**Use Case: Apply for a Philippine Passport (KYC on Registration)**

**Actor:** Citizen (User)

**Preconditions:**

* The citizen has access to the JuanID ServiceNow portal.
* The citizen has basic digital literacy and access to necessary documents (physical or digital copies).

**Main Flow:**

1. **User Accesses JuanID Portal:** The citizen navigates to the JuanID portal via a web browser.
2. **User Registration:**
   * The citizen creates a new account.
   * **KYC Verification:** The user is prompted to upload **at least one** of the following for identity verification:
     + Valid Government-Issued ID (e.g., National ID, Driver's License, Passport if they have one)
     + Philippine Birth Certificate (if no valid government-issued ID is available).
   * **System Analyzes KYC Document (GenAI):**
     + The system uses Generative AI to extract key information from the uploaded KYC document (name, date of birth, place of birth, issuing authority).
     + The system attempts to verify (basic level) if the uploaded document meets certain criteria (clear image, all necessary fields present, matches information provided during registration).
   * User provides other basic personal information.
   * User creates a password and agrees to terms and conditions.
   * Account is created and the user is logged in.
3. **User Selects 'Philippine Passport':** The citizen navigates to the "Apply for ID" section and selects 'Philippine Passport'.
4. **System Displays Requirements:** The system displays the official list of requirements for a Philippine Passport.
5. **System Checks Pre-verified Documents:** The system indicates which requirements might already be partially fulfilled based on the KYC document uploaded during registration.
6. **User Uploads Remaining Required Documents:** The citizen uploads any additional documents required for the passport application that were not part of the initial KYC.
7. **System Analyzes Uploaded Documents (GenAI):** The system uses Generative AI to extract information and perform basic validation on the newly uploaded documents.
8. **System Validates All Documents (Basic Level):** The system performs a basic check on all relevant documents (KYC and application-specific) for completeness and format.
9. **System Displays Fees and Payment Options:** The system shows the applicable passport processing fees and available payment methods.
10. **User Pays Fees (Potentially Integrated/Simulated):** The user completes the payment process through the platform (this could be a simulated step for the capstone).
11. **System Auto-Fills Application Form (Where Possible/Simulated):** Based on the extracted information and uploaded documents, the system automatically fills out the online passport application form (or presents a pre-filled form for review).
12. **User Reviews and Confirms Information:** The citizen reviews the pre-filled information and confirms its accuracy.
13. **User Submits Application:** The citizen submits the application through the JuanID portal.
14. **System Acknowledges Submission:** The system displays a confirmation message with a unique application tracking number and informs the user about the next steps (e.g., potential physical appearance requirement).
15. **System Sends Initial Submission (Simulated):** The system simulates the initial submission of the application data and documents to the DFA's system.
16. **User Tracks Application Status:** The citizen can track the status of their application through the JuanID portal.
17. **System Provides Notifications:** The system sends notifications to the user regarding application updates (e.g., "Application Received," "Schedule for Physical Appearance - if required," "Passport Ready for Pickup").

**Alternative/Error Flows:**

* **Invalid KYC Document:** The system notifies the user if the uploaded KYC document is invalid or unclear and prompts for a valid replacement. Account creation may be blocked until valid KYC is provided.
* **Mismatch in KYC Information:** If the information extracted from the KYC document doesn't match the registration details, the system prompts the user to review and correct the information.
* **Other document-related errors** (as in the previous use case).
* **Payment failure** (if payment integration is simulated).

**Postconditions:**

* The citizen has registered on the JuanID platform with their identity (partially) verified.
* The citizen has submitted their passport application through a unified platform.
* The citizen has a record of their application and a tracking number within the JuanID system.
* The citizen receives updates on their application status and any requirements for physical appearance or ID pickup.

technologies and potential APIs for JuanID application  
Frontend (Vite React with MUI):  
  
  - Vite: A fast build tool for modern web applications. It's excellent for development speed and hot module replacement.  
  - React: A JavaScript library for building user interfaces. It's component-based, making it efficient for building complex UIs like the JuanID portal.  
  - MUI (Material UI): A React UI framework that implements Google's Material Design. It provides a set of ready-to-use, customizable components that will help you build a visually appealing and consistent user interface quickly. MUI is a great choice for creating a clean and user-friendly experience.  
  
Backend (Spring Boot):  
  
  - Spring Boot: A Java-based framework for building microservices and web applications. It simplifies the setup and configuration of Spring applications, making it efficient for backend development. Spring Boot is well-suited for handling the application logic, data management, and API endpoints for JuanID.  
  
