BLG 411E Software Engineering Term Project

Project: Witch Puzzles

Group: Witch

Members

Ertuğrul Şentürk

Utku Biçer

Hüseyin Şimşek

Oğuzhan Altıntaş

Oğuz Eren Kacar

Lucas Holczinger

Ata Türköz

17.11.2024

CHANGE LOG TABLE

- Add UI mock-ups.
- Add flow diagrams.

Software Requirements Specification (SRS)

1. INTRODUCTION

Goal

The purpose of this document is to outline the functional and non-functional requirements for a puzzle-solving website. This platform is designed to engage users in solving puzzles, tracking their progress, and interacting with other users through a leaderboard and profile features.

Contents and Organization

The document is structured as follows: - System Requirements: Lists both functional and non-functional requirements. - Use Cases: Outlines the various types of users, scenarios, and use cases, with diagrams illustrating the flow of actions within the system.

2. SYSTEM REQUIREMENTS

Functional Requirements

1. **User Registration and Authentication**
    - The system will provide secure user registration, including a unique email requirement.
    - Users will have access to secure login, logout
2. **Puzzle Display and Solving**
    - Users can view puzzles and select their preferred difficulty level.
    - A timer will start with the puzzle, enabling competitive gameplay on leaderboards.
    - Users will be able to submit solutions to the leaderboard.
3. **Leaderboard and User Progress Tracking**
    - The platform will rank users on a leaderboard based on times achieved in puzzle-solving.
    - Users can view their scores and progress in the profile panel.
    - Scores and rankings will be updated in real-time based on solved puzzles and recorded times.
4. **Profile Panel**
    - Users can track their progress and view past performance.
    - The profile page will display completed puzzles and times, along with current rankings.

Non-Functional Requirements

1. **Performance**
    - The system should maintain page load times under 2 seconds under typical load.
2. **Security**

- Sensitive data, including passwords, will be encrypted. All sensitive actions (e.g., login, registration) will use HTTPS.

3. **Usability**
   - The interface should be user-friendly and accessible on both desktop and mobile devices.
4. **Reliability**
   - The platform aims for 99.9% uptime.

3. USE CASES

3.1 User Types

1. **Visitor**
   - Can view available puzzles, view the leaderboard, and register for an account.
2. **Registered User**
   - Can log in, solve puzzles, track progress, view the leaderboard, and update personal information.

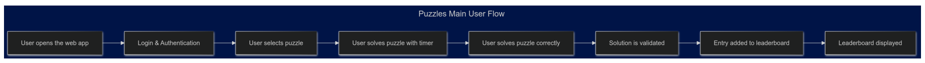
3.2 User Scenarios

1. **Persona: Rümeysa, a casual puzzle enthusiast**
   - Rümeysa logs in and selects an easy puzzle to start. She finishes the puzzle and checks the leaderboard to see where she ranks compared to others.
2. **Persona: Batuhan, a competitive puzzle solver**
   - Batuhan loves challenging puzzles and tries to reach the top of the leaderboard. He completes multiple puzzles of increasing difficulty, tracking his performance in his profile panel.

3.3 Use Case Diagram

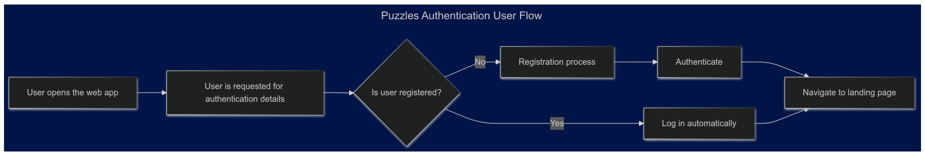
Overall System Use Case Diagram

The main use case diagram will depict interactions between user types and the core functionalities: Registration and Authentication, Puzzle Display and Solving, Progress Tracking, and Leaderboard Viewing.

