

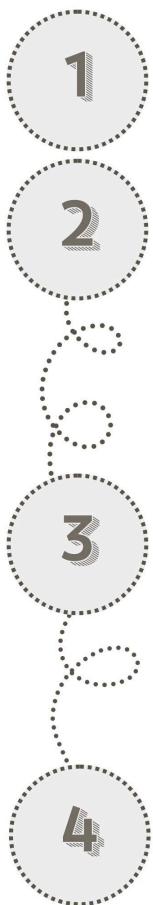


SE Project

BrainStack - Design Report

Submitted To :- Dr. Santosh Singh Rathore

TABLE OF CONTENTS



INTRODUCTION

DESIGN CONSIDERATIONS

Assumptions
Design Constraints
Design Methodology
System Environment

ARCHITECTURE

System Design
Functional Decomposition Tree
Data Flow Diagram

USER INTERFACE DESIGN



INTRODUCTION

OVERVIEW OF THE PROJECT

The project involves the development of Brainstack, a platform inspired by Discord, designed to provide a collaborative communication environment. This platform aims to facilitate seamless interaction among users, enabling them to create and participate in multiple custom rooms, join video channels, edit their profiles, and much more. One of the key features of BrainStack is the ability to share files over the platform, enhancing the collaborative experience and making it easier for users to exchange important documents, images, and other media.

Brainstack is designed to cater to a diverse user base, from casual social groups to professional teams, providing a versatile space for communication and collaboration. The platform emphasizes user-friendliness and accessibility, ensuring that individuals with varying levels of technical proficiency can navigate and utilize its features effectively. By providing a secure and engaging environment, Brainstack seeks to become a go-to platform for those looking to connect, collaborate, and create in a digital space.

OVERVIEW OF EXISTING SYSTEMS AND TECHNOLOGIES

CUSTOM ROOMS: Users can create and join custom rooms tailored to specific topics, interests, or projects, fostering targeted discussions and collaborations.

VIDEO CHANNELS: The platform supports video channels, allowing users to engage in face-to-face conversations, virtual meetings, and live streams, enhancing the sense of connection among participants.

PROFILE CUSTOMISATION: Users can personalise their profiles with avatars, bios, and other details, enabling them to express their identity and preferences within the community.

FILE SHARING: Brainstack enables users to share files directly within the platform, facilitating easy access to important documents, images, and other media, thus streamlining collaboration.

CHATBOT: Brainstack incorporates a chatbot named BrainBot that provides automated assistance, answers user queries, and enhances the overall user experience, making the platform more interactive and user-friendly.



DESIGN CONSIDERATIONS

Design considerations are important factors to take into account when creating a software system, as they influence the decisions made throughout the design process. In the case of our BrainStack, some of the main design considerations include the assumptions made, design constraints, design methodology, and system environment.

ASSUMPTIONS

- **USER DIVERSITY:** We anticipate a diverse user base with varying technical skills and communication needs. The platform is designed to be user-friendly and accessible to everyone, from casual users to professional teams.
- **DEVICE COMPATIBILITY:** Brainstack is expected to be used on various devices, including desktops, laptops, and mobile phones. We aim for compatibility across different operating systems and browsers to ensure a consistent user experience.
- **STABLE INTERNET CONNECTION:** We assume users have stable internet connections for uninterrupted communication. Brainstack is optimized for performance, but a minimum level of connectivity is necessary for smooth operation of features like video calls and real-time messaging.

DESIGN CONSTRAINTS

- **TECHNICAL CONSTRAINTS:** The choice of technology stack and development tools may impose limitations on Brainstack's functionality and performance.
- **BUDGET CONSTRAINTS:** Budget Constraints: Financial limitations may affect the development scope, impacting the range of features and overall quality of the platform.
- **TIME CONSTRAINTS:** Deadlines may restrict the development process, influencing the depth of testing and refinement of features.
- **LEGAL AND COMPLIANCE CONSTRAINTS:** Brainstack must comply with data protection regulations and copyright laws, which could impact its design and functionality.



DESIGN METHODOLOGY

AGILE METHODOLOGY: Brainstack utilised an agile approach, prioritising flexibility and iterative development. This allowed for regular incorporation of user feedback and continuous improvement throughout the design process.

USER-CENTRED DESIGN: The development of Brainstack was guided by a user-centered design philosophy. Emphasis was placed on understanding user needs and preferences, with design decisions informed by usability testing and user feedback.

