Universidad Nacional de Colombia
School of Engineering
Course: Software Engineering II

Alvaro Andres Romero Castro
Baruj Vladimir Ramirez Escalante
Brayan Alejandro Munoz Perez
Jenny Catherine Herrera Garzon
Juan David Ladino Triana

Workshop No. 2

1. CRC Cards

User	
• Register and authenticate the user. • Access and manage active habits. • View progress and achievements. • Maintain basic data (email, display name, creation date).	• AuthService • UserHabitStats • StatsWeekly • UserAchievement

AuthService	
• Create new user accounts. • Validate credentials and handle sign-in/sign-out. • Generate and verify JWT tokens. • Hash and manage passwords securely.	• User

Habit	
• Represent a predefined habit from the catalog. • Store category, name, and default frequency.	• UserHabit

● Serve as a template for user habits.	

UserHabit	
● Register which habits are active for each user. ● Store custom frequency and target time. ● Allow activation/deactivation of habits. ● Act as the link between *User* and *Habit*.	● User ● Habit ● Checkin

Checkin	
● Record whether a habit was completed or missed on a specific date. ● Prevent duplicate daily entries. ● Serve as the basis for statistics and achievements.	● UserHabit ● GamificationService

StatsWeekly	
● Calculate weekly completion percentage. ● Track the user’s global streak. ● Provide progress data for the dashboard.	● User ● Dashboard

Achievement	
● Define achievement rules and conditions (e.g., 7 days in a row, ≥70% weekly rate). ● Store achievement code, name, and type.	● UserAchievement ● GamificationService

UserAchievement	
● Associate unlocked achievements with users. ● Store the unlock date.	● User ● Achievement ● Dashboard ● GamificationService

GamificationService	
● Evaluate conditions for achievements and symbolic levels. ● Assign consistency points. ● Unlock achievements when rules are met. ● Notify users about new rewards.	● Checkin ● Achievement ● UserAchievement

Dashboard	
● Display weekly progress, streaks, and achievements. ● Retrieve user statistics and achievements. ● Render visual indicators (percentage, streak, symbolic level).	● StatsWeekly ● UserAchievement

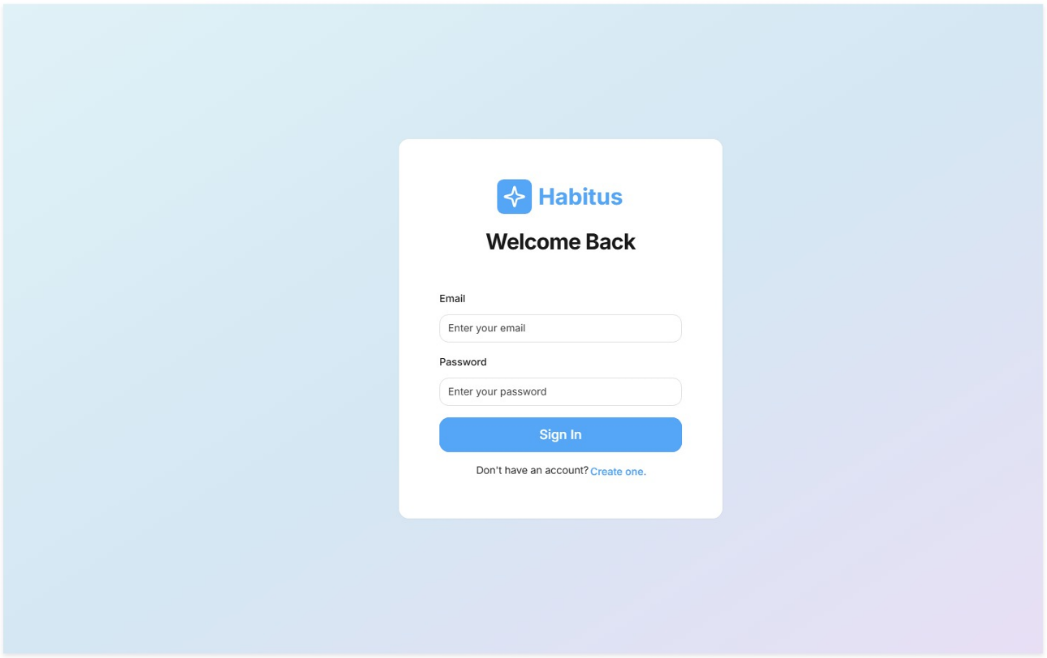
StreakAchievement	
● Define achievements based on consecutive successful check-ins. ● Evaluate streak length required for unlocking (e.g., 7-day streak).	● GamificationService ● UserAchievement

ConsistencyAchievement	
● Define achievements based on average completion percentage over a week. ● Evaluate if user's weekly completion rate ≥ threshold (e.g., 70%).	● GamificationService ● UserAchievement

