind users can effortlessly compose, edit, and publish blog posts using voice commands. Furthermore, the system employs advanced audio feedback mechanisms and tactile interfaces to enhance user engagement and navigation. The project's implementation and evaluation encompass usability testing, user feedback collection, and iterative refinement processes to ensure optimal functionality and user satisfaction. By empowering blind individuals with the tools to participate actively in the digital discourse, this initiative contributes to the promotion of inclusivity and diversity in the online community.

Chapter 1: Project Feasibility Report

1.1. Introduction

Our project introduces a groundbreaking voice-based blogging website tailored explicitly for the visually impaired community. In a digital landscape primarily dominated by visual content, we recognize the critical need to create inclusive platforms that cater to diverse user needs. Our platform revolutionizes the blogging experience by providing an intuitive interface where blind individuals can effortlessly create, share, and engage with content using voice commands. By eliminating the barriers posed by traditional text-based interfaces, we aim to empower blind individuals to express themselves, share their stories, and connect with others in a seamless and meaningful manner.

Through the power of voice technology, our platform fosters a sense of community and belonging among visually impaired users, offering them a dedicated space to showcase their perspectives and experiences. With an emphasis on accessibility and usability, our website features intuitive navigation, robust voice recognition capabilities, and inclusive design elements tailored to the unique needs of blind individuals. By amplifying the voices of the visually impaired, we aspire to promote greater understanding, empathy, and inclusion in the digital sphere, ultimately paving the way for a more equitable and accessible online world. Join us as we embark on this transformative journey to harness the power of technology for the betterment of blind individuals worldwide.

1.2. Project Feasibility Report

The project aims to develop a voice-based blogging website tailored to meet the needs of blind individuals, facilitating their active participation in online communication and content creation. There are many types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Specification
- Information
- Motivational
- Legal and Ethical

1.2.1. Technical Feasibility

Technical Feasibility deals with asking the question as to whether the system can be developed or not. It is one of the most important questions before starting the project because it is assessing the limits of theory or technology applicable to the project. Another important query to be answered is to evaluate whether you (the project members or organization) possess the technology and technical expertise.

1.2.2. Operational Feasibility

Evaluation of technical ability of the staff to operate the project is the main aim of operational feasibility. In this area the question arises as to whether the problem is worth solving and if the solution provided for the problem works or not. How do end users and managers feel about the problem or solution is another query to be answered.

1.2.3. Economic Feasibility

Justification for the benefit/cost analysis relative to the project is to be measured in economic feasibility. Therefore, economic feasibility can be divided into two parts; cost estimates and benefit estimates. Cost estimates can further be alienated into development or acquisition costs (one time) and maintenance and operation costs (ongoing). In order to find development costs, break the project into tasks and use the lifecycle cost models. Experienced costs gained from similar projects should then be used to make estimates. The function point metric should be calculated.

Benefit estimates enclose tangible benefits and intangible benefits. Tangible benefits would include reduced costs and increased revenues. However, information quality, job satisfaction, and external standing are examples of intangible benefits.

1.2.4. Schedule Feasibility

Time is an important factor. The assessment and evaluation of the completion of a project with the available staff and resources within time is very essential. Meeting deadlines and milestones should always be kept in mind.

1.2.5. Specification Feasibility

Requirements are the features that the system must have or a constraint that must be accepted for the customer. The question arises as to whether the requirements are clear and definite. The scope boundaries must also be assessed.

1.2.6. Information Feasibility

The feasibility of information must be assessed regarding its completion, reliability, and meaningfulness.

1.2.7. Motivational Feasibility

Evaluation of the client staff regarding the motivation to perform the necessary steps correctly and promptly must occur.

1.2.8. Legal & Ethical Feasibility

”Do any infringements or liabilities arise from this project? “ is the main focus of this feasibility.

1.3. Project/Product Scope

The "voice-based blogging website for blind people" project entails defining the boundaries and requirements specific to the website's features and functionalities tailored for blind users, including voice recognition, text-to-speech, and accessible interface components. This involves aligning available resources such as time, budget, and human resources with the defined requirements and breaking down the project scope into

manageable milestones through iterative development. Throughout the process, continuous monitoring and control of scope changes are essential, with a focus on negotiating the inclusion of requirements based on attributes like priority, effort, and risk to maintain project feasibility and alignment with stakeholder needs. By prioritizing requirements and adopting an iterative approach, the project can adapt to feedback and evolve to meet the needs of blind users effectively while ensuring the project's overall success.

1.4. Project/Product Costing

The "voice-based blogging website for blind people" project, metrics play a crucial role in assessing both the knowledge-oriented and achievement-oriented aspects of development. Knowledge-oriented metrics focus on tracking and evaluating the development process itself, aiding in prediction and monitoring. Achievement-oriented metrics, on the other hand, measure product aspects related to overall quality. Cost estimation primarily relies on algorithmic modeling, where costs are analyzed using mathematical formulas that link inputs with metrics to produce estimates. These formulas are derived from historical data analysis. To enhance the accuracy of cost estimation, the model needs calibration to the specific development environment, which involves adjusting the weightings of the metrics to reflect the project's unique characteristics and requirements accurately.

1.4.1. Project Cost Estimation By Function Point Analysis

Information domain values are defined in the following manner:

Number of user inputs:

- Each distinct voice command or input providing application-oriented data to the software.
- Formula for counting: Count of unique voice commands recognized by the system.
- For example:
    - Create a new article
    - Open User Profile
    - Edit an article
    - Help
    - Open all articles
    - Open my articles
    - Navigate to pages(7)

Number of user outputs:

- Each output delivering application-oriented information to the user through speech or text.
- Formula for counting: Count of unique responses or outputs generated by the system.
- For example:
    - "Your post has been created."
    - "Comment added."
    - "Going to Home Page."
    - "You liked this article."
    - "Your post has been deleted."(5)

Number of user inquiries:

- Each inquiry initiated by the user resulting in immediate software response.
- Formula for counting: Count of distinct user queries or requests handled by the system.
- For example:
    - "Open article*."
    - "Scroll up and Scroll down."
    - "Navigate to the home page."
    - "Log out"(4)

Number of files:

- Each logical grouping of data, such as user profiles, posts, comments, etc.
- Formula for counting: Count of distinct data files managed by the system.
- For example:
    - Admin
    - Configuration Settings
    - Articles
    - Users(4)

Number of external interfaces:

- All interfaces used for transmitting information to and from other systems.
- Formula for counting: Count of machine-readable interfaces facilitating data exchange with external systems.
- For example:
    - Database interface
    - Internal and External API’s
    - Libraries(3)

To compute function points (FP), the following relationship is used:

Count Total = 7+5+4+4+3 = 23
FP est. = 23 * [0.65 + 0.01 * (Fi)]

For example, if rated each GSC(General System Characteristics) as follows:

1. Data communications: 4
2. Distributed data processing: 3
3. Performance: 5
4. Heavily used configuration: 2
5. Transaction rate: 3
6. On-Line data entry: 4
7. End-user efficiency: 5
8. On-Line update: 3
9. Complex processing: 4
10. Reusability: 2
11. Installation ease: 3
12. Operational ease: 4
13. Multiple sites: 1
14. Facilitate change: 3

The **sum of GSC ratings** would be:

Fi = 4+3+5+2+3+4+5+3+4+2+3+4+1+3=46

This total sum of GSC ratings, in this case, is 46. You would then use this sum to calculate the value adjustment factor (Fi) as per the formula provided in the Function Point Analysis methodology.

FP est. = 23 * [0.65 + 0.01 * (Fi)]
= 23 * [0.65 + 0.01 * 46]
= 25.53

Finally, Total Project Cost and Total Project Effort are calculated given the average productivity parameter for the system.

The formulae are given as follows:

Cost / FP = labor rate / productivity parameter = 50/2 = 25/FP

Total Project Cost = FP est. * (cost / FP) = 25.53*25 = 638.25

Total Estimated Effort = FP est. / productivity parameter = 25.53/2 = 12.765

1.4.2. Project Cost Estimation by using COCOMO'81 (Constructive Cost Model)

Basic COCOMO

Type	Effort	Schedule
Organic	PM= 10	TD= 5
Semi-Detached	PM= 15	TD= 7
Embedded	PM= 20	TD= 9

Intermediate COCOMO

Type	Effort
Organic	M= 1.05
Semi-Detached	M= 1.10
Embedded	M= 1.15

1.4.3. Activity Based Costing

Activity-Based Management (ABM) can be conceptualized as follows:

Measuring Business Process Performance:

- Assessing the efficiency and effectiveness of each activity within a business process through verbal evaluations and analysis.
- Example: Analyzing the time taken and resources utilized for voice command recognition and response generation.

Estimating Cost of Outputs:

- Determining the cost associated with producing business process outputs by considering the resources consumed during voice-based operations.
- Example: Estimating the cost of generating a voice-based blog post by accounting for the time and effort spent by speech recognition software and content creation resources.

Identifying Efficiency Opportunities:

- Recognizing opportunities to enhance process efficiency and effectiveness by identifying inefficient or costly activities.
- Example: Detecting patterns of inefficiency in voice command processing and seeking ways to optimize speech recognition algorithms and resource allocation.

Basic Cost Drivers:

For each activity state in an activity diagram, the basic cost drivers are:

- **Resources:**
  - Identifying the business workers and entities involved in voice-based activities and quantifying their participation.

    - Example: Determining the number of speech recognition software instances and content creators engaged in producing voice-based blog posts.
- **Cost rate:**
    - Assigning a cost per unit of time for each business worker or entity involved in voice-based activities.
    - Example: Calculating the cost per hour of speech recognition software usage and content creation services.

- **Duration**:
    - Assessing the time duration for each activity within a voice-based workflow and allocating resources accordingly.
    - Example: Allocating speech recognition resources for the duration of voice command processing and content creation resources for the duration of blog post generation.

- **Overhead**:
    - Considering fixed costs associated with voice-based workflows and activities, such as infrastructure maintenance or software licensing fees.
    - Example: Accounting for fixed costs incurred by maintaining speech recognition servers and software licenses for content creation tools.

1.5. Task Dependency Table

Tasks	**Dependencies**	**Duration(days)**
Gather Functional requirements	-	1
Develop Functional specifications	-	1
Use case diagrams	1	1
Writing used cases	2	1
SRS Submission	1,3	1
UML Diagrams	5	5
Design Database	4,6	5
Design Review	6,7	2
Design Document Submission	8	1
Design the User Interface	9	31
Code of Authentication	8,10	6
Developing System Module & Configuration	11	13
Designing Different Test Cases	17	5
Unit Testing	18	3
Performance Testing	18,19	4
Acceptance Testing	18,20	2
Reporting Issues Found	21	1
Corrected	22	1
Verified	23	1

Table 1.1: Task dependency Table

1.6. CPM - Critical Path Method

1.6.1. Specify the Individual Activities

In our project we have divided our tasks in following activities. These activities are performed on weekly bases and their Gantt chart is also developed along with their Staff allocation.

1. Feasibility Study
2. Requirement Specification
3. Design Document
4. Interface Design
5. Backend Development
6. Database design
7. Web Development.

1.6.2. Determine the Sequence of the Activities

Activity	Name (Activity)	Immediate Predecessor	Duration (Weeks)
A	Feasibility	None	1
B	Requirement Specification	A	2
C	Design Document	B	2
D	Interface Design	B, C	3
E	Back end Development	B, C, D	5
F	Database design	E	2
G	Website Development	F	3
H	Testing	G	4

Table 1.2: CPM

1.6.3. Draw the Network Diagram

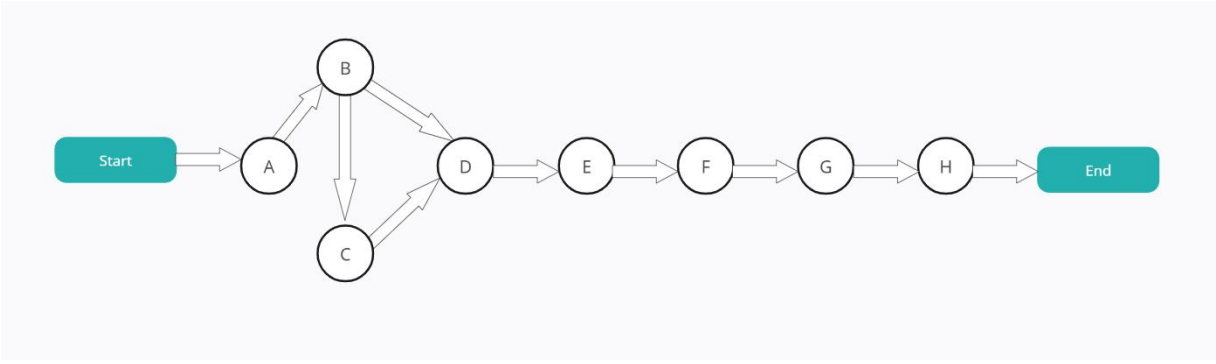


Figure 1.1: Network diagram

1.6.4. Estimate Activity Completion Time

Activity	Duration	ES	EF	LS	LF	TS	FS	Slack
A	1	0	1	0	1	0	0	0
B	1	1	3	1	3	0	0	0
C	2	3	5	3	5	0	0	0
D	3	3	6	3	6	0	0	0
E	5	6	11	6	11	0	0	0
F	2	11	13	11	13	0	0	0
G	3	13	16	15	18	0	0	2
H	4	16	20	18	22	0	0	2

Table 1.3: Activity Completion Time

1.6.5. Identify the Critical Path

As the critical path is the path with the highest value of duration so: Critical path of Our Project is:

A → B → C → D → E → F → G → H = 15 Weeks

1.7. Gantt chart

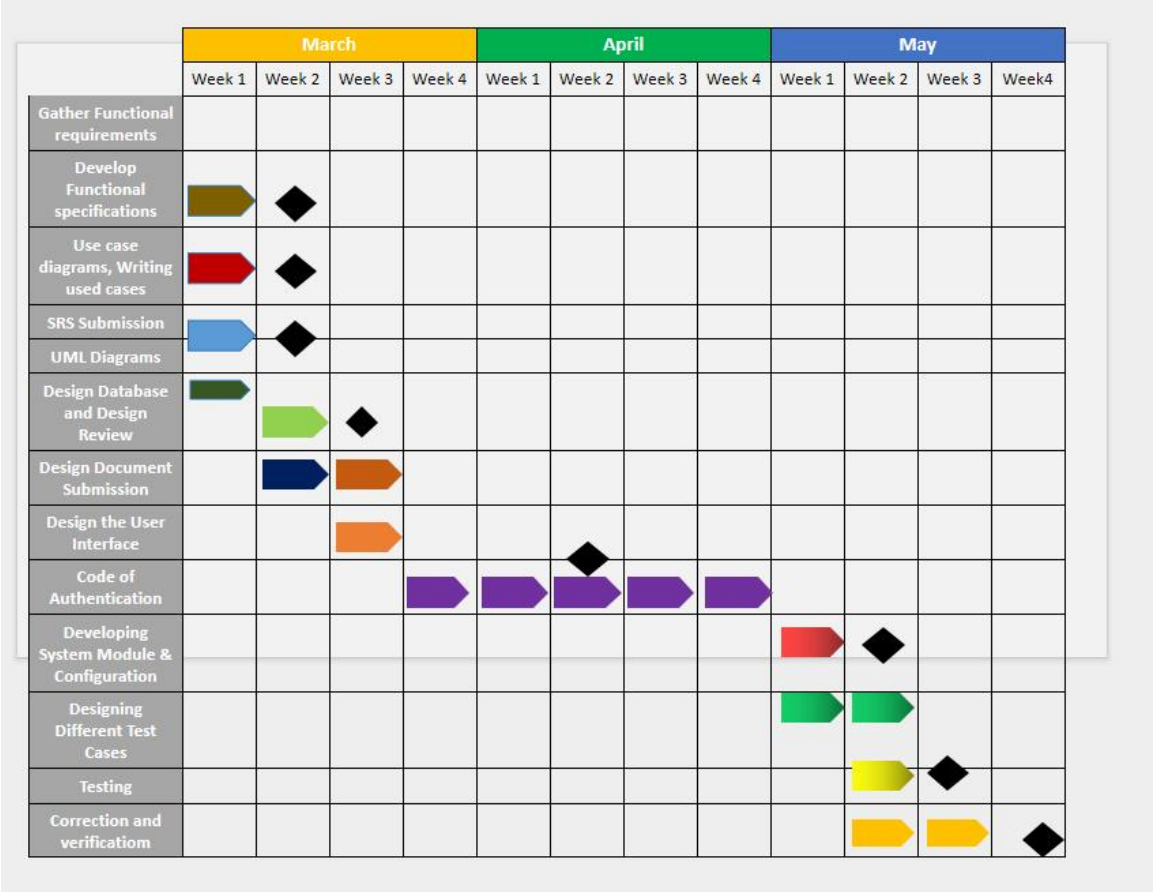


Figure 1.2: Gantt Chart

1.8. Introduction to Team member and their skill set

As I am working on my project alone I have no team mates or project leader. I have a wide skill set like I'm great problem analysis, problem identifying, critical solution finding job, hard working, great at both front-end and back-end development with database connectivity and API's integration, and also I'm an intermediate level of AI and ML code generator and flutter developer as well as python developer along with AI integration.

1.9. Task and Member Assessment Table

M1 member name **Amina Fayyaz**.

The task assignment to the development team is given below: Task durations and dependencies:

Task durations and dependencies

Tasks	Dependencies	Duration(days)
T1	-	1
T2	-	1
T3	1	1
T4	2	1
T5	1,3	1
T6	5	5
T7	4,6	5
T8	6,7	2
T9	8	1
T10	9	31
T11	8,10	6
T12	11	13
T13	17	5
T14	18	3
T15	18,19	4
T16	18,20	2
T17	21	1
T18	22	1
T19	23	1

Table 1.9.1: Task Duration and Dependency

1.10. Tools and Technology with reasoning

The application tools, which are to be used on front and back end of the system to be developed, should be listed. The reasons for these tools should also be described.
Identify what the needs for tool support are, and what the constraints are, by looking at the following:

- Development Process: For iterative development, automated testing tools like FLUTTER or NODE.JS can ensure efficient testing throughout the project, enabling rapid iterations and quality assurance.
- Host Platform(s): Cloud hosting platforms such as AWS (Amazon Web Services) or Google Cloud Platform provide scalability, reliability, and accessibility, making them suitable choices for hosting a voice blogging website.

- Target Platform(s): Considering the target platforms, web development frameworks like React.js for frontend and Node.js for backend can offer versatility and compatibility across different devices and browsers.
- Programming Language(s): JavaScript is a versatile language commonly used for web development. It offers libraries and frameworks like React.js and Node.js that facilitate the development of interactive and scalable web applications.
- Existing Tools: Leveraging existing tools such as Git for version control, Docker for containerization, and Jira for project management can streamline development processes and enhance collaboration within the team.
- Distribution of Development Organization: Communication and collaboration tools like Slack, Microsoft Teams, or Zoom can support effective coordination and communication among distributed development teams, ensuring seamless workflow and productivity.
- Size of Development Effort: Integrated development environments (IDEs) like Visual Studio Code or JetBrains WebStorm are suitable for projects of any size, offering robust features, extensions, and support for various programming languages and frameworks.
- Budget and Time Constraints: Open-source tools and frameworks like React.js, Node.js, and Express.js can significantly reduce development costs while offering powerful features and community support, aligning with budgetary constraints and time limitations.

1.11. Vision Document

The Vision defines the stockholder's view of the product to be developed, specified in terms of the stockholder's key needs and features. Containing an outline of the envisioned core requirements, it provides the contractual basis for the more detailed technical requirements.