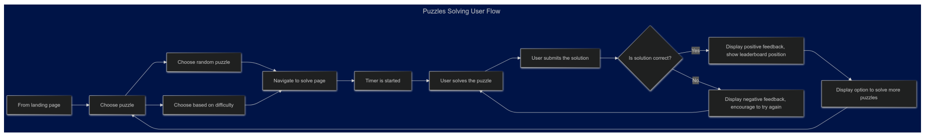


Additional Use Case Diagrams

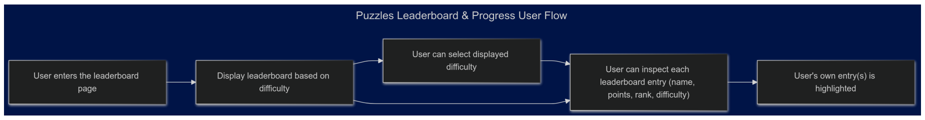
- Diagram for Registration and Authentication



- Diagram for Puzzle Display and Solving


Puzzles Solving User Flow
From landing page
Choose puzzle
Choose random puzzle
Choose based on difficulty
Navigate to solve page
Timer is started
User solves the puzzle
User submits the solution
Is solution correct?
Yes
No
Display positive feedback, show leaderboard position
Display negative feedback, encourage to try again
Display option to solve more puzzles

- Diagram for Leaderboard and User Progress Tracking


Puzzles Leaderboard & Progress User Flow
User enters the leaderboard page
Display leaderboard based on difficulty
User can select displayed difficulty
User can inspect each leaderboard entry (name, points, rank, difficulty)
User's own entry(s) is highlighted

3.4 Use Cases

Use Case 1: User Registration and Login Main Flow 1. User navigates to the registration page. 2. User enters a unique email and password. 3. The system verifies and creates the account. 4. User logs in with credentials, and the system redirects them to the main puzzle page.

Alternative Flow - If the email is already in use, the system notifies the user and prompts for a different email.

Use Case 2: Puzzle Display and Solving Main Flow
1. User selects a puzzle from the list based on difficulty. 2. The puzzle interface loads, and the timer starts. 3. User solves the puzzle and submits the answer. 4. The system validates answers and provides feedback. 5. Upon completion, the system logs the user's time and updates the leaderboard.

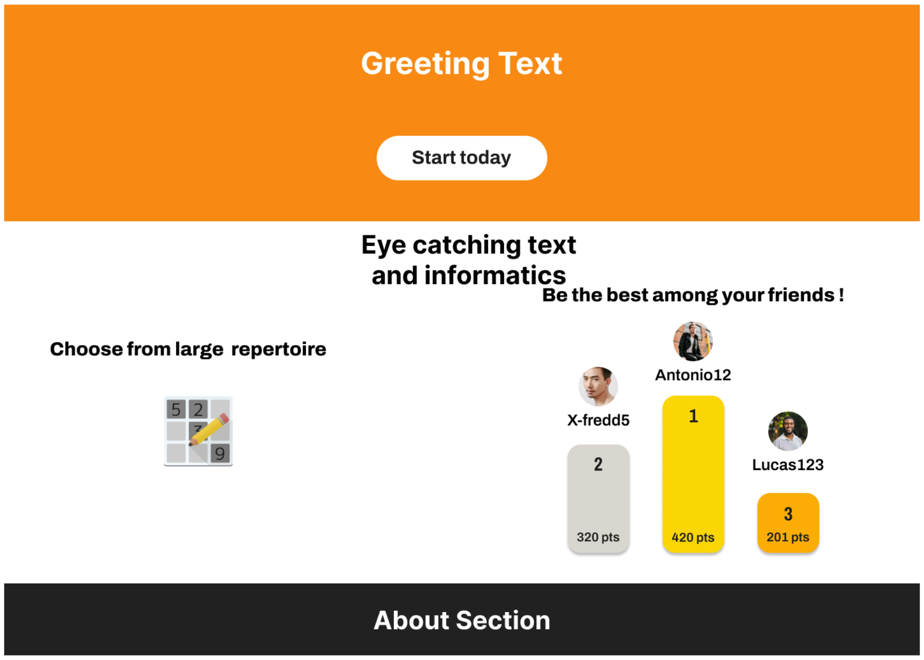
Alternative Flow
- If the user submits an incorrect answer, they are given the option to retry.

Use Case 3: Leaderboard and Progress Tracking Main Flow
1. User views the leaderboard to see their rank based on completion times. 2. The system displays rankings in real-time. 3. User checks their progress in their profile panel.