2. Mockups

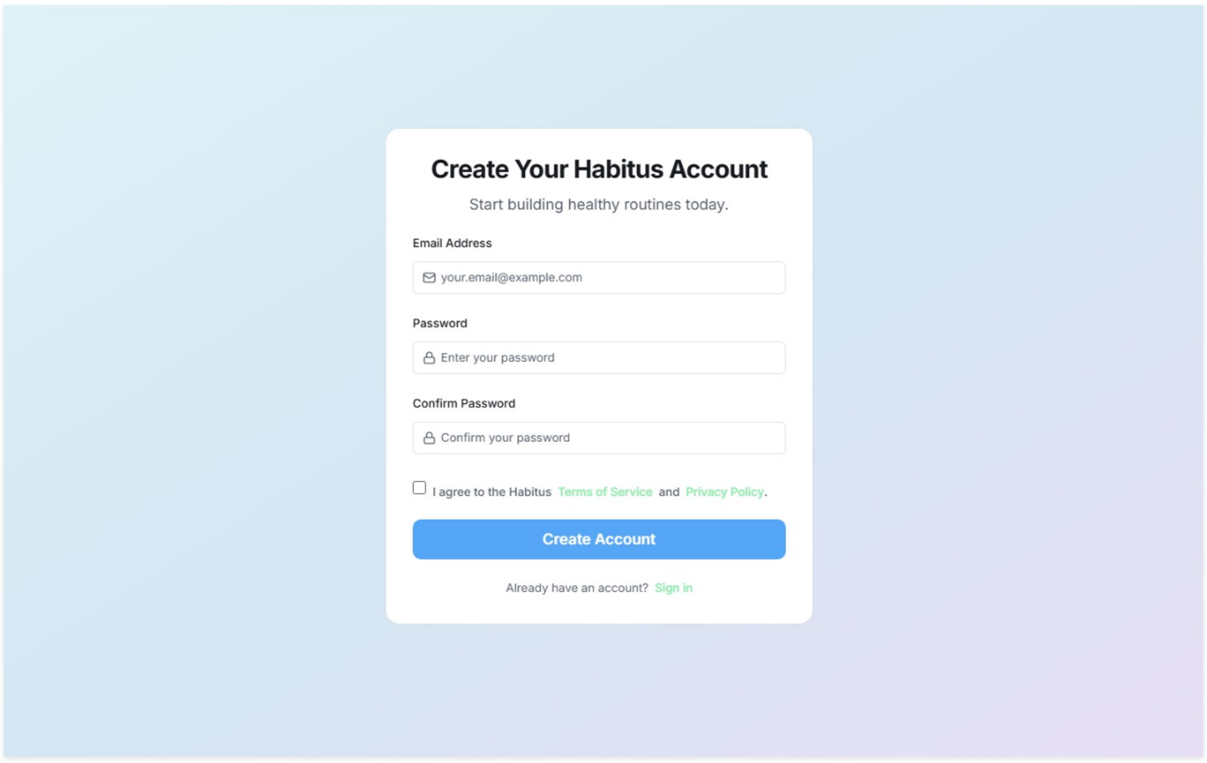
The following mockups represent the main user interfaces of the **Habitus** web application.
These screens were designed using **Visily** and reflect the user flow defined in Workshop 1: registration, habit selection, daily tracking, progress visualization, and symbolic achievements. Each screen is briefly explained below.

2.1. Login Screen



Purpose:
This screen allows registered users to **log in** using their email and password.
Its purpose is to provide a **secure entry point** to the system, granting access to the user's personalized habits and progress data.
It also includes navigation options for creating a new account or recovering a forgotten password.

2.2. Registration Screen



Create Your Habitus Account

Start building healthy routines today.