- **Problem Statement:** Clearly articulate the challenges faced by blind individuals in accessing and contributing to online content, emphasizing the need for a voice-based solution.
- **Stakeholders:** Identify and involve blind individuals, accessibility experts, developers, content creators, and platform administrators as stakeholders to ensure their needs and perspectives are considered.
- **System Boundaries:** Define the scope and limitations of the voice-based blogging platform, including functionalities related to content creation, browsing, interaction, and administration.
- **Use Cases and Primary Scenarios:** Illustrate how blind users will navigate and interact with the platform through voice commands, highlighting key scenarios like creating posts, exploring content, and managing settings.
- **Constraints:** Explore constraints such as accessibility standards, technological limitations, budgetary constraints, and regulatory requirements to ensure compliance and feasibility.
- **Key Features:** Identify essential features like voice recognition, text-to-speech capabilities, intuitive navigation, content moderation, and accessibility settings to address the specific needs of blind users.

- **Consistency and Validation:** Ensure that the proposed features align with the identified problems and constraints, validating their effectiveness in addressing the needs of blind individuals.
- **Evolution and Adaptability:** Recognize that the vision document is subject to evolution as requirements, technology, and user feedback evolve. Prioritize adaptability and flexibility to accommodate future enhancements and changes.
- **System Boundaries:** Define the scope and limitations of the voice-based blogging platform, including functionalities related to content creation, browsing, interaction, and administration.

1.12. Risk List

For a voice-based blogging website tailored to blind users, the Risk List comprises potential hazards and uncertainties that may impede the project's success. These risks encompass challenges such as accessibility limitations, potential inaccuracies in speech recognition, insufficient content creation tools, technical integration hurdles, and low user adoption rates. Additionally, concerns regarding data security, platform compatibility with assistive technologies, resource constraints, regulatory compliance, and feedback iteration pose significant risks. By prioritizing these risks and developing robust mitigation strategies, the project team can navigate potential obstacles effectively, ensuring the platform's accessibility, usability, and overall success in serving the needs of blind individuals in the blogging community.

1.13. Product Features/ Product Decomposition

For a voice-based blogging website designed for blind users, the product features would revolve around intuitive voice navigation, content creation, and community engagement. These functionalities would enable users to seamlessly browse, search, and access blog posts and comments using voice commands. The platform should facilitate voice-enabled content creation, allowing users to compose, edit, and manage their blog posts and profiles through spoken prompts. Essential accessibility features such as text-to-speech capabilities for content consumption and voice recognition for user authentication would ensure an inclusive experience. Additionally, interactive features like voice-based commenting, liking, and sharing would promote community engagement among blind users, fostering a dynamic and accessible blogging environment.

Chapter 2: Software Requirement Specification (For Object Oriented Approach)

2.1 Introduction:

Requirements engineering process provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system. The task of capturing, structuring, and accurately representing the user's requirements so that they can be correctly embodied in systems which meet those requirements (i.e. are of good quality).

Here, **requirements specification** is to be discussed. Requirements specification would lead to the following four steps:

- Identify external interfaces
- Development of context diagram
- Capture “shall statements
- Allocate requirements
- Prioritize requirements
- Development of requirements traceability matrix

2.1.1 Systems Specifications

The following are the clauses that must be included while describing the system specifications.

Introduction

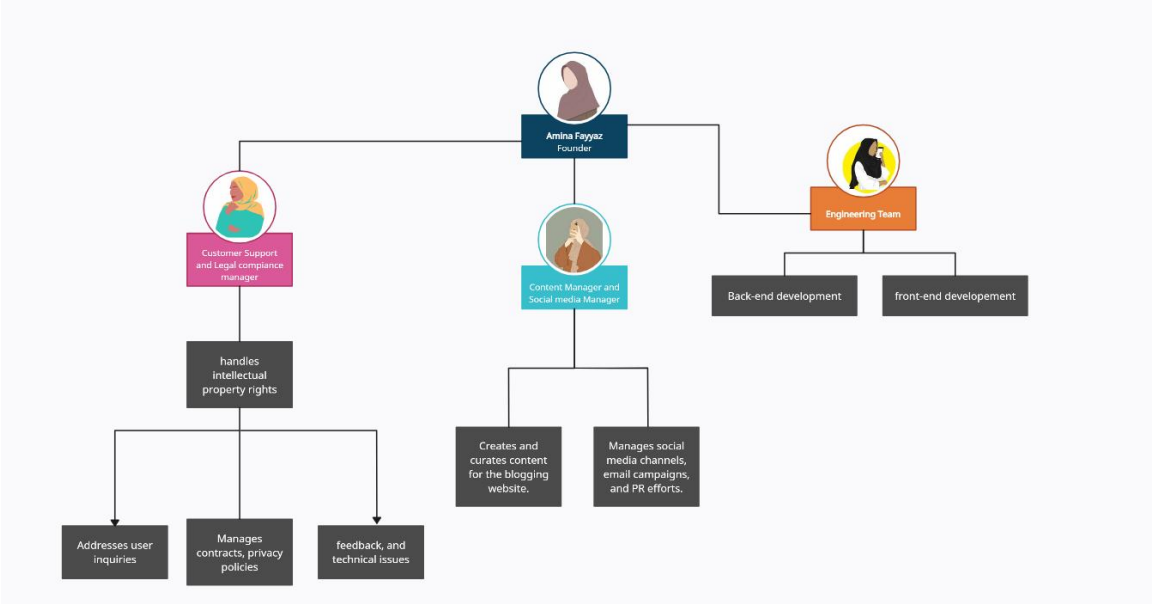
The introduction should provide a brief overview of the voice-based blogging platform, including its purpose, target audience (blind users), and the key features it aims to offer. This section may also include information about the organization behind the platform, its mission, and its commitment to accessibility and inclusivity.

Existing System

Here, the focus would be on elaborating the main functionalities and components of the existing voice-based blogging platform (if any) and identifying areas for improvement or enhancement. This could involve discussing the current challenges faced by blind users in accessing and interacting with blogging platforms.

Organizational Chart

In the context of a voice-based blogging platform, the organizational chart may not be directly applicable. However, this section could discuss the internal structure of the development team or the organizational structure of any supporting entities involved in the platform's development and maintenance.



Organizational Chart

Scope of the System

The scope of the voice-based blogging platform should clearly define its boundaries and the specific features and functionalities it intends to include. This section may also outline any limitations or constraints that need to be considered during the development process, such as technology compatibility or resource constraints.

Summary of Requirements: (Initial Requirements)

The summary of requirements should provide an abstract outlining the initial set of high-level requirements for the voice-based blogging platform. This may include requirements related to user authentication, content creation and management, accessibility features (such as voice navigation and screen reader compatibility), social interaction features, and security measures. The abstract serves as a foundation for further analysis and detailed requirement gathering.

2.1.2. Identifying External Entities

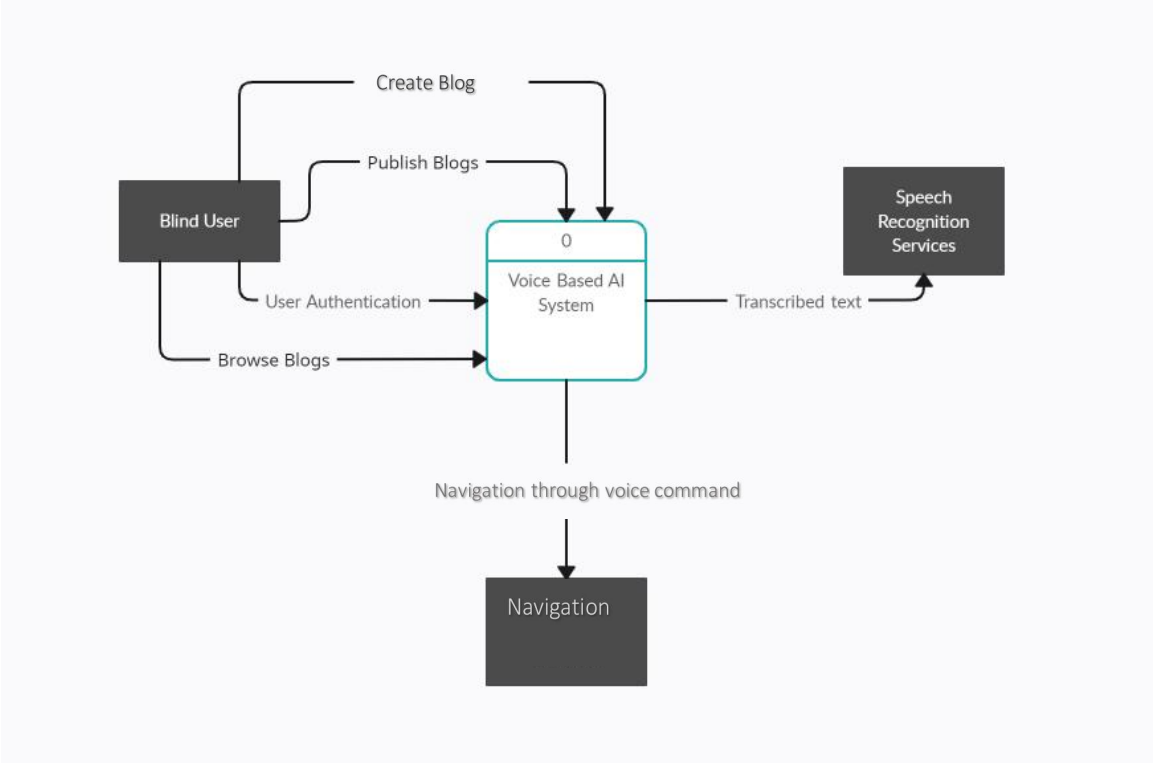
The process of identifying external entities in the context of a voice-based blogging platform for blind users involves first identifying potential entities based on the abstract and then refining them based on the platform's specific requirements and user interactions. This ensures that the platform effectively accommodates the needs of blind users and facilitates seamless interaction and engagement with the blogging environment.
The Identification of External Entities is done in two phases:

a. **Over Specify Entities from Abstract:** Based on the abstract of the voice-based blogging platform, potential external entities can be identified. These entities may include users (both blind and sighted), content creators, administrators, third-party integrations (such as social media platforms), and external services

(such as speech-to-text and text-to-speech engines). This initial identification provides a broad overview of the entities that interact with the platform.

b. **Perform Refinement:** Once the initial external entities are identified, they can be refined based on the platform's business logic and user requirements. For example, in the case of blind users interacting with the platform, the external entities may include assistive technology devices, accessibility standards, user preferences, and user support services. Refining the entities ensures that they align closely with the platform's goals and functionalities, taking into account the unique needs and challenges of blind users.

2.1.3. Context Level Data Flow Diagram:



Context Level Data Flow Diagram

2.1.4. Capture "shall" Statements:

Identify “shall” statements, as they would be all functional requirements.

Para#	Functional Requirements
1.0	A user "shall" be able to create and publish blog posts via voice commands.
1.0	Users "shall" have the ability to register and authenticate using voice recognition .
1.0	The system "shall" offer registration processes accessible through voice recognition.
1.0	Users "shall" be allowed to open/view user’s profile.
1.0	The platform "shall" open an article upon request via voice input.
1.0	The platform "shall" be able to edit a post via voice input.
1.0	The platform "shall" be able to delete a post via voice input.

2.1.5. Allocate Requirements:

Allocate the requirements in the use cases.

Para #	Functional Requirements	Use Case Name
1.0	A user "shall" be able to create and publish blog posts via voice commands.	UC_Create_Publish_Post
1.0	Users "shall" have the ability to register and authenticate.	UC_Registration_Auth
1.0	The system "shall" offer registration processes accessible.	UC_Registration_Request
1.0	User “shall” be able to sign in to the system using voice commands.	UC_Login
1.0	Users "shall" be allowed to open/view user’s profile.	UC_Open_View_User_ Profile
1.0	The platform "shall" open an article upon request via voice input.	UC_Open_Article
1.0	The platform "shall" be able to edit a post via voice input.	UC_Edit_Post
1.0	The platform "shall" be able to delete a post via voice input.	UC_Delete_Post

2.1.6. Prioritize Requirements:

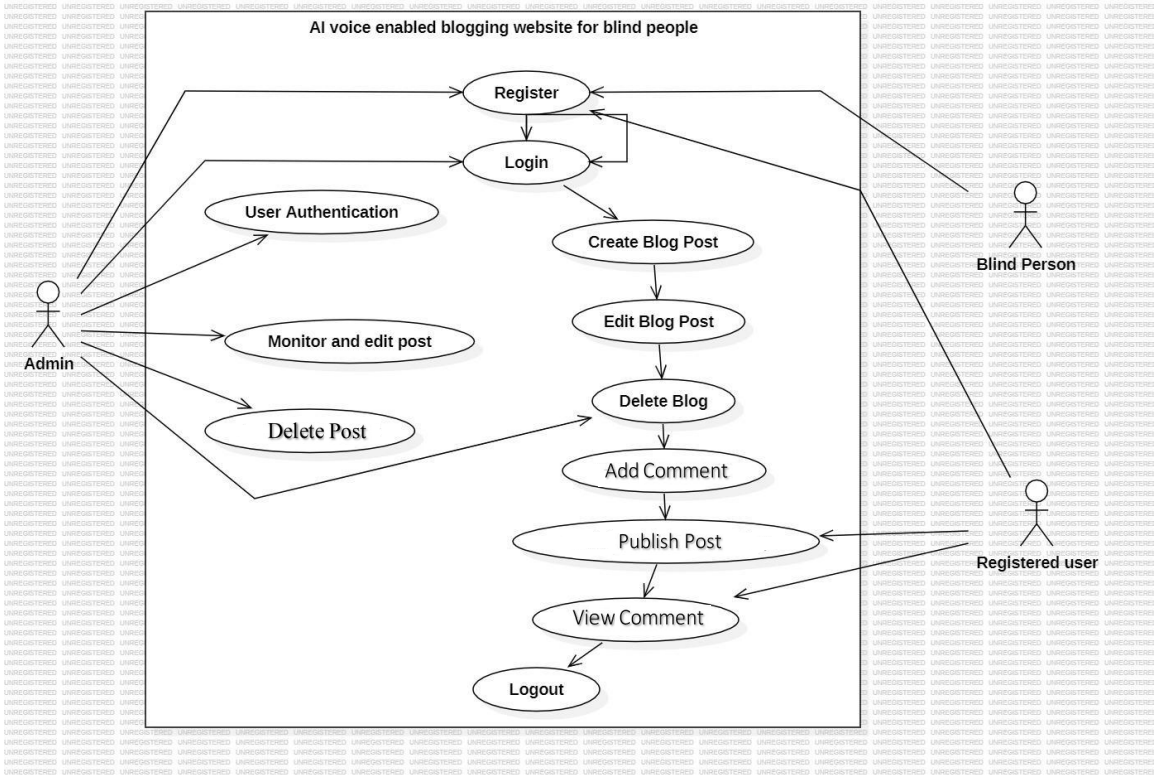
Para #	Priority	Functional Requirements	Use Case ID	Use Case Name
1.0	Highest	A user "shall" be able to create and publish blog posts via voice commands.	UC_1	UC_Create_Publish_Post
1.0	Highest	Users "shall" have the ability to register and authenticates.	UC_2	UC_Registration_Auth
1.0	Highest	The system "shall" offer registration processes accessible.	UC_3	UC_Registration_Request
1.0	Highest	User “shall” be able to sign in to the system using voice commands.	UC_4	UC_Login
1.0	Highest	Users "shall" be allowed to open/view user’s profile.	UC_5	UC_Open_View_User _Profile
1.0	Medium	The platform "shall" open an article upon request via voice input.	UC_6	UC_Open_Article
1.0	Medium	The platform "shall" be able to edit a post via voice input.	UC_7	UC_Edit_Post
1.0	Medium	The platform "shall" be able to delete a post via voice input.	UC_8	UC_Delete_Post

2.1.7. Requirements Trace-ability Matrix:

Para #	System Specification Text	Build	Use Case Name	Category
1.0	A user "shall" be able to create and publish blog posts via voice commands.	B1	UC_Create_ Publish_Post	Business
1.0	Users "shall" have the ability to register and authenticate.	B1	UC_Registrati on_Auth	Business
1.0	The system "shall" offer registration processes accessible.	B1	UC_Registrati on_Request	Business
1.0	User “shall” be able to sign in to the system using voice commands.	B1	UC_Login	Business
1.0	Users "shall" be allowed to open/view user’s profile.	B1	UC_Open_ View_User_ Profile	Business
1.0	The platform "shall" open an article upon request via voice input.	B1	UC_Open_ Article	Business
1.0	The platform "shall" be able to edit a post via voice input.	B1	UC_Edit_Post	Business
1.0	The platform "shall" be able to delete a post via voice input.	B1	UC_Delete_Po st	Business

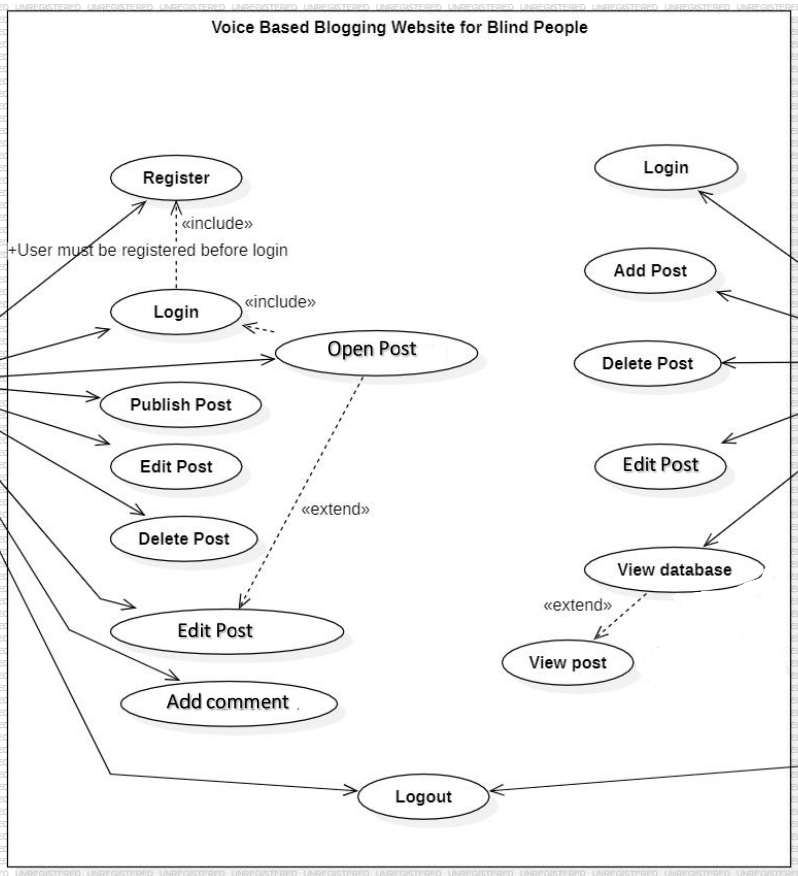
-

2.1.8. High Level Use case Diagram:



Use Case Diagram

2.1.9. Analysis Level Use case Diagram:



2.1.10. Use case Description:

While technically not part of UML, use case documents are closely related to UML use cases. A use case document is text that captures the detailed functionality of a use case. Such documents typically contain the following parts:

Create/Publish Post

Actor: User, Admin **UC ID:** UC_1 **UC Name:** UC_Create_Publish_Post	**Developer:** Amina Fayyaz **Date:** 17-March-2024
Precondition: User must create login first.	
Main Scenario: **Actor's Action** **1.** Actor use voice commands to publish post.	**System Response** **2.** System Publish the post.
Post Condition: Post is published if actor's actions are successful.	
Alternatives: Error if no post is created	

Figure 2.1: Create Post Usecase Description

Create User

Actor: User **UC ID:** UC_2 and UC_3 **UC Name:** UC_Registration_Auth UC_Registration_Request	**Developer:** Amina Fayyaz **Date: 30**-March-2024
Precondition: User must be connected through internet.	
Main Scenario: **Actor's Action** **1.** Actor clicks/Voice commands to Sign up button	**System Response** **2.** System creates his account.
Post Condition: User is created if actor's actions are successful.	
Alternatives: Error if user is already created.	