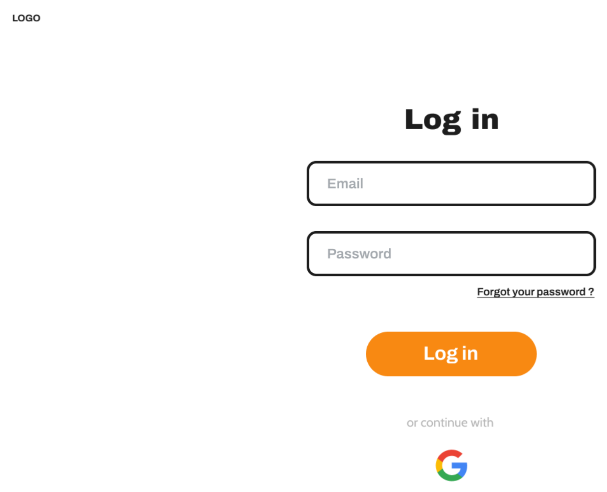
Alternative Flow
- If no puzzles are completed, a prompt will encourage the user to try a puzzle.

4. USER INTERFACE MODEL

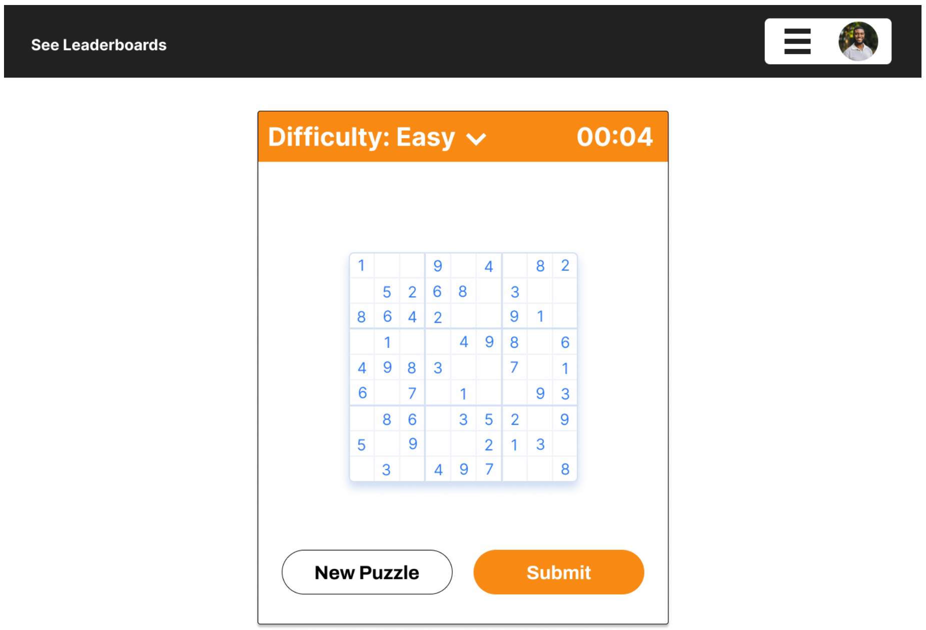
4.1 Landing Page


Greeting Text
Start today
Eye catching text and informatics
Be the best among your friends !
Choose from large repertoire
Antonio12
X-fredd5
Lucas123
1
420 pts
2
320 pts
3
201 pts
About Section