Email Address

your.email@example.com

Password

Enter your password

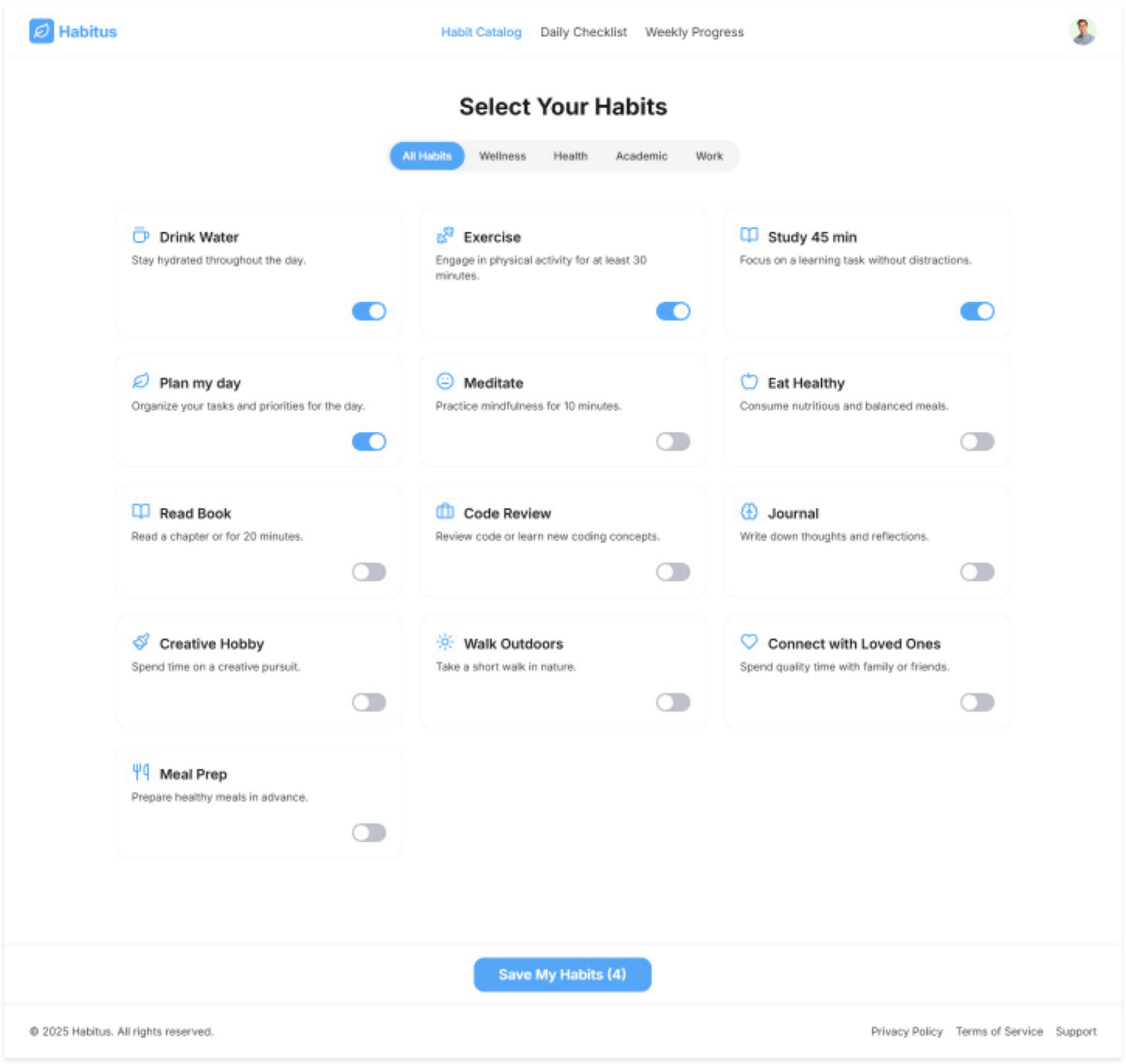
Confirm Password

Confirm your password

I agree to the Habitus Terms of Service and Privacy Policy.

Purpose:
The registration screen enables new users to **create an account** by entering their email and setting a password.
Its purpose is to **add new users** to the system safely and easily, starting their journey toward building consistent habits within the Habitus platform.

2.3. Habit Catalog Screen



Select Your Habits

Drink Water — Stay hydrated throughout the day. [✓ on]

Exercise — Engage in physical activity for at least 30 minutes. [✓ on]

Study 45 min — Focus on a learning task without distractions. [✓ on]

Plan my day — Organize your tasks and priorities for the day. [✓ on]

Meditate — Practice mindfulness for 10 minutes.

Eat Healthy — Consume nutritious and balanced meals.

Read Book — Read a chapter or for 20 minutes.

Code Review — Review code or learn new coding concepts.

Journal — Write down thoughts and reflections.

Creative Hobby — Spend time on a creative pursuit.

Walk Outdoors — Take a short walk in nature.

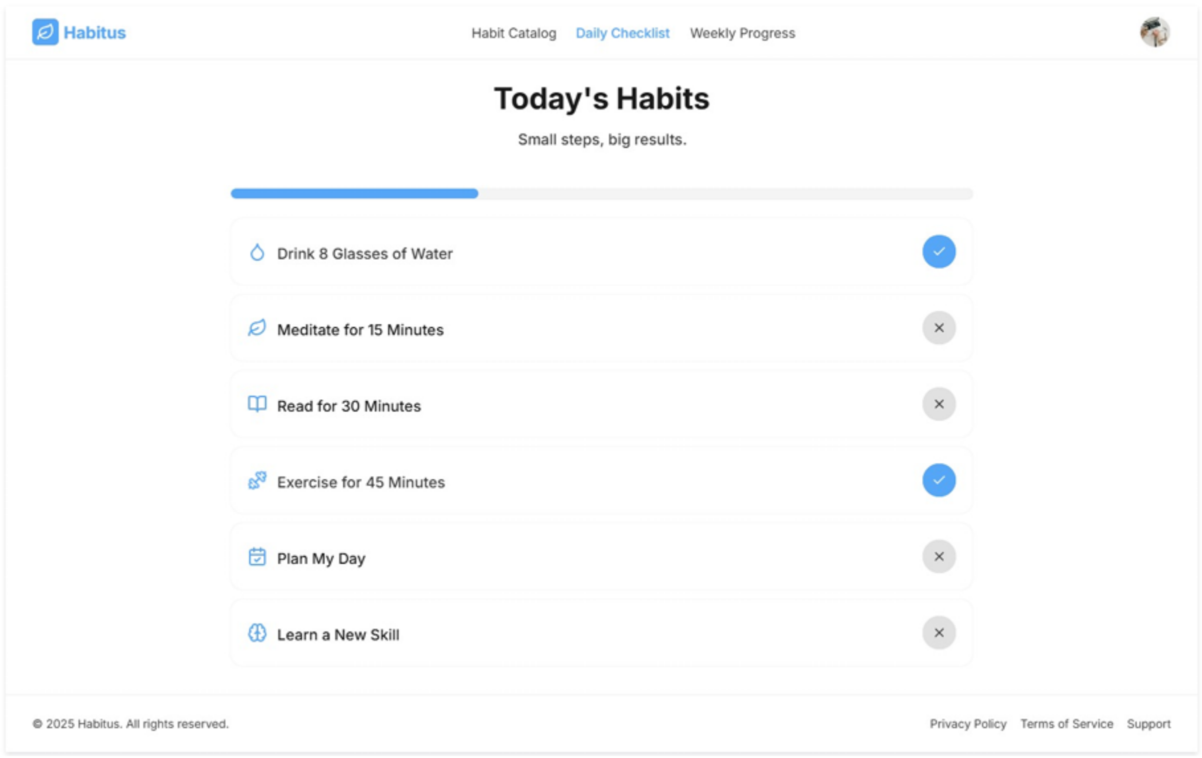
Connect with Loved Ones — Spend quality time with family or friends.

