ÇANKAYA UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

CENG 407

Software Design Document (SDD)

GoatCapella

Tarık Şen

Çağdaş Güleç

Beyza Tozman

İrem Beyza Aydoğan

Şeref Berkay Kaptan

Table of Contents

1. Introduction

This Software Design Document (SDD) presents a technical overview of the GoatCapella project, an innovative platform aimed at elevating the experience of competitive programming and algorithmic problem-solving. Drawing upon the insights and requirements laid out in the Software Requirements Specification (SRS) and the educational methodologies explored in the Literature Review, this SDD brings together architectural decisions, system decompositions, user interface considerations, and database design strategies. GoatCapella seeks to consolidate challenges and data from multiple existing coding platforms—such as LeetCode, CodeWars, and CodeForces—into a single, immersive environment where learners can both hone their technical skills and engage with peers. In doing so, GoatCapella incorporates a variety of learning and motivational frameworks, including challenge-based learning, gamification, and social learning methodologies. These frameworks, described in detail in the Literature Review, help shape the platform's core features and ensure that it serves as an engaging, enjoyable, and educational ecosystem for diverse user groups.

1.1. Purpose

The purpose of this Software Design Document (SDD) is to define the architectural and design approach for developing GoatCapella, a web-based platform for coding challenges and competitive programming. This document provides the technical details required for developers, testers, and stakeholders to understand the system's design and ensure its implementation aligns with defined goals. GoatCapella integrates various coding platforms (e.g., LeetCode, CodeWars) and introduces features like gamification, challenge-based learning, and social collaboration to enhance user experience.

1.2. Scope

GoatCapella's scope extends beyond merely aggregating challenges from existing coding platforms. Its mission is to offer a unified and enhanced problem-solving environment where users can interact with multiple challenge sources, create customized challenges, track their own progress, and receive AI driven recommendations. Key functionalities include:

- **API Integrations:** Securely connecting to and extracting challenge-related data from third party platforms (LeetCode, CodeWars, CodeForces, etc.).
- **Challenge Management**: Enabling users and administrators to create, join, or monitor various types of challenges (public, private, speed-run, or adaptive challenges).
- **Adaptive Learning & AI Components:** Analyzing user performance to adjust question difficulty and recommend new challenges based on collaborative filtering, reinforcement learning, or content-based models.
- **Gamification Mechanics:** Introducing badges, coins, daily streaks, and leaderboards to foster

ongoing engagement and reflect progress.

- **Community & Feedback Features:** Offering commentary tools, solution reviews, and collaborative spaces for each challenge, thus promoting the social learning methodology highlighted in the Literature Review.

Potential future additions to the platform's roadmap, as hinted at in both the SRS and Literature Review, could include exploring the development of a mobile app to improve accessibility and considering the creation of a proprietary problem-solving system to possibly reduce reliance on external platforms.

1.3. Definitions, Acronyms, and Abbreviations

Although many of the key terms used in GoatCapella (e.g., SRS, CBL, Gamification) have been defined in the Software Requirements Specification (SRS), the Software Design Document (SDD) introduces a few additional, design-focused terms that warrant clarification:

- **Architectural Diagram:** A high-level representation of the system's structure, illustrating how various components (modules, services, third-party integrations) interact. This diagram helps stakeholders visualize the "big picture" of the application and understand major communication pathways.
- **ER Diagram (Entity-Relationship Diagram):** A graphical illustration of the database's entities (e.g., Users, Challenges, Solutions) and the relationships among them. In the SDD, ER Diagrams help convey how data is stored and interconnected in relational databases (e.g., PostgreSQL).
- **Sequence Diagram:** A type of UML (Unified Modeling Language) diagram that shows how processes interact through time. In this document, sequence diagrams map out the flow of messages or events between objects (e.g., the application server, the database, the user's browser) in a specific scenario, such as creating a new challenge or fetching user progress.
- **Activity Diagram:** A type of UML diagram that represents the flow of activities or tasks in a system. It is used to visualize the sequence and conditions of workflows, such as a user registering for the platform, submitting a challenge, or receiving feedback. Activity diagrams are particularly useful for identifying decision points and parallel processes, providing a detailed view of the system's functional behavior.
- **Use Case Diagram:** A UML diagram that depicts the interactions between the system and its various actors (e.g., regular users, sub-admins, main admins). Use case diagrams help define functional requirements by illustrating the distinct actions each actor can perform.
- **Class Diagram:** Another UML construct used to represent the static structure of the system by showing its classes, attributes, operations, and the relationships among them. In the SDD, class diagrams clarify how the application's main objects (e.g., User, Challenge, Submission) are

modeled in code.

- **UML (Unified Modeling Language):** A standardized modeling language that offers a variety of diagrams (such as Sequence Diagrams, Class Diagrams, and Use Case Diagrams) for describing the design, structure, and behavior of a software system.

These design-oriented terms are referenced throughout the SDD to ensure clarity around how GoatCapella's architecture, data models, and workflows are documented and communicated.

1.4. Overview

This document is structured to offer a clear progression from high-level design objectives to more granular details.

A. Introduction

Provides the high-level background and motivations behind GoatCapella, referencing the SRS and Literature Review for context. It outlines the purpose, scope, and key terminology relevant to the project's implementation.

B. System Design

- **Architectural Design:** Discusses the system architecture, highlighting the division of responsibilities among modules.
- **Decomposition Description:** Breaks down the solution into smaller components or services, detailing how each module communicates within the broader ecosystem.
- **System Modeling:** Shows activity diagrams and sequence diagrams to illustrate the operational lifecycle of features (e.g., challenge creation, AI-based question suggestion, user progress tracking).
- **Database Design:** Explains the rationale for choosing both a relational database (PostgreSQL) for structured data and a NoSQL store (MongoDB) for unstructured or flexible data needs.

C. User Interface Design

Presents the primary page flows (landing page, challenge overview, challenge creation, profile page, admin dashboard, etc.).

1.5. Version History

A summary of the document's revision history ensures transparency and traceability:

Version	Date	Description
1.0	27.12.2024	Initial Release

2. System Design

2.1. Architectural Design

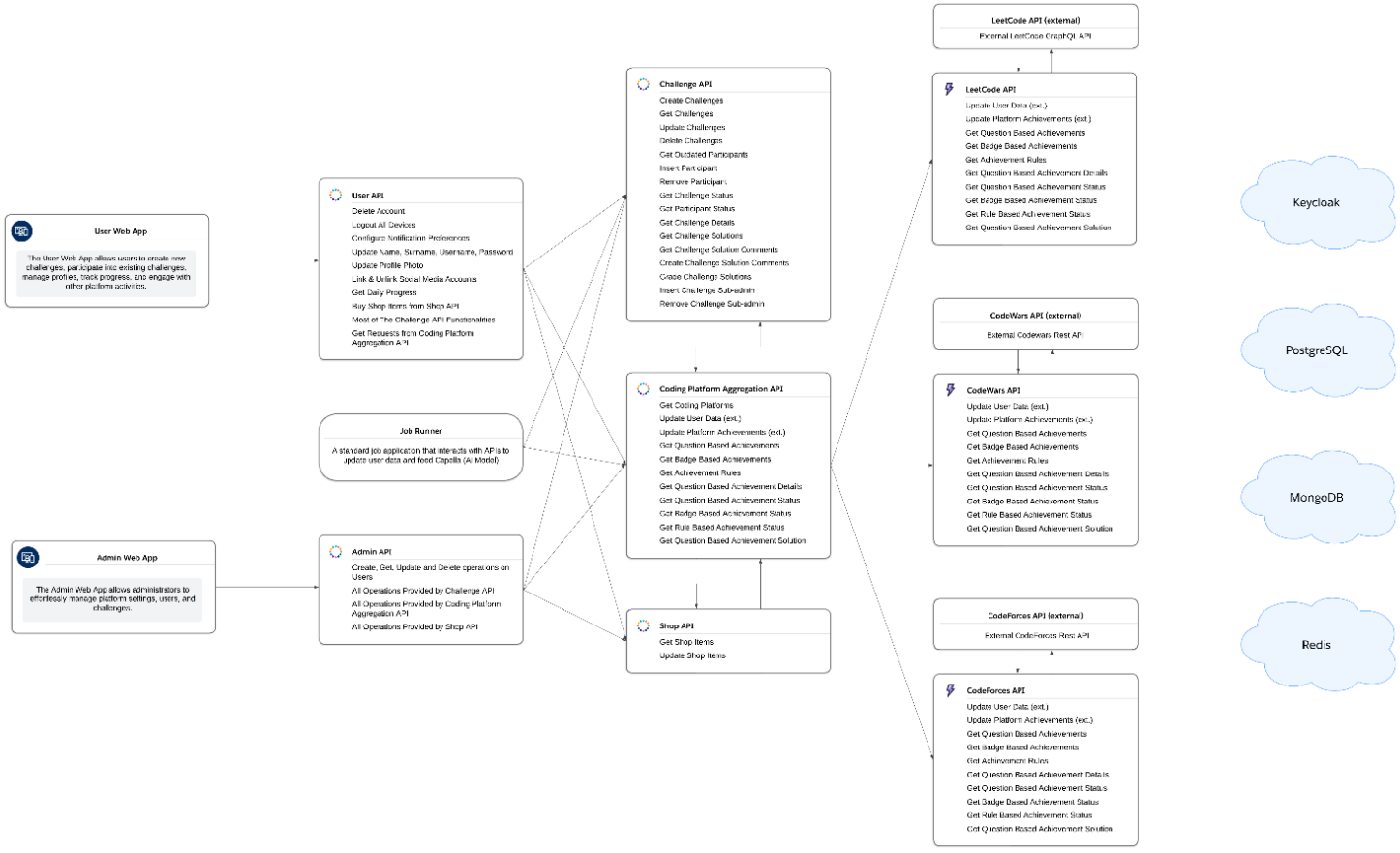

Figure 1. High-Level Architectural Design

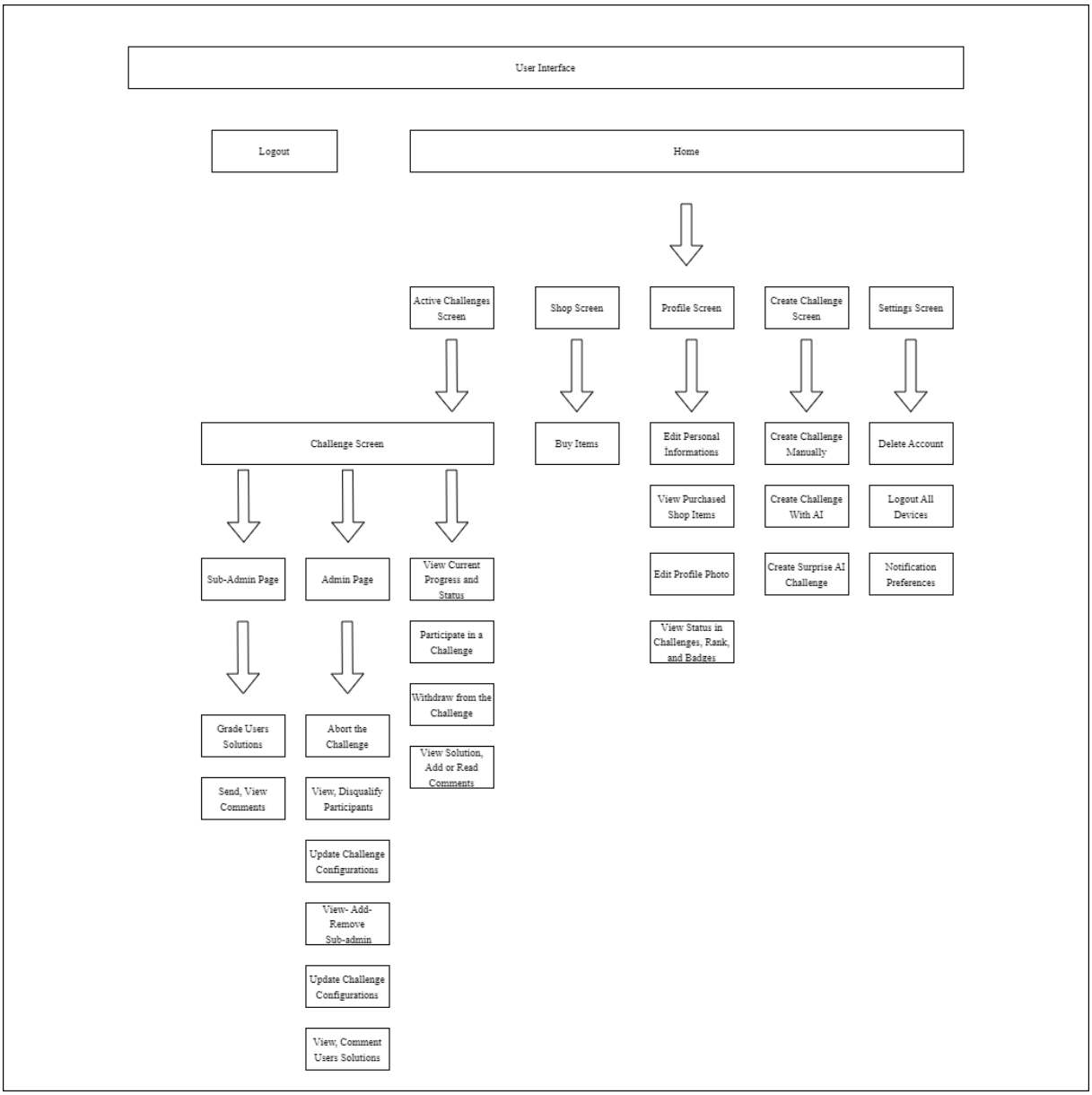

Figure 2. Layered Architecture Design for System

2.2. Decomposition Description

1. User Interface

Main Screen:

- It is the section where users can access the shop, profile, active challenges, participated challenges, and settings.
- Facilitate navigation between different challenges and features.

View Components:

- Displays challenges, leaderboards, and user achievements.
- Includes screens for initiating, participating in, and tracking challenges.

2. **Analytics and Personalization**

AI Data Analytics:

- It collects and analyzes user data to understand the user's preferences and interactions.

AI-Driven Personalization Manager:

- Utilizes artificial intelligence to perform data analysis, creating new challenges or recommending existing challenges based on the user's previous participation in challenges.

3. **Shop**

Shop Manager:

- Provides an interface where users can view items such as challenge boosters, extra attempts, or special content.
- Handles purchase transactions exclusively using in-app coins and updates the coin balance accordingly.

Item Catalog:

- Stores prices (in coins) and descriptions of challenge-related items.

4. **Settings**

User Preferences:

- Allows users to configure their application preferences.
- Provides options to enable or disable specific types of notifications.

Account Manager:

- Allows users to permanently delete their accounts.
- Enables users to log out from all active sessions on multiple devices.

5. **Security and Privacy**

Data Protection:

- Advanced encryption techniques are used to secure user data.

- Privacy and security are maintained during data transmission and storage.

Privacy Controls:

- It provides privacy settings that allow users to control their personal information and manage how it is shared.

6. Challenge Process

Manual Challenge Creation:

- Challenge admins can create their own challenges by defining settings, duration, difficulty, and rewards.
- Challenge admins can add a sub-admin and invite other users to participate in the challenge.

AI-Assisted Challenge Creation:

- Challenge admins can customize a few settings and let AI generate the rest for optimal user participation and engagement.

Surprise AI-Generated Challenges:

- The AI Agent analyzes the user's preferences, historical performance data, and activity patterns to create a challenge configuration tailored to the user’s needs.

Active Challenges:

- Users can view a list of active challenges they are currently participating in and track their progress.
- They can see details of existing challenges, including their names, durations, participants, and rewards.

Features:

- Users can filter or search for challenges based on criteria such as type, duration, or rewards.

7. Challenge Management Responsibilities

Admin Responsibilities:

- Create or abort challenges.
- View and manage participants, assign/remove sub-admins.
- Monitor progress and grade submissions.

Sub-Admin Responsibilities:

- Grade submissions, manage comments, and track progress.

Collaboration:

- Admins oversee sub-admin actions to maintain control and accountability.

8. Notifications and Alerts

Notification Manager:

- Manages in-app notifications and real-time alerts for challenge updates, deadlines, or results.
- Customizes notifications based on user preferences.

Event Triggers:

- Sends notifications for significant events such as the addition of a new challenge, deadline reminders, and results of completed challenges.

9. User Engagement and Social Features

Engagement Tools:

- Provides features like leaderboards and badges to motivate users.

Social Interaction:

- Competing with other users also increases motivation and engagement.

