

# Pay to Go Shower System

K.O.D Inc

Nicoy Smith  
Kenneth Anglin

Ottor Mills  
David Thomas

---

# Introduction

---

# Introduction to Software Modelling

- Software Modelling is the abstraction of a Software Application before coding.
- A better understanding of any system can be obtained by considering it from different perspectives such as **Requirements Models, Static Models, and Dynamic Models** of the Software System.

# Introduction to Pay to Go Shower System

- The idea of Pay to Go Shower System was first conceptualized after the realization that it is really difficult to use a public bathroom in comfort.
- It allows individuals to be cleansed quickly and thoroughly through the payment of a small fee.
- It allocates the optimal amount of soap and water based on the showerers dimensions.

# Requirements Analysis

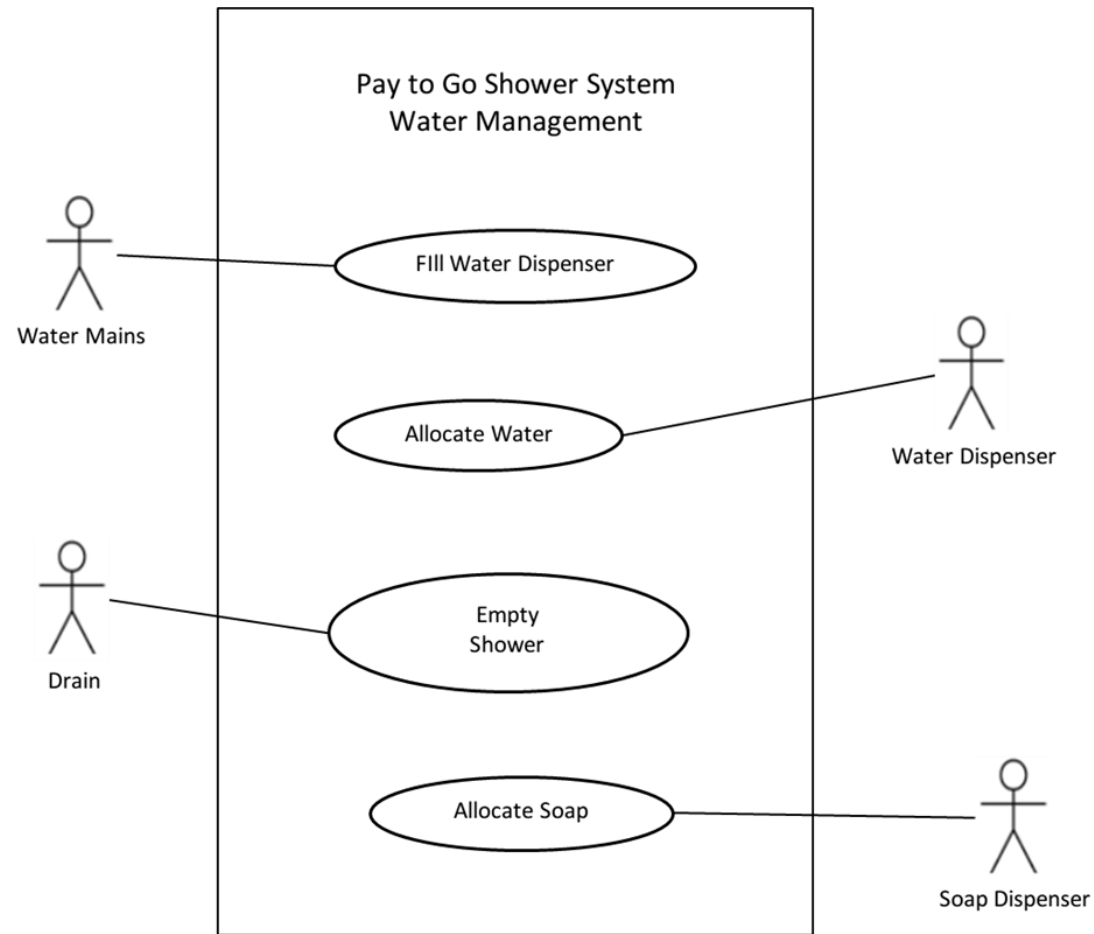
---

## What is Requirements Analysis ?

- Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new product.