Figure 2.2: Create User Usecase Description

Login User

Actor: User, Admin **UC ID:** UC_4 **UC Name:** UC_Login	**Developer:** Amina Fayyaz **Date:** 3-April-2024
Precondition: User must be registered first.	
Main Scenario: **Actor's Action** **1.** Actor use voice commands to login.	**System Response** 2. System login the user 3. .
Post Condition: User is logged in if actor's actions are successful.	
Alternatives: Error if user is not registered	

Figure 2.3: Login User Usecase Description

Open/View User Profile

Actor: User, Admin **UC ID:** UC_5 **UC Name:** UC_Open_View_User_profile	**Developer:** Amina Fayyaz **Date:** 6-April-2024
Precondition: User must be logged in first.	
Main Scenario: **Actor's Action** **1.** Actor use voice commands to view/open his profile.	**System Response** 2. System opens the user profile
Post Condition: User profile is open if actor's actions are successful.	
Alternatives: Error if user profile is not visible.	

Figure 2.4:Open/View User Profile Usecase Description

Open Article

Actor: User, Admin **UC ID:** UC_6 **UC Name:** UC_Open_Article	**Developer:** Amina Fayyaz **Date:** 23-April-2024
Precondition: User must be logged in first.	
Main Scenario: **Actor's Action** **1.** Actor use voice commands to open article.	**System Response** 3. System opens the user profile
Post Condition: User profile is open if actor's actions are successful.	
Alternatives: Error if there is no article with the specified name.	

Figure 2.5:Open Article Usecase Description

Edit Post

Actor: User, Admin **UC ID:** UC_7 **UC Name:** UC_Edit_Post	**Developer:** Amina Fayyaz **Date:** 15-May-2024
Precondition: User must interact with voice.	
Main Scenario: **Actor's Action** **1.** Actor edits the post using voice command	**System Response** **2.** System opens the editor
Post Condition: Post is edited if actor's actions are successful.	
Alternatives: Error if no post is edited.	

Figure 2.6: Edit Post Usecase Description

Delete Post

Actor: User, Admin **UC ID:** UC_8 **UC Name:** UC_Delete_Post	**Developer:** Amina Fayyaz **Date: 1**5-June-2024
Precondition: User must create post first.	
Main Scenario: **Actor's Action** **1.** Actor clicks/voice commands on delete post.	**System Response** **2.** System deletes the post.
Post Condition: Post is deleted if actor's actions are successful.	
Alternatives: Error if no post is deleted	

Figure 2.7: Delete Post Usecase Description

Chapter 3: Design Document (For Object Oriented Approach)

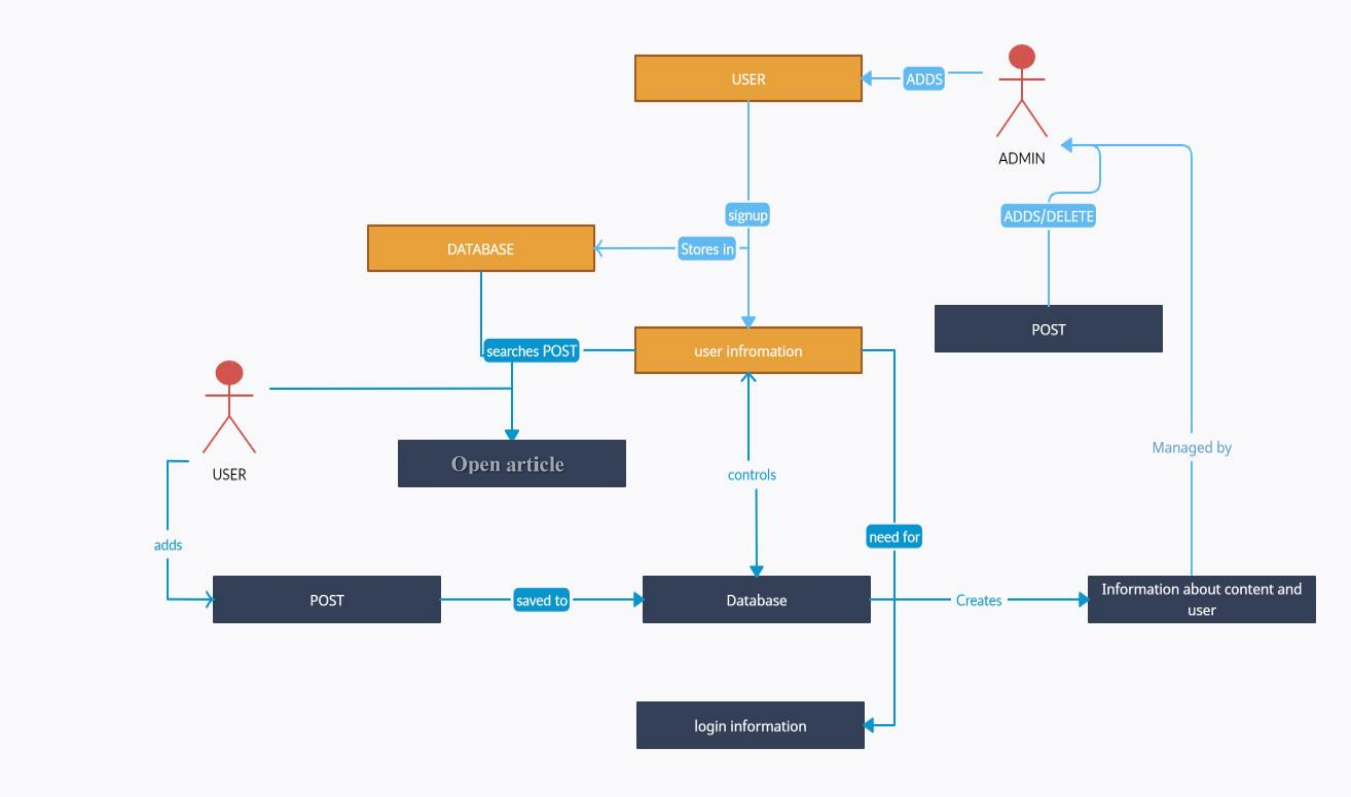
3.1. Introduction:

The Design Document marks the transition from system analysis to the actual solution development. Having comprehensively analyzed the problem domain in the previous phase, the focus now shifts to crafting a solution using object-oriented principles tailored for voice-based interaction. This phase outlines the various artifacts essential for designing a robust, intuitive, and efficient voice-based application.

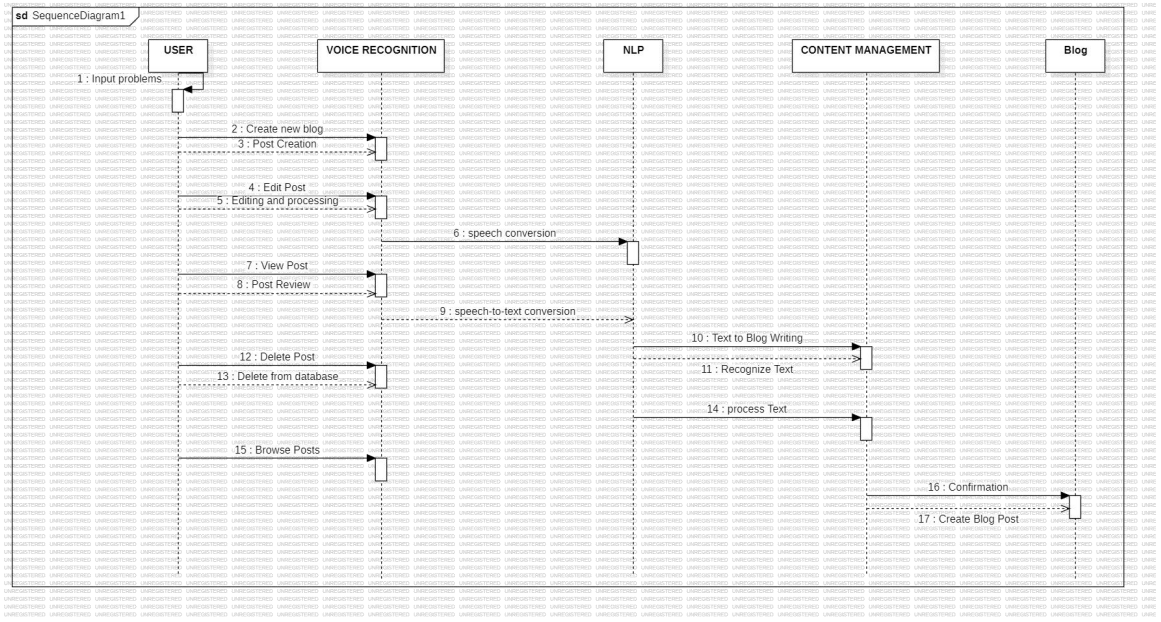
1. Domain Model
2. System Sequence Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. Operation Contracts
6. Design Class Diagram
7. State Transition Diagram

Now we discuss these artifacts one by one as follows:

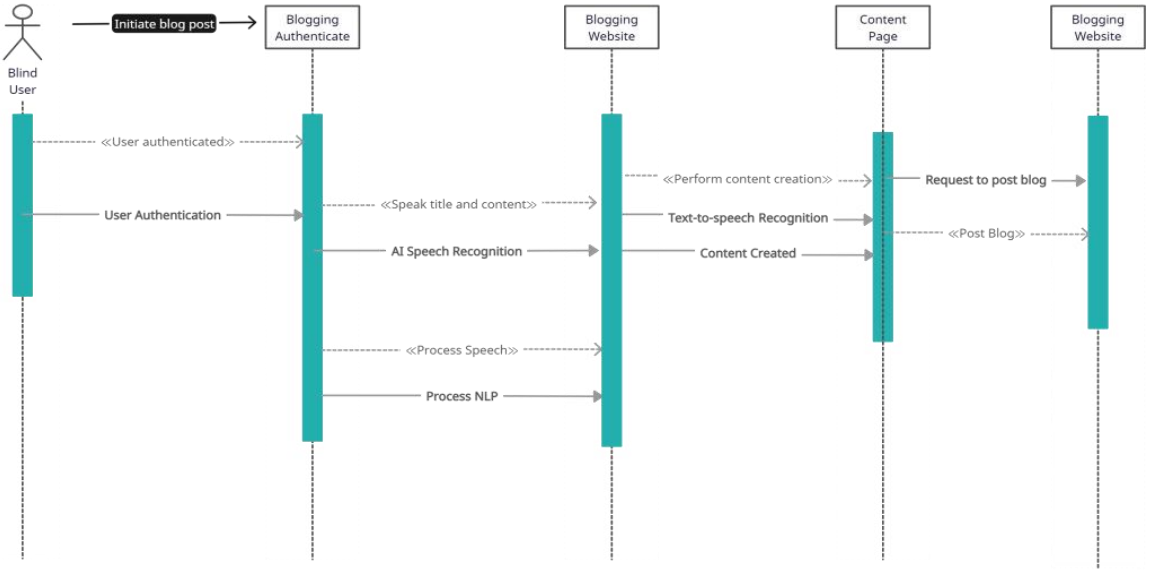
3.2. Domain Model



3.3. System Sequence Diagram

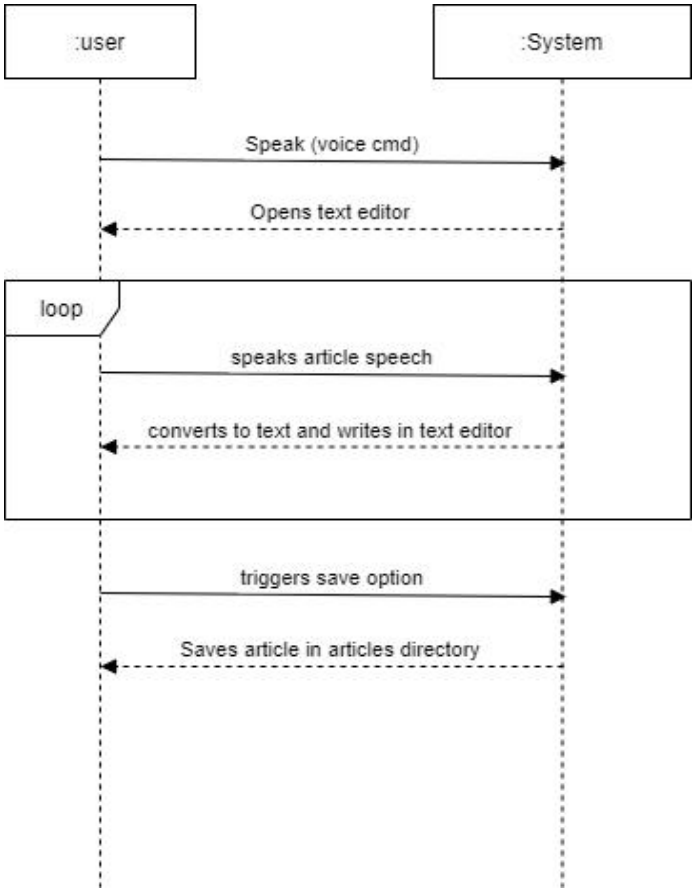


3.4. Sequence Diagram

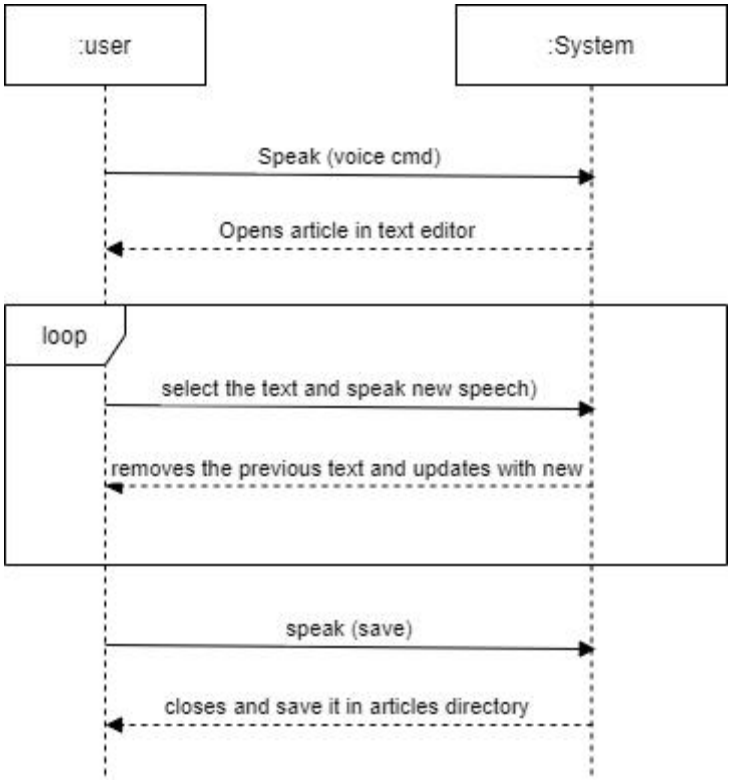


System Sequence Diagram

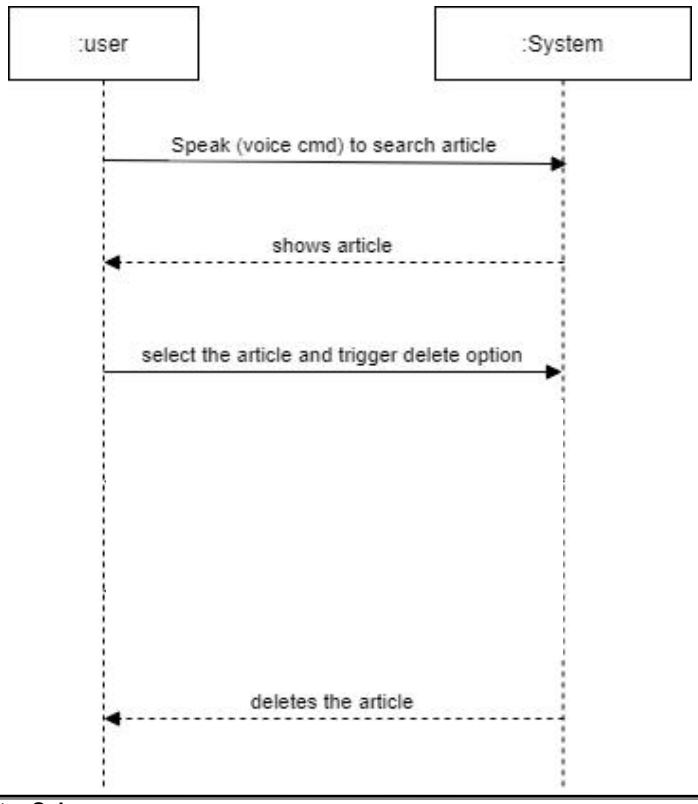
Create Article



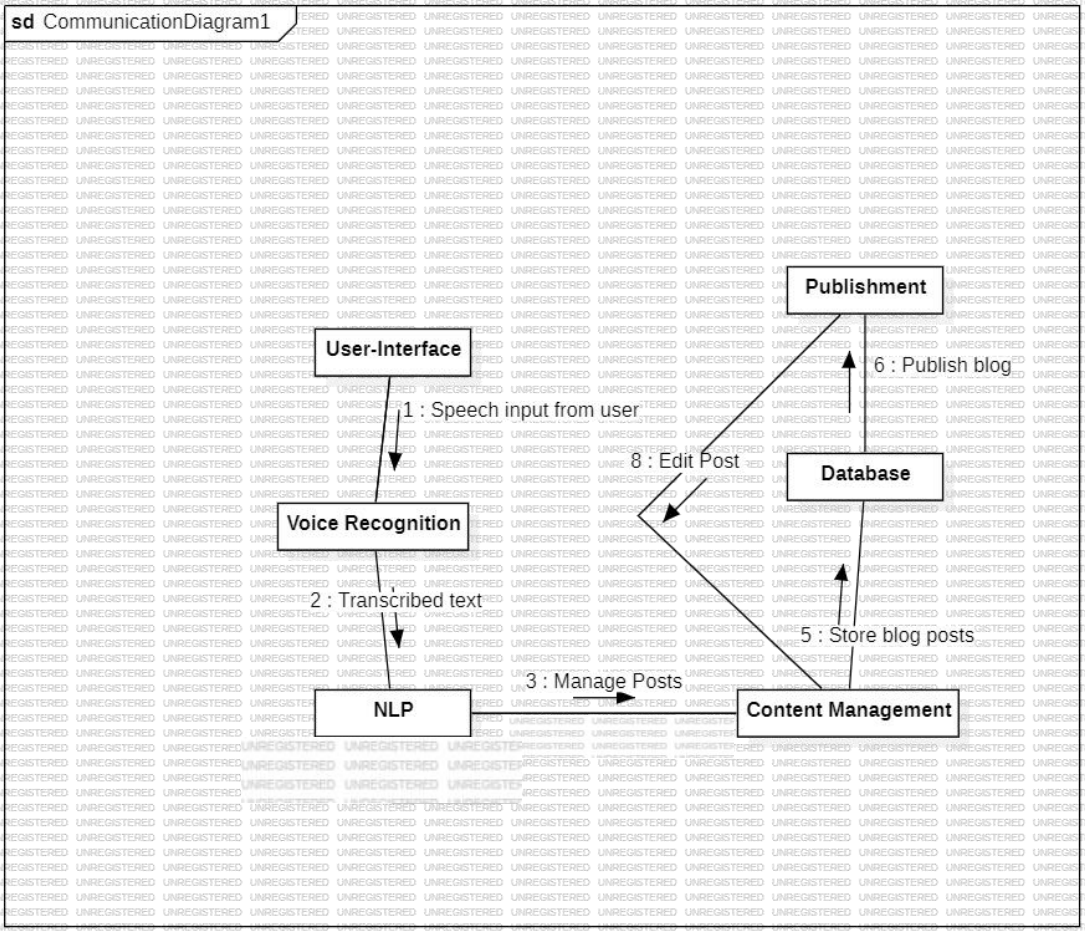
Edit Article



Delete Article



3.5. Collaboration Diagram



3.6. Operation Contracts

Contract C01- Write Article

Name	Write article
Operation	Write_Article(speech) Publish/Post(article)
Type	User, Admin
Cross-Reference	UC_1 - Create_Publish_Article
Pre-conditions	1. System is connected online on a web browser. 2. User must be logged in.
Post-conditions	1. Article is written with user's speech 2. Text styling is done with voice commands 3. Article is published when finished

