CUBETIMER PRO

PRESENTATION OF THE ADVANCED PROGRAMMING PROJECT

Santiago Andrés Benavides Coral – 20232020036

Programación Avanzada 020-84

Docente: Carlos Andrés Sierra Virguez

Ingeniería de sistemas

Universidad Distrital Francisco José de Caldas

THE BUSINESS MODEL CANVAS

Key Activities	**Key Resources**	**Channels**
• Implementing a database. • Adding more cube types. • Incorporating new statistics. • User engagement and feedback collection.	• FastAPI framework.	• Web platform.
Value Propositions	**Customer Relationships**	**Customer Segments**
• Precision timer for speedcubing, • Automatic scramble generation for different cube types • Automatic stadistics update.	• Feedback loop for continuos improvement.	• Casual speedcubers. • Casual cubers.

USER STORIES

1. **User Registration**
   As a new user; I want to create an account so that I can log in and track my progress over time.
2. **Cube selection**
   Cube a casual cuber, I want to choose the type of cube so that I can time my solves for different type of cubes.
3. **Scramble Generation**
   As a speedcuber, I want to receive a random scramble for my selected cube so that I can practice under standard competition conditions.
4. **Start and Stop timer**
   As a user, I want to start and stop the timer so that I can record the time it takes me to solve the cube.
5. **Automatic Statistics Update**
   As a user, I want my statistics to update automatically after each solve so that I can track my performance easily.
6. **View Past Performance**
   As a user, I want to see a history of my past solves that I can measure my improvement over time.
7. **Log Out**
   As a user, I want to log out of the app so that my session is securely closed.
8. **Error Handling**
   As a user, I want the app to notify me if I make a mistake (e.g., enter the application incorrectly) so that I can correct it quickly and continue using the app.

HOW DO I GET TO THESE USER STORIES?

1. **Identify User Roles:**
   Focused on the main user (the speedcuber) who interacts with the app.
2. **Analyze Core Features:**
   Reviewed the app's core functionality (cube selection, timing, scramble generation, stats).
3. **Prioritize Essential Features:**
   Focused on relevant features like timing, scramble generation, and automatic stats.
4. **Map the User Journey:**
   Traced the steps a user takes from logging in, chossing a cube, solving, and viewing results.
5. **Define User Goals:**
   Created stories by linking each feature to a real users goal and its benefit.