Meal Prep — Prepare healthy meals in advance.

© 2025 Habitus. All rights reserved.

Purpose:
This screen displays a list of **predefined habits** organized by categories such as wellness, health, academic, and work.
Users can **activate or deactivate** the habits they want to follow.
Its purpose is to allow **initial personalization**, letting users select the routines they will track daily.

2.4. Daily Tracking Screen



Today's Habits

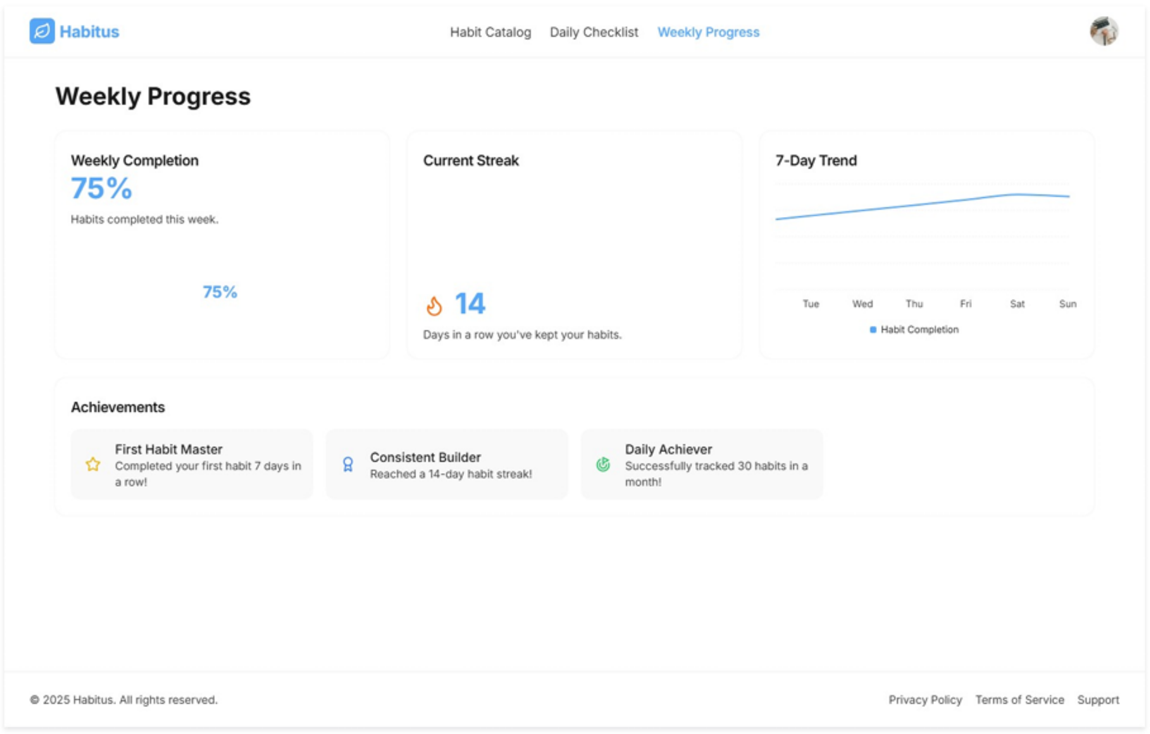
Small steps, big results.

- Drink 8 Glasses of Water ✓
- Meditate for 15 Minutes ✕
- Read for 30 Minutes ✕
- Exercise for 45 Minutes ✓
- Plan My Day ✕
- Learn a New Skill ✕

© 2025 Habitus. All rights reserved.

Purpose:
The daily tracking view shows the user’s **active habits** for the current day with options to mark each as **“Completed”** or **“Missed.”**
A progress bar reflects the percentage of completed habits.
Its purpose is to provide a **quick and intuitive interface** for daily habit check-ins.

2.5. Weekly Progress and Achievements Screen



Weekly Progress

Weekly Completion
75%
Habits completed this week.
75%

Current Streak
14
Days in a row you've kept your habits.

7-Day Trend

Achievements

First Habit Master
Completed your first habit 7 days in a row!

Consistent Builder
Reached a 14-day habit streak!

Daily Achiever
Successfully tracked 30 habits in a month!

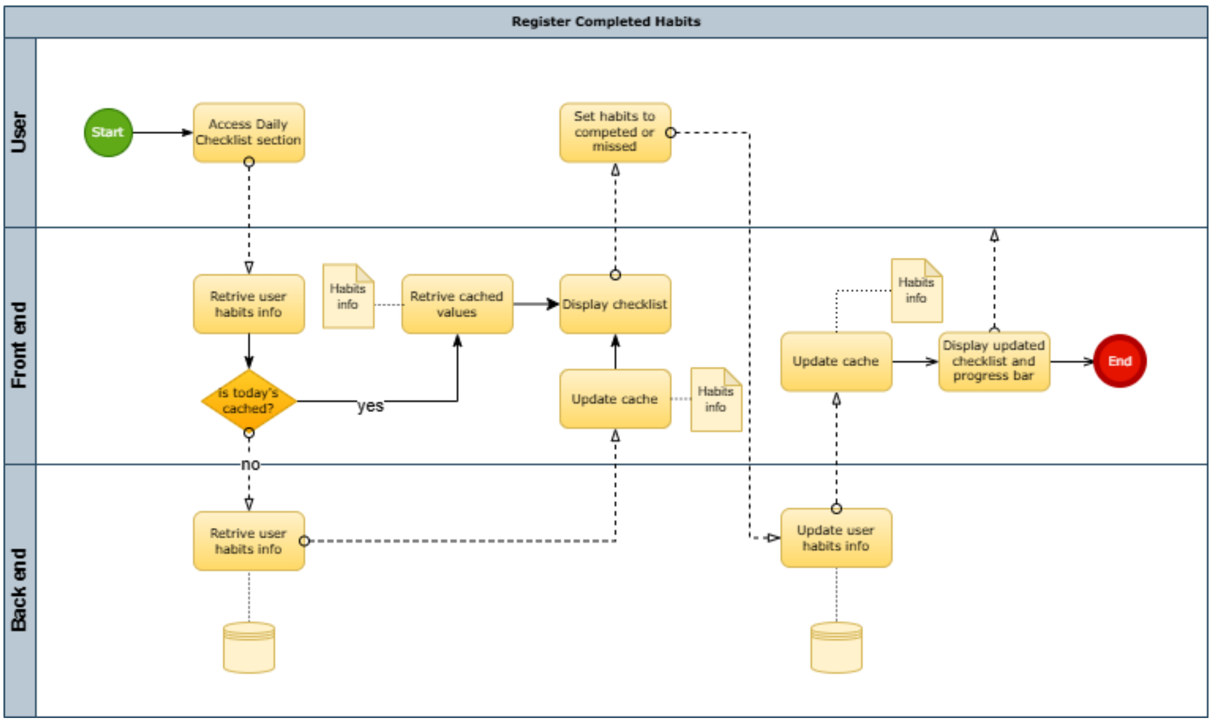
© 2025 Habitus. All rights reserved.

Privacy Policy Terms of Service Support

Purpose:
This combined screen provides users with a **comprehensive overview of their weekly progress and achievements** in one place.
It displays the **weekly completion rate**, **current streak**, and a **7-day trend chart** to visualize consistency over time.
Additionally, it showcases **symbolic achievements**—badges such as *First Habit Master*, *Consistent Builder*, or *Daily Achiever*—to celebrate milestones and motivate continuous engagement.
Its purpose is to **encourage reflection and reward progress**, helping users maintain motivation through clear and positive visual feedback.

3. Business Model Process of Daily Check-In

BPMN Diagram:



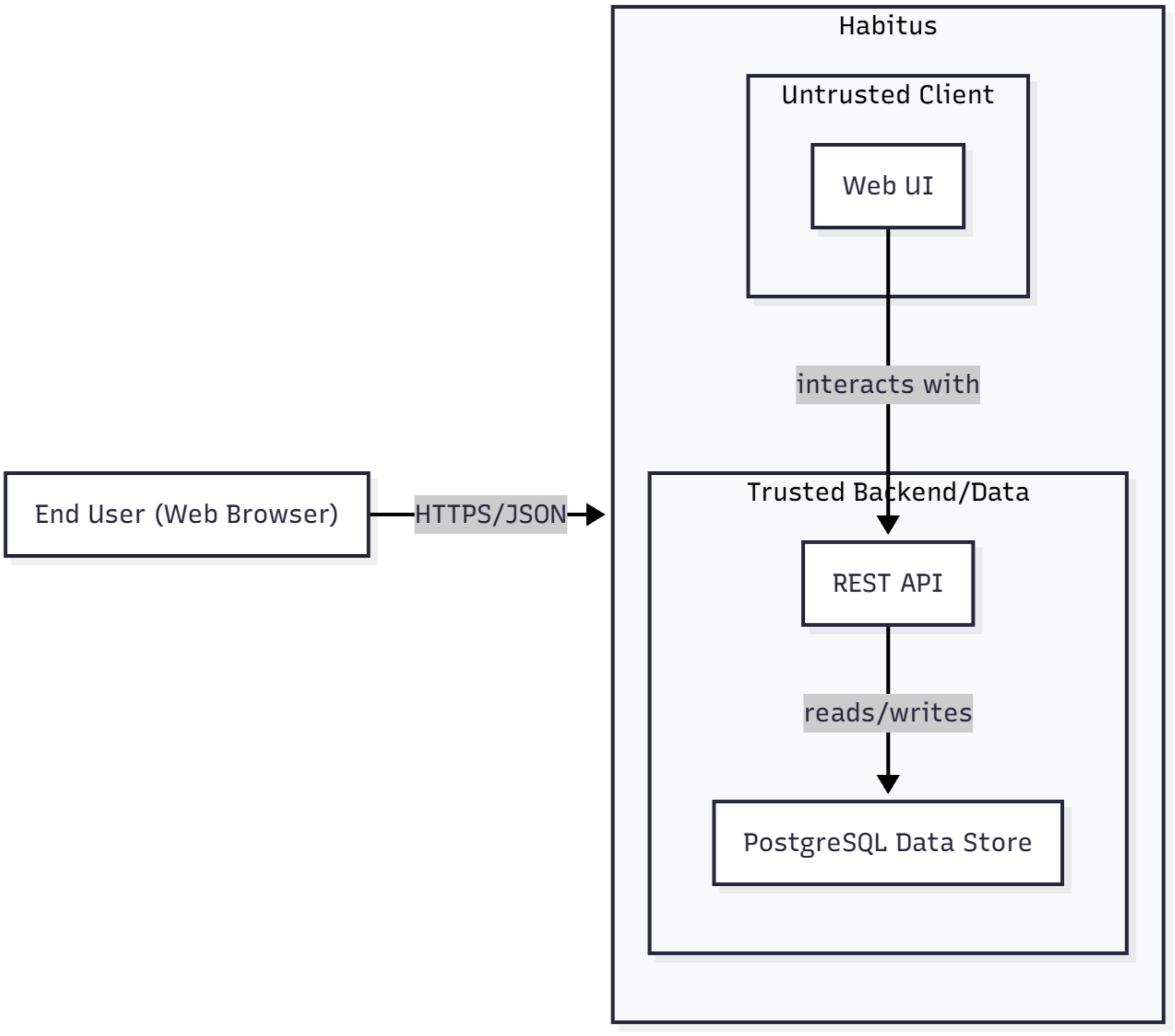
Process description:

This process allows the user to register the habits they have completed through the day, based on the habits they have decided to follow. With this information the system may track the users progress and award rewards based on their performance.

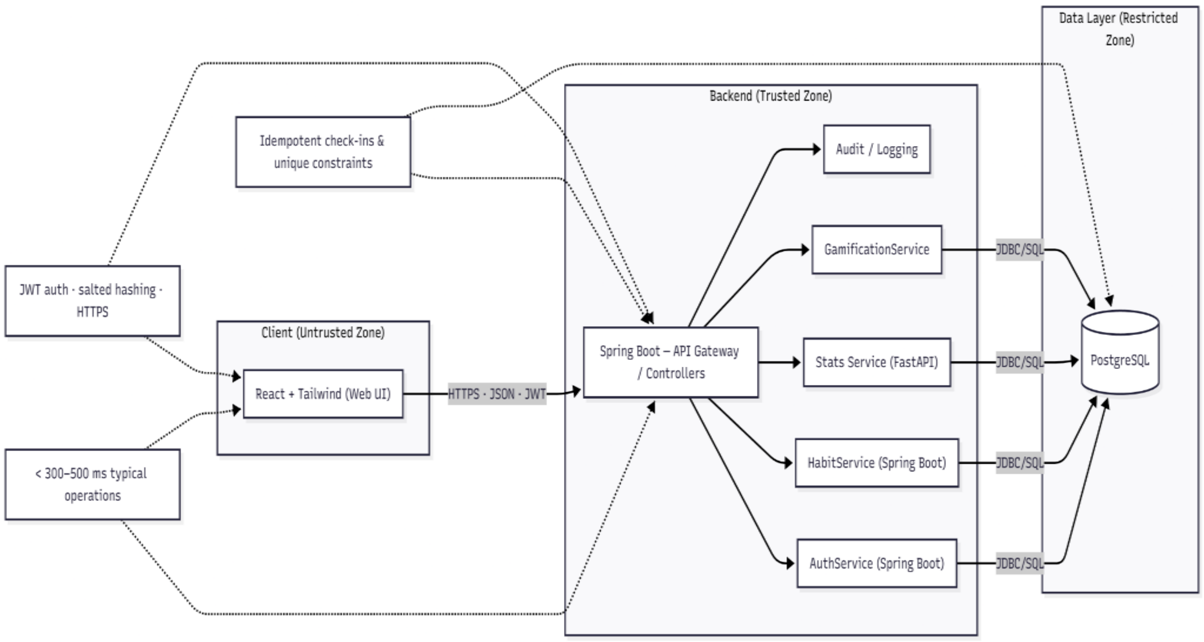
It essentially operates to capture the user's progress by giving them a checklist. This allows for an intuitive interface with the user, and easy parsing of information for the system.

4. Architecture Diagram

4.1 Context (C1)



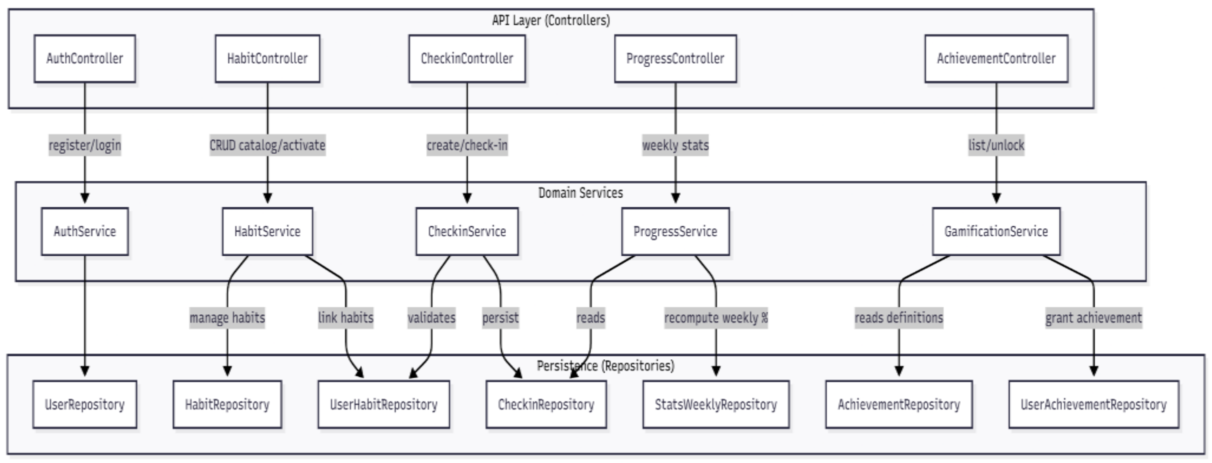
4.2 Container / Layered View


Data Layer (Restricted Zone)
Backend (Trusted Zone)
Idempotent check-ins & unique constraints
Audit / Logging
GamificationService
JWT auth · salted hashing · HTTPS
Client (Untrusted Zone)
Spring Boot – API Gateway / Controllers
Stats Service (FastAPI)
PostgreSQL
React + Tailwind (Web UI)
HTTPS · JSON · JWT
< 300–500 ms typical operations
HabitService (Spring Boot)
AuthService (Spring Boot)
JDBC/SQL
JDBC/SQL
JDBC/SQL
JDBC/SQL

Notes.

- API Gateway/Controllers terminate TLS, validate JWT, and route to domain services.
- AuthService handles registration/sign-in, password hashing (salted), and token issuance.
- HabitService manages the catalog and user activations; enforces uniqueness.
- Stats Service (FastAPI) computes weekly completion % and streaks (simple CPU-bound calc).
- GamificationService evaluates rules (7-day streak, ≥70% weekly, etc.) and stores badges.
- PostgreSQL stores users, habits, check-ins, weekly stats, and achievements.
- Audit/Logging centralizes request logs, security events, and domain audits.

4.3 Component View (C3 — backend slice)



4.4 Security & Quality Alignment

Security: HTTPS end-to-end; JWT for auth; salted password hashing; least-privilege DB roles.