SYSTEM ENVIRONMENT

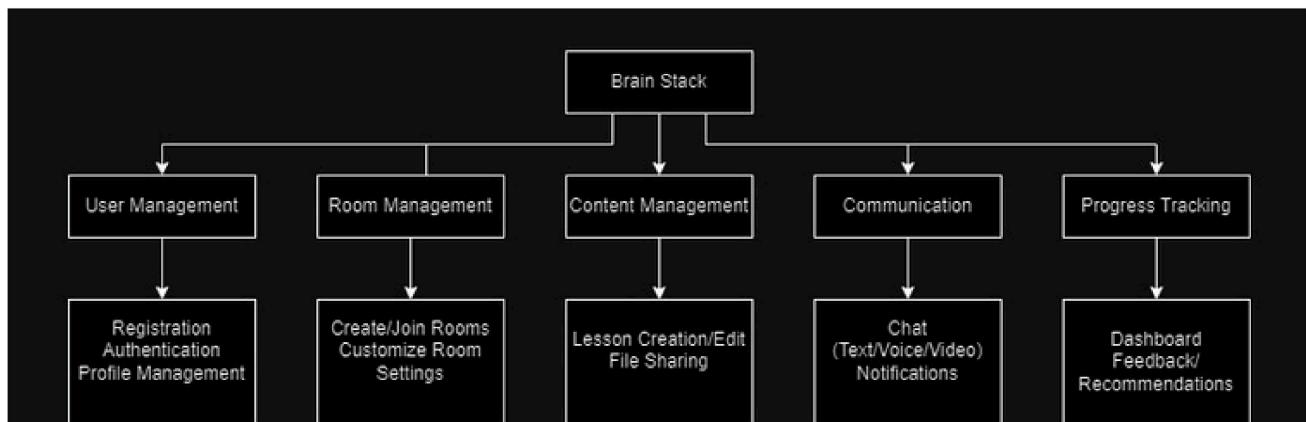
- **DEPLOYMENT:** Brainstack is hosted on cloud servers for scalability and reliability, allowing seamless global access.
- **COMPATIBILITY:** The platform is compatible with various devices and operating systems, including Windows, macOS, iOS, and Android.
- **DEPENDENCIES:** Brainstack uses third-party services, libraries, and APIs for features like authentication, multimedia processing, and communication.
- **SCALABILITY:** The cloud-based environment allows for easy scaling to accommodate growth and increased user activity.

