**Technical Design Document: Albert Heijn Collectibles Mobile Website**

**1. Introduction**

This document outlines the technical design for a mobile website that allows users to collect virtual Albert Heijn characters. The website will feature an interactive 3D environment representing different sections of an Albert Heijn store. Users can discover characters by navigating these 3D snippets, and physically collect them by scanning NFC tags embedded in 3D-printed figures obtained in-store. The platform will provide clues for uncollected characters and food waste-related information for collected ones.

**2. Goals**

* Create an engaging and interactive mobile website using 3D elements powered by Three.js.
* Allow users to virtually explore different store sections to discover collectible characters.
* Implement a system where uncollected characters are visually represented as blacked-out with a question mark.
* Provide a riddle/hint for uncollected characters upon clicking their placeholder.
* Display product information with a food waste focus for collected characters upon clicking their 3D representation.
* Enable character collection through NFC scanning.
* Utilize Vue.js for the frontend framework to manage the application state and UI components.
* Maintain the visual style and branding of Albert Heijn.
* Store application data in a JSON file for this initial project phase.

**3. Target Audience**

Customers of Albert Heijn, particularly children interested in collecting the in-store promotional items.

**4. Technical Stack**

* **Frontend Framework:** Vue.js
* **3D Graphics Library:** Three.js
* **NFC Interaction:** Web NFC API
* **State Management(optional):** Vuex
* **Styling:** CSS
* **Data Storage (Temporary):** JSON file

**5. System Architecture**

The application will primarily be a frontend-heavy single-page application (SPA) built with Vue.js and Three.js.

* **3D Environment:** Three.js will be used to render interactive 3D snippets of different store sections (e.g., bakery, produce, dairy). These will be provided as 3D models by the MV team.
* **Character Representation:** Each collectible character will have a 3D model. The application will manage the state of each character (collected or uncollected).
* **UI Components:** Vue.js components will be responsible for rendering the store navigation, character displays, information modals, and any other interactive elements.
* **Data Management:** A JSON file will store information about the collectibles, including:
  + Character ID
  + Name
  + Store Section Association
  + 3D Model Path
  + Riddle/Hint (for uncollected)
  + Product Information
  + Food Waste Tip (for collected)
  + NFC Tag ID (linking the physical NFC tag to the digital character)
* **NFC Handling:** The Web NFC API will be used to read the unique ID from the NFC tags embedded in the physical collectibles. This ID will be matched against the data in the JSON file to mark the corresponding character as collected.

**6. Detailed Design**

**6.1. 3D Store Navigation**

* The main view will present a way to swipe horizontally through different 3D snippets representing store areas.
* Each snippet will visually suggest the type of collectibles that might be found there (e.g., bread in the bakery section).
* Performance on mobile will be a key consideration for the complexity and optimization of these 3D scenes.

**6.2. Character Display**

* Within each 3D store section, the collectible characters associated with that section will be displayed.
* **Uncollected Characters:** Will be rendered as a blacked-out silhouette with a question mark overlay.
* **Collected Characters:** Will be rendered in full color and detail based on the 3D models provided.
* Clicking on a character (or its placeholder) will trigger a modal or overlay.

**6.3. Information Modals/Overlays**

* **Uncollected Character:** The modal will display the riddle/hint associated with that character, providing a clue about the Albert Heijn product needed to "collect" it (via the physical NFC figure).
* **Collected Character:** The modal will display information about the corresponding product and a food waste-related tip.

**6.4. NFC Scanning**

* The application will need a clear call-to-action (e.g., a button or prompt) to initiate the NFC scanning process.
* Upon scanning an NFC tag, the application will read the unique ID.
* This ID will be compared against the NFC Tag ID stored in the JSON data.
* If a match is found and the character is not yet collected, the character's state will be updated to "collected," and its visual representation in the 3D scene will change.
* The collected state should be persistent (within the browser's local storage, given the JSON-based backend).