2.3. System Modeling

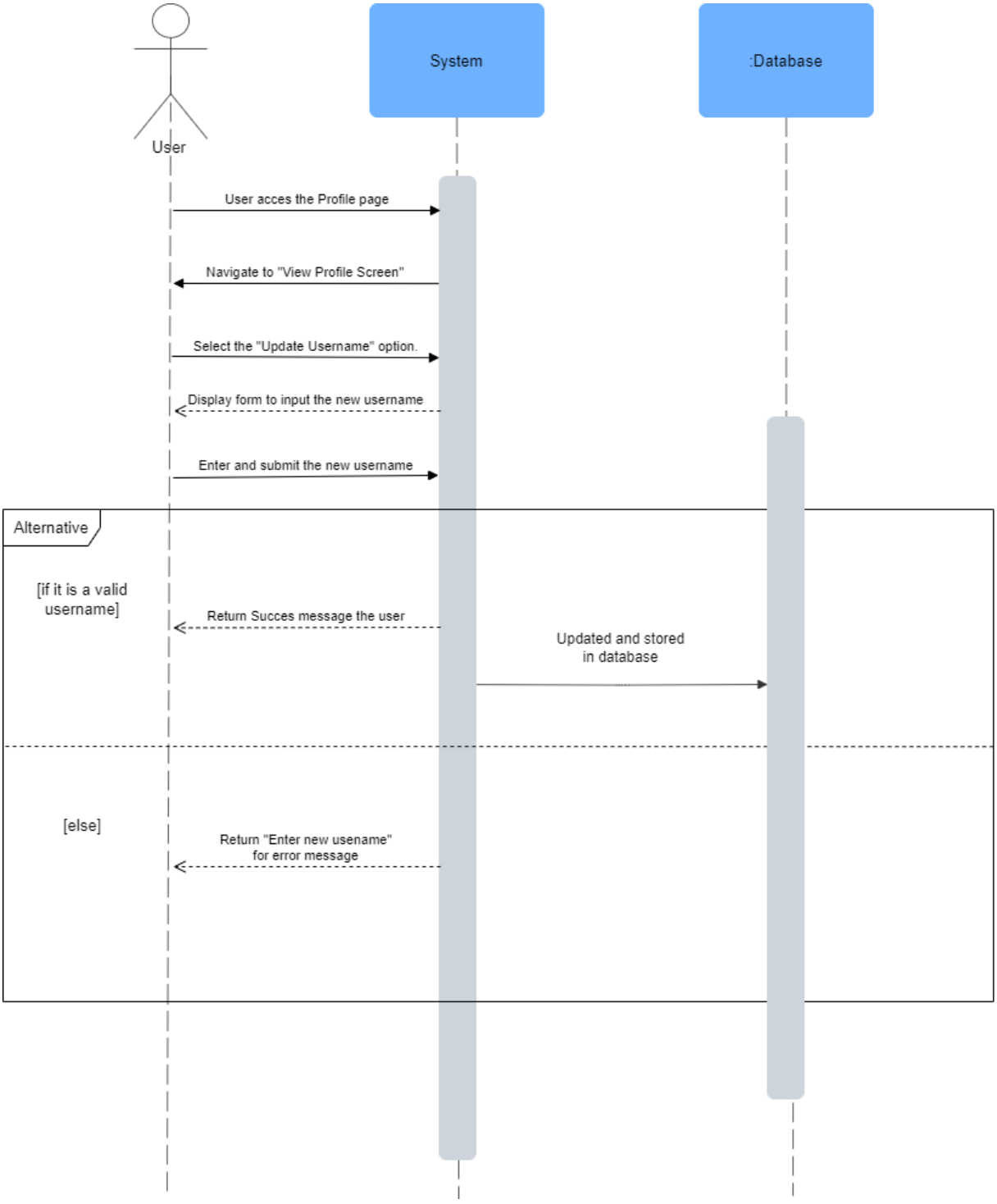

Figure 3. Sequence Diagram of Update Username

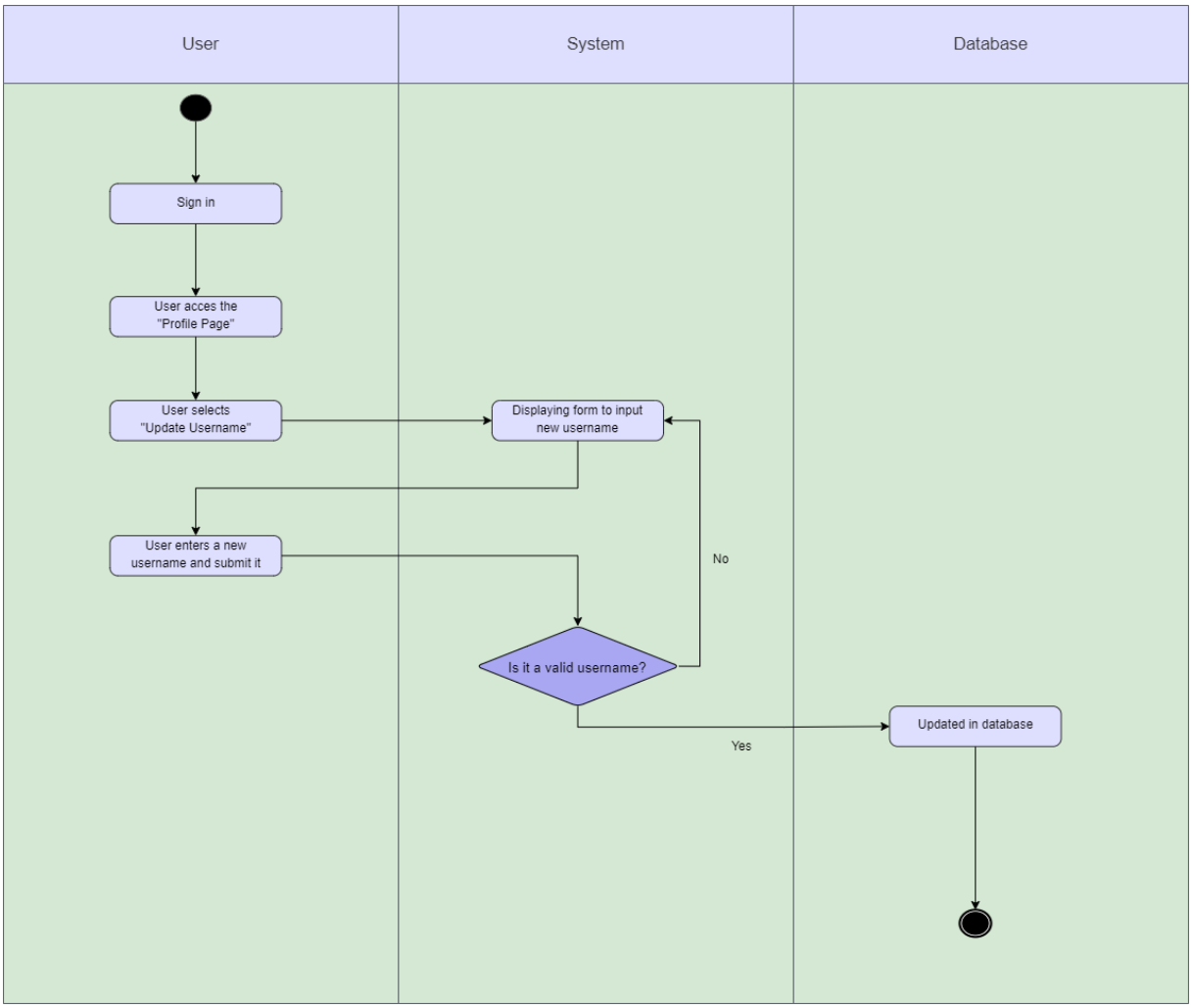

Figure 4. Activity Diagram of Update Username