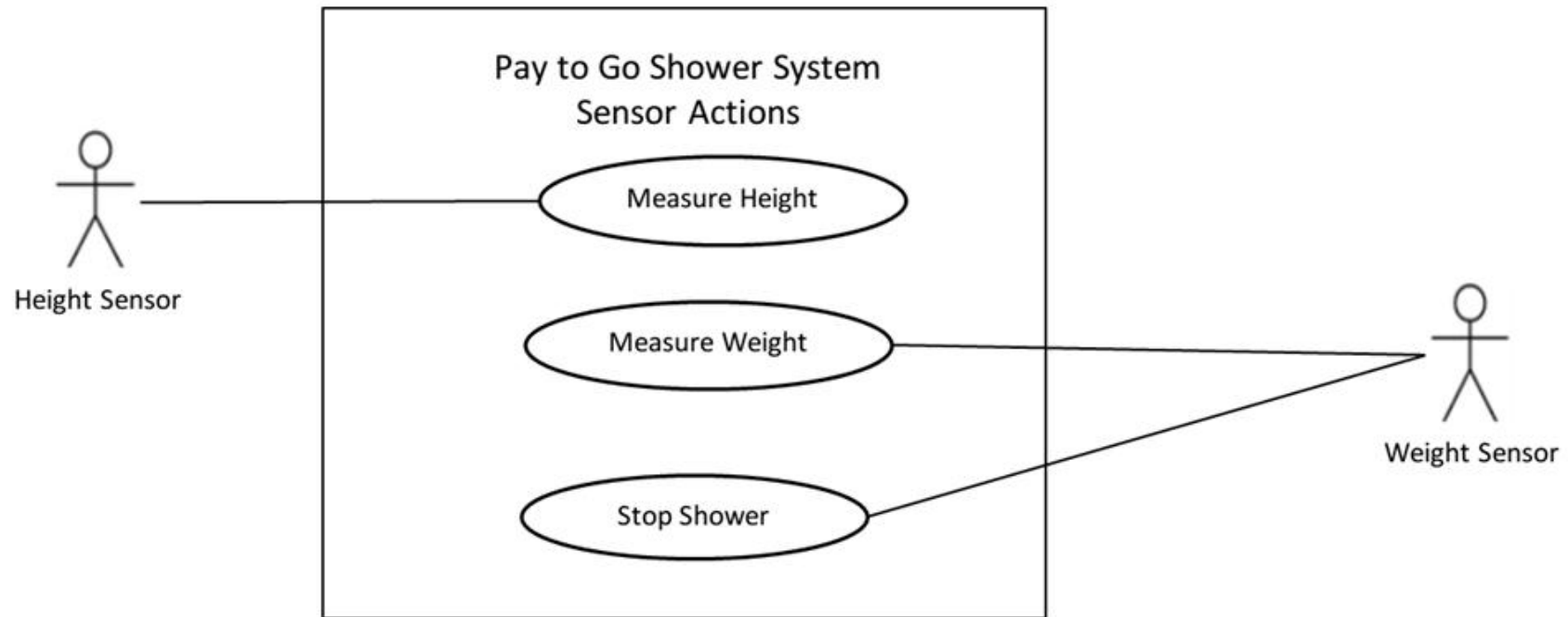
# Requirements Analysis

## Use Case Diagrams



# Requirements Analysis

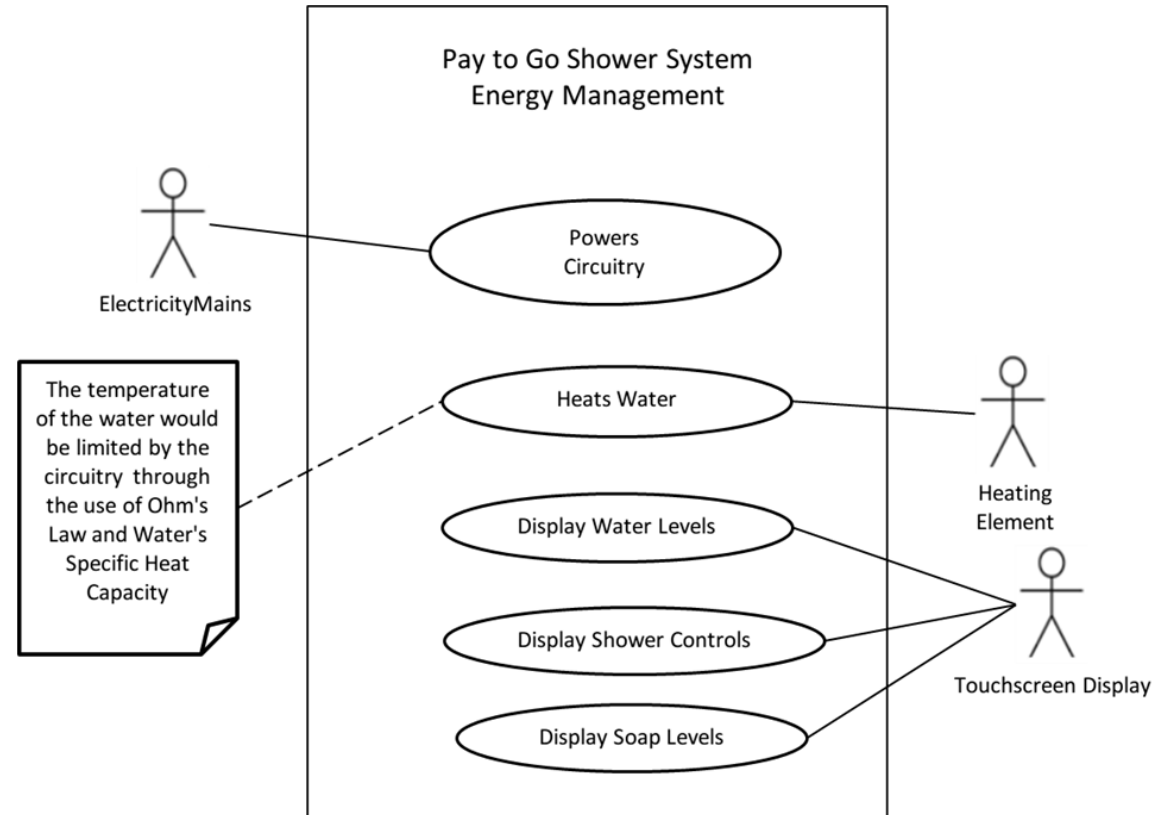
## Use Case Diagrams





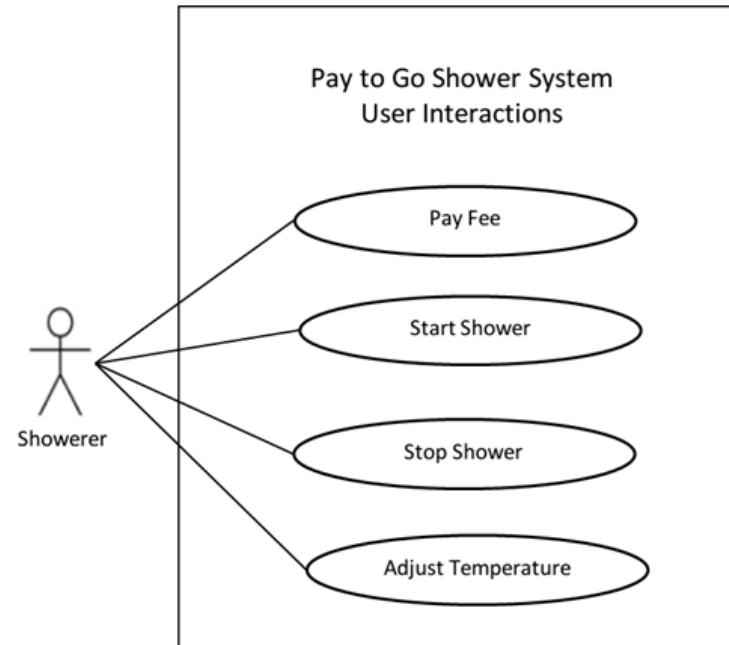
# Requirements Analysis

## Use Case Diagrams



# Requirements Analysis

## Use Case Diagrams



# Requirements Analysis

## Use Case Textual Descriptions

- **Use case name:** Pay Fee
- **Summary:** The shower transaction is paid for
- **Actor:** Showerer
- **Precondition:** User has enough money
- **Main Flow:**
  - 1. User uses smartphone to scan QR Code
  - 2. User completes payment process

# Requirements Analysis

## Use Case Textual Descriptions

- **Use case name:** Start Shower
- **Summary:** The User starts the Shower Process
- **Actor:** Showerer
- **Precondition:** User has paid their showering fee
- **Main Flow:**
  - 1. User steps into the shower.
  - 2. The weight sensor is activated
  - 3. The user's weight is determined
  - 4. The user's weight data is sent to the system