**6.5. Data Structure (JSON)**

JSON

{

"collectibles": [

{

"id": "baguette\_01",

"name": "Baguette Buddy",

"storeSection": "bakery",

"modelPath": "/assets/models/baguette.glb",

"riddle": "I'm long and crusty, a French delight, perfect with cheese, morning, noon, or night.",

"productInfo": "Our freshly baked baguette is a customer favorite. Enjoy it within 2 days for the best taste.",

"foodWasteTip": "Stale baguette can be turned into delicious croutons or breadcrumbs. Don't let it go to waste!",

"nfcTagId": "ABC123XYZ",

"isCollected": false

},

{

"id": "tomato\_02",

"name": "Tommy Tomato",

"storeSection": "produce",

"modelPath": "/assets/models/tomato.glb",

"riddle": "I'm round and red, juicy and bright, in salads and sauces, I'm always a delight.",

"productInfo": "Our vine-ripened tomatoes are packed with flavor. Store them at room temperature for optimal taste.",

"foodWasteTip": "Overripe tomatoes can be used to make a flavorful pasta sauce or soup.",

"nfcTagId": "DEF456UVW",

"isCollected": true

},

// ... more collectibles

],

"collectedCharacters": ["tomato\_02"] // Array of collected character IDs for persistence

}

**6.6. State Management (Vuex)**

Even with a JSON backend, Vuex can help manage the application's state (e.g., the list of collectibles, the collected status of each, the currently displayed store section). This will make the application more organized and maintainable as it grows.

**6.7. Albert Heijn Styling**

* The color palette, typography, and overall visual language of the website should closely resemble Albert Heijn's branding. The MV team will play a crucial role in providing assets and guidance on this.

**7. User Interface (UI) Considerations**

* **Mobile-First Design:** The website must be designed and developed with mobile devices as the primary target.
* **Intuitive Navigation:** Swiping between store sections should be smooth and responsive.
* **Clear Visual Cues:** Collected and uncollected characters should be easily distinguishable.
* **Accessibility:** Consider basic accessibility guidelines to ensure usability for a wider audience.
* **NFC Interaction Guidance:** Provide clear instructions to the user on how to scan the NFC tags.

**8. Development Process & Team Collaboration**

* **Mediavormgeving Team:** Responsible for creating and optimizing the 3D models of the store snippets and collectible characters, as well as providing branding assets.
* **Software Development Team:** Responsible for the frontend development using Vue.js and Three.js, implementing the NFC functionality, managing the data, and ensuring mobile performance.
* Close collaboration between both teams will be essential to ensure the 3D assets are integrated seamlessly into the web application and that the visual style aligns with Albert Heijn's branding.

**9. Potential Challenges & Considerations**

* **Mobile Performance:** Rendering complex 3D scenes on mobile devices can be performance-intensive. Optimization of 3D models, efficient Three.js usage, and careful consideration of rendering techniques will be crucial.
* **Web NFC API Support:** The Web NFC API might not be universally supported across all mobile browsers. A fallback mechanism or clear communication about browser compatibility might be necessary.
* **NFC Tag Reading Reliability:** Ensure robust error handling and clear feedback to the user during the NFC scanning process.
* **Maintaining Albert Heijn's Style:** Close collaboration with the design team is needed to accurately replicate their visual identity.
* **Scalability (Beyond School Project):** If this project were to scale beyond a school exercise, a proper backend database and API would be necessary for managing data and user accounts.

**10. Future Enhancements (Beyond Initial Scope)**

* Integration with a user account system (if Albert Heijn API access were available or a custom system was built).
* Gamification elements (e.g., badges, leaderboards).
* Social sharing features.
* Small minigame(s)

**11. MoSCoW**

|  |  |  |  |
| --- | --- | --- | --- |
| Must Have | Should Have | Could Have | Wont Have |
| Swipeable 3D Store Sections | Vue.js Framework Implementation | Vuex for State Management | Backend Database |
| Collectible Character Display (3D) | Three.js for 3D Rendering | Smoother 3D Transitions/Animations | Trading or Social Features |
| Uncollected Character Placeholder | Basic Albert Heijn Styling | NFC Scanning Instructions/Feedback | Augmented Reality (AR) Features |
| Riddle/Hint for Uncollected Characters |  | More Advanced 3D Interactions |  |
| Collected Character Information |  | Integration with Albert Heijn Account via API |  |
| NFC Scanning for Collection |  | User Accounts |  |
| Updating Collection Status |  |  |  |
| Persistence of Collected Data (Local Storage) |  |  |  |
| Data Storage in JSON |  |  |  |