Contract C02- Edit articles

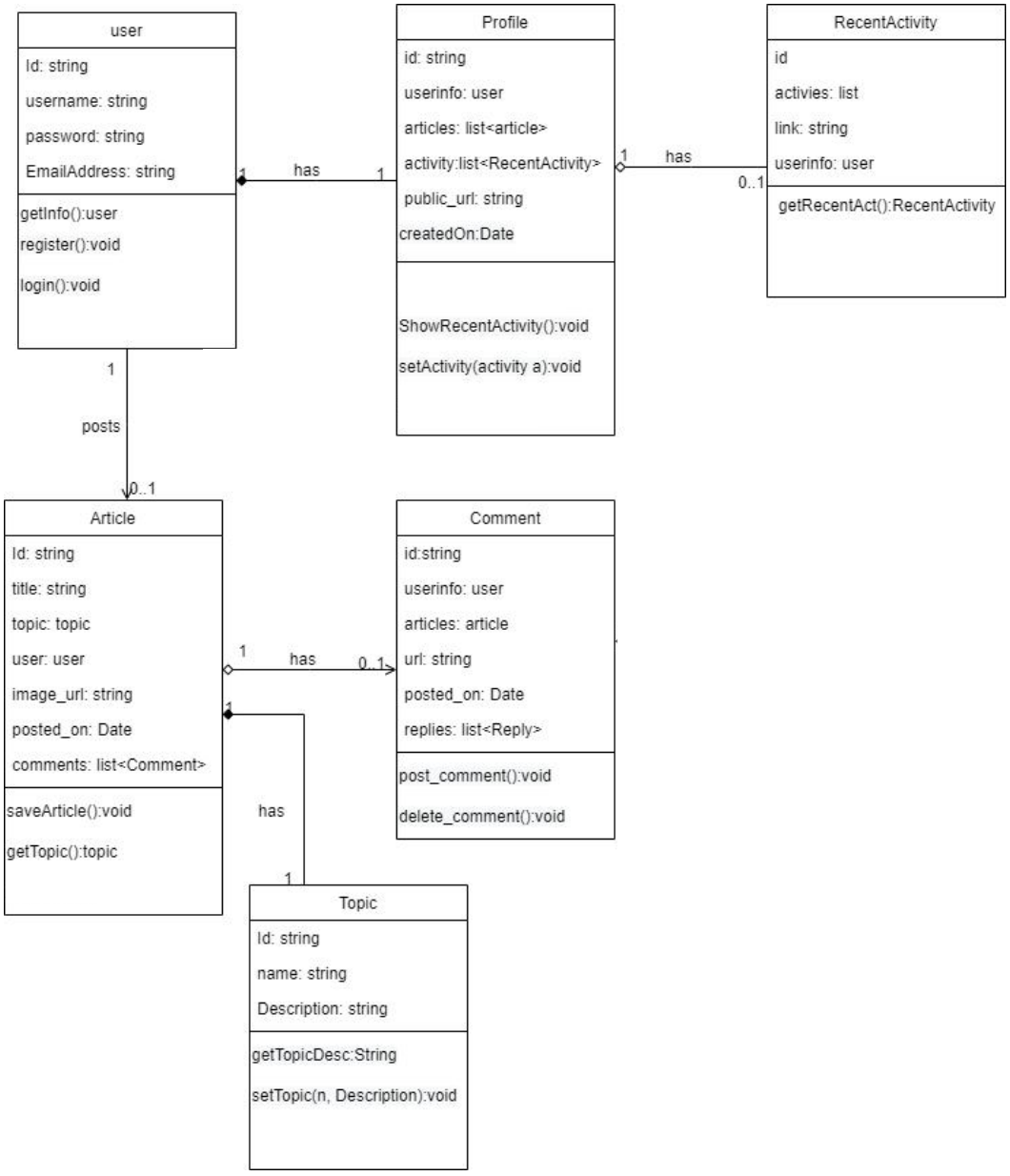
Name	Edit article
Operation	Edit(article)
Type	User, Admin
Cross-Reference	UC_7 - Edit_Article
Pre-conditions	1. System is connected online on a web browser 2. System is logged in and authenticated 3. User has opened articles directory

Post-conditions	1. Article is opened and is scrolled up/down page with voice 2. Text is updated with user's new speech and article is saved

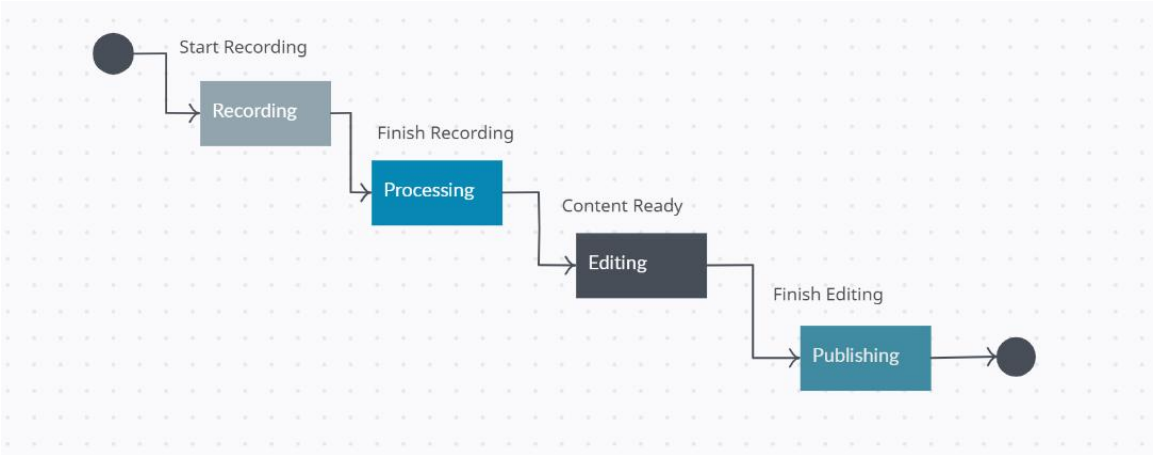
Contract C03- Delete articles

Name	Delete article
Operation	Delete(article)
Type	User, Admin
Cross-Reference	UC_8 - Delete_Article
Pre-conditions	1. System is connected online on a web browser/ 2. System is logged in and authenticated. 3. User has opened articles directory. 4. If user is ADMIN, he can delete any article. 5. If user is registered user, he can only delete his article.
Post-conditions	1. Article is deleted.

3.7. Design Class Diagram



3.8. State chart diagram



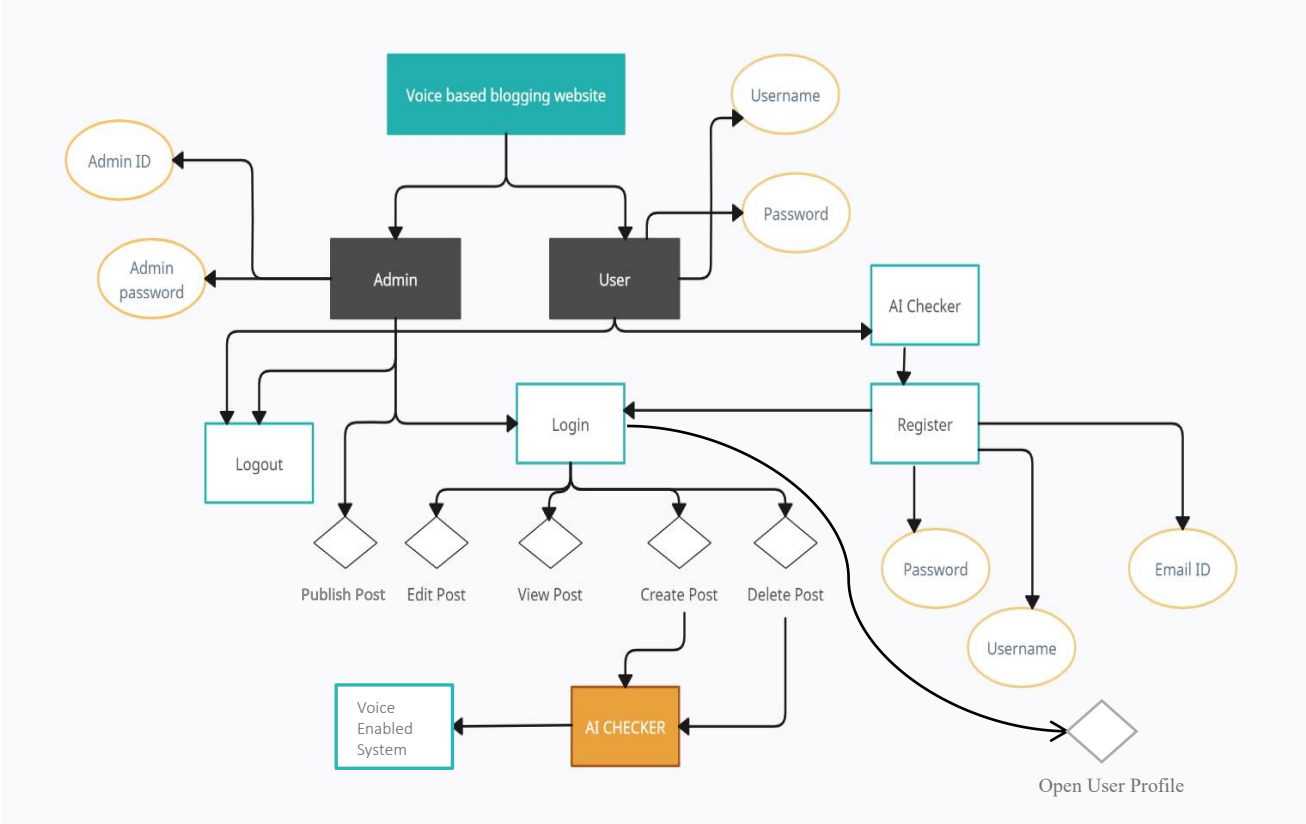
Chapter 3: Software Requirement Specification and Design Document (For structured Approach)

3.1. Introduction:

Analysis & Design Model for structured approach must contain following artifacts:

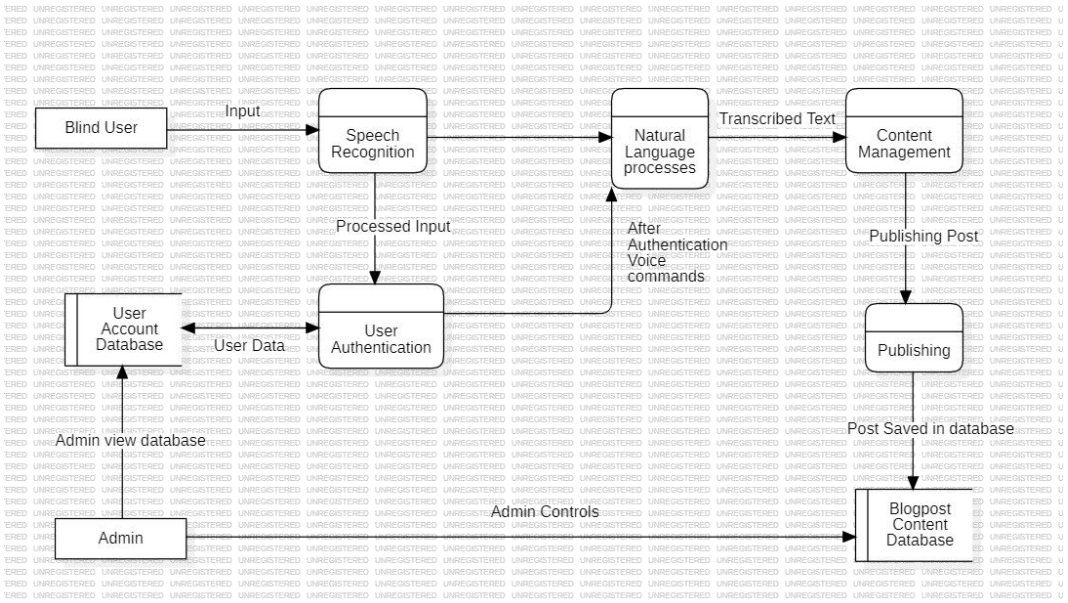
1. Entity Relationship Diagram
2. Data Flow Diagram (Functional Model)
3. State Transition Diagram (Behavioral Model)
4. Architecture Design
5. Component Level Design

3.1.1. Entity Relationship Diagram:

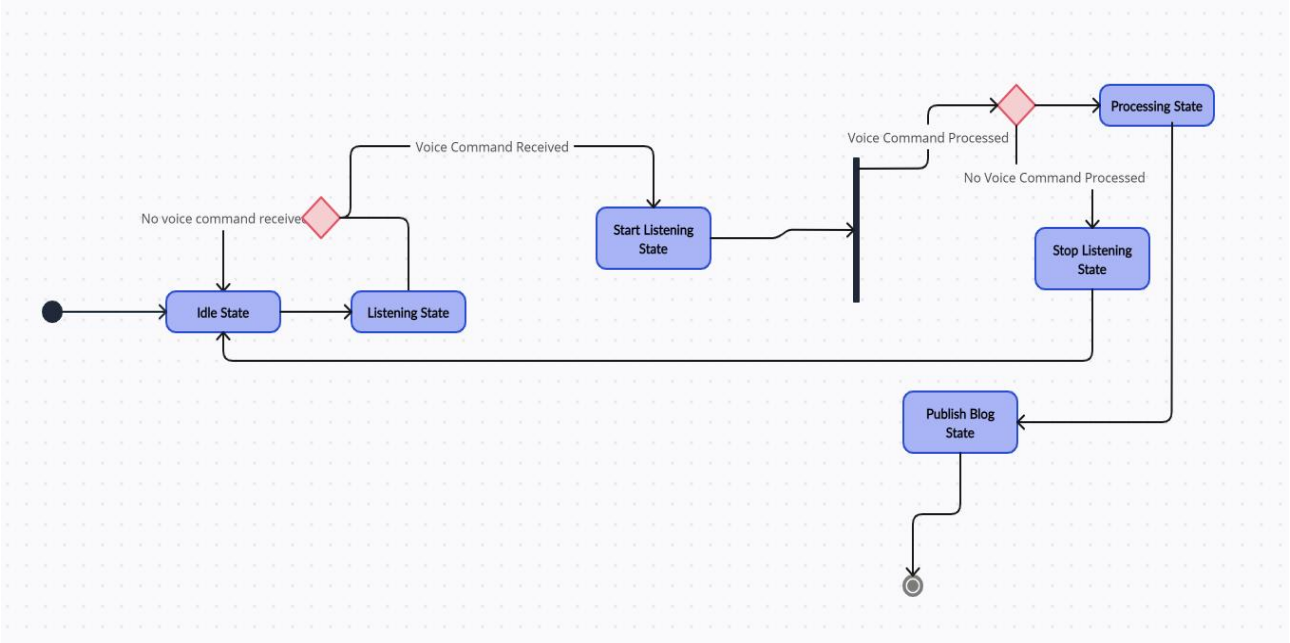


3.1.2. Data flow diagram (Functional Model)

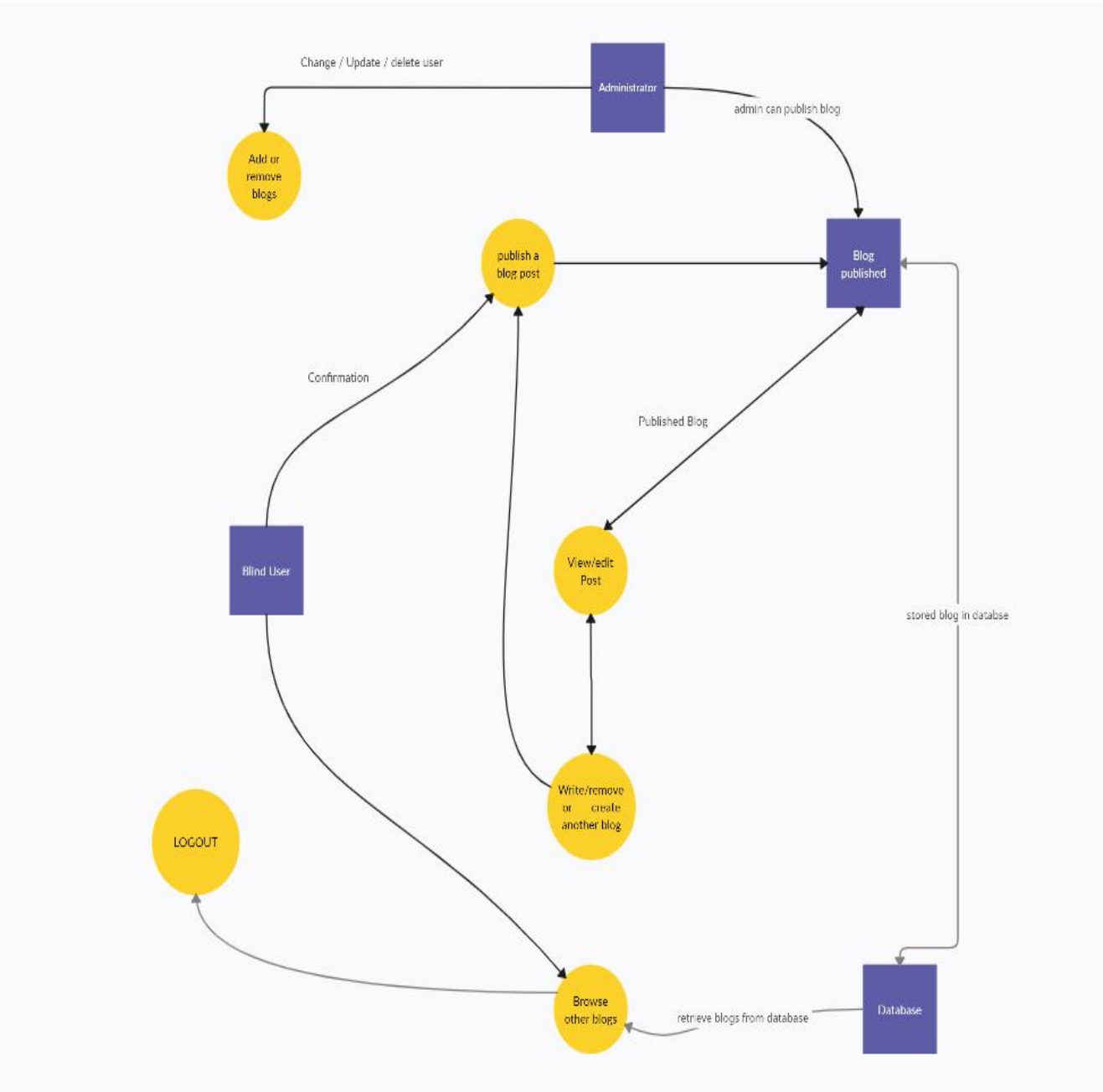
DFD is all about to identify the major processes in your system and develop Data Flow Diagram up to required level.