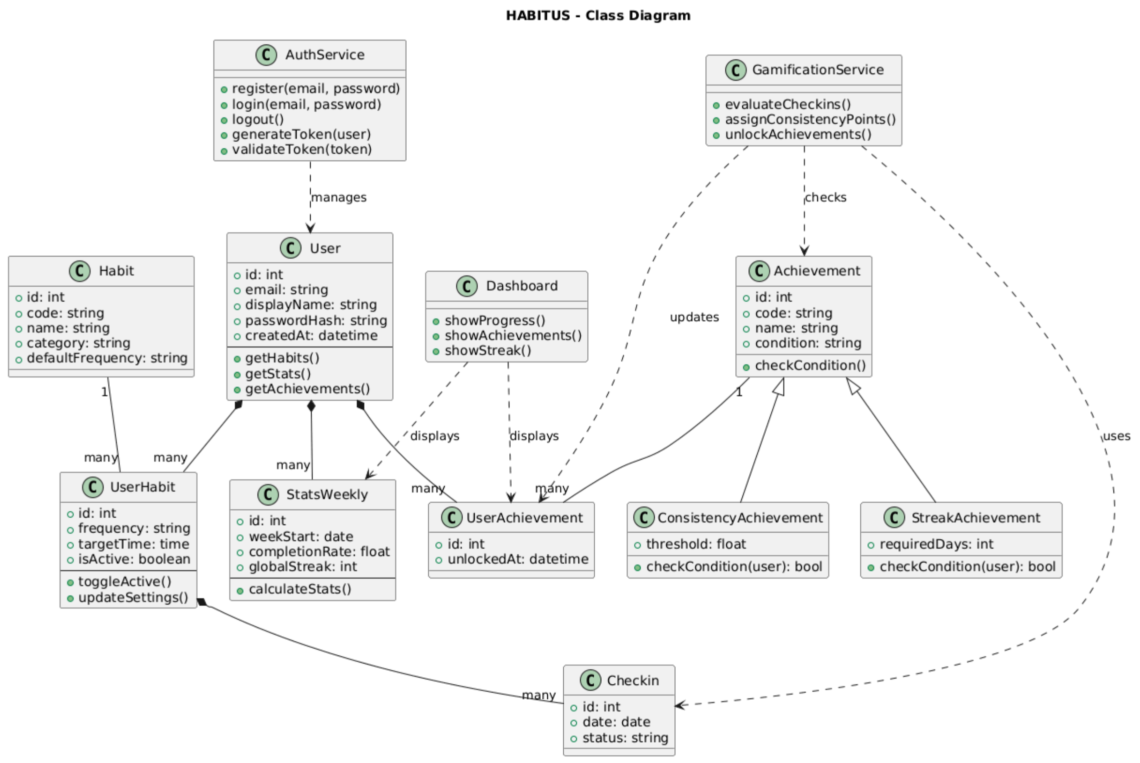
Reliability: idempotent check-ins; unique constraints per (user_habit, date); transactions around write flows.

Performance: cache weekly stats; target <300–500 ms for common operations; lightweight services.

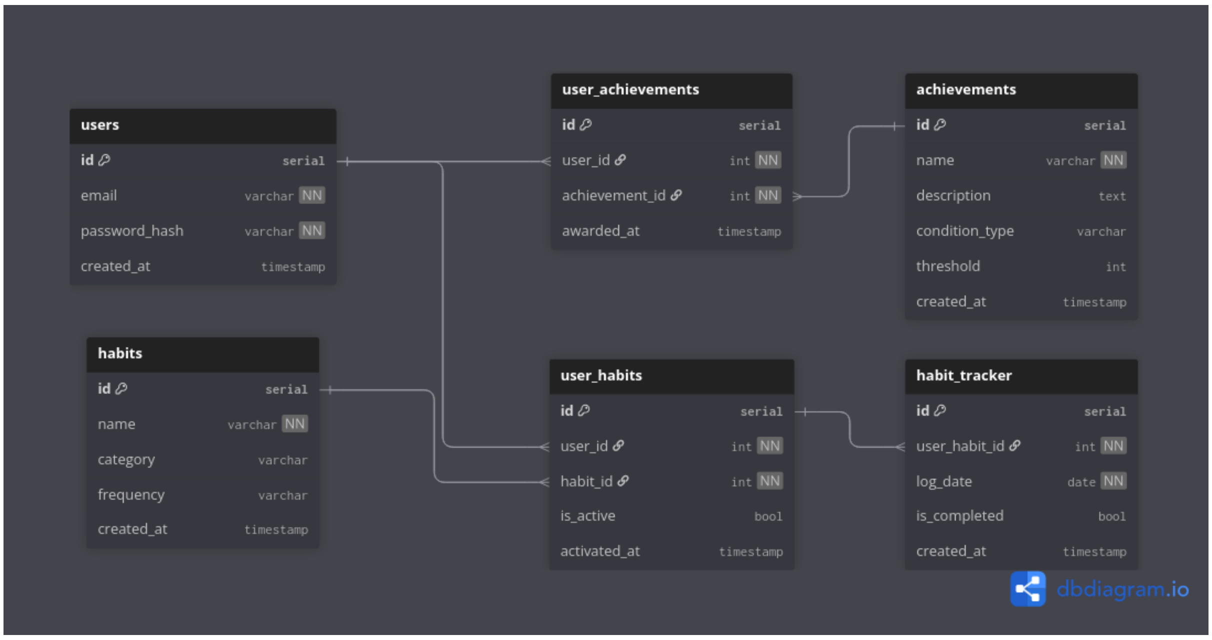
Scalability: stateless API (horizontal scaling); DB normalized with proper indexes.

Maintainability: layered boundaries enable unit tests and independent evolution (services).

5. Class Diagram



6. Relation Database Model



User: Represents each registered person who uses the Habitus system.
Stores authentication and basic account data.
User_achievements: Represents the achievements that a user has already unlocked.
Acts as a junction table linking users and achievements.
Achievements: Defines all possible achievements that can be unlocked in the system.
Each one specifies a condition (like “7-day streak” or “70% weekly consistency”).
Habits: Stores the catalog of predefined habits that users can choose from. Each habit includes its category (e.g., health, academic, productivity) and frequency (daily or weekly).
User Habits: Links a user to a habit they decided to follow.
It indicates which predefined habits each user is actively tracking.
Habit_tracker: Records whether a user completed a habit on a specific day (or week). Used to calculate progress, streaks, and achievement conditions.