4.2 Login Page

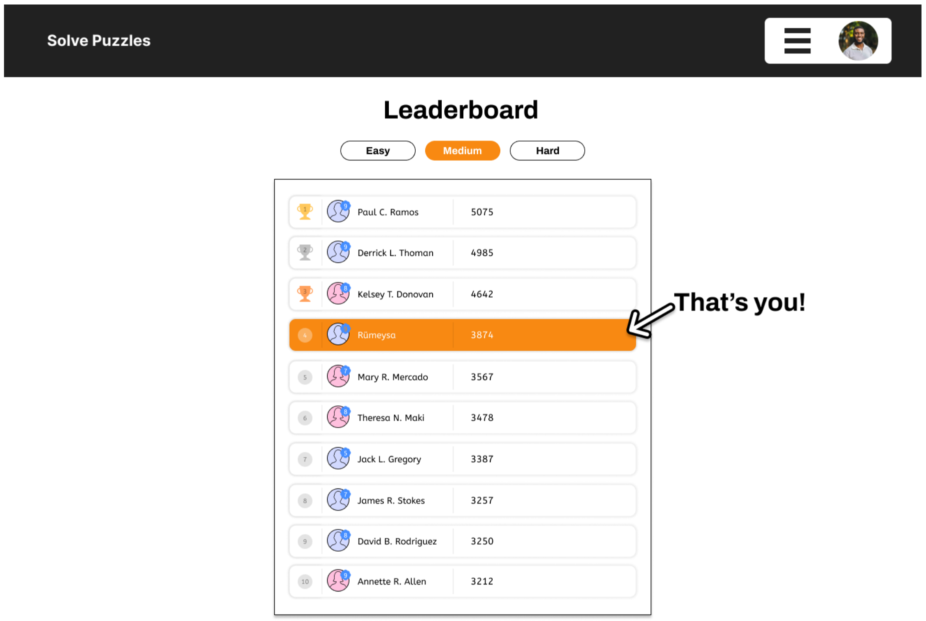


4.3 Puzzle Page



1			9		4		8	2
	5	2	6	8		3		
8	6	4	2			9	1	
	1			4	9	8		6
4	9	8	3			7		1
6		7		1			9	3
	8	6		3	5	2		9
5		9			2	1	3	
	3		4	9	7			8

4.4 Leaderboard Page

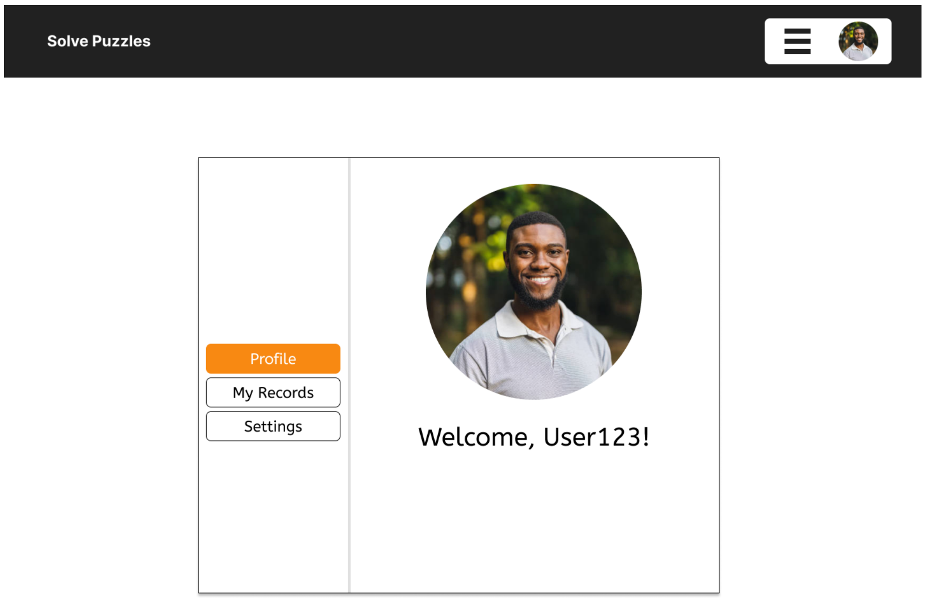


Leaderboard

Easy | Medium | Hard

1	9	Paul C. Ramos	5075
2	9	Derrick L. Thoman	4985
3	8	Kelsey T. Donovan	4642
4	9	Rümeysa	3874
5	7	Mary R. Mercado	3567
6	8	Theresa N. Maki	3478
7	5	Jack L. Gregory	3387
8	7	James R. Stokes	3257
9	8	David B. Rodriguez	3250
10	9	Annette R. Allen	3212