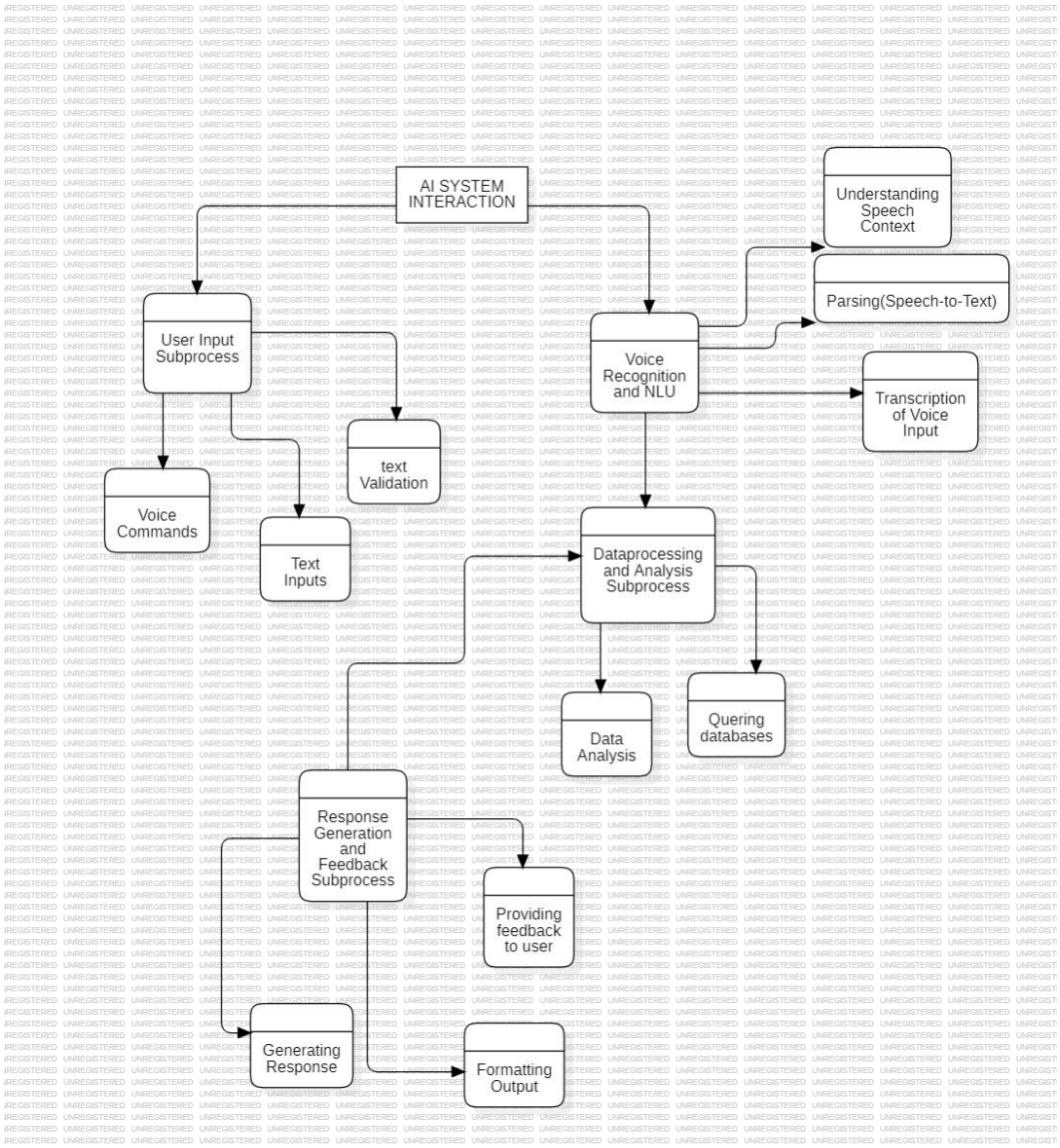
3.1.3. State Transition Diagram



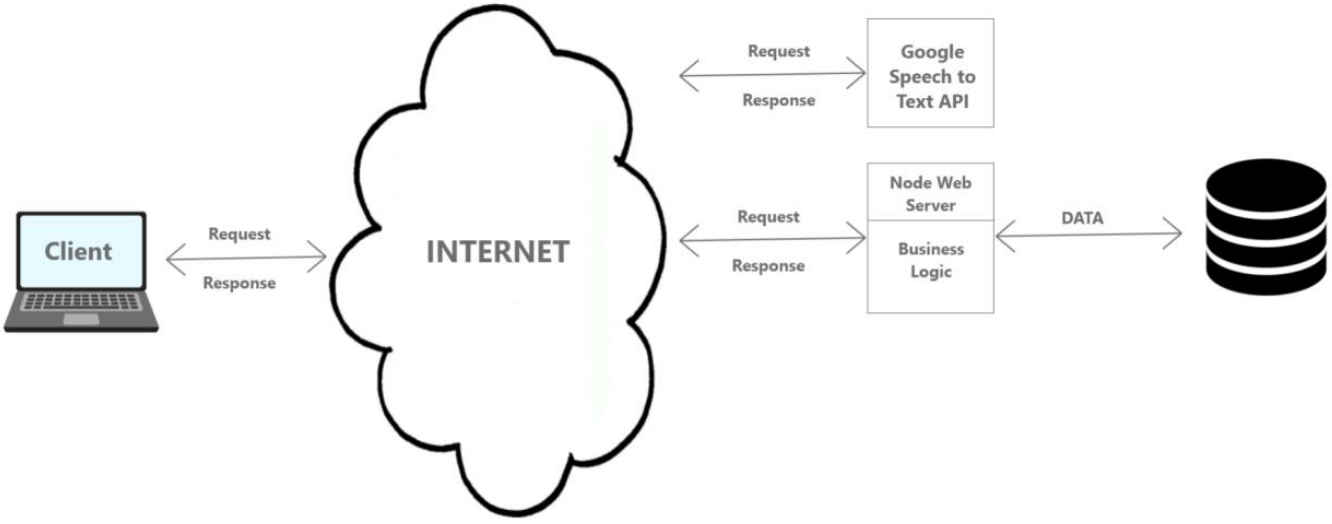
3.1.4. Architectural design



Level 1 DFD

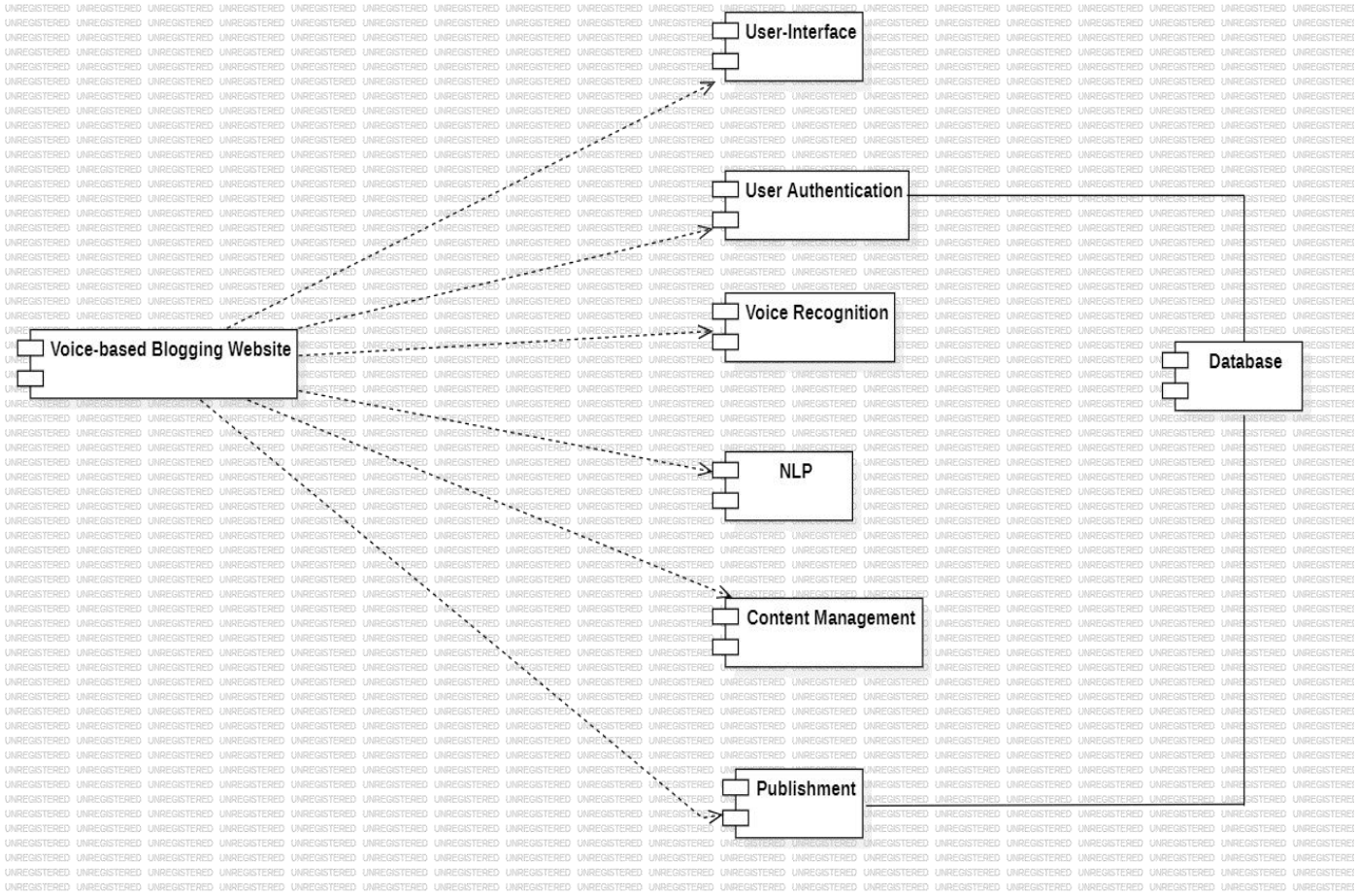

AI SYSTEM INTERACTION
User Input Subprocess
Voice Commands
Text Inputs
text Validation
Voice Recognition and NLU
Understanding Speech Context
Parsing(Speech-to-Text)
Transcription of Voice Input
Dataprocessing and Analysis Subprocess
Data Analysis
Quering databases
Response Generation and Feedback Subprocess
Providing feedback to user
Generating Response
Formatting Output

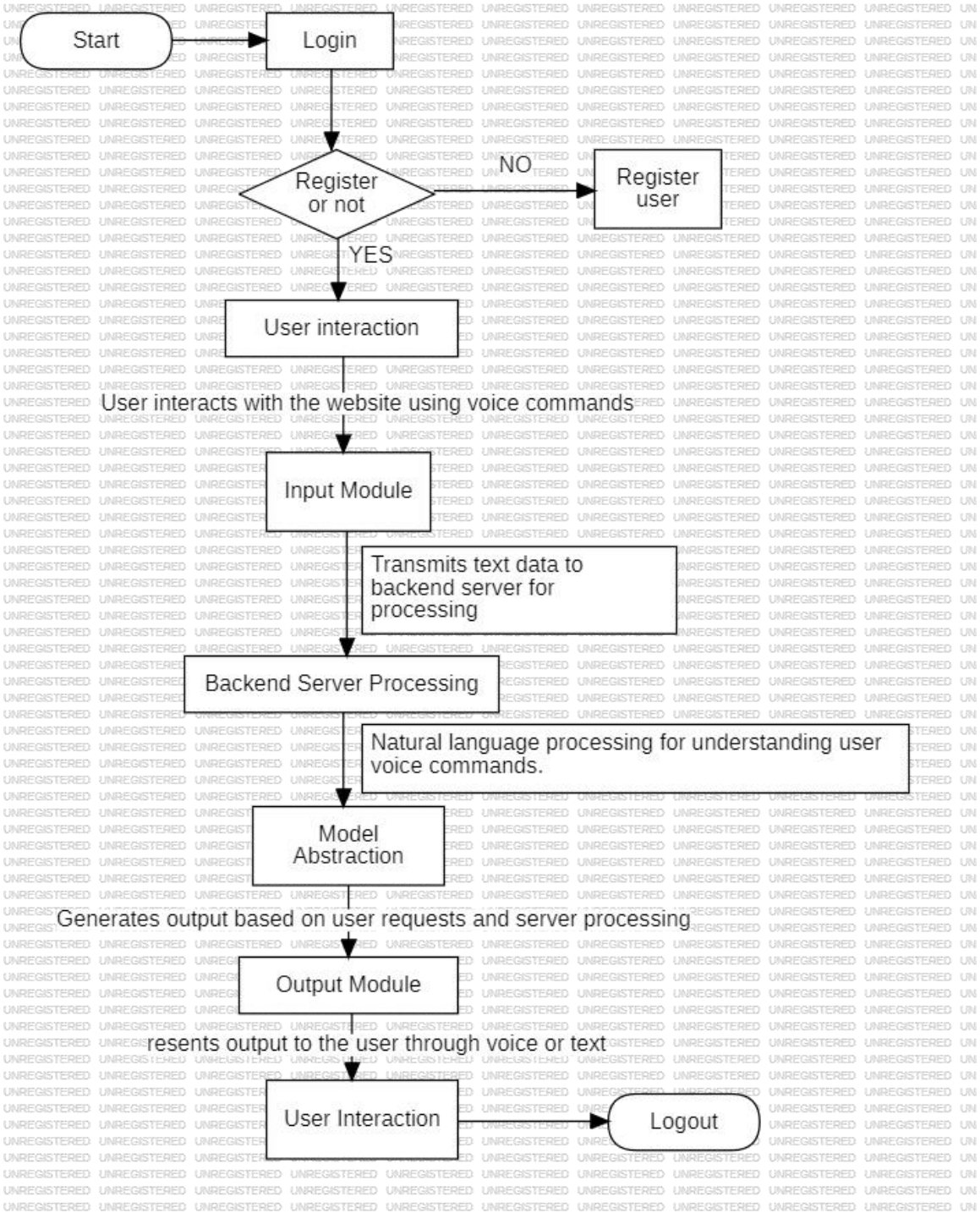
Level 2 DFD



Program Structure/Design Architecture(State Diagram)

3.1.5. Component Level Design




Start
Login
Register or not
NO
Register user
YES
User interaction
User interacts with the website using voice commands
Input Module
Transmits text data to backend server for processing
Backend Server Processing
Natural language processing for understanding user voice commands.
Model Abstraction
Generates output based on user requests and server processing
Output Module
resents output to the user through voice or text
User Interaction
Logout

Flow Chart

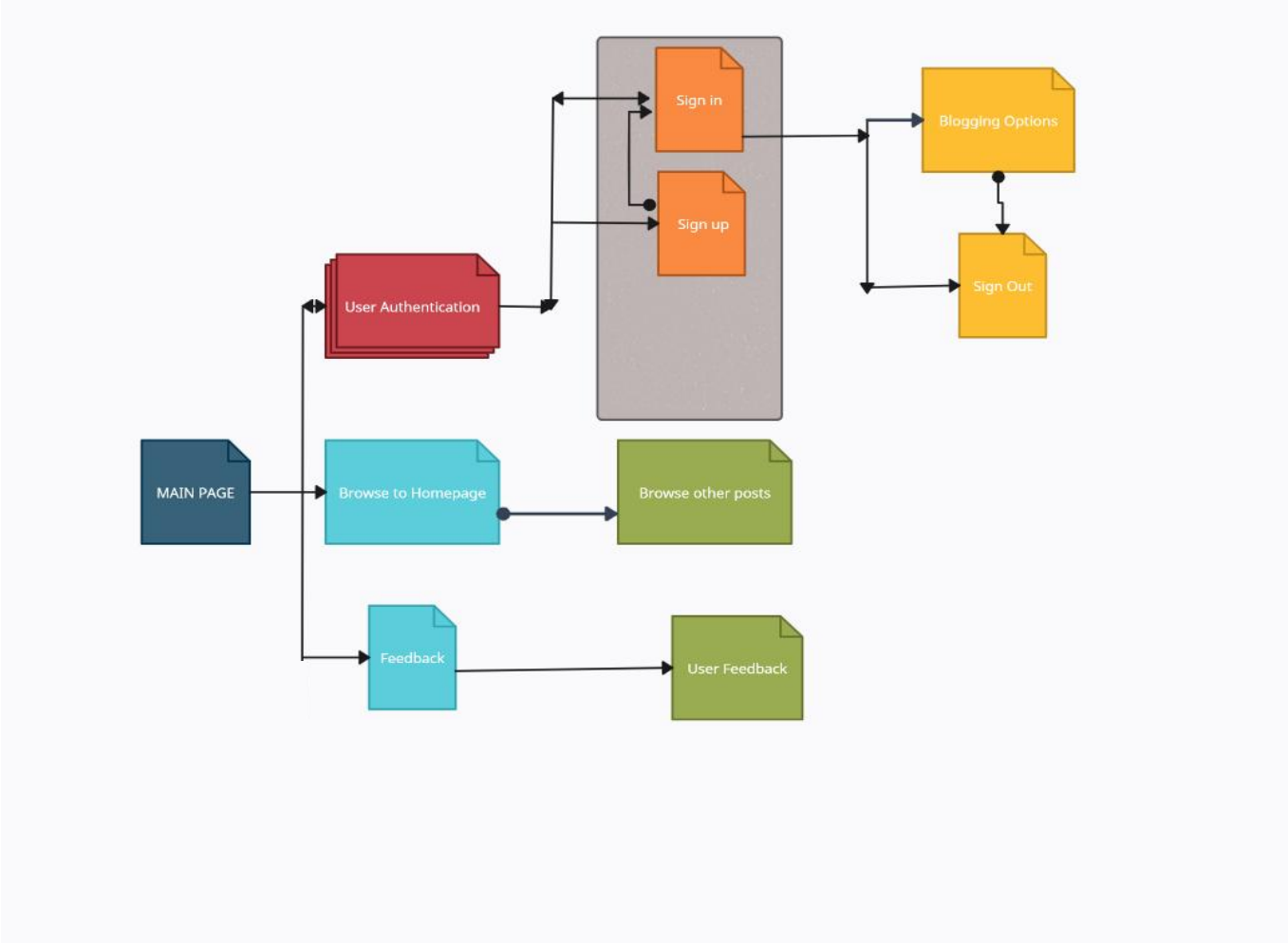
Chapter 4: User Interface Design

4.1. Introduction

Designing a user interface for voice-based interactions presents unique challenges and opportunities. Unlike traditional interfaces where elements are visualized visually, in voice-based interfaces, the emphasis is on crafting intuitive and efficient auditory experiences. Rather than sketching page elements, designers utilize techniques