APPLICATION PROCESS AND COMPONENTS

COMPONENTS

1. FRONTEND

   - **Timer component** → Handles the start/stop functionality of the timer and updates resolution times for statistics.
   - **Cube selection component** → Allows users to choose between different cube types (2x2, 3x3, 4x4).
   - **Scramble display component** → Shows the scramble sequence.
   - **Statistics component** → Displays and updates performance metrics automatically after each solve.
   - **User interface component** → Manages user interactions and navigation within the application.
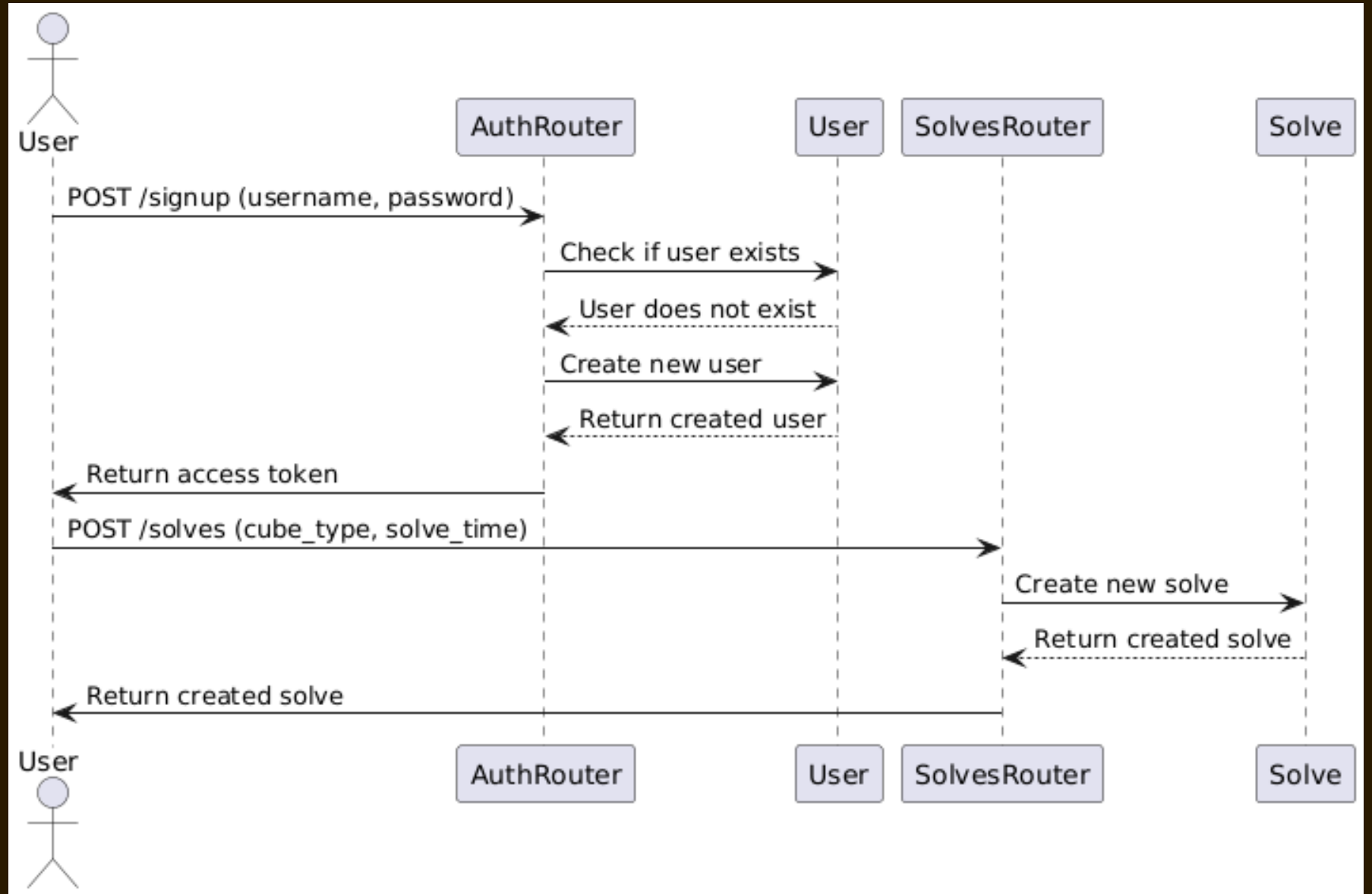
2. BACKEND

   - **Authentication component** → Handles user signup, login, password hashing, authentication, and token generation.
   - **Scramble component** → Generates random scramble sequences for 2x2, 3x3, and 4x4 cubes, ensuring valid moves.
   - **Solves component** → Manages solve submissions, retrieves user solve history, and calculates statistics.
   - **Database component** → Stores and retrieves user data, solves, and authentication credentials.
   - **Dependencies component** → Provides utility functions for password hashing, token creation, and user authentication.
   - **Schemas component** → Defines data models and request/response validation for API endpoints.
   - **Config component** → Stores environment variables and application settings like token expiration time.