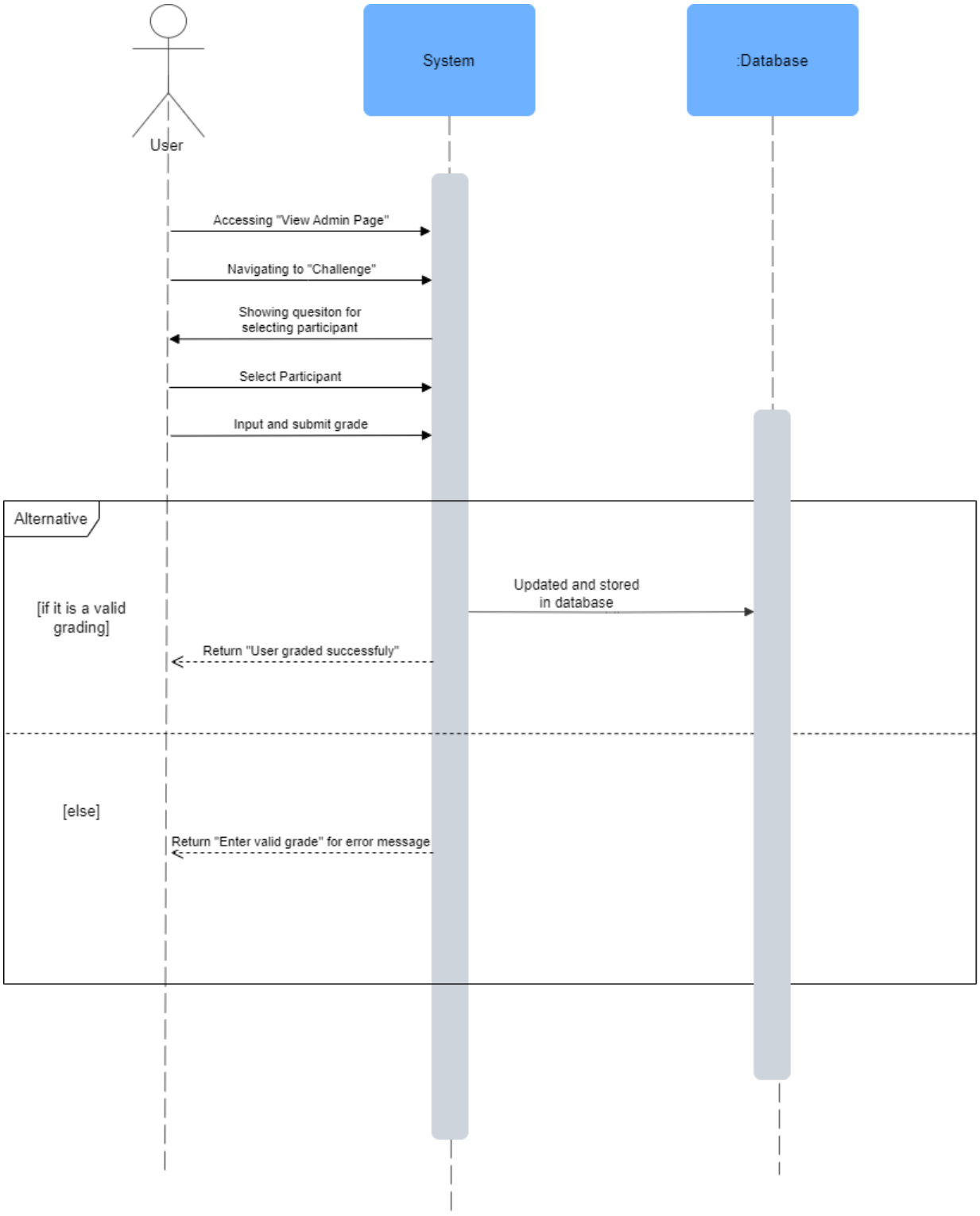

Figure 5. Sequence Diagram of Admin Grading

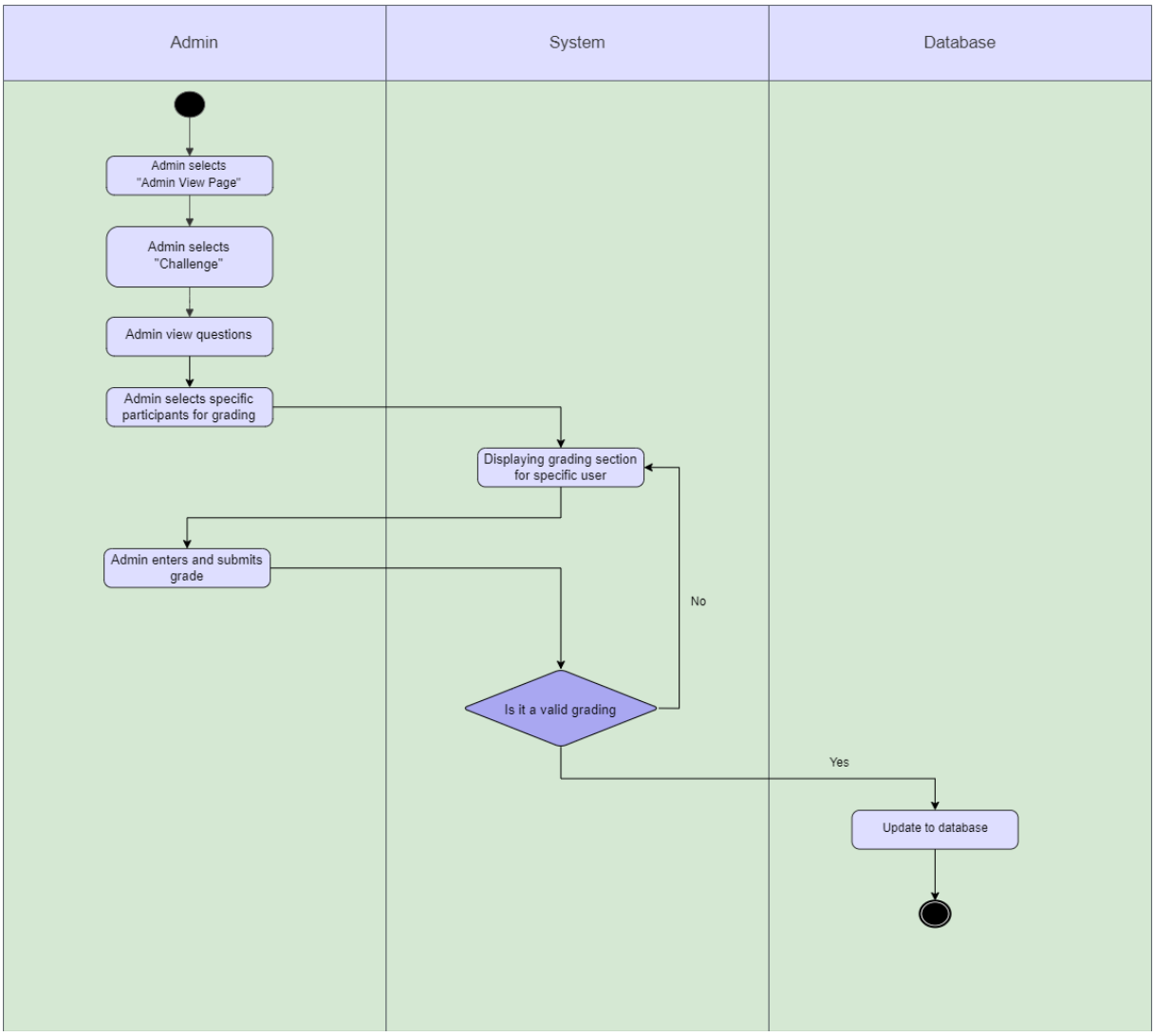

Figure 6. Activity Diagram of Admin Grading

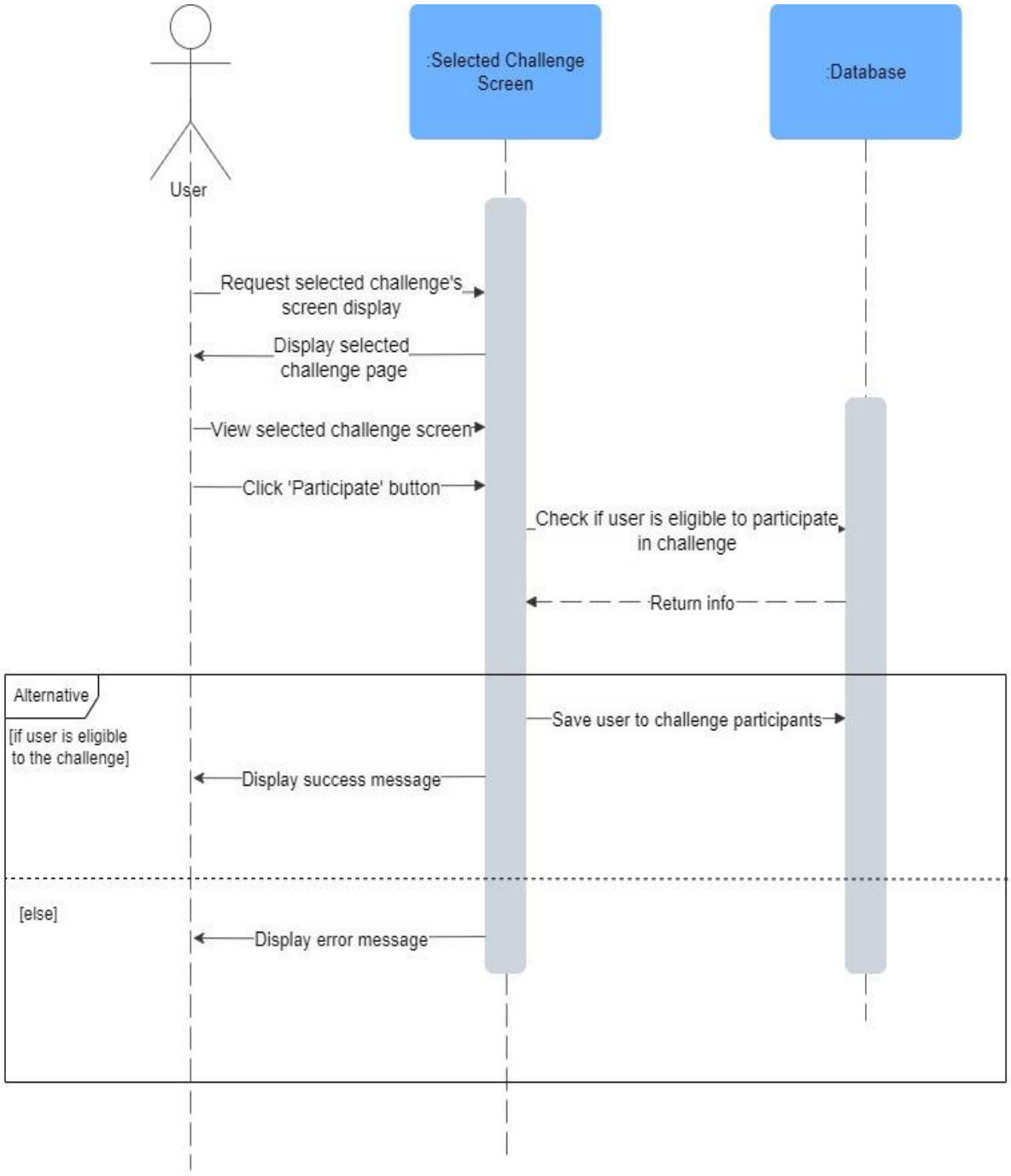