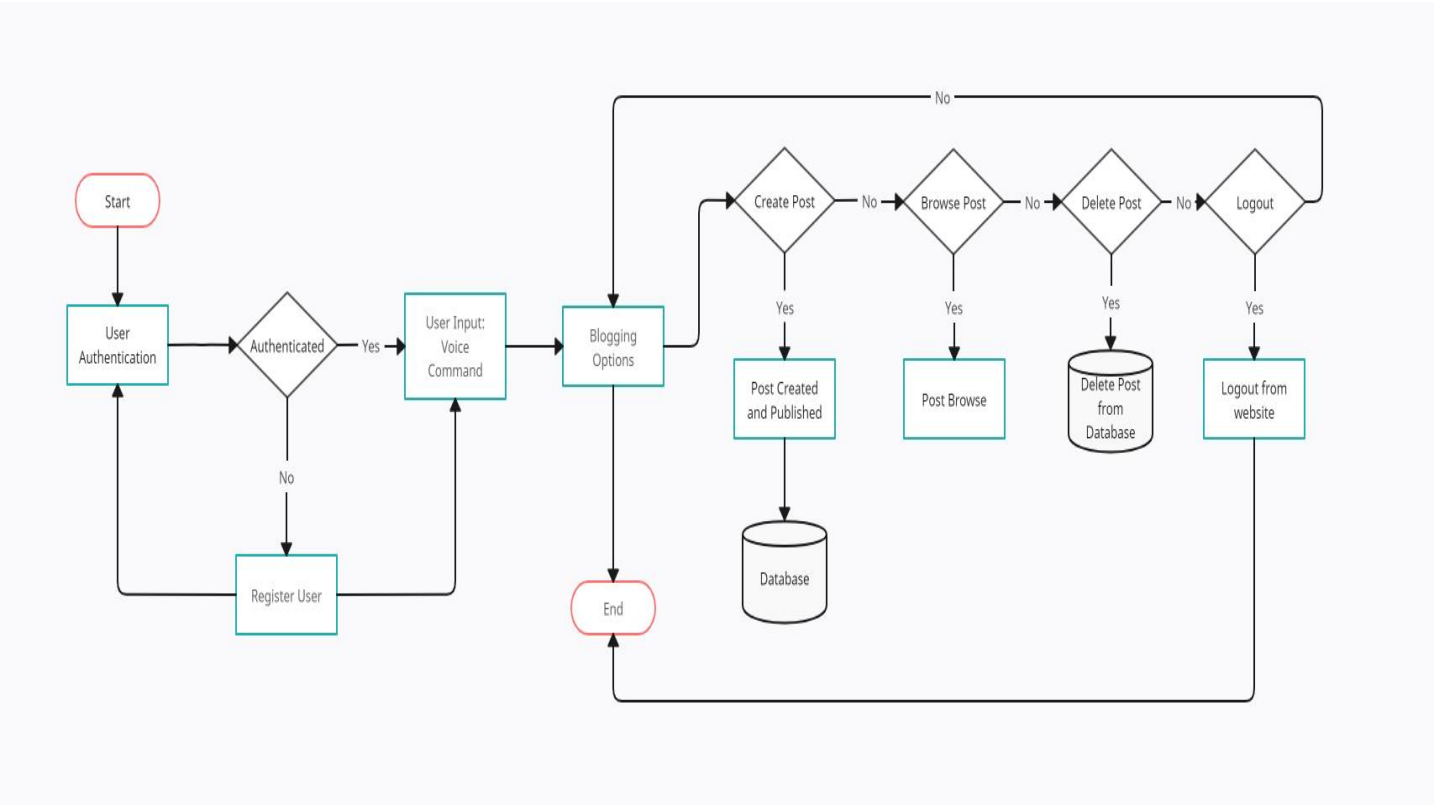
1. Site maps
2. Storyboards
3. Navigational maps
4. Traceability Matrix

4.1.1. Site Maps



4.1.2. Story boards

4.1.3. Navigational maps:



4.1.4. Trace-ability Matrix

Feature	Use Case ID	UI ID	Priority	Use Case Cross Ref	DB Table ID	Elaborated Use Case IDs	Dependent Classes
User Registration	UC_REG_001	UI_REG_001	High	UC_REG_002	DB_REG_001	UC_REG_002	User Credential
Create New Post	UC_POST_001	UI_POST_001	High	UC_POST_002	DB_POST_001	UC_POST_002	Post
Edit Existing Post	UC_POST_002	UI_POST_002	High	UC_POST_001	DB_POST_001	UC_POST_001	Post
Delete Post	UC_POST_003	UI_POST_003	Medium	UC_POST_001	DB_POST_001	UC_POST_001	Post
User Authentication	UC_AUTH_001	UI_AUTH_001	High	UC_AUTH_002	DB_AUTH_001	UC_AUTH_002	User Credential
View User Profile	UC_PROFILE_001	UI_PROFILE_001	Medium	UC_PROFILE_002	DB_PROFILE_001	UC_PROFILE_001	Profile

Chapter 5: Software Testing

5.1. Introduction:

This deliverable is based on the IEEE standard of software testing i.e., IEEE SOFTWARE TEST DOCUMENTATION Std 829-1998. This standard describes a set of basic test documents that are associated with the dynamic aspects of software testing (i.e., the execution of procedures and code). The standard defines the purpose, outline, and content of each basic document. While the documents described in the standard focus on dynamic testing, several of them may be applicable to other testing activities (e.g., the test plan and test incident report may be used for design and code reviews). This standard may be applied to commercial, scientific, or military software that runs on any digital computer. Applicability is not restricted by the size, complexity, or criticality of the software. However, the standard does not specify any class of software to which it must be applied. The standard addresses the documentation of both initial development testing and the testing of subsequent software releases. For a particular software release, it may be applied to all phases of testing from module testing through user acceptance. However, since all of the basic test documents may not be useful in each test phase, the particular documents to be used in a phase are not specified. Each organization using the standard will need to specify the classes of software to which it applies and the specific documents required for a particular test phase.

The standard does not call for specific testing methodologies, approaches, techniques, facilities, or tools, and does not specify the documentation of their use. Additional test documentation may be required (e.g., code inspection checklists and reports). The standard also does not imply or impose specific methodologies for documentation control, configuration management, or quality assurance. Additional documentation (e.g., a quality assurance plan) may be needed depending on the particular methodologies used.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

5.2. Test plan

5.2.1. Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.2.2. Outline

A test plan shall have the following structure:

a. Test plan identifier
b. Introduction
c. Test items
d. Features to be tested
e. Features not to be tested
f. Approach
g. Item pass/fail criteria
h. Suspension criteria and resumption requirements
i. Test deliverables
j. Testing tasks
k. Environmental needs
l. Responsibilities
m. Staffing and training needs
n. Schedule
o. Risks and contingencies
p. Approvals

5.2.2.1. Test plan identifier

The identifier for test plan is **VBBWBP_AFTP.01.1**

The abbreviation for the identifier is as:
VBBWBP is Voice-Based Blogging Website for Blind People
AFTP is All features to be tested in test plan
01.1 is version 1 and revision 1

5.2.2.2. Introduction

The testing phase mainly depends upon the Software test planning. All other phases depend on it either directly or in directly.
Test planning phases defines and elaborate following points:

1) What tester has to perform for testing.
2) Standards of quality to use in testing.
3) Resources to be employed for testing.
4) Schedule and time scale for the testing phase.
5) Describe all the risks and contingencies involves in testing and how to overcome them.

5.2.2.3. Test items

Following are the test items which are going to be tested:

1) Login
2) Register users
3) Post Creation
4) Edit post
5) View post
6) Delete post
7) View/Open Profile
8) Voice recognition
9) Voice testing
10) Navigation through voice commands

All these are generalized items which will in fact contain different tests to remove anomalies and defects from the Project.

5.2.2.4. Features to be tested

1) Login
2) Register users
3) Post Creation
4) Edit post
5) View post
6) Delete post
7) View/Open Profile
8) Voice recognition
9) Voice testing
10) Navigation through voice commands

5.2.2.5. Features not to be tested

Test will be performed on every feature and will make sure of it that nothing would be left.

5.2.2.6. Approach

We are going to use Following testing technique to test our project.

1) Unit testing
2) Integration testing

Now unit testing covers almost 70 percent of program test. So, unit testing will be used extensively to test each and every component of project possibly. Moreover, Integration tests are used to test the UI of the website. They cover 30 percent of total test coverage.

5.2.2.7. Item pass/fail criteria

There will be different criteria for the Unit testing and Integration testing for the passing and failure of a testing. Test driven approach in development of website will be used. It means, first write the test and the execute it. When test will fail, it will be rewritten for the code required to pass the test. After that refactor it again and test till it passes all the tests which are designed and written.

Pass/Fail Criteria for Unit Testing

1) A unit test will be considered as completed(pass) if it evaluates that the functionality which is tested is also there in the project and works properly as well.
2) A unit test will be considered as fail if the time out happens during testing and it does not provide the required functionalities.

Pass/Fail Criteria for Integration Testing

1) Integration testing will be successful if it passes the combination of units of program as a group.

Now all the tests which will not follow above criteria will be considered as fail test.

5.2.2.8. Suspension criteria and resumption requirements

The criteria for the suspension and resumption of testing shall be as follow:

1) If number or types of defects reach a point where the follow-on testing has no value. The difficulty level increases and it will cause the wastage of time working on that poorly refined test case then it will make no sense to continue testing. In this case tests will be sent for further reviews and development.
2) After completion of development and reviews and refactoring, testing shall resume.

5.2.2.9. Test deliverables

Identify the deliverable documents. The following documents should be included:

a. Test plan
b. Test design specifications
c. Test case specifications
d. Test procedure specifications
e. Test item transmittal reports
f. Test logs
g. Test incident reports
h. Test summary reports

5.2.2.10. Testing tasks

Following tasks should be completed for testing:

a. Test plan prepared.
b. Functional specifications written and delivered to the testing team.
c. Environment should be ready for testing.
d. Perform the Tests.
e. Prepare Test summary report

5.2.2.11. Environmental needs

1. **Unit Testing:** For execution of unit test, Python 3.6+, Django 3.0+ and pip dependencies must be installed in computer
2. **Integration Testing:** Django server is necessary to smoothly run integration tests.

5.2.2.12. Responsibilities

The main person which is responsible for the testing execution and management is called test manager. In our documentation and project, the testing is performed by 'Amina Fayyaz'.

Testing Module	Performed By
Testing Planning	Amina Fayyaz
Test Specification	Amina Fayyaz
Test Case Development	Amina Fayyaz
Test Writing	Amina Fayyaz
Test Execution	Amina Fayyaz

Table 5: Responsibilities

5.2.2.13 Staffing and training needs

1. To apply systematic testing methods, I should require a complete understanding and core knowledge of software testing
2. To apply test and to understand them a person requires complete knowledge of testing components provided.
3. I must have the knowledge of python and libraries working with it

5.2.2.14. Schedule

Testing is scheduled to be performed from 8 to 17 June 2024.

5.2.2.15. Risks and contingencies

Some of the potential actions to be taken as a part of determination of planning risks and contingencies are listed below.

1. The system will be rejected by the end user if they found it less interactive and difficult to use.
2. A proper documentation of requirement specification which is further known as system specification is crucial. In the absence of these documents test failure can be the resultant and all our efforts will be in vain.
3. All requirements and feature should be provided and cross checked repeatedly to assure perfection before the website is published.
4. If defects remain at the end of elected development period that do not fall into above categories, then we will conduct a review of the defect to determine that which action should be taken.
5. Delayed delivery of test items might require increased night shift scheduling to meet the delivery date.

5.2.2.16 Approvals
Test manager needs to approve the overall test strategy.

	July 21, 2024
---	---
AMINA FAYYAZ	DATE

5.3. Test design specification

5.3.1. Purpose
The purpose of the Test Design Specification for the voice-based blogging project is to define the specific scope, approach, and schedule of the testing activities. It aims to identify the key features and functionalities of the voice-based blogging platform that will be tested. Additionally, the Test Design Specification outlines the testing tasks to be performed, the resources required for testing, and the personnel responsible for executing each task. Moreover, it addresses the potential risks associated with the testing plan and outlines strategies for mitigating these risks.

5.3.2. Outline
A test plan shall have the following structure:

1. Test design specification identifier
2. Feature to be Tested
3. Approach Refinement
4. Testing identification
5. Feature Pass Fail Criteria
6. Approvals

5.3.2.1 Test design specification identifier
The identifier for the test design specification is **VBBWBP_DSI.**
DSI is Design Specification Identifier.

5.3.2.2. Features to be tested

1) Login
2) Register users
3) Post Creation
4) Edit post
5) View post
6) Delete post
7) Voice recognition
8) Voice testing
9) Navigation through voice command
10) Open/View Profile

5.3.2.3. Approach Refinement
We will be using the same strategy and approach as mentioned in the test plan phase.

5.3.2.4. Testing Identification
In this phase we make a list of testing features which can be tested and can be taken as a test case in the future steps of testing. In our testing strategies we are taking all the test items as test cases.

5.3.2.5. Features pass/fail criteria
There will be different criteria for the Unit testing and Integration testing for the passing and failure of a testing. Moreover, test driven approach in development of website will be used. It means, first write the test and the execute it. When test will fail, it will be rewritten for the code required to pass the test. After that refactor it again and test till it passes all the tests which are designed and written.

Pass/Fail Criteria for Unit Testing

3) A unit test will be considered as completed(pass) if it evaluates that the functionality which is tested is also there in the project and works properly as well.
4) A unit test will be considered as fail if the time out happens during testing and it does not provide the required functionalities.

Pass/Fail Criteria for Integration Testing

2) Integration testing will be successful if it passes the combination of units of program as a group.

Now all the tests which will not follow above criteria will be considered as fail test.

5.3.2.6. Approvals
Name of test plan manager who approves the design phase.

	July 21, 2024
----------	----------
AMINA FAYYAZ	DATE

5.4. Test Case Specification

5.4.1. Purpose
To define a test case identified by a test design specification.

5.4.2. Outline
A test case specification shall have the following structure:

a. Test case specification identifier
b. Test items
c. Input specifications
d. Output specifications
e. Environmental needs
f. Special procedural requirements
g. Inter case dependencies

5.4.2.1. Test case specification identifier

LOGIN	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC1
Purpose:	To test whether Login work or not
Pre-Requirement:	User must be registered
Test Feature:	Login
Steps:	Username, password File execution
Status:	Passed

Test Case 5.1: Login

REGISTER USER	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC2
Purpose:	To test whether user registers or not
Pre-Requirement:	User must have Id and know his/her Id
Test Feature:	User Registration
Steps:	Username, password, name, email File execution
Status:	Passed

Test Case 5.2: User registration

POST CREATION	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC3
Purpose:	To test whether POST created or not
Pre-Requirement:	User must be logged in
Test Feature:	Creation of post
Steps:	Create a post using voice commands
Status:	Passed

Test Case 5.3: Post Creation

VIEW POST	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC4
Purpose:	To test whether POST is visible after creation or not
Pre-Requirement:	Post must be created first
Test Feature:	Post Visibility
Steps:	View post through voice commands
Status:	Passed

Test Case 5.4: View Post

EDIT POST	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC5
Purpose:	To test whether POST is editable after creation or not
Pre-Requirement:	Post must be created first
Test Feature:	Post Editing
Steps:	Edit post through voice commands
Status:	Passed

Test Case 5.5: Edit Post

DELETE POST	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC6
Purpose:	To test whether POST can be deleted after creation or not
Pre-Requirement:	Post must be created first
Test Feature:	Post Deletion
Steps:	Delete post through voice commands
Status:	Passed

Test Case 5.6: Delete Post

VOICE RECOGNITION	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC7
Purpose:	Verify that the system understands user voice commands or not
Pre-Requirement:	User must be login first
Test Feature:	Voice Recognition
Steps:	Navigate through website sections with voice
Status:	Working

Test Case 5.7: Voice Recognition

VOICE TESTING	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC8
Purpose:	Verify that the system accurately interprets and executes the commands.
Pre-Requirement:	User must be login first
Test Feature:	Voice Testing
Steps:	Visit website as Admin Navigate through website sections with voice.
Status:	Passed

Test Case 5.8: Voice Testing

VIEW/OPEN PROFILE	
Test Engineer:	Amina Fayyaz
Test Case ID:	TC10
Purpose:	To test whether USER PROFILE is visible or not
Pre-Requirement:	User must be Logged in first
Test Feature:	Profile Visibility
Steps:	View/Open profile through voice commands
Status:	Passed

Test Case 5.10: View/Open Profile

Serial No	Test Features	Test Case Identifier
1	Login	TC1
2	User Register	TC2
3	Post Creation	TC3
4	View Post	TC4
5	Edit Post	TC5
6	Delete Post	TC6
7	Voice Recognition	TC7
8	Voice Testing	TC8
9	Navigate through Voice Commands	TC9
10	Open/View User profile	TC10

Table 5.3: Features and test cases

5.4.2.2 Test items

1) Login
2) Register users
3) Post Creation
4) Edit post
5) View post
6) Delete post
7) Voice recognition
8) Voice testing
9) Navigation through voice command
10) Open/View Profile

5.4.2.3. Input specifications

Serial No	Test Case Input	Test Case Identifier
1	Login	TC1
2	Post Creation	TC3
3	View Post	TC4
4	Edit Post	TC5
5	Voice Recognition	TC7
6	Voice Testing	TC8
7	Navigate through Voice Commands	TC9
8	Open/View User Profile	TC10

Table 5.3: Input Specification

5.4.2.4. Output specifications

Serial No	Test Features	Test Case Identifier
1	To test whether Login work or not	TC1
2	To test whether user registers or not	TC2
3	To test whether POST created or not	TC3
4	To test whether POST is visible after creation or not	TC4
5	To test whether POST is editable after creation or not	TC5
6	To test whether POST can be deleted after creation or not	TC6
8	Verify that the system understands user voice commands or not	TC7
9	Verify that the system accurately interprets and executes the commands	TC8
10	Verify whether website pages are navigating through voice commands or not	TC9
11	To test whether User profile is visible or not	TC10

Table 5.4: Output Specification

5.4.2.5. Environmental needs

5.4.2.5.1. Hardware

Hardware requirements are:

1. Laptop with internet access
2. Modern multi-core processor
3. A minimum of 8 GB of RAM

5.4.2.5.2. Software

Software requirements are:

1. Django Server
2. Android Virtual Device (AVD)
3. Visual Studio Code
4. Any Web Browser
5. Linux OS

5.4.2.5.3. Other

There are no other special requirements for the testing other than software and hardware requirements.

5.4.2.6. Special procedural requirements

While executing the test, it must have chromium-based browser.

5.4.2.7. Inter case dependencies

Serial No	Test Features	Test Case Identifier	Test Case Dependency
1	Login	TC1	TC2
2	User Register	TC2	-
3	Post Creation	TC3	TC1
4	View Post	TC4	TC3
5	Edit Post	TC5	TC3
6	Delete Post	TC6	TC3
8	Voice Recognition	TC7	TC1
9	Voice Testing	TC8	TC1
10	Navigate through Voice Commands	TC9	TC1
11	Open/View User profile	TC10	TC1

Table 5.5: Inter test cases dependencies

5.5. Test procedure specification

5.5.1. Purpose
To specify the steps for executing a set of test cases or, more generally, the steps used to analyze a software item in order to evaluate a set of features.

5.5.2 Outline
A test procedure specification shall have the following structure:

a. Test procedure specification identifier
b. Purpose
c. Special requirements
d. Procedure steps

5.5.2.1. Test procedure specification identifier
The identifier for the test procedure specification is **VBBWBP_TPSI**.

5.5.2.2. Purpose
To define a test procedure specified by a test design specification

5.5.2.3. Special requirement
To proceed with the testing, one should have knowledge of following techniques:

1. Unit testing
2. Integration Testing
3. Python frameworks
4. Django server

5.5.2.4. Procedure steps
This section details the steps involved in executing the test procedure:

5.5.2.4.1. Log
Logging be performed using Django built-in logging system

5.5.2.4.2. Set up
Install visual studio code

1. Install Django
2. Add testing dependencies
3. Execute the Test
4. Generate test coverage report.

5.5.2.4.3. Start
Need a plan and design of the test before executing them in visual studio code. Once all the test written, run scripts through visual studio code.

5.5.2.4.4. Proceed

Execute test cases in the order of their dependency. Test cases with no dependencies or solved dependencies first

5.5.2.4.5. Measure

The measurement whether a test is correct or not is solely based on the human observation. Logs will be displayed in interpreter which will display that whether a test is successful or not

5.5.2.4.6. Shut down

To shut down the testing, stop testing forcefully using python or run a command on visual studio code writer.

5.5.2.4.7. Restart

To restart the test simply click on the php script folder where the test resides and click the to restart test by shell scripts commands.

5.5.2.4.8. Stop

To stop the test orderly, first we will have to wait for the competition of execution.

5.5.2.4.9. Wrap up

Now to test the code again and again with different inputs and finalize the result

5.5.2.4..10. Contingencies

Contingencies can be reported or tested whenever you maintain test logs. So, in order to maintain your test logs, you will have generated them python shell prompt.

5.6. Test item transmittal report

5.6.1. Purpose

To identify the test items being transmitted for testing. It includes the person responsible for each item, its physical location, and its status. Any variations from the current item requirements and designs are noted in this report.