# Requirements Analysis

## Use Case Textual Descriptions

- **Use case name:** Stop Shower
- **Summary:** The User stops the Shower Process
- **Actor:** Showerer
- **Precondition:** The User has started the shower
- **Main Flow:**
  - 1. User starts the shower process
  - 2. User stops the shower process

# Requirements Analysis

## Use Case Textual Descriptions

- **Use case name:** Adjust Temperature
- **Summary:** The user changes the temperature of the water.
- **Actor:** Showerer
- **Precondition:** Fee has been paid.
- **Main Flow:**
  - 1. User pays fee
  - 2. User enters Shower
  - 3. User sets desired temperature of water

# Static Modelling

---

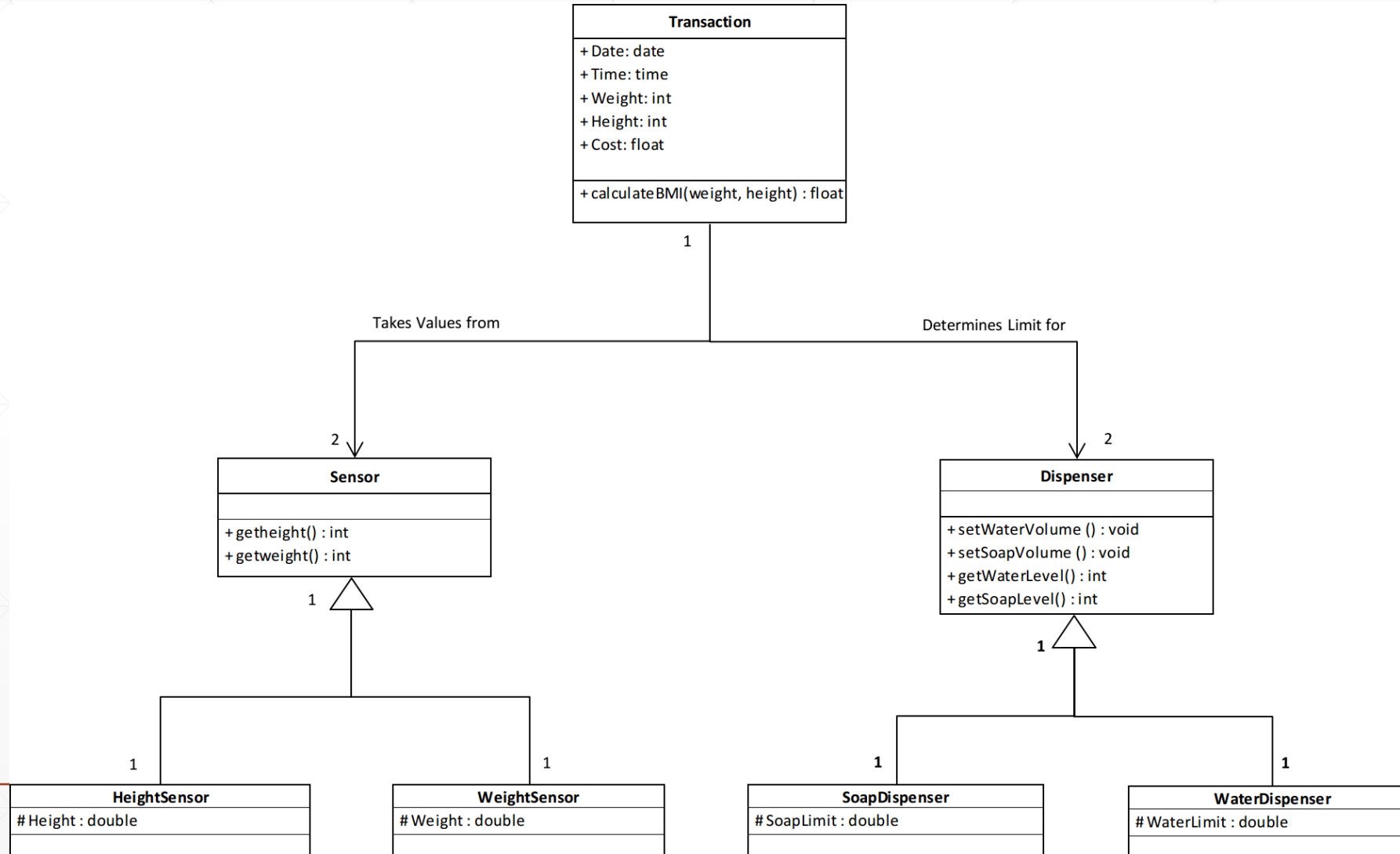
## What is Static Modelling ?