APPLICATION PROCESS AND COMPONENTS

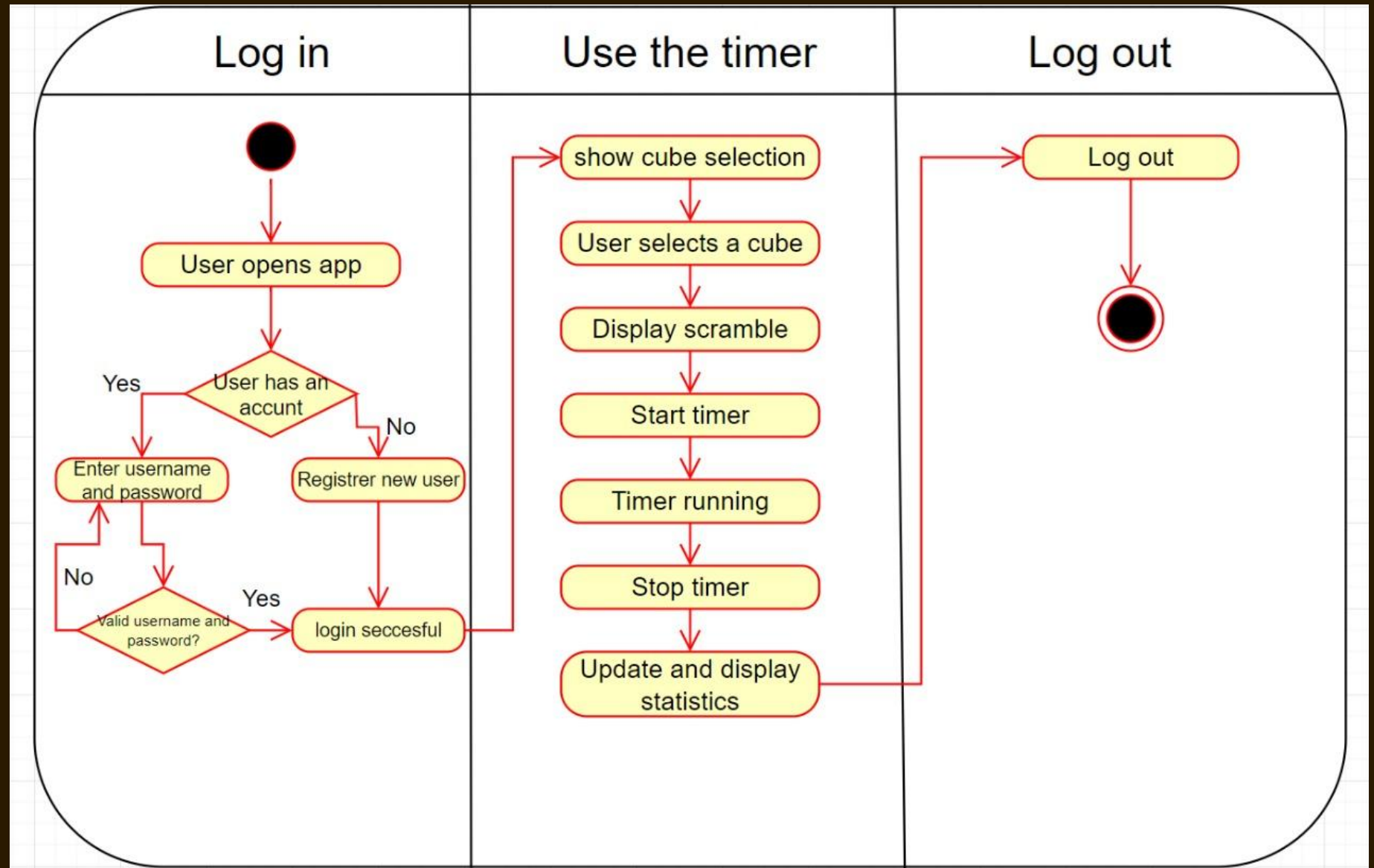**PROCESSES**

- **Timer Start and Stop** → The user starts and stops the timer to record solve times.
- **Statistics Update** → After the user stops the timer, the app automatically updates the user's statistics.
- **View Solve History** → The user can Access their past solve times to track progress.
- **Log Out** → The user can log out of the application.

SEQUENCE DIAGRAM

ACTIVITY DIAGRAM

CLASS DIAGRAM

AuthRouter

+ signup(user_create: UserCreate): Token
+ login(form_data: OAuth2PasswordRequestForm):
Token

ScrambleRouter

+ get_scramble(cube_type: string, length: int): Scramble

Interacts with

SolvesRouter

+ create_solve(solve: SolveCreate, current_user: dict):
SolveOut
+ get_solves(current_user: dict): List[SolveOut]
+ get_stats(current_user: dict): List[Stats]

User

- int id
- String username
- String hashed_password

+ get_user_by_username(username: String): User
+ create_user(username: string, hashed_password:
string): User

Scramble

- List[string] faces
- List[string] modifiers

+ generate_scramble(cube_type: string, length: int):
string

Interacts with

Interacts with

1

0..*

Solve

- int id
- string cube_type
- float solve_time
- string timestamp
- string username

+ add_solve(username: string, cube_type: string,
solve_time: float): Solve
+ get_solves_for_user(username: string): List[Solve]