ARCHITECTURE

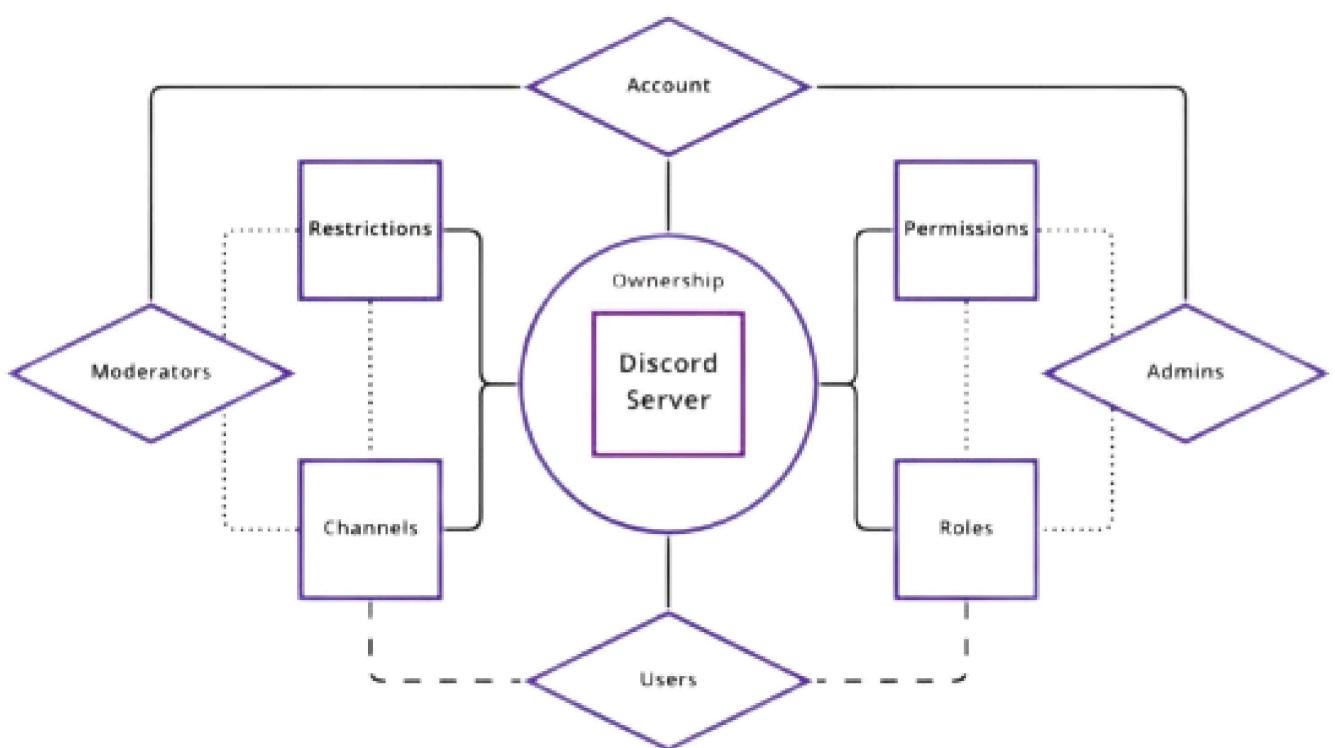
SYSTEM DESIGN

- **USER REGISTRATION MODEL:** This module allows individuals to create learner accounts on the platform. Users can provide essential information during registration, such as their name, email address, and password. Additional user profile details can be added to personalize the learning experience.
- **CUSTOMISED ROOMS:** Users can create and join custom rooms tailored to specific topics, interests, or projects, fostering targeted discussions and collaborations. Some of these rooms can be designated as video rooms, allowing users to engage in face-to-face conversations, virtual meetings, and live streams, enhancing the sense of connection among participants. Each room can have its own set of features and permissions, allowing users to customise their experience based on their needs.
- **BACKEND ARCHITECTURE:** The system architecture would likely involve a backend database to store room configurations, user permissions, and video room settings. APIs would be used to communicate between the frontend and backend, allowing for seamless interactions and updates to room information and user settings. The frontend would be designed to provide a user-friendly interface for both learners and administrators, allowing them to easily access and manage their respective areas of the platform.
- **PERSONALISED USER PROFILE:** Users can personalise their profiles with avatars, bios, and other details, enabling them to express their identity and preferences within the community. This allows users to showcase their interests and expertise, making it easier for others to connect and collaborate with them.

FUNCTIONAL DECOMPOSITION TREE



INSPIRATIONAL DATAFLOW DIAGRAM

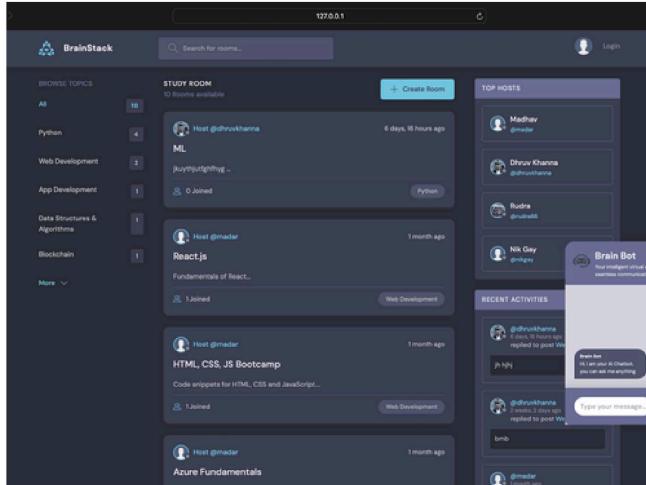


USER INTERFACE DESIGN

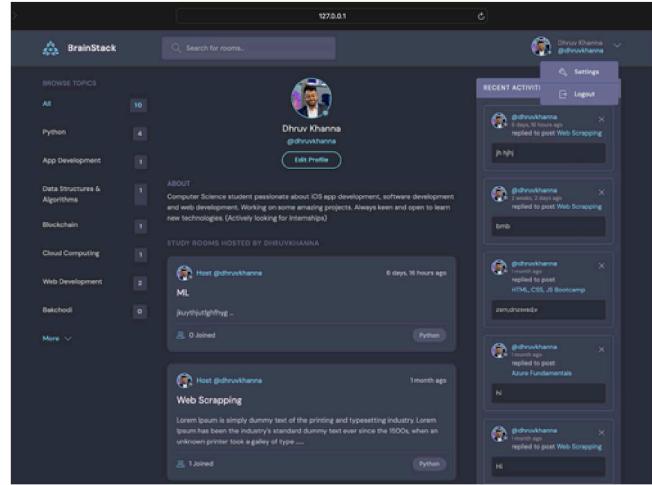
- **SIMPLE NAVIGATION:** Used a clean and intuitive layout with easy-to-understand navigation menus and buttons to help users move between different sections of the platform effortlessly.
- **CLEAR COMMUNICATION:** Ensured that important information, such as notifications and feedback, is clearly visible and easy to understand. Used concise language and visual cues to convey messages effectively.
- **RESPONSIVE DESIGN:** Designed the UI to be responsive, ensuring that it adapts seamlessly to different screen sizes and devices. This ensured a consistent user experience across desktop, tablet, and mobile devices.
- **PERSONALISATION:** Allowed users to personalise their experience by customising their profiles and settings. This included choosing profile pictures, setting notification preferences, and selecting room themes, enhancing user engagement.

User interface will consist of the following main screens:

i) Home Page



ii) User Profile Page





iii) One of the Room Page

iv) Dashboard to join VideoRooms

v) Dashboard to share files

vi) Create new room

vii) Edit user profile