Figure 7. Sequence Diagram of Participant Challenge

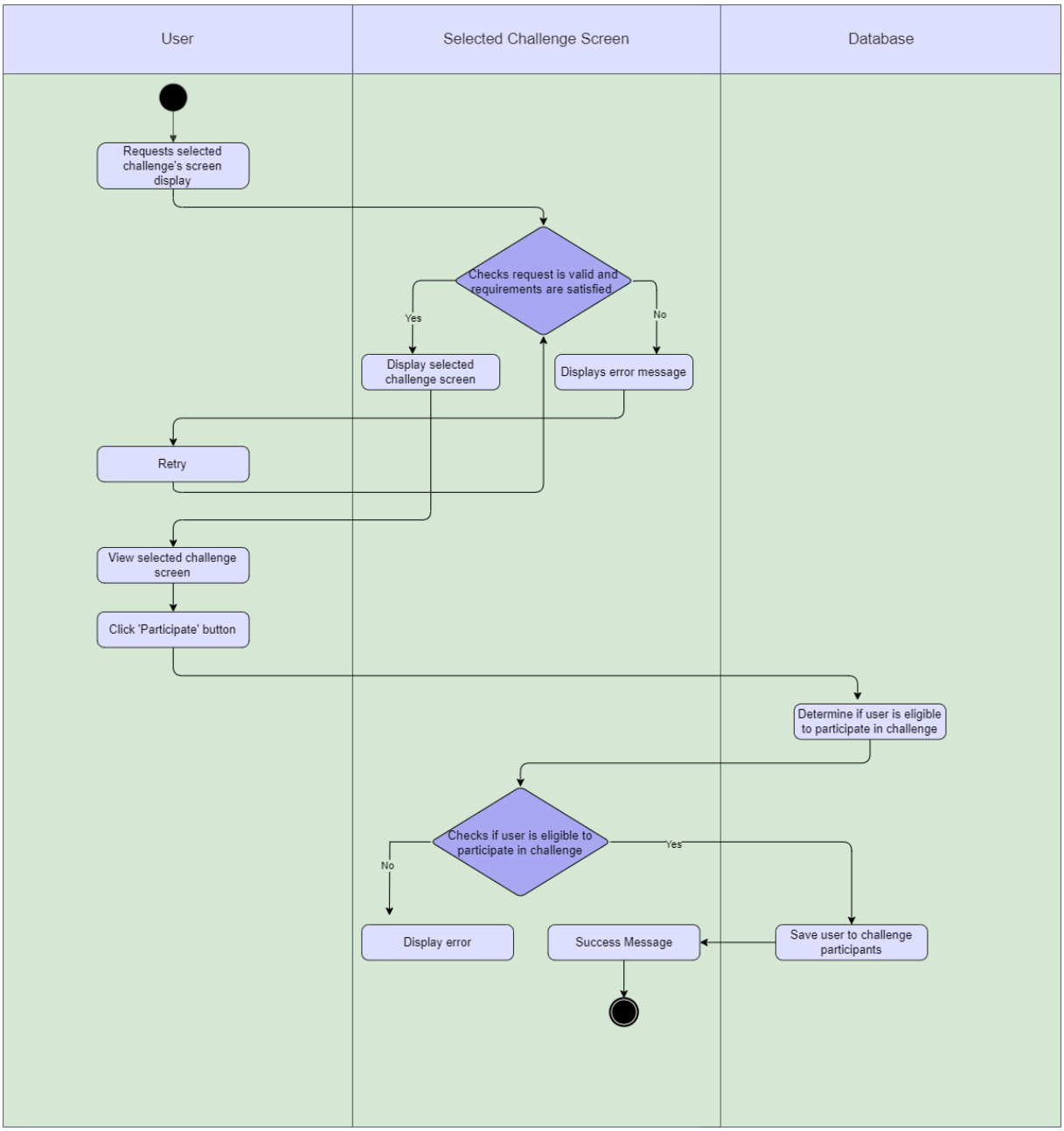

Figure 8. Activity Diagram of Participant Challenge

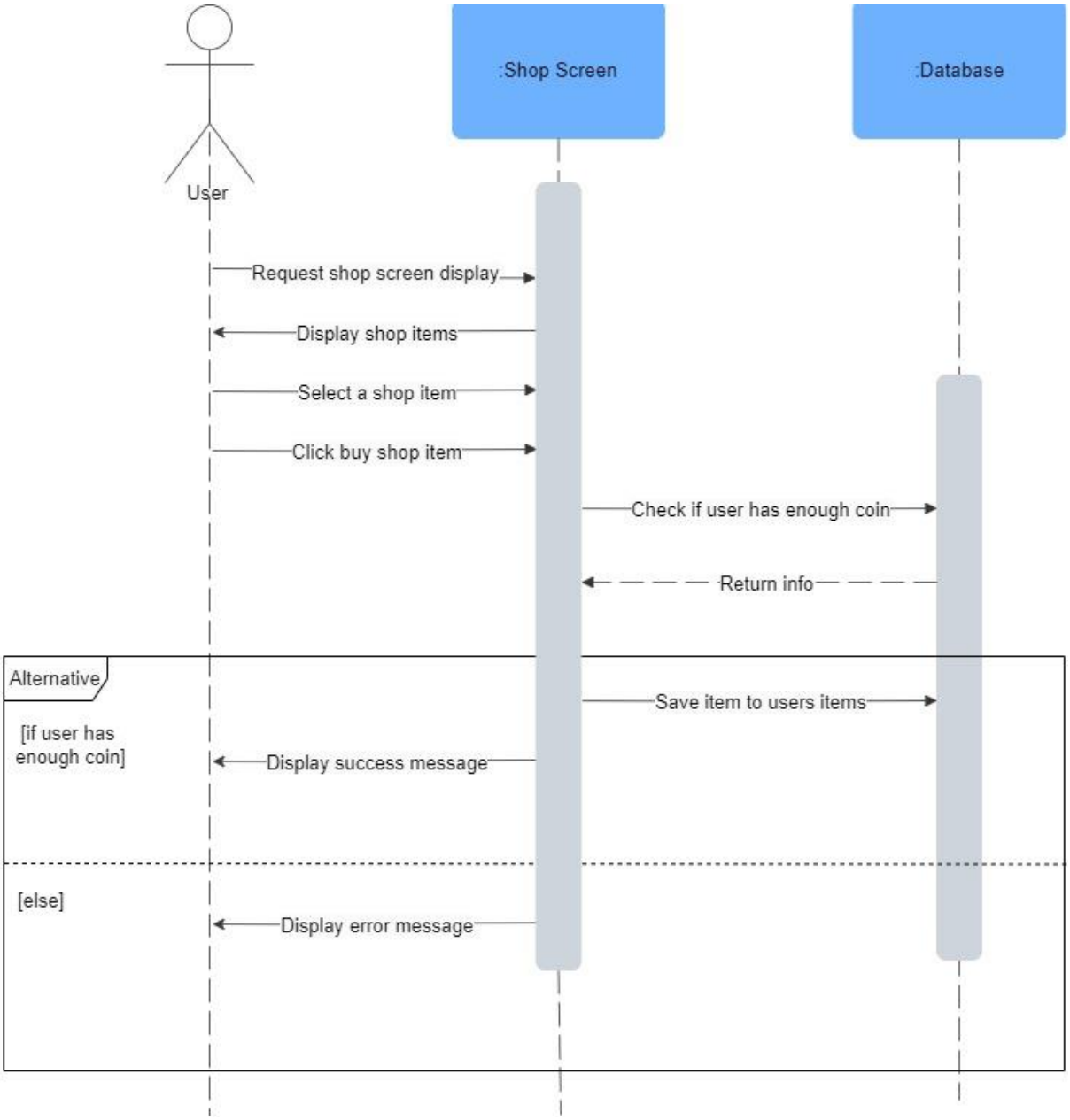

Figure 9. Sequence Diagram of Shop Page