4.5 Profile Page

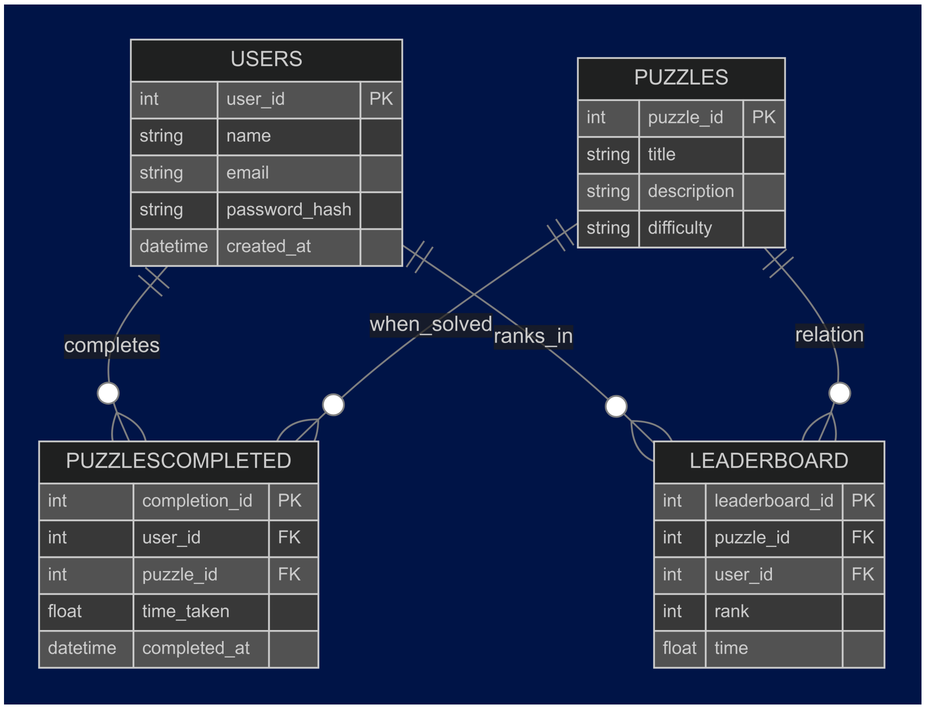


Welcome, User123!

5. FLOW DIAGRAMS

5.1 General Data Model

The entity relationship diagram for our database.



5.2 Important Data Considerations

1. JSON (JavaScript Object Notation)

- *Use Case*: Used to exchange data between the front-end and back-end of the website.
- *Advantages*:
  - Lightweight and easy to parse.
  - Can be stored in SQL databases using JSON or JSONB data types for flexible and semi-structured data storage.
- *Example*: Fetching puzzles or sending user progress updates between client and server.

SQL-Specific Considerations

- *Structured Data: All core entities (e.g., User, Puzzle, Leaderboard) are stored in **SQL tables* with proper relationships.

- *Data Formats*:
  - *JSON*: To store semi-structured data like user preferences or dynamic puzzle metadata.
- *Queries*:
  - SQL will handle structured data operations like fetching leaderboards (SELECT), updating user progress (UPDATE), or inserting new puzzles (INSERT).

Considerations for Choosing a Format

- *Efficiency*: JSON for dynamic or flexible fields; standard SQL tables for structured data.
- *Readability*: JSON is human-readable for semi-structured data, while SQL schemas ensure clear organization for structured data.
- *Ease of Use*: SQL ensures robust data management and relationships, while formats like JSON simplify data interchange.

