Design Rationale req4

Changes made inside uml:

- **Added magical item abstract class, consumable item interface, consume action, consumed item manager.**

Changes made inside design rationale:

- Removed **1 action for 1 consumable item** and added **1 action class needed for multiple consumable items.**
- **Items that were consumed no longer stays in actor inventory. Consumed items will be stored inside consumed item manager.**
- **Player will store the instance of Consumed item manager.**

1 action class needed for multiple consumable items. When the player has consumable item in the inventory, the player has option to consume action. For the new design, when item returns new consume action to the player allowable actions, it passes itself (consumable item) into consume action. Since consume action stores instance of consumable item, it can execute buffs of the item to actor by calling consumable item interface method. The buffs are then added to the actor inside the item class (such as super mushroom, power star). By doing this, it no longer needs to create 1 action for 1 consumable item.

Consumed item manager. When the actor consumes the consumable item, consume action will add consumable item into consume item manager and remove item from actor's inventory. By doing this, inventory will no longer contain item that is consumed which makes the design cleaner and easier to manage.

Player will store the instance of Consumed item manager. Our assumption is only the player can consume items. So, player stores consumed item manager and manages item buffs by calling a ticker for consumed item manager every round.

Magical item implements consumable item interface. Magical item abstract is created and implements consumable item interface. If we want to create new item that implements consumable item, that new item can just inherit magical item abstract class. When we want to add 10 more consumable item, each of them does not need to implement consumable item interface (reduces work).

Consume action and consumable item interface. Any item that implements consumable item interface should have buff effect which runs for several turns until condition has met. Consume action will store the consumable item into the consumed item manager because consumable item will always have buff effect and consume action manages only the item that has buff effect.(Which should be stored inside the consumed item manager to check if buffed effects should be removed every turn)