Buyer Account

Buyer Account Creation

**1. Initialize**

* **Usernames List**:
  + Create a list called existing\_usernames which holds all the usernames that are already taken. This ensures we can check the uniqueness of a new username.
* **Password Validation Function**:
  + Define a function is\_valid\_password(password) that checks whether the password meets the specified constraints. This function will return True if the password is valid and False otherwise.
* **Username Uniqueness Function**:
  + Define a function is\_unique\_username(username) that checks if the provided username is unique by comparing it against the existing\_usernames list. It returns True if the username is unique and False if it is already taken.

**2. Get Input**

* **Username**:
  + Prompt the user to enter a Username.
  + This input will be checked for uniqueness.
* **Password**:
  + Prompt the user to enter a Password.
  + This input will be validated to ensure it meets the constraints.
* **Other Details**:
  + Prompt the user to enter First Name, Last Name, Email, Address, and Phone Number.

**3. Validate Inputs**

* **Username Validation**:
  + Check if the entered Username is unique by using the is\_unique\_username function.
  + If the username is not unique, prompt the user to enter a different Username and repeat the check until a unique username is provided.
* **Password Validation**:
  + Validate the entered Password using the is\_valid\_password function.
  + If the password does not meet the constraints, prompt the user to enter a valid Password and repeat the validation until a valid password is provided.

**4. Store Inputs**

* **Account Storage**:
  + If all inputs (Username, Password, First Name, Last Name, Email, Address, Phone Number) are valid, store the user's details in a user account database or list.

**5. Output**

* **Confirmation**:
  + Once the account is successfully created, provide a confirmation message to the user indicating that the account creation was successful.

**Detailed Steps**

**Step-by-Step Process**

1. **Initialization**:
   * Prepare a list of existing usernames.
   * Define functions to validate the password and check username uniqueness.
2. **Getting Inputs**:
   * Prompt the user to enter each required piece of information.
3. **Username Validation**:
   * Use the is\_unique\_username function to ensure the username is not already taken.
   * If the username is not unique, inform the user and prompt them to choose another.
4. **Password Validation**:
   * Use the is\_valid\_password function to ensure the password meets the constraints:
     + Length between 8 and 16 characters.
     + Includes at least one lowercase letter.
     + Includes at least one uppercase letter.
     + Includes at least one number.
     + Includes at least one special character.
   * If the password is invalid, inform the user and prompt them to enter a new one.
5. **Input Storage**:
   * Once all inputs are validated, store them in the system's user database.
6. **Confirmation**:
   * Provide feedback to the user that their account has been created successfully.

**Key Considerations**

* **Data Validation**: Ensuring that each piece of user input meets the required criteria is crucial for security and data integrity.
* **User Feedback**: Informing the user of any errors and providing clear instructions on how to correct them enhances the user experience.
* **Security**: Password constraints help in creating strong passwords, protecting user accounts from potential breaches.
* **Uniqueness**: Ensuring unique usernames prevents user confusion and potential conflicts in the system.

### Dealer Verification Process for Buyers

**1. Initialize**

* Prepare a list to store existing business registration numbers and tax IDs.
* Define a function to validate business registration details.
* Define a function to validate supporting documents.

**2. Get Input**

* Prompt the user to input business registration details: Business Name, Registration Number, and Tax ID.
* Prompt the user to input business contact information: Business Address, Phone Number, and Email.
* Prompt the user to upload supporting documents.

**3. Validate Business Registration Details**

* Check if the Registration Number and Tax ID are unique and not already used by another business.
* Validate the format and correctness of Registration Number and Tax ID.

**4. Validate Supporting Documents**

* Check if the uploaded documents are valid and meet the required criteria.
* Ensure that the documents match the provided business registration details.

**5. Validate Business Contact Information**

* Ensure that Business Address, Phone Number, and Email are correctly formatted and valid.

**6. Verification Process**

* If all inputs are valid and the documents are verified, mark the user as a verified dealer.
* If any validation fails, provide feedback to the user to correct the errors and resubmit the necessary details.