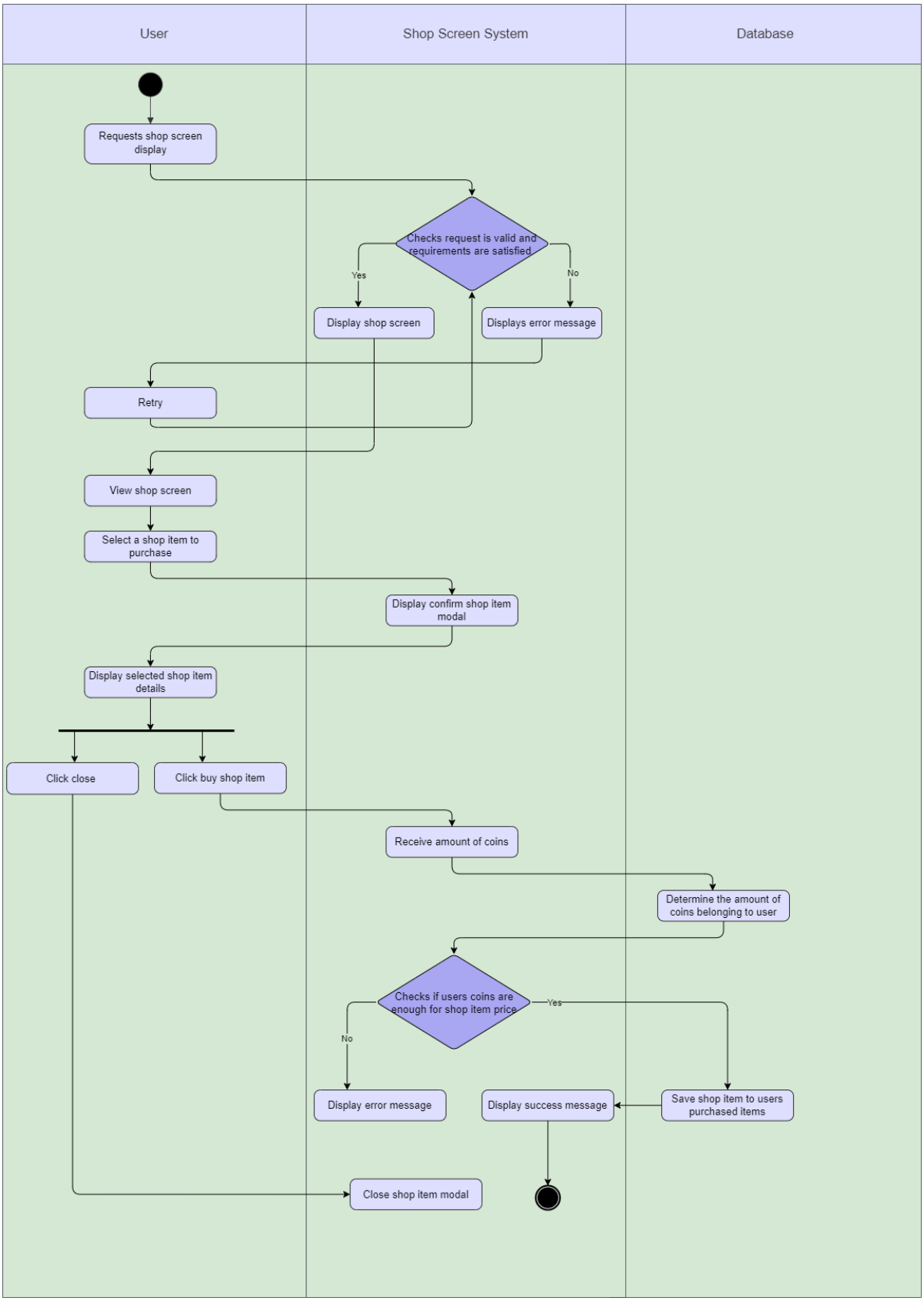

Figure 10. Activity Diagram of Shop Page

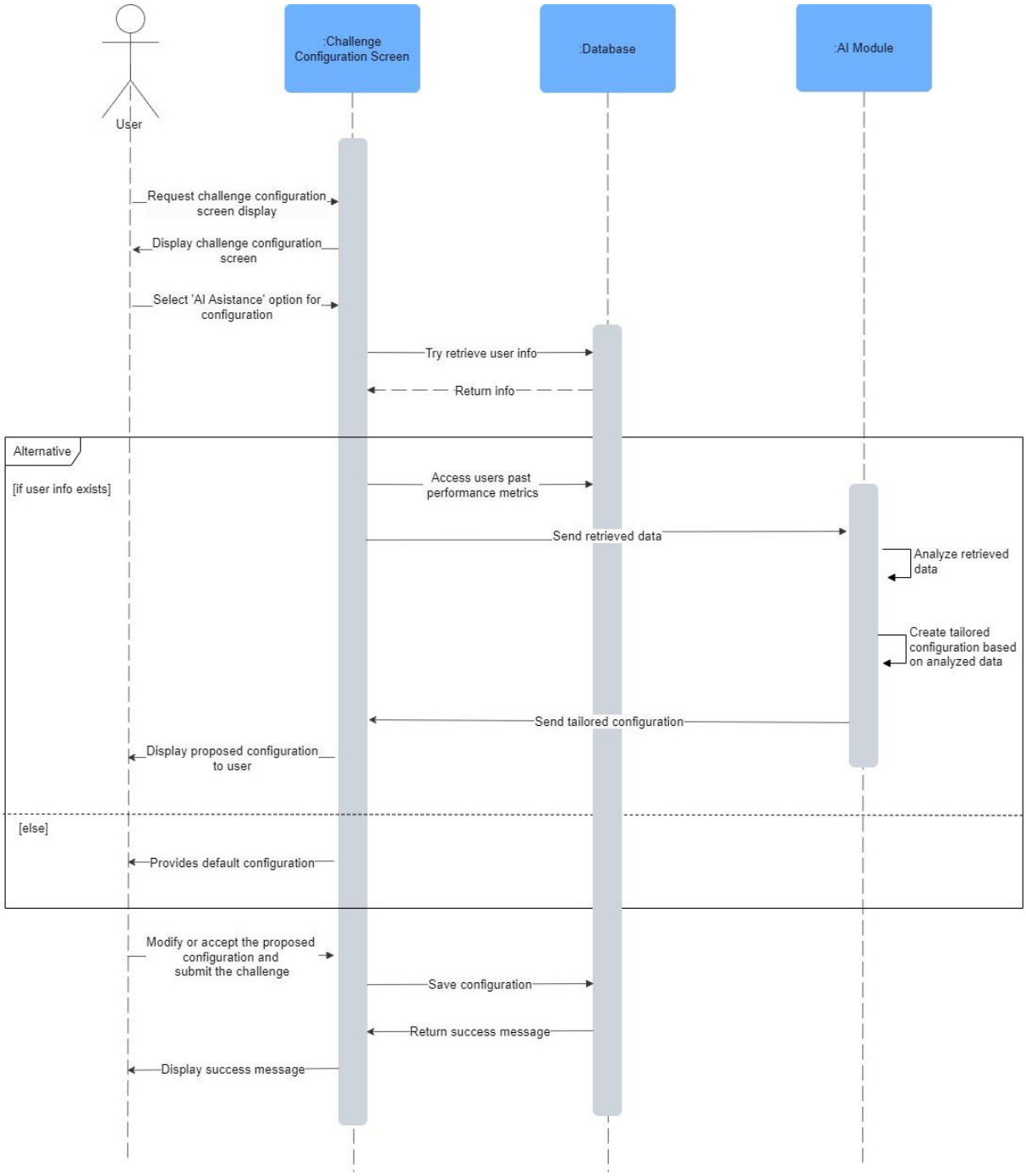

Figure 11. Sequence Diagram of AI Create Challenge