Potential APIs and Integrations:  
  
1.  GenAI Service:  
      - You'll need an API to handle the Generative AI tasks, primarily document analysis (extracting information from uploaded IDs and birth certificates). Some options include:  
          - Google Cloud Document AI: A powerful service for document parsing and understanding. It can extract text, identify fields, and even understand the structure of various document types.  
          - AWS Textract: Similar to Google Cloud Document AI, AWS Textract can extract text and structured data from documents.  
          - Azure Form Recognizer: Microsoft's offering for intelligent document processing.  
          - Custom-Trained Model: If you have the resources and expertise, you could train your own GenAI model, but this is likely out of scope for a 3-week project.  
      - For your capstone, you might simulate this API if a full integration is too complex. You could create a simplified service that takes sample document data and returns pre-defined extracted information.  
2.  Payment Gateway (Simulated for Capstone):  
      - For handling application fees, you'd typically integrate with a payment gateway like:  
          - PayMaya (Philippines)  
          - GCash (Philippines)  
          - Stripe  
          - PayPal  
      - For your capstone, you'll likely need to simulate the payment process. You could create a mock payment service that accepts a payment amount and returns a successful transaction response.  
3.  Government APIs (Conceptual for Capstone):  
      - In a real-world scenario, you'd need to integrate with government APIs to:  
          - Verify the authenticity of uploaded IDs.  
          - Submit application data directly to the relevant agencies (e.g., Department of Foreign Affairs for passport applications).  
          - Retrieve application status updates.  
      - These APIs are often not publicly available or require specific agreements. For your capstone, you'll need to conceptually outline how these integrations would work. You can simulate the API responses for demonstration purposes.  
4.  ServiceNow APIs:  
    \* You can utilize ServiceNow APIs to interact with the platform for data retrieval, workflow management, and user authentication.  
  
Architecture Overview:  
  
1.  Frontend (React/MUI): The user interacts with the JuanID portal built with React and MUI. The frontend makes API requests to the Spring Boot backend.  
2.  Backend (Spring Boot): The Spring Boot application handles user authentication, manages application data, orchestrates the calls to the GenAI service (or simulated service), and (in a real-world scenario) would interact with government APIs. It exposes RESTful APIs for the frontend to consume.  
3.  GenAI Service: This service (either a real API or a simulated one) analyzes uploaded documents.  
4.  Database (within ServiceNow or external): You'll need a database to store user information, application data, and document metadata. This could be within ServiceNow or an external database like PostgreSQL or MySQL.  
  
Key Considerations for your Capstone:  
  
  - Focus on Core Functionality: Given the 3-week timeframe, prioritize the core features: user registration with KYC, passport application with document analysis, and a basic workflow.  
  - Simulation: Don't try to build everything perfectly. Simulate complex integrations like payment gateways and government APIs.  
  - Scalability: While scalability isn't a primary concern for a capstone project, keep it in mind when designing your architecture.  
  - Security: Pay attention to security, especially when handling sensitive user data and ID documents. Use secure coding practices and follow security guidelines for your chosen frameworks.  
  
This provides a solid starting point for your tech stack and API considerations. Remember to break down the development into smaller, manageable tasks and focus on delivering a functional MVP within the 3-week timeframe.

welp Ai rasd ni hahha

but atleast we have some sort of guide

as for the usecase scenario, I asked gemini what will be gen AI will do with the requirements:

Okay, these are the detailed requirements for a new adult Philippine passport application. This is excellent information for building the logic within JuanID.

Here's how we can structure this information within the app's flow and how the GenAI can assist:

**JuanID Passport Application Flow with Detailed Requirements:**

1. **User Selects 'Philippine Passport' (as before).**
2. **System Presents Core Requirements:** JuanID will display the "A. CORE REQUIREMENTS" clearly to the user:
   * **Confirmed Online Appointment:** (Information only - JuanID might provide a link to the official appointment site but won't handle the scheduling itself initially).
   * **Personal appearance:** (Information only - informs the user this is mandatory).
   * **Accomplished Application Form:** JuanID can potentially:
     + Provide a link to the official downloadable form.
     + **(Future Enhancement with GenAI)** Pre-fill parts of the form based on the user's JuanID profile and uploaded KYC/documents.
   * **PSA Authenticated Birth Certificate:**
     + Clearly state the requirement for the **original and photocopy on Security Paper**.
     + Inform users about the **Local Civil Registrar Copy** requirement if the PSA copy isn't clear.
     + **GenAI Assistance:** Analyze the uploaded Birth Certificate (from KYC or specifically for the application) to:
       - Extract key details (name, date of birth, place of birth).
       - Potentially identify if it's a PSA-authenticated copy (based on visual cues, though this might be limited).
       - Prompt the user if the uploaded copy seems unclear.
   * **Married Females (using spouse's last name):**
     + Clearly state the requirement for the **Original and photocopy of PSA Authenticated Marriage Contract on Security Paper or Report of Marriage.**
     + **GenAI Assistance:** If a Marriage Contract is uploaded, analyze it to extract relevant details (names, marriage date).
   * **Valid IDs:**
     + Provide a clear link to the "List of Acceptable IDs for Passport Processing."
     + Require the user to upload a photocopy of **one (1)** valid ID.
     + **GenAI Assistance:** Analyze the uploaded valid ID to extract key information (name, date of birth, issuing authority, expiry date) and potentially perform basic checks for validity (e.g., not expired, clear image).
   * **Note on Maiden Name:** Display this note clearly.
3. **System Detects Potential Special Cases (Based on User Input/KYC):**
   * **Late Registered Birth Certificate:**
     + If the birth date on the uploaded Birth Certificate indicates a potentially late registration (e.g., registered significantly after birth), JuanID can ask the user a clarifying question: "Was your Birth Certificate registered more than 10 years after your birth?"
     + Based on the answer, it will then present the relevant supporting document requirements (IDs pre-dating registration OR current IDs with NBI Clearance).
     + **GenAI Assistance:** Analyze the registration date on the Birth Certificate (if discernible) to pre-suggest the relevant supporting documents.
   * **No Birth Certificate or Report of Birth:**
     + If the user indicates they don't have a Birth Certificate, JuanID will:
       - Present the different procedures based on the year of birth (before or after 1950).
       - Guide them on filing for late registration or the requirements for those born on/before 1950 (PSA Certificate of No Birth Record, Affidavit of Two Disinterested Persons).
   * **Dissolved Marriage (Wanting to Revert to Maiden Name):**
     + JuanID can ask: "Have you been married and is your marriage now dissolved, and do you want to use your maiden name?"
     + If yes, it will present the required documents (PSA Birth Certificate, acceptable IDs, and either annotated PSA Marriage Certificate/ROM OR Court Order and Certificate of Finality).
     + **GenAI Assistance:** Analyze uploaded Marriage Certificates/Court Orders to identify annotations or dissolution details (though legal document analysis can be complex).
   * **Lacking Data in Birth Certificate/Report of Birth:**
     + JuanID can ask: "Does your Birth Certificate have any missing information?"
     + If yes, it will require proof of filing of a supplemental report or correction from the LCR.
   * **Discrepancy in Data:**
     + JuanID can ask: "Is there any difference in the information (like name or birthdate) between your Birth Certificate and other documents?"
     + Based on the user's response and the specific discrepancy, it will present the relevant requirements (annotated PSA Birth Certificate, petition for correction of first name).
   * **Dual Citizen, Naturalized Citizen, Citizenship by Election, Citizenship by Act of Legislation:**
     + JuanID can include a section asking about the user's citizenship status beyond being a regular Filipino citizen.
     + Based on the selection, it will display the corresponding "B. SUPPORTING DOCUMENTS" requirements.
     + **GenAI Assistance:** For documents like foreign passports or identification certificates, GenAI can extract key details for basic verification.
4. **Document Upload and Analysis (as before, but now specific to each requirement).**
5. **Fee Payment (as before).**
6. **Application Submission (as before).**
7. **Guidance on Next Steps:** JuanID will clearly inform the user that **personal appearance is mandatory** and they will need to schedule an online appointment via the official DFA website (provide a link). It will also mention the potential need for physical appearance for claiming the passport.

**Key Role of GenAI:**

* **Information Extraction:** Efficiently pull data from uploaded documents to pre-fill forms and compare against requirements.
* **Basic Validation:** Perform initial checks on document clarity, completeness (presence of key fields), and potential expiry (for IDs).
* **Contextual Guidance:** Based on the uploaded documents and user responses, provide tailored information about required supporting documents.
* **(Future Potential)** Assist in identifying the *type* of document uploaded (e.g., "This looks like a PSA Birth Certificate").

By meticulously mapping these requirements into the JuanID flow and leveraging GenAI for analysis and guidance, you can create a very helpful and intelligent application for passport processing. Remember to clearly indicate to the user the limitations of the AI's verification capabilities and the importance of providing accurate and authentic documents.

Todo:  
what I want to do is create a simple user account and all of its data is stored on a single string variable called, "userdata" since I'm kinda lazy to make all of it and in the ai part we will just parse every document into a markdown language and save it in here, so potentially this is the workflow

Technology:

Servicenow (backend & frontend)

gemini free api key(with vision model since this deals with pdf, image for the document)

imgurl: for saving the images like the actual pdf or image on that way only the link is saved to the userdata so only the matadata and link will be save

database schema:

first name

last name

email

password

gender

userdata: empty by default

The user creates an account, on the first login since the knowledge is empty, the app will automatically redirect into uploading a document mode.

upon receiving the file, the pdf is then converted into an image png for gemini compatibility since they are typical photocopy anyway

then the image is then given into the gemini api with a systemprompt to store it into a markdown format with proper styling, the images is then also sent to the imgurl and then once the link has been recieved, its then also appended to the userdata for historical sake

The page that is responsible for displaying the ai output should support markdown format