LEARNING TASK 01

ACTION 01: Resetting a forgotten password.

Observation and Decomposition:

- Who is acting:
  User who tries to login in existing account.
- What is being attempted:
  The user attempts to reset the password for their account.
- What must already be true:
  User account must exist, password reset mechanism must exist.
- Under what conditions it should be allowed:
  Some of the information match with user account records.
  Password reset request must be verified
- What happens if it is not allowed:
  Password remains unchanged.
  Reset attempt fails.

Constraint Identification:

- Who is acting:
  User who tries to login in existing account.
- What is being attempted:
  The user attempts to reset the password for their account.
- What must already be true:
  User account must exist,
  password reset mechanism must exist.
  Verification mechanism should exist.
- Under what conditions it should be allowed:
  Some of the information match with user account records.
  Password reset request must be verified
- What happens if it is not allowed:
  Password remains unchanged.
  Reset attempt fails.

ACTION 02: Installing an application

Observation and Decomposition:

- Who is acting:
  A user using some sort of device.
- What is being attempted:
  User try to install a specific app on device.
- What must already be true:
  Existence of that app must be true
  System should not have it already and the specs to run that app.
- Under what conditions it should be allowed:
  When user initialize the download
- What happens if it is not allowed:
  The device remains unchanged.

Constraint Identification:

- Who is acting:
  A user using some sort of device, which they should have control over it.
- What is being attempted:
  User try to install a specific app on device.
- What must already be true:
  A device must exist and in working condition.
  App is installable.
  A device with installation mechanism
  Storage
  Existence of that app must be true
  System should not have it already and the specs to run that app.
- Under what conditions it should be allowed:
  The installation mechanism accepts the application package.
  When user initialize the download
- What happens if it is not allowed:
  The device remains unchanged.

ACTION 03: Transferring money

Observation and Decomposition:

- Who is acting:
  A user who controls the account.
- What is being attempted:
  Transfer of specific amount from source account to destination.

- What must already be true:
  Both source and destination should exist
  the source account contains a balance
  Transfer mechanism
- Under what conditions it should be allowed:
  Source account must have amount equal to or greater than the transfer amount
- What happens if it is not allowed:
  The balance of both account remains unchanged.

Constraint Identification:

- Who is acting:
  A user who controls the account.
- What is being attempted:
  Transfer of defined amount from source account to destination.
- What must already be true:
  Both source and destination should exist
  the source account contains a balance
  Transfer mechanism
- Under what conditions it should be allowed:
  Source account must have amount equal to or greater than the transfer amount
  The transfer mechanism accepts the transfer request as valid.
- What happens if it is not allowed:
  The balance of both account remains unchanged.

ACTION 04: Submitting an assignment

Observation and Decomposition:

- Who is acting:
  A user with an assignment
- What is being attempted:
  Submitting a specific assignment
- What must already be true:
  User having a assignment
  Submission mechanism
- Under what conditions it should be allowed:
  The submission mechanism accepts the assignment submission.

- What happens if it is not allowed:
  Assignment is not submitted

Constraint Identification:

- Who is acting:
  A user with an assignment
- What is being attempted:
  Submitting a specific assignment for evaluation.
- What must already be true:
  User having a assignment
  Submission mechanism
  Assignment is in state of submission
- Under what conditions it should be allowed:
  The submission mechanism confirms that the assignment is acceptable.
- What happens if it is not allowed:
  Assignment is not submitted.
  Assignment remains in previous state

ACTION 05: Adding an item to a cart

Observation and Decomposition:

- Who is acting:
  A user interacting with a shopping system
- What is being attempted:
  A specific item is added to cart
- What must already be true:
  Item must exist
  User must exit
  A cart exist
  Mechanism to add in cart
- Under what conditions it should be allowed:
  Add to cart mechanism accepts the item.
- What happens if it is not allowed:
  Item is not added in cart
  Cart remains unchanged or not created

Constraint Identification:

- Who is acting:
  A user associated with cart
- What is being attempted:
  A specific item is added to cart
- What must already be true:
  Item must exist
  User must exit
  A cart exist and linked to user
  Mechanism to add in cart
- Under what conditions it should be allowed:
  Add to cart mechanism accepts the item.
  Item is available when request is made
- What happens if it is not allowed:
  Item is not added in cart
  Cart remains unchanged

Structure Formation:

1. Actor: The one who initiates the action
2. Goal: What the action tries to achieve
3. Required setup: Things that should be already present
4. Decision making condition
5. Result if conditions pass
6. Result if conditions fails

Consistency Testing:

ACTION 01: Resetting a forgotten password.

