Queueing Algorithm:

1. Purpose of the Queueing Algorithm:

The Queueing Algorithm manages and organizes the order in which students can access washing machines and dryers fairly.
The “First In First Out” (FIFO) queuing algorithm is being used in our app in the laundry queue system. There is also the “Last in First Out” (LIFO) algorithm which is a stack algorithm that also achieves the task of putting all the users in a queue, allowing only one person to be allotted the machine at any one point in time. However on the basis of fairness, it makes more sense to have a first come first served policy instead of making the person who queued first wait for a long time therefore we chose the FIFO queuing algorithm.

2. Workflow of the Queueing Algorithm:

Inputs:

- Request to join the washing machine or dryer queue.
- Wallet balance (credits).
- Available slots in the queue.
- Position in the queue.

Outputs:

- Placement in the queue.
- Credit deduction from the user's wallet.
- Notifying the user that it’s their turn.

Algorithm Steps:

1. The user requests to join the washing machine queue through the app.
2. The algorithm checks whether the user has at least 1 credit.
3. If the user has 1 or more credits, they are assigned a position in the queue.
4. One credit is deducted from the user's wallet.
5. The algorithm places the user at the last of the queue by assigning a number.
6. The number displayed would be how many people are in front of the user in the queue. The algorithm derives the displayed number by subtracting the number assigned to the user from the person who is the first in the line.
7. The algorithm monitors the queue and notifies the user when it's their turn.

3. Limitations

The major problem with the queueing algorithm is how to combine the queue for the washing machine with the queue for the dryer. When the user is done using the washing machine, (typically) the user will use the dryer to dry their clothes. Thus, we were initially thinking about adding a separate queue for the dryer. However, the primary goal of our app is to simplify and streamline the laundry process. Introducing a separate queue for dryers could lead to a less efficient and user-friendly experience, which contradicts our core objective.
A solution would be an automatic queue transition where upon queuing for their washing machine, users could be automatically placed in the dryer queue. This would simplify the process and ensure a smooth transition without user intervention. However, if a user who was supposed to use the dryer forgets to use it or take it out, it would start a “traffic jam” which would result in inconvenience for the following users. We would need to make our algorithm more complex in order to prevent and solve these “traffic jams”.

Punishment Algorithm:

1. Purpose of the Punishment Algorithm:

The Punishment Algorithm is designed to discourage delays and ensure that students promptly start and finish their laundry cycles. Whenever the user does not start the machine or take out their clothes in a given time, they will get punished. It serves as a mechanism to maintain accountability and it motivates users to adhere to the established timeframes.
A similar algorithm (although much more complicated) would be credit scoring algorithms used in the credit lending (finance) sector that assess individual behaviors and histories. However, our approach is intentionally simplified because we will not (and should not) restrict users with a history of bad behavior from using washing machines since every student should be entitled to wash their clothes on campus. Additionally, as laundry is just a tiny part of life, compared to money, users would not care as much about getting good "creditworthiness" on our app.

2. Workflow of the Punishment Algorithm:

Inputs:

- User delay in starting or taking out a laundry.
- Previous user position in the queue.
- The severity of the delay.

Outputs:

- Potential actions taken (placing the user back in the queue, deducting more credits).

Algorithm Steps:

1. The algorithm detects if a user has delayed in starting or finishing their laundry cycle.
2. The severity of the delay is assessed, determining the level of punishment.
3. Based on the output, the algorithm takes action, such as placing the user back in the queue or deducting additional credits.

3. Limitations

The algorithm may not always distinguish between deliberate delays and unintentional ones and users may not always perceive the algorithm's punishments as fair or just, especially if they believe the delay was due to a legitimate reason (e.g., elevator not coming on time or fire alarm). Additionally, the algorithm does not directly address the problem of the low turnover rate of washing machines because it just indirectly encourages users to take their clothes on time not to get punished. Therefore, it will be ineffective with users who don't fear the punishment because they rarely use the washing machine.