- Static Modelling represents a system in the form where it does change or vary with time. In other words it defines the classes of a system, the relationship between these classes and the functionalities of each class.



# Static Modelling

## Class Diagram



# Static Modelling

## Class Diagram Textual Description

- The class diagram shows an overview of the system. A transaction consists of a flat fee however the allocated soap and water to a user determined by the values returned by the height and weight sensors. By default an optimal amount for the average person is allocated to the dispensers upon payment as this prevents unnecessary wastage, however if the client's weight is too high or body mass index (BMI) calculated from the values returned by the height and weight sensors is above normal levels (overweight or obese) then more soap and water will be allocated to the dispensers to cover the larger surface area of the client. What is not used by the current client remains in the dispenser for the next client which promotes the reservation of resources.

# Dynamic Modelling

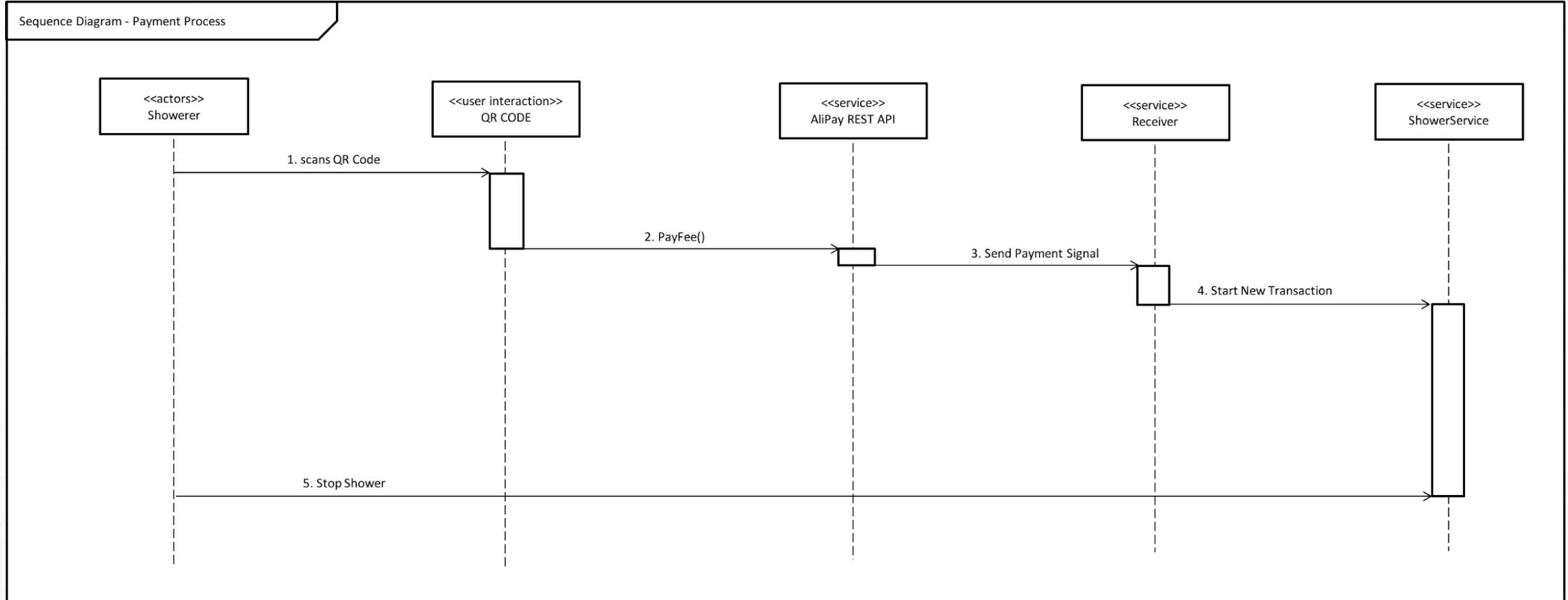
---

## What is Dynamic Modelling ?

- Dynamic Modelling Represents a System at runtime through the instantiation of each class and their interactions that conform to the use cases of the system.

# Dynamic Modelling

## Sequence Diagram



