

ASE Pair Assignment

Laundry Problem

Akanksha Gupta (ag3749)

Ishan Guru (ig2333)

Link to Trello Boards:

1. User Stories:

<https://trello.com/b/LdTsh6lR/asepair-laundrystem-userstories>

2. Use Cases:

<https://trello.com/b/f77AcJHC/pairprog-laundrystem-usecases>

User Story:

As a user, I should be able to add the laundry feature to the software controlling an existing robot. The robot should collect dirty clothes, wash each load, dry them, fold and iron them if needed and put them back to the rack. If not needed, then the robot should discard the unwanted items.

Multiple User Stories:

1. Gather Clothes: As a user, I should be able to program the robot to collect clothes to be washed from the customer
2. Segregate Clothes: As a user, I should be able to program my robot to segregate colored clothes and white clothes, so as not to mix them.
3. Washing Clothes: As a user, I should be able to program my robot to wash clothes. The robot should be able to put the appropriate amount of clothes and the washing liquid in the washer and press the wash button.
4. Check Status of Clothes: As a user, I should be able to program my robot to check the status of clothes time to time.
5. Dry Clothes: As a user I should be able to program my robot to pick clothes from washer and put them to dryer once they are done washing. The robot should be able to set the appropriate temperature for drying and press the drying button
6. Pickup, Iron and Fold: As a user, I should be able to program my robot to pick up clothes from dryer, iron them, fold them and put them in a shelf.
7. Delivery Alert: As a user, I should be able to program my robot to send alert to customers once the order for washing is completed to pick up the clothes.

Do you think it would be useful to break this up into multiple user stories? Why or why not? If so, write those user stories and include them on your task board.

Yes, after we came up with the first one user story, we found that it was giving a more holistic idea and lacked details. Also, the more the granular user stories allow us to break the tasks down into more modular steps, since each individual task is a part of the larger one.

How would you organize the work among your team members to develop and deliver this feature?

Since the current user story is generic and provides a holistic view, we would modularize our user story into multiple user stories as shown above and then divide them among the team. We will consider the prior experience of my team members in a particular technology. If it maps to the technology we are using, we will get for Knowledge Transfer. Further if one of my team member has experience in cloud, then we would like that member to design the architecture related to cloud for our robot programming. Also, the individual modular user stories can be divided among team members, with each team member working on an individual user story. This would allow for a larger amount of tasks to be done simultaneously. Additionally, we will

follow the Agile model as discussed in class. We will work together as a team in sprints. With each sprint, we will plan to finish few user stories based on team member's availability and expertise.

Use cases

• Use Case: Laundry Washing

– Collecting the laundry items

Actor: Robot

Prerequisites: The robot controller gets the dirty clothes from the customer to initiate cleaning of clothes.

Overview: The Robot gathers the dirty clothes.

– Segregating the laundry items

Actor: Robot

Prerequisites: The robot segregates the clothes into colored and white clothes to feed them into the washer.

Overview: The Robot segregates the laundry items.

– Washing the clothes

Actor: Robot

Prerequisites: Laundry items are segregated into categories based on color.

Overview: The robot puts the clothes in the washer, puts the needed amount of detergent solution and presses the washing option.

• Use Case: Re-Washing of clothes

– If a re-wash of clothes is needed, the above process is repeated.

– Collecting the laundry items

Actor: Robot

Prerequisites: The robot controller gets the dirty clothes from the customer to initiate cleaning of clothes.

Overview: The Robot gathers the dirty clothes.

– Segregating the laundry items

Actor: Robot

Prerequisites: The robot segregates the clothes into colored and white clothes to feed them into the washer.

Overview: The Robot segregates the laundry items.

– Washing the clothes

Actor: Robot

Prerequisites: Laundry items are segregated into categories based on color.

Overview: The robot puts the clothes in the washer, puts the needed amount of detergent solution and presses the washing option.

• Use Case: Check status of clothes

– Check status of clothes being washed

Actor: Robot

Prerequisites: The laundry is in process of being washed and re-washed, if required.

Overview: The robot checks status of the clothes being washed and dried.

• Use Case: Folding and ironing the clothes and putting them back to shelf

– Segregate the laundry items based on folding, ironing needed or to be discarded.

Actor: Robot

Prerequisites: The items are dry.

Overview: Robot identifies and categorizes the items into the following:

Use Case: Iron Clothes

-The clothes are to be ironed

Actor: Robot

Prerequisites: The items are dry.

Overview: The robot irons the clothes that need ironing.

Use Case: Fold Clothes

-The clothes are to be folded

Actor: Robot

Prerequisites: The items are dry.

Overview: The robot irons the clothes that need ironing.

Use Case: Discard Clothes

-The clothes that are to be discarded, should be discarded

Actor: Robot

Prerequisites: The items are dry.

Overview: The robot discards the clothes that are not needed.

- **Use Case: Send alert to customer about the washing completion**

- **Send alert for completion to customer**

Actor: Robot

Prerequisites: The items are sorted.

Overview: The robot proceeds to send alerts to the customer and let them know that they can come and pick up the clothes.

Multiple Use Cases:

We preferred the multiple use cases method instead of single one. The multiple use cases method is more detailed and gives more clarity about the tasks.

Multiple use cases helped us breakdown the problem into manageable chunks.

Discuss your two task boards. Compare the two approaches. Which do you think is better? Why?

We created two Trello boards, one depicts the user stories and the other depicts the use cases. The user stories provided a generic over view of the tasks to be done. We personally believe that the user story approach is much better than the user tasks approach since it allows us to follow a more AGILE based development process. Since a user story is more of a holistic view at the task that needs to be accomplished, the developer is at the liberty of splitting it into subdivided tasks that the team can work towards. Hence, this allows for much more creativity and malleability with the solution; which is why we think it's better.

Additionally, we feel like the process of leading a user story to the final product is much more flexible and affective than following use cases. This is because with use cases, you are only getting a set of interactions that are expected with the system, however; with a user story the developer is able to identify the general scenarios expected out of the system. Once a user story is provided, custom use cases can be defined and modified, however; the other way around cannot be done. Since they're written in the point of view of the user, the developer is better able to understand the final product from the perspective of the client instead of the engineers. This, ultimately, leads to more satisfied customers and better products.