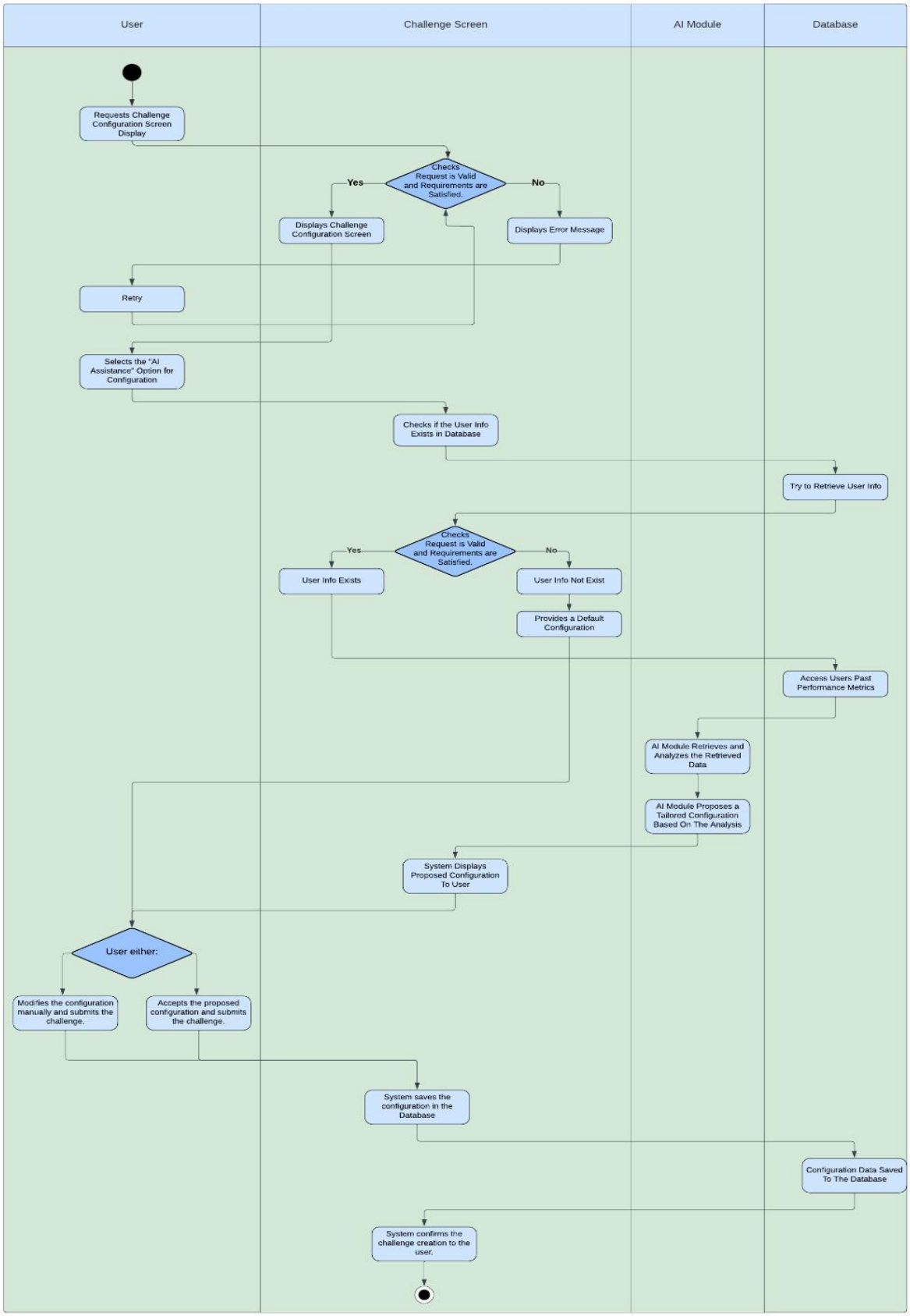

Figure 12. Activity Diagram of AI Create Challenge

2.4. Database Design

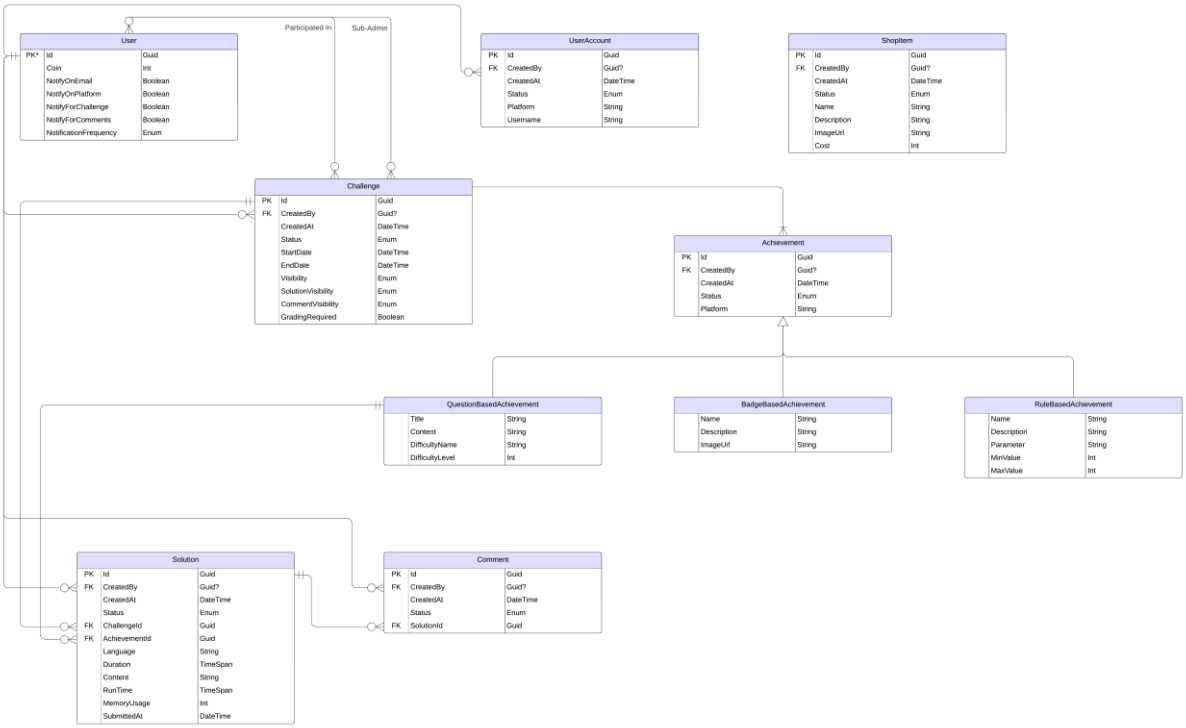

Figure 13. ER Diagram of Database Design

3. User Interface Design