**7. Store Inputs and Update Status**

* Store the validated business registration details, contact information, and supporting documents.
* Update the user's account status to 'verified dealer' in the system.

**8. Output**

* Provide feedback to the user about their verification status.
* If verification is successful, activate the user account and provide access to the system.
* If verification fails, inform the user of the reasons and prompt them to correct and resubmit the required information.

### Function Definitions

#### is\_valid\_registration\_details(registration\_number, tax\_id, existing\_registration\_numbers, existing\_tax\_ids)

* Checks the uniqueness and format of the registration number and tax ID.

#### are\_valid\_supporting\_documents(documents)

* Checks if the uploaded documents are valid and meet the required criteria.

### Pseudocode

pseudo

Copy code

START

existing\_registration\_numbers = ["reg123", "reg456", ...]

existing\_tax\_ids = ["tax123", "tax456", ...]

FUNCTION is\_valid\_registration\_details(registration\_number, tax\_id, existing\_registration\_numbers, existing\_tax\_ids)

IF registration\_number IN existing\_registration\_numbers THEN

RETURN False

END IF

IF tax\_id IN existing\_tax\_ids THEN

RETURN False

END IF

# Additional format checks can be implemented here

RETURN True

END FUNCTION

FUNCTION are\_valid\_supporting\_documents(documents)

# Implement document validation logic

RETURN True or False based on validation

END FUNCTION

PROMPT "Enter Business Name: "

READ business\_name

PROMPT "Enter Registration Number: "

READ registration\_number

PROMPT "Enter Tax ID: "

READ tax\_id

PROMPT "Enter Business Address: "

READ business\_address

PROMPT "Enter Phone Number: "

READ phone\_number

PROMPT "Enter Email: "

READ email

PROMPT "Upload Supporting Documents: "

READ documents

IF is\_valid\_registration\_details(registration\_number, tax\_id, existing\_registration\_numbers, existing\_tax\_ids) THEN

IF are\_valid\_supporting\_documents(documents) THEN

# Store the details in the system

existing\_registration\_numbers.append(registration\_number)

existing\_tax\_ids.append(tax\_id)

# Update user status to verified dealer

user\_status = "verified dealer"

PRINT "Verification successful. Your account is now active."

ELSE

PRINT "Supporting documents are not valid. Please resubmit the documents."

END IF

ELSE

PRINT "Registration details are not valid or already in use. Please provide correct details."

END IF

END

### Detailed Steps

**Step-by-Step Process**

1. **Initialization**:
   * Prepare lists of existing registration numbers and tax IDs to ensure uniqueness.
   * Define functions to validate registration details and supporting documents.
2. **Getting Inputs**:
   * Prompt the user to enter business registration details and contact information.
   * Prompt the user to upload supporting documents.
3. **Validate Business Registration Details**:
   * Check if the registration number and tax ID are unique and correctly formatted.
4. **Validate Supporting Documents**:
   * Ensure the uploaded documents are valid and match the business registration details provided.
5. **Validate Business Contact Information**:
   * Check the format and validity of the business address, phone number, and email.
6. **Verification Process**:
   * If all validations pass, mark the user as a verified dealer.
   * If any validation fails, inform the user of the issues and prompt them to correct and resubmit the necessary information.
7. **Store Inputs and Update Status**:
   * Store the validated details and update the user's account status in the system.
8. **Output**:
   * Provide feedback on the verification status.
   * If successful, activate the user account.
   * If unsuccessful, provide reasons and instructions for resubmission.

### Key Considerations

* **Data Validation**: Ensuring all inputs and documents meet the required criteria is essential for the verification process.
* **User Feedback**: Providing clear and actionable feedback to users helps them understand and correct any issues.
* **Security**: Verifying business details and supporting documents helps in maintaining the integrity and security of the system.
* **Uniqueness**: Ensuring unique registration numbers and tax IDs prevents duplication and potential fraud.