Software Requirements Specification

Version <<NULL>>

Sep 07, 2024

Discussify

by

Toe

**Table of Contents**

Table of Contents

List of Figures

1. Introduction
2. Overall Description
3. Requirements Specification

**List of Figures**

[Figure 1 - System Environment 4](#_Toc77487669)

[Figure 2 - Article Submission Process 6](#_Toc77487670)

[Figure 3 - Editor Use Cases 8](#_Toc77487671)

[Figure 4 - Logical Structure of the Article Manager Data 23](#_Toc77487672)

**1. Introduction**

***1.1. Purpose***

The purpose of this document is to present a detailed description of the Discussify. It will explain the purpose and features of the system, MAYBE the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

***1.2. Scope of Project***

Discussify is a discussion platform allowing users to post, comment, vote, and manage communities. Also, about managing a bustling community, ensuring everyone’s voice is heard, and delivering personalized content to keep users engaged. The application will initially launch as mobile app, the webapp is planned for future development.

***1.3. Glossary/Convention***

|  |  |
| --- | --- |
| **Term** | **Definition** |
| DB | Database |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

***1.4. References***

***1.5. Overview of Documents***

In the Overall Description section, this document gives an overview of the functionality of this project. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

**2. Overall Description**

***2.1. Product Overview***

Discussify will be a social platform using microservices architecture, allowing users to post, comment, vote, and follow various user-created communities.

The backend will be built with .**NET Core Microservices**. The system will consist of multiple backend services, each responsible for different core functionalities. Each service will have its own dedicated database engine to ensure scalability and separation of concerns. **Discussify** will use **RabbitMQ** for inter-service communication and **SignalR** for real-time notifications. The authentication and authorization mechanisms will be managed using .**NET Identity** in one of the services.

The frontend of Discussify will initially be a **mobile app** built using **React Native**. React Native allows the development of a single codebase that can run on both iOS and Android, ensuring consistency across platforms and reducing development time.

***2.2. Product Features***

***2.3. User Classes and Characteristics***

***2.4. Operating Environment***

***2.5. Design and Implementation Constraint***

***2.6. Assumptions and Dependencies***

**3. System Features**

***3.1. Functional Requirements***

**4. External Interface Requirements**

***4.1. User Interfaces***

***4.2. Hardware Interfaces***

***4.3. Software Interfaces***

***4.4. Communication Interfaces***

**5. Nonfunctional Requirement**

***5.1. Performance Requirements***

***5.2. Safety Requirements***

***5.3. Security Requirements***

***5.4. Software Quality Requirements***