5.6.2. Outline

A test item transmittal report shall have the following structure:

a. Transmittal report identifier
b. Transmitted items
c. Location
d. Status
e. Approvals

5.6.2.1. Transmittal report identifier

The identifier for the transmittal report is **VBBWBP_TRI.**
TRI is Transmittal report identifier.

5.6.2.2. Transmitted items

All the items which are mentioned in the Test Plan are part of the test transmittal report which provides all those tests which are failed and again revised and developed for refactoring for further testing.
As I am working on this project alone so, I am responsible for test planning and specification development as indicated in Testing Responsibilities. Whereas these test case specifications then will be transmitted for successful running of the test.

5.6.2.3. Location

All the test document including most important test case specification will be placed in a PDF file and will be transmitted to HDD external hard drive to save these documents.

5.6.2.4. Status

At this phase all the test cases which are provided are not tested prior so their status is unresolved or we can say false. After the testing will be done then if there will be any problem in the code, the code will be again corrected and refactored.

5.6.2.5. Approvals

Names and signatures are required for approvals.

July 21, 2024

---------- ----------

AMINA FAYYAZ DATE

5.7. Test log

5.7.1. Purpose

To provide a chronological record of relevant details about the execution of tests.

5.7.2. Outline

A test log shall have the following structure:

a. Test log identifier;
b. Description;
c. Activity and event entries.

5.7.2.1. Test log identifier

The identifier for Test Log is **VBBWBP_TLI**
TLI is Test Log Identifier.

5.7.2.2. Description

Here are all the test items which are planned to be tested and also are transmitted to the test manager.

SERIAL NO	TEST FEATURES	TEST CASE IDENTIFIER
1	Login	TC1
2	User Register	TC2
3	Post Creation	TC3
4	View Post	TC4
5	Edit Post	TC5
6	Delete Post	TC6
7	Voice Recognition	TC7
8	Voice Testing	TC8
9	Navigate through Voice Commands	TC9
10	Open/View User Profile	TC10

Table 5.6: Test features

5.7.2.3. Activity and event entries

Sr No.	Test Features	Test Case Identifier	Date	Name
1	To test whether Login work or not	**TC1**	8th June	Amina Fayyaz
2	To test whether user registers or not	**TC2**	8th June	Amina Fayyaz
3	To test whether POST created or not	**TC3**	9th June	Amina Fayyaz
4	To test whether POST is visible after creation or not	**TC4**	10th June	Amina Fayyaz
5	To test whether POST is editable after creation or not	**TC5**	12th June	Amina Fayyaz
6	To test whether POST can be deleted after creation or not	**TC6**	12th June	Amina Fayyaz
7	Verify that the system understands user voice commands or not	**TC7**	12th June	Amina Fayyaz
8	Verify that the system accurately interprets and executes the commands.	**TC8**	12th June	Amina Fayyaz
9	Verify whether website pages are navigating through voice commands or not	**TC9**	12th June	Amina Fayyaz
10	To test whethe user profile is visible or not.	**TC10**	10th July	Amina Fayyaz

Table 5.7: Activity and Event Entities

5.7.2.3.1. Execution description

1. Linux
2. Ram 2GB (minimum) and 4GB (recommended)
3. Django server
4. Visual code studio

5.7.2.3.2. Procedure results

Sr No.	Test Features	Test Case Identifier	Status
1	To test whether Login work or not	**TC1**	**Pass**
2	To test whether user registers or not	**TC2**	**Pass**
3	To test whether POST created or not	**TC3**	**Pass**
4	To test whether POST is visible after creation or not	**TC4**	**Pass**
5	To test whether POST is editable after creation or not	**TC5**	**Pass**
6	To test whether POST can be deleted after creation or not	**TC6**	**Pass**
7	Verify that the system understands user voice commands or not	**TC7**	**Pass**
8	Verify that the system accurately interprets and executes the commands.	**TC8**	**Pass**
9	Verify whether website pages are navigating through voice commands or not	**TC9**	**Pass**
10	To test whethe user profile is visible or not.	**TC10**	**Pass**

5.7.2.3.3. Environmental information

1. Linux
2. Ram 2GB (minimum) and 4GB (recommended)
3. Django server
4. Visual code

5.7.2.3.4. Anomalous events

Testing each and every test case which will be later investigated. But once developer start writing more code, problems can occur at any time. So, it is necessary to use test driven approach.

5.7.2.3.5. Incident report identifiers

Record the identifier of each test incident report, whenever one is generated.

5.8. Test incident report

5.8.1. Purpose

To document any event that occurs during the testing process that requires investigation.

5.8.2. Outline

A test incident report shall have the following structure:

a. Test incident report identifier
b. Summary
c. Incident description
d. Impact

5.8.2.1. Test incident report identifier

The identifier is **VBBWBP_TIRI.**
TIRI is Test Incident Report Identifier.

5.8.2.2. Summary

In this section we will analyze and investigate the reason behind failure of test case if any.

5.8.2.3. Incident description

No incident report was observed during test execution.

5.8.2.4.Impact

All tests passed successfully.

5.9. Test summary report

5.9.1. Purpose

To summarize the results of the designated testing activities and to provide evaluations based on these results.

5.9.2. Outline

A test summary report shall have the following structure:

a. Test summary report identifier
b. Summary
c. Variances
d. Comprehensive assessment
e. Summary of results
f. Evaluation
g. Summary of activities
h. Approvals

5.9.2.1. Test summary report identifier

Identifier for test summary report will be **VBBWBP_TSRI**
TSRI is test summary report identifier.

5.9.2.2. Summary

All the test are successfully completed and passed.

5.9.2.3. Variances

No test case failed.

5.9.2.4. Comprehensiveness assessment

All test cases are tested and no test case is left behind.

5.9.2.5. Summary of results

All tests were successfully tested and no incident was happened. All features have been tested and all test have been passed the criteria.

5.9.2.6. Evaluation

All the test are successfully completed and passed.

5.9.2.7. Summary of activities

There are three development phases in testing. The first one was test planning followed by test specification and designing and test execution. All these test activities and phases are planned and executed according to Gantt chart. Here is the breakdown of all the activities with their time and staff who was responsible.

Testing Activities	Name	Time Elapsed
Test Planning	Amina Fayyaz	5 days
Test Designing	Amina Fayyaz	4 days
Test Execution	Amina Fayyaz	7 days

Table 5.8: Activities

5.9.2.8. Approvals
Specify the names and titles of all persons who must approve this report.

July 21, 2024

---------- ----------

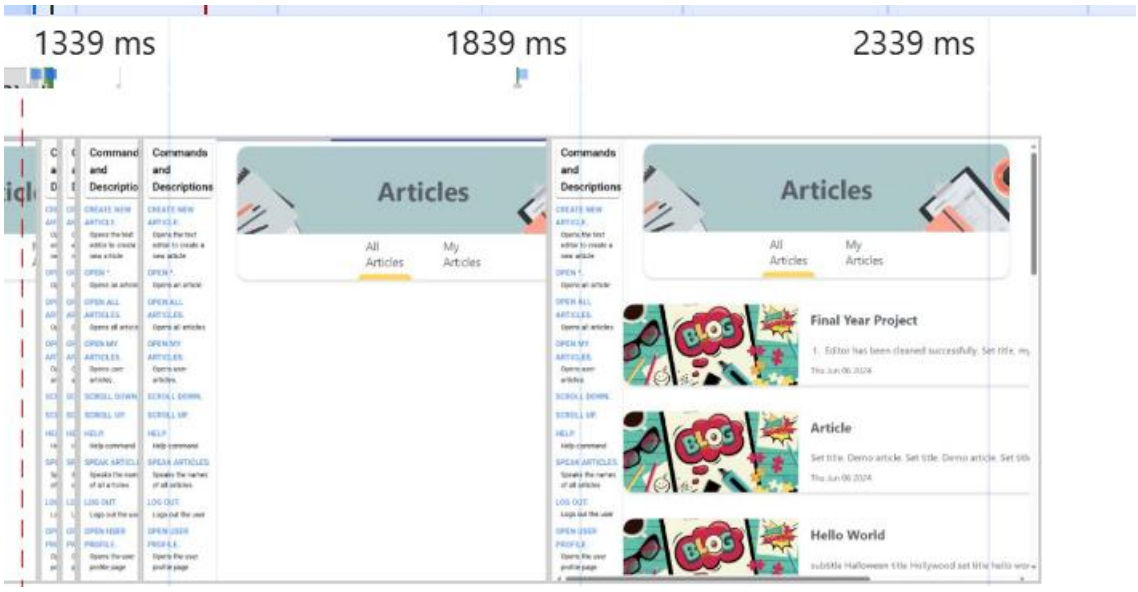
MR. M.ABRAR DATE

Chapter 6: Results

The "**Voice based Blogging Website For Blind People**" project aimed to develop a comprehensive web-based application designed to facilitate blogging for visually impaired individuals. This chapter details the outcomes and findings of the project, evaluating the effectiveness and performance of the app in meeting its specified goals and objectives. The results are based on various tests, manual testing, and performance metrics.

1. User Registration and User Logged In Results:

The system efficiently handles user data and ensures secure access. User is authenticated using voice commands. The system handles the voice commands efficiently. From sign in page to Home page 500-1000ms time is estimated:



2. Article Creation:

Article is created after the user is logged in to the system. User navigates to the articles directory where he can create an article using his voice. His voice is converted into text to write the article and various styling can be applied using his/her voice. The system converts the speech into text.

3. View User profile:

User can navigate to his profile using voice command "Open user profile". The system handles the command and navigates the user to his profile where he can view his published articles and his favorite articles which he has liked. User profile is updated according to the user activity.

4. Open Article:
User can open any article, read any article using the voice command "**Speak**" and the system will change text-to-speech and reads the article for the user. User can like or unlike any article using his voice. User can also add comment on any article using his voice.

5. Edit Article:
User can edit his article using voice command **"Edit Article".** The system will authenticate him if the opened article is written by the user or not then the command **"Edit Article"** will be visible in the sidebar. System will edit the article by converting speech to text.

6. Add Comment:
User can add comment to any article. The system will handle the add command functionality. User will add comment using his voice. When the user says 'Submit Comment', comment will be added.

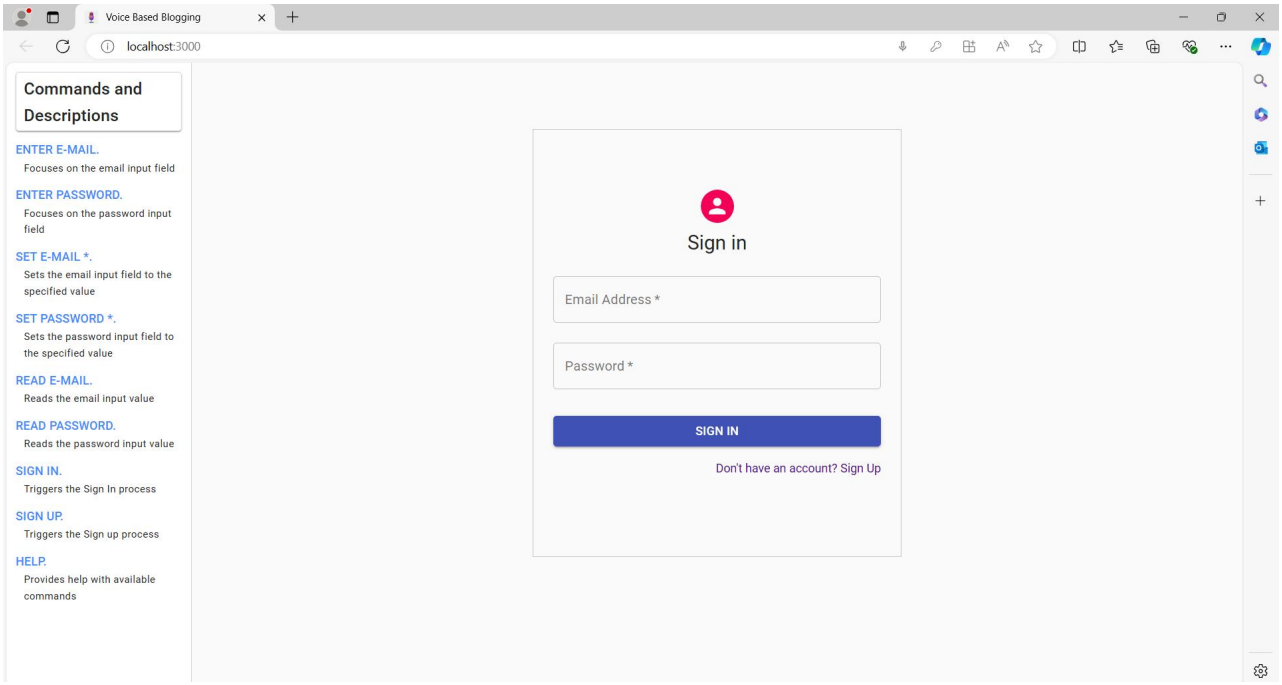
7. Logout:
User can logout from his account using his voice. The system will handle the logout functionality and his activity will be saved. The system will speak to the user **"You have been logged out successfully"** in order to give him the feedback that he has logged out of the system.

8. Delete Article:
User can only delete his own article using voice. The system will authenticate the author of the article then delete article command will be visible in the sidebar. **Admin** can delete any article despite of the author authentication or permission.

9. ToolBar:
User can enjoy all the styling by just using his/her voice. He/She can apply **'bold', 'italics', strike through', heading levels *, 'block quote', 'code block'** and can set title of his/her article.

Chapter 7: User Manual

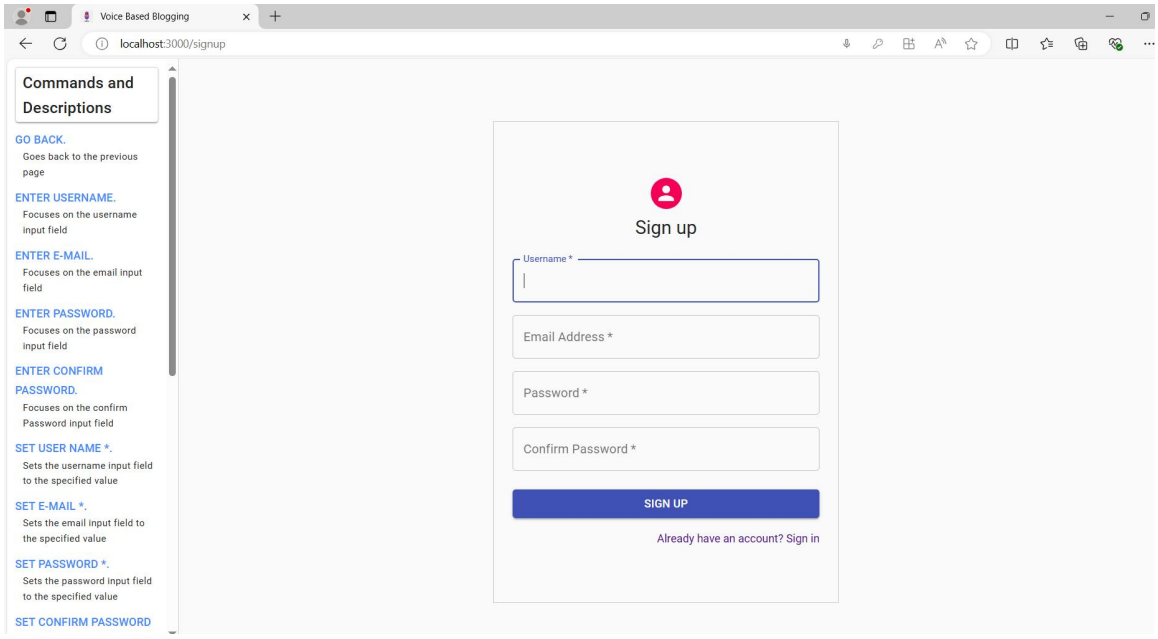
7.1. Login



Description:

1. When sign in page is shown system will trigger tts(**text-to-speech**) to tell the blind user that "**Welcome to the sign in page. What would you like to do say help and I will guide you through the website**."
2. User can enter in email text field to enter email using his voice command **"Enter E-mail "**.
3. User can enter in password text field to enter password using his voice command **"Enter Password "**.
4. User can set email using his voice command **"Set E-mail *"**.
5. User can set password using his voice command **"Set Password *"**.
6. User can also sign in to the system using his voice command **"Sign In."**
7. **If user is not registered he can say "Sign Up." or click on the sign up text and he will be navigated to the sign up page by the system.**

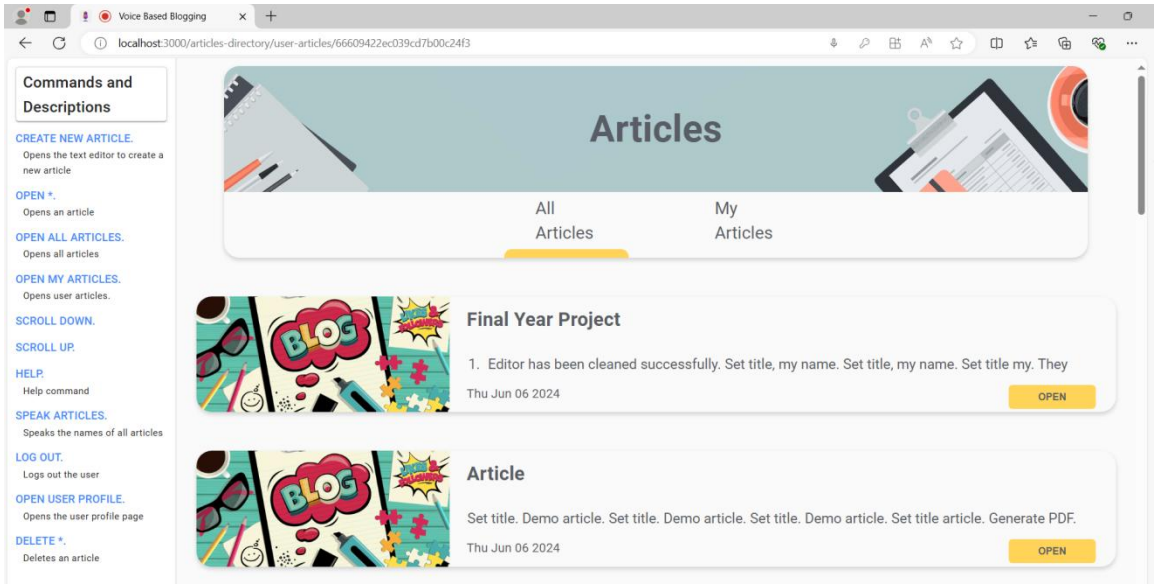
7.1.1. Sign Up



Description:

1. When sign up page is shown system will trigger tts(**text-to-speech**) to tell the blind user that "**Welcome to the sign up page. What would you like to do say help and I will guide you through the website**."
2. User can enter in username text field to enter username using his voice command **"Enter User Name "**.
3. User can enter in email text field to enter email using his voice command **"Enter E-mail "**
4. User can enter in password text field to enter password using his voice command **"Enter Password "**.
5. User can enter in password text field to enter password using his voice command **"Enter Confirm Password "**
6. User can set username using his voice command **"Set Username*"**
7. User can set email using his voice command **"Set E-mail *"**.
8. User can set password using his voice command **"Set Password *"**.
9. User can set confirm password using his voice command **"Set Confirm Password *"**
10. User can also sign in to the system using his voice command **"Sign In."**
11. If user is not registered he can say "Sign Up." or click on the sign up text and he will be registered to the system and system will navigate him to the sign in page.