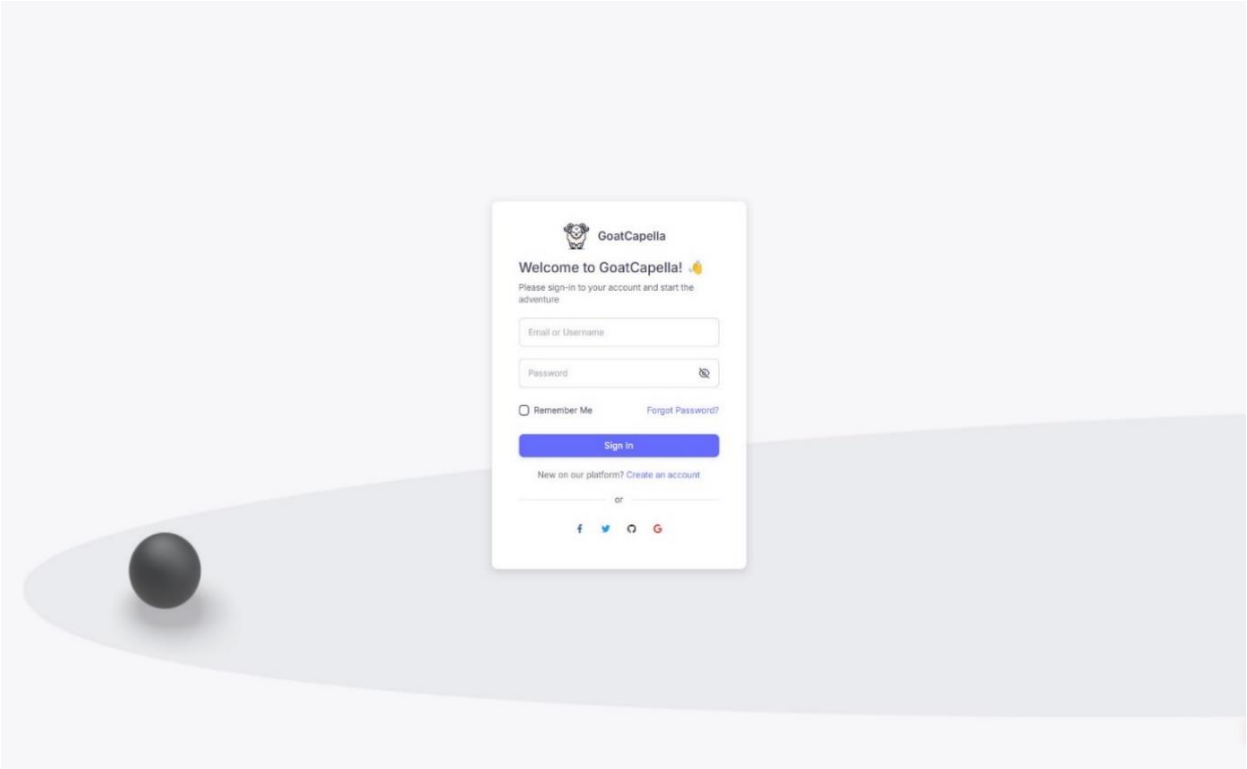

Figure 14. Sign Up Page

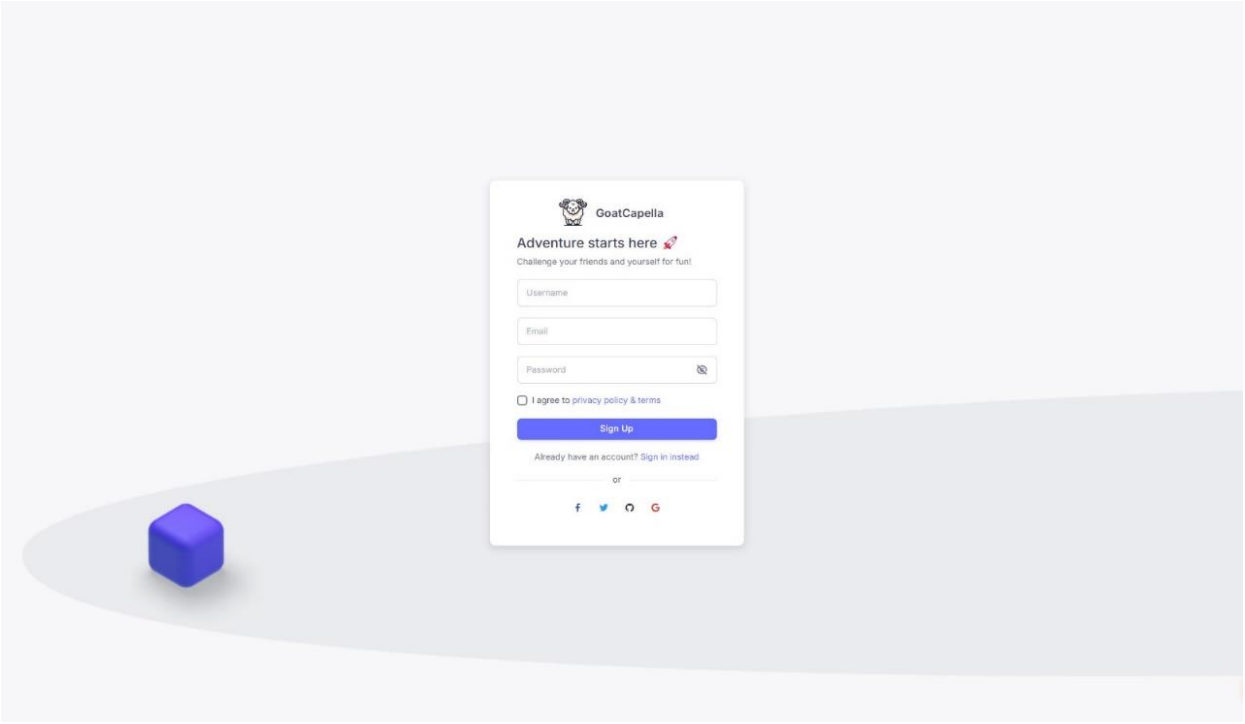

Figure 15. Sign In Page

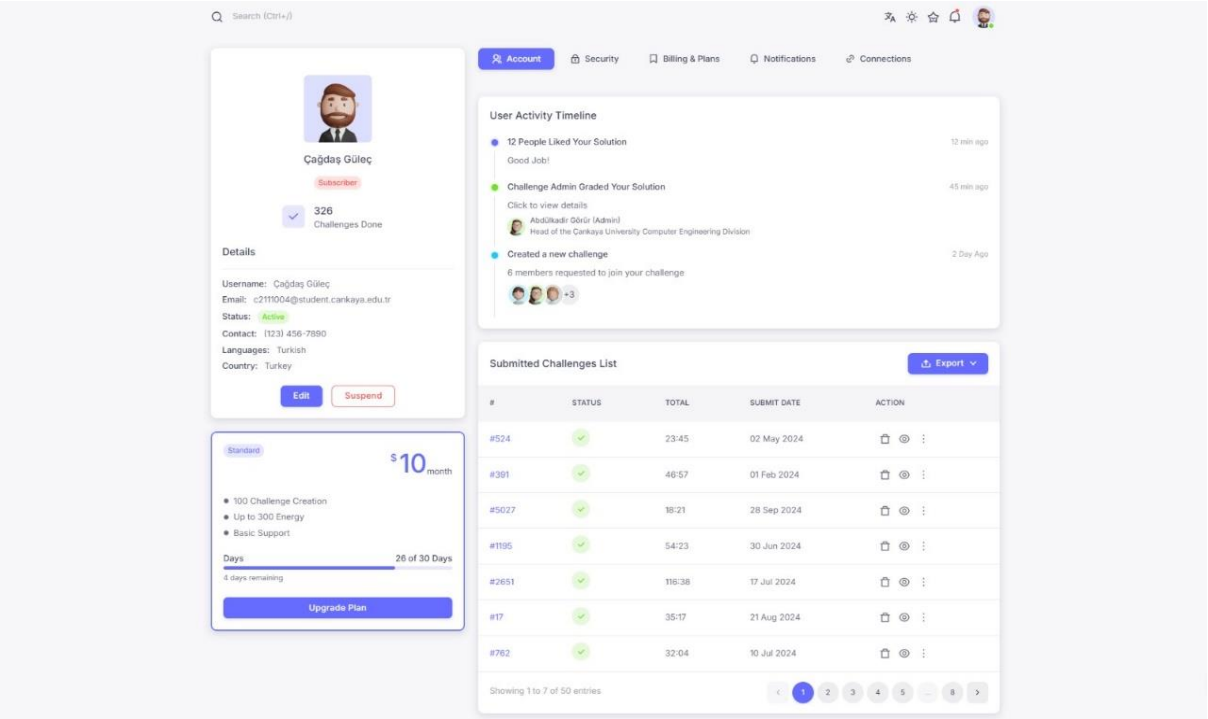

Figure 16. User Profile Page