5.3 Data Flow Diagram

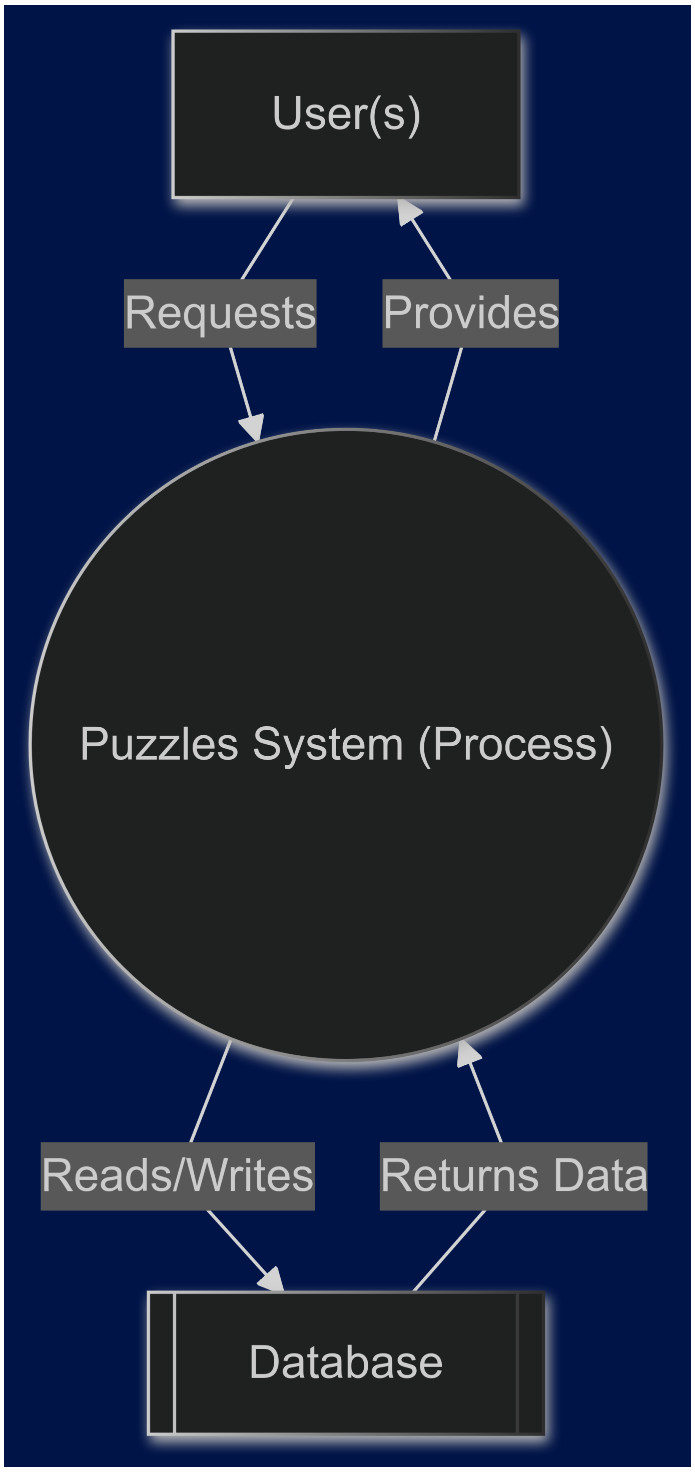
Level 0

External Entities:

- **Users**: The end-users who interact with the platform to register, solve puzzles, and check their standings on the leaderboard.
- **Database**: Represents the data storage for the platform (e.g., user information, puzzle data, leaderboard entries).

Data Flows:

- **Users to System**:
  - Registration/Login details.
  - Puzzle solutions and interactions.
  - Requests for leaderboard or profile data.
- **System to Users**:
  - Response to login or registration attempts.
  - Puzzle data for display.
  - Updated leaderboard or profile information.
- **System to Database**:
  - Stores user credentials and puzzle completion data.
  - Retrieves leaderboard rankings and puzzle metadata.
- **Database to System**:
  - Provides data needed for user requests.


User(s)
Requests
Provides
Puzzles System (Process)
Reads/Writes
Returns Data
Database

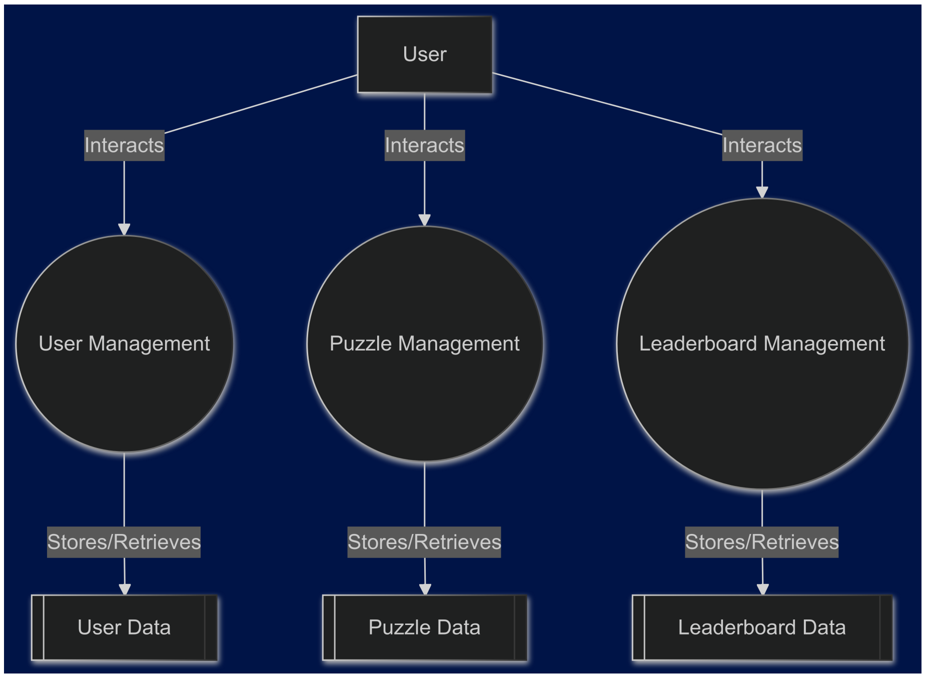
Level 1

Processes:

- **User Management**:
  Handles user-related activities such as:
  - Registration (stores name, email, password hash).
  - Login (validates credentials).
  - Profile updates.
- **Puzzle Management**:
  - Retrieves puzzle data.
  - Validates puzzle solutions.
  - Updates puzzle completion records.
- **Leaderboard Management**:
  - Retrieves leaderboard rankings for specific puzzles.
  - Updates rankings when new completions are logged.