1. Actor: User connected to an account
2. Goal: Reset the password
3. Required setup: Account exists, User-account link exist, reset process exists
4. Decision making condition: Verification confirms the connection with user

5. Result if conditions pass: Password is changed and old one becomes invalid
6. Result if conditions fails: Password remains unchanged

Checks

- **Ambiguity:** None
- **Redundancy:** None
- **Missing information:** None

ACTION 02: Installing an application

1. Actor: User controlling device
2. Goal: Install a app in device
3. Required setup: Existing of app
4. Decision making condition: App is compatible and have sufficient storage
5. Result if conditions pass: App is added to device
6. Result if conditions fails: App is not added to device

Checks

- **Ambiguity:** None
- **Redundancy:** None
- **Missing information:** None

ACTION 03: Transferring money

1. Actor: User linked to a source account
2. Goal: Transfer a specific amount to a destination account
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition: Source account balance ≥ transfer amount at request time
5. Result if conditions pass: Amount is deducted from source and amount is added to destination
6. Result if conditions fails: Balance unchanged

Checks

- **Ambiguity:** None
- **Redundancy:** None
- **Missing information:** None

ACTION 04: Submitting an assignment
1. Actor: User with assigned assignment
2. Goal: Submit the assignment
3. Required setup: Assignment and submission mechanism
4. Decision making condition: Assignment is in submittable stage
5. Result if conditions pass: Assignment state changes to submitted
6. Result if conditions fails: Assignment state remains unchanged

Checks
- **Ambiguity:** None
- **Redundancy:** None
- **Missing information:** None

ACTION 05: Adding an item to a cart
1. Actor: User using shopping system
2. Goal: Add item to cart
3. Required setup: Item, cart and add to cart mechanism
4. Decision making condition: Item is available
5. Result if conditions pass: Item is added to cart
6. Result if conditions fails: Item is not added to cart

Checks
- **Ambiguity:** None
- **Redundancy:** None
- **Missing information:** None

Refusal Modeling

REFUSAL 1: Resetting a Forgotten Password
The attempted action: Resetting of password
The exact reason it fails: The verification does not confirms the connection with user
Where in the structure the failure is detected: Decision condition

REFUSAL 2: Transferring Money
The attempted action: Transferring money from source to destination account
The exact reason it fails: The source account does not have sufficient balance

Where in the structure the failure is detected: Decision condition

REFUSAL 3: Adding an Item to a Cart
The attempted action: Adding an item into cart
The exact reason it fails: Item is not available
Where in the structure the failure is detected: Decision condition

Explanation for refusal

REFUSAL REASON 1: Resetting a Forgotten Password
The action was not allowed because the verification condition required to confirm the user's association with the account was not satisfied at the time of the reset attempt.

REFUSAL REASON 2: Transferring Money
The action was not allowed because the source account did not have an available balance equal to or greater than the requested transfer amount at the time of the request.

REFUSAL REASON 3: Adding an Item to a Cart
The action was not allowed because the item was not available when the add-to-cart was requested

Reflection and Compression

What information must always be explicit for an action to be safe?

For an action to be safe, the actor, the goal of the action, and the conditions under which it is allowed must always be explicit. It must be clear who is performing the action, what they are trying to do, and what already exists before the action is evaluated. Most importantly, the conditions that decide whether the action succeeds or fails must be written in a way that can be checked without interpretation. If any of these are implied instead of stated, the action becomes open to multiple interpretations and cannot be reliably evaluated.

What felt unnecessary in your structure?

At first, it felt unnecessary to separate “required setup” from “decision conditions,” because as a human, both are often thought of together. However, while working through multiple actions, this separation proved useful. It prevented mixing existence with evaluation. No part of the final structure felt truly unnecessary, but the process showed that adding more fields would not have improved clarity and would likely have introduced redundancy.

Where did ambiguity try to enter?

Ambiguity mostly tried to enter through assumptions that felt obvious, such as assuming ownership, availability, or validity without stating how they are confirmed. Words like “authorized,” “valid,” or “allowed” initially seemed sufficient, but they hid important checks. Ambiguity also appeared when timing was not specified, such as when a balance or availability should be evaluated. These gaps only became visible when trying to apply the same structure consistently across different actions.

What breaks if humans skip structure and rely on intuition?

When humans rely on intuition instead of structure, decisions become inconsistent. The same action may be allowed in one case and denied in another without a clear reason. Failures become difficult to explain because the exact point of failure is not known. Over time, this leads to systems that behave unpredictably and cannot be trusted. Structure forces clarity by making every decision traceable to a specific condition, which intuition alone cannot guarantee.