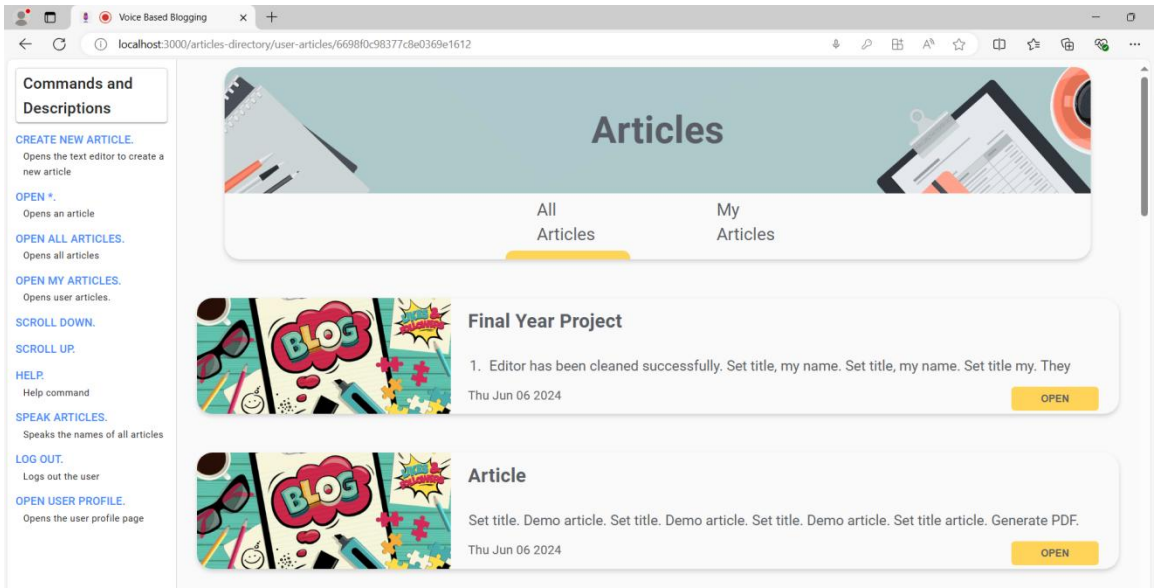
7.2. Main Interface



Description:

1. After Successfully login into application, application shows main interface page.
2. In main interface, user has given controls according to their roles e.g. (Blind User, Administrator).
3. User can create article, open any article, open his/her own article, open user profile, **Delete* is for Admin.**

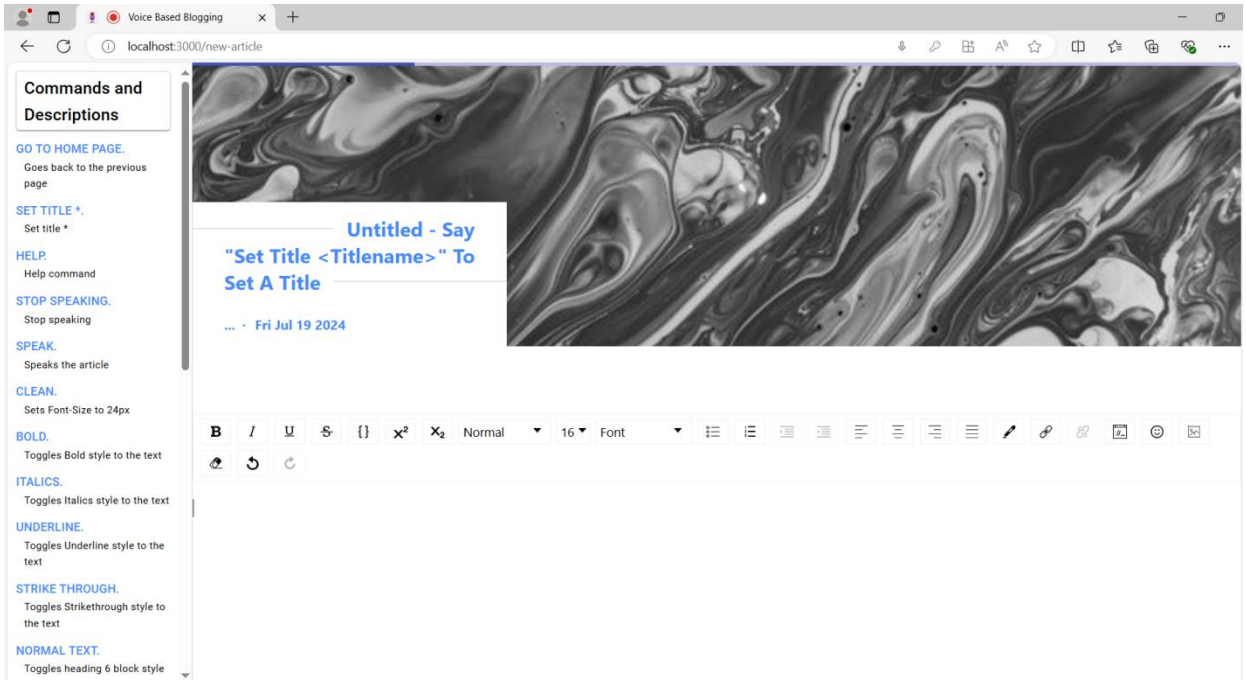
7.2.1. Articles Directory For Blind User



Description:

1. On first encounter of the main interface, system will tell welcome using ‘speak-tts’.
2. User can say ‘help’ and system will speak all the commands to the user.
3. User can perform the tasks using his/her voice and the system will navigate or respond to the user according to his voice command.
4. User can create new article, open his profile, logs out of the system open an article, speak articles, help and open all and my articles.

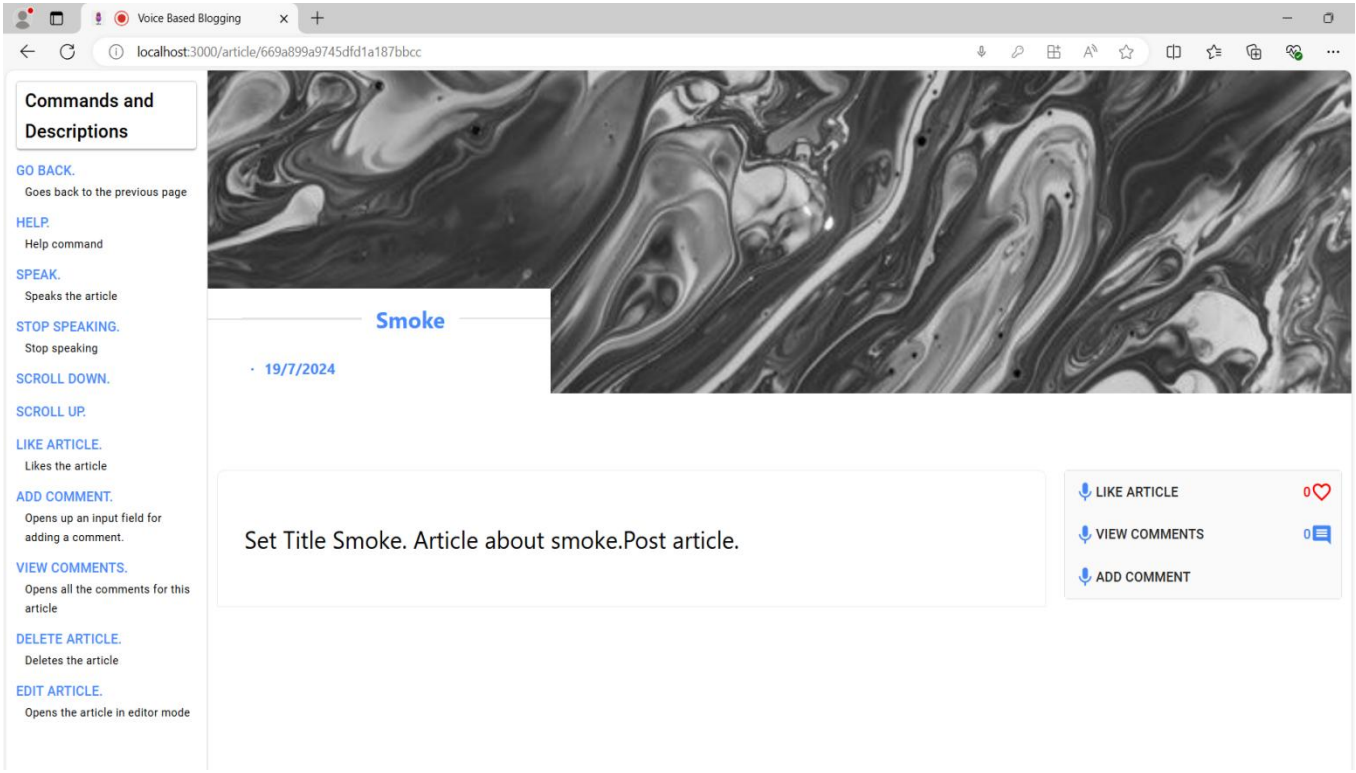
7.3 Text Editor



Description:

1. On first encounter of the text editor interface, system will tell welcome using ‘speak tts’.
2. User can perform the tasks using his/her voice and the system will navigate or respond to the user according to his voice.
3. User can add styling to the article like bold, italics, underline, set font, insert new line or clear the whole tool bar.
4. User can set title of his/her article using voice command “**Set Title ***”.
5. User can go back to home page.
6. User can post/publish article.
7. System can read the article when user say the command “**Speak**”

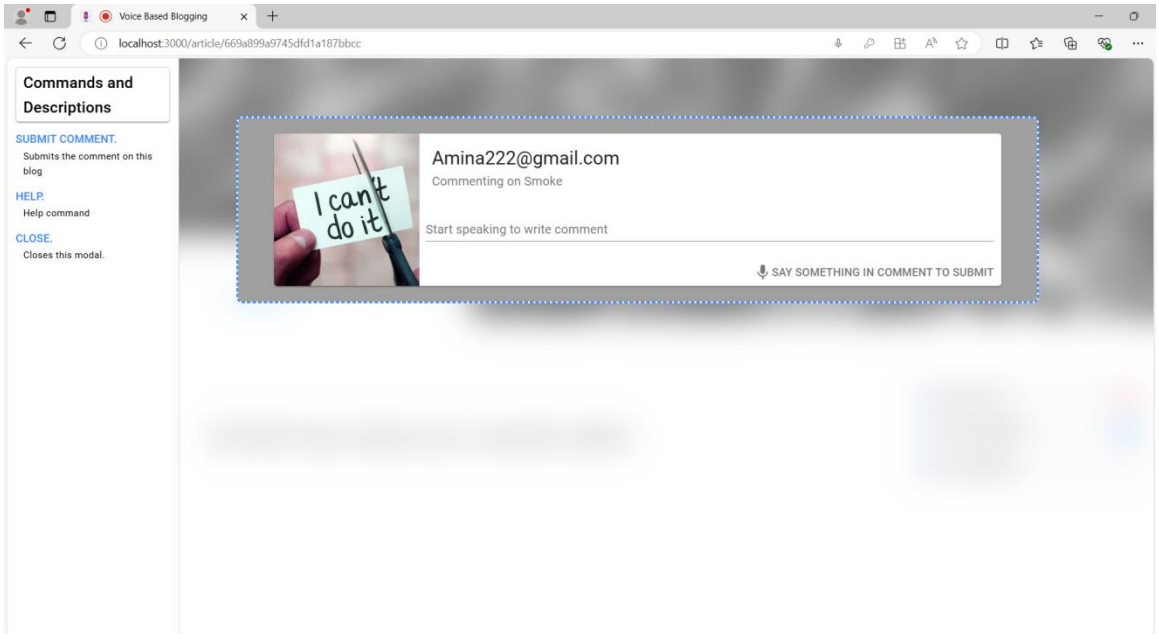
7.4 Published Article



Description:

1. On first encounter of the published article interface, system will tell about the page using ‘speak tts’.
2. User can perform the tasks using his/her voice and the system will navigate or respond to the user according to his voice.
3. User can edit the article.
4. User can go back.
5. User can delete article.
6. User can add feedback as like article, add comment and view comment.
7. User can say the command ‘speak’ and the system will read the published article.

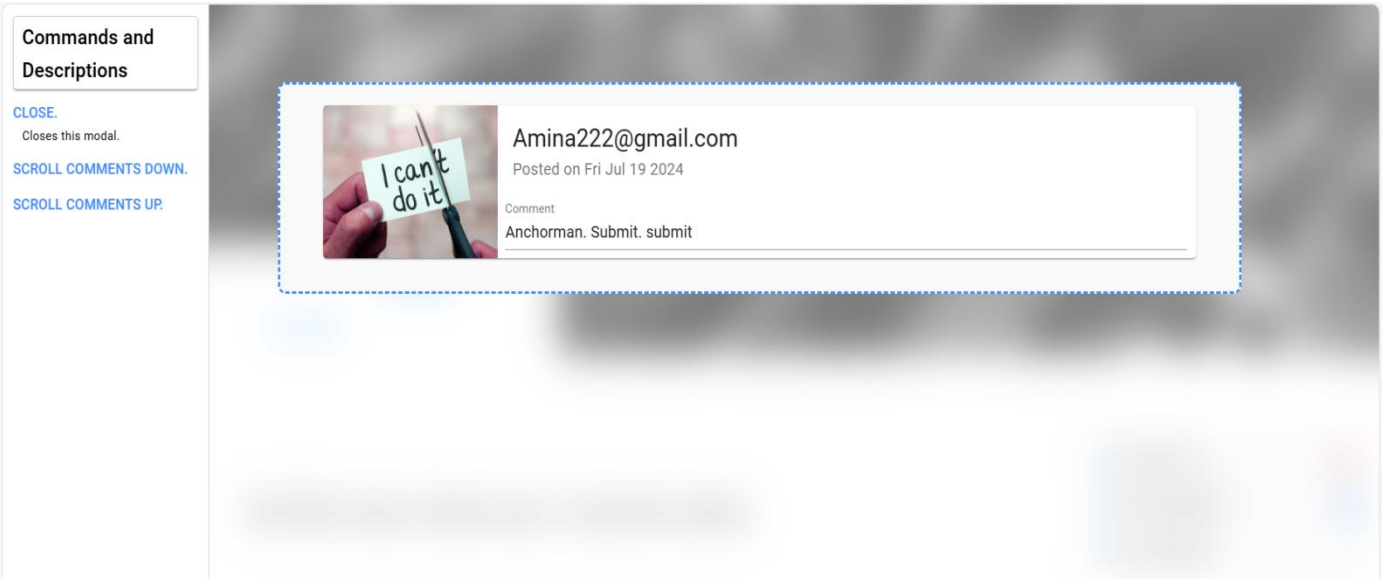
7.4.1. Add Comment



Description:

1. On first encounter of the add comment, system will tell about the page using 'speak tts'.
2. User can perform the tasks using his/her voice and the system will navigate or respond to the user according to his voice.
3. User can speak to add his/her comment.
4. User can close the modal.
5. User can say help and system will speak the available commands.
6. User can say speak and the system will write the comment on the published article.
7. User say "**submit comment"** and comment will be added, system will close the comment section.

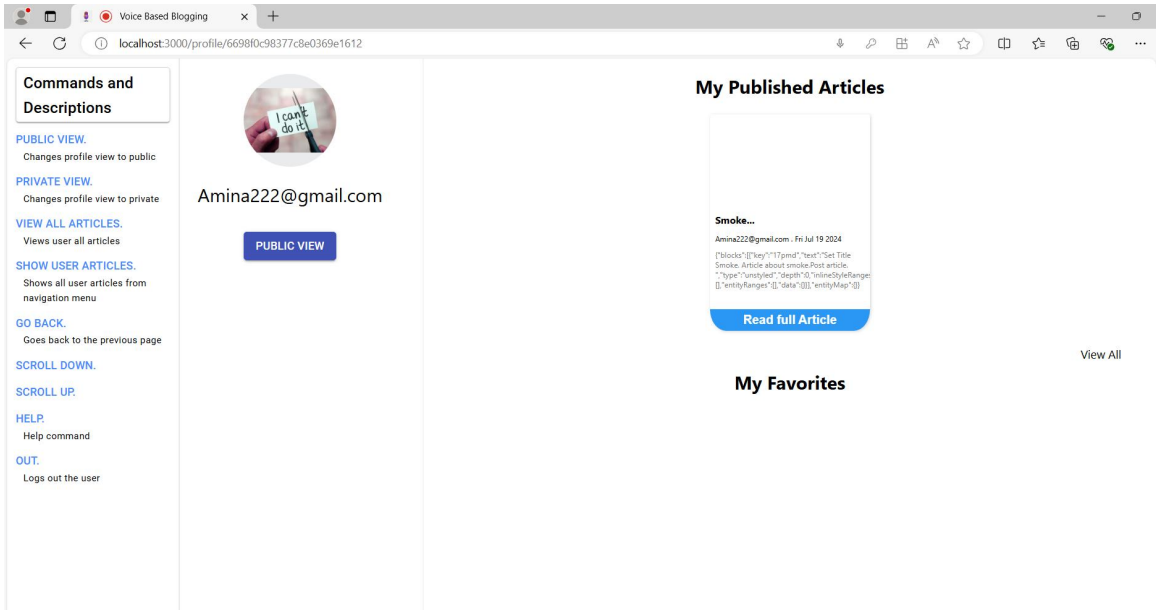
7.4.2. View Comment



Description:

1. User can perform the tasks using his/her voice and the system will navigate or respond to the user according to his voice.
2. User can scroll up and down the comments.
3. User can close the modal.

7.4 User Profile



Description:

1. User can perform the tasks using his/her voice and the system will navigate or respond to the user according to his voice.
2. User can scroll up and down the profile.
3. User can go back to the articles directory.
4. User can log himself out of the website.
5. User can say help and it will guide him to the available commands.

Chapter 8: Conclusion and Future Work

8.1. Conclusion:

The **"Voice Based Blogging Website For Blind People"** project aims to empower blind individuals by providing a voice-based blogging platform that is accessible and user-friendly. Through features such as voice command navigation and speech-to-text.

The development process leverages modern web technologies and accessibility standards to create a robust and scalable application. The project prioritizes continuous improvement, user-centric design, and regular updates based on user feedback. Emphasis on non-functional aspects such as performance, security, usability, scalability, reliability, compatibility, regulatory compliance, data backup, and maintenance ensures that the app meets high standards of quality and reliability.

"Voice Based Blogging Website For Blind People" is not just a tool for blogging; it is a comprehensive platform that integrates various aspects of digital accessibility into a single, easy-to-use application. The app's design prioritizes user experience, security, and reliability, making it a valuable tool for users seeking to share their voices and experiences through blogging.

8.2. Future Work:

The future directions for the **"Voice Based Blogging Website For Blind People"** include following key areas for development and enhancement:

1. **Enhanced Voice Recognition**:
    - Implement advanced voice recognition technologies to improve the accuracy and responsiveness of the speech-to-text functionality.
    - Support for multiple languages and dialects to cater to a diverse user base.

2. **Machine Learning Integration**:

    - Use machine learning algorithms to personalize the user experience based

on individual preferences and usage patterns.
- Develop predictive text and voice input suggestions to streamline the blogging process.

3. **Community and Social Features**:

    - Introduce community features such as user profiles, followers, and comments to foster a sense of community among users.
    - Enable social media integration to allow users to share their blog posts on various platforms directly from the app.

4. **Accessibility Enhancements**:

    - Continuously improve the accessibility features based on user feedback and emerging best practices in web accessibility.
    - Include additional assistive technologies like Braille display compatibility and tactile feedback for touchscreen devices.

5. **Content Moderation and Safety**:

    - Develop automated and manual content moderation systems to ensure a safe and respectful environment for all users.
    - Implement privacy and security measures to protect user data and prevent unauthorized access.

6. **Monetization Options**:

    - Explore monetization strategies such as premium features, ad placements, and partnerships with relevant organizations.
    - Provide opportunities for users to monetize their content through sponsorship and affiliate marketing.

7. **Data Analytics and Insights**:

    - Incorporate data analytics tools to provide users with insights into their blog performance, reader demographics, and engagement metrics.
    - Use these insights to offer personalized recommendations and growth strategies for content creators.

8. **Educational Resources and Training**:

    - Develop educational resources, tutorials, and training sessions to help users maximize the platform's potential.
    - Partner with organizations to offer workshops and webinars on digital accessibility and effective blogging techniques.