Data Stores:

- **User Data Store**: Stores user information.
- **Puzzle Data Store**: Contains puzzle details.
- **Leaderboard Data Store**: Stores rankings and completion times for puzzles.



Level 2

Step 1: Puzzle Selection

- The user selects a puzzle from a list displayed on the platform.
- The system queries the Puzzle Data Store to fetch puzzle data and displays it to the user.

Step 2: Solution Submission

- The user submits a solution to the selected puzzle.
- The system validates the solution:
  - Ensures the user completed the puzzle within allowed parameters (e.g., time limits, no duplication).

Step 3: Update Records

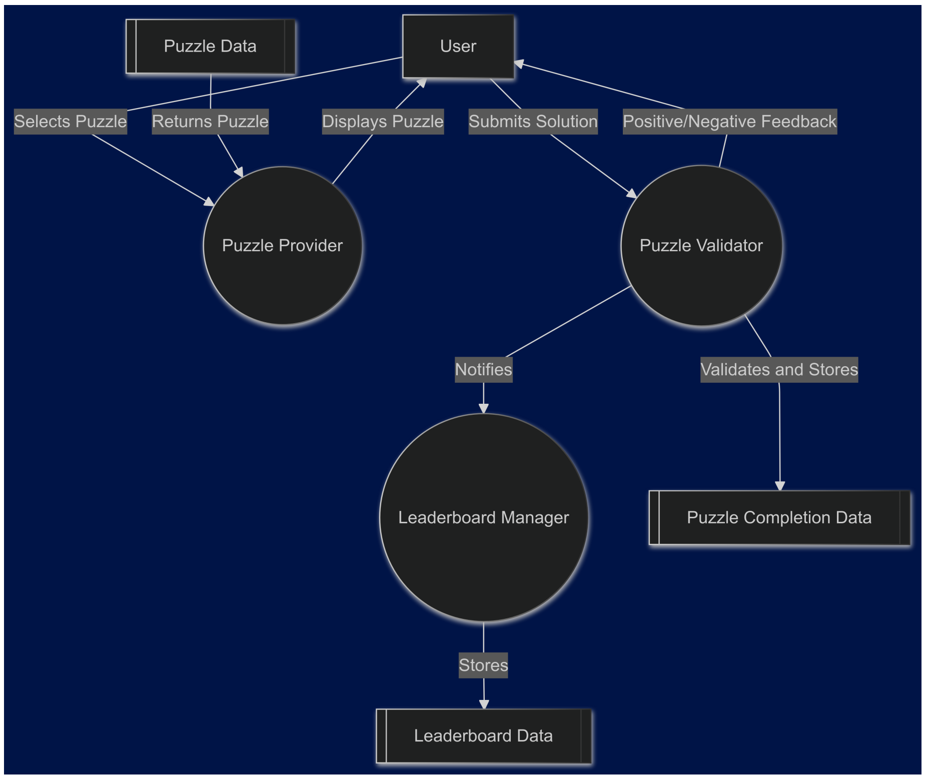
- If the solution is valid, the system:
  - Logs the completion in the `PuzzlesCompleted` Data Store (records `user_id`, `puzzle_id`, `time_taken`, and `completed_at` timestamp).
  - Updates the `Leaderboard` Data Store for the puzzle:
    - Recalculates rankings based on the new completion time.
    - Saves the updated rankings.

Step 4: Leaderboard Retrieval

- The updated leaderboard is retrieved from the Leaderboard Data Store.
- The system formats the leaderboard data and sends it to the user interface.

Step 5: Response to User

- The user sees the leaderboard, including their rank, time, and comparison with others.


Puzzle Data
User
Selects Puzzle
Returns Puzzle
Displays Puzzle
Submits Solution
Positive/Negative Feedback
Puzzle Provider
Puzzle Validator
Notifies
Validates and Stores
Leaderboard Manager
Puzzle Completion Data
Stores
Leaderboard Data