**Campus Recycling**  
**Description:** Users need to collect recyclables and go to the designated recycling station on campus, and scan the QR code to sign in. After the back-end verification check-in, the task is completed and the user receives points, which can be completed up to one time per day.

**User Story (if applicable):**  
*"As a player, I want to scan a QR code at the recycling station, so that I can complete my cycling task and earn points."*

**1. Acceptance Criteria**

Define what needs to be met for this item to be considered **complete**.

✅ The user enters the game interface to view the specified recycle bin location.

✅ The user clicks the “SCAN” button to enter the scan code interface.

✅ The user scans the QR code, and completes the task after system verification.

✅ The user can earn 10 points for completing the task.

✅ The user can complete the scan at the most once a day. And the repeated scanning is invalid.

✅ If the user succeeds in scanning, a success message will be displayed. If the user fails to scan, a failure message will be displayed (eg, “QR code is invalid, please try again”).

✅ If the user scans the code but has completed the task today, the system will display “Task completed, come back tomorrow” prompt to prevent duplicate submissions.

✅ [Option1]: After the user scans the code, a prompt pops up, and automatically jumps back to the game interface after 3 seconds. (Optional: If the code scanning fails, the “Try again” button is displayed in the pop-up window. The user can click the button to return to the code scanning page.)

[Option2]: After the user scans the code, a prompt pops up, and the “Return” button appears at the same time. The user can return to the game interface (the interface with the “SCAN” button) by clicking the “Return” button.

**2. Requirements & Specifications**

Provide clear **functional** and **non-functional** details.

* **Functional:**

1. Task flow

The user enters the game screen, which displays the address of the designated recycling station on campus.

The user clicks the “SCAN” button to enter the scan code interface.

After the scan is successful, the system sends a sign-in request to the back-end. After the backend verification is successful, the task is completed, the points are included in the user’s total points, and the leaderboard is updated.

1. Scan code verification

By default, the camera opens automatically.

QR code recognition: If the user successfully scans, there will be a scanning success prompt and display 10 points; If the scan fails, a scanning failure message (eg, “QR code is invalid, please try again”) is displayed. If the user scans the code but has completed the task today, the system will display “Task completed, come back tomorrow”.

[Option1]: The user can return to the game screen after clicking “Return”.

[Option2]: After 3 seconds, it will automatically jump back to the game interface. (Optional: If the code scanning fails, the “Try again” button is displayed in the pop-up window. The user can click the button to return to the code scanning page.)

1. Result calculation

The user can earn 10 points for completing the task.

The user can complete this task only once a day.

The system stores user sign-in records for code scanning.

1. Data submission

POST /api/complete\_task

{user\_id, task\_id, verification, timestamp}

1. Leaderboard update

GET /leaderboard/recycling

* **Non-functional:**

Scanning identification time <1 seconds.

UI adapts to the mobile terminal.

QR code scanning must be stable and stalling free.

**Optional:** Include wireframes or process flow diagrams.

**3. Dependencies & Constraints**

* Task API gets game task information.
* QR code API for check-in verification.
* Database storage score.
* Leaderboard API gets ranking data.
* The back-end needs to record the user task completion status.

**4. GWT**

Provide Given-Then-When for main cases and edge cases.

**Scene1:** User successfully scanned check-in

**Given:** The user enters the page for scanning codes.

**When:** The user scans the correct QR code. And the system recognizes the QR code and verifies it successfully.

**Then:** The message indicating that the scanning task is completed successfully is displayed. The user obtains 10 points and the system submits the task data. (After 3 seconds, the page will automatically jump back or the user will click the “Return” button.)

**Scene2:** User fail to scan the code

**Given:** The user enters the page for scanning codes.

**When:** The user scans the QR code provided by the non-specified recycling station or the scan code is incomplete.

**Then:** The message “QR code is invalid, please try again” is displayed, and a “Try again” button is provided. (or automatically jump in three seconds)

**Scene3:** User try to complete the task repeatedly

**Given:** The user has successfully completed this task once today.

**When:** The user scans code again.

**Then:** The system displays “Task completed, come back tomorrow” prompt. (After 3 seconds, the page will automatically jump back or the user will click the “Return” button.)