**Campus Recycling**  
**Description:** Users participate in campus sustainability initiatives by collecting recyclables and storing them at designated recycling stations. The challenge not only rewards recycling behavior, but also educates users on the impact of reducing waste. Users will not only receive points for successfully recycling items, but will also see how their recycling actions contribute to the planet's environment.

**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 once per recycling station per day. Repeated scans at the same station on the same day are invalid.

✅ If the user succeeds in scanning, a success message will be displayed “You’ve earned 10 points for recycling! Every bin helps save our planet”, let users feel that their recycling behavior has a positive impact on the earth's environment.

If the user fails to scan, a failure message will be displayed “Invalid QR Code”.

✅ If the user scans the code from a station they have already scanned today, the system will display “Already Scanned” prompt to prevent duplicate submissions.

✅ After the user scans the code, a prompt pops up, and the “Return home” 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. Users can click the Recycle Bin button to see the location of the site.

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

Users must enable the camera rights.

QR code recognition: If the user successfully scans, there will be a scanning success prompt and display “Every bin helps save our planet”; If the scan fails, a scanning failure message “Invalid QR Code” is displayed. If the user scans the code but has completed the task today, the system will display “Already Scanned”.

The user can return to the game screen after clicking “Return Home”.

The backend must track which station codes the user has scanned each day to ensure the user can only scan once per station per day.

1. Result calculation

The user can earn 10 points for completing the task.

The user can complete this task once per recycling station per 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. Then the user can click the “Return Home” button to return.

**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 “Invalid QR Code” is displayed, and the “Try Again” and “Return Home” buttons are provided.

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

**Given:** The user has successfully scanned at this specific recycling station today.

**When:** The user scans the QR code of the same station again on the same day.

**Then:** The system displays “Already Scanned” prompt. Then the user can click the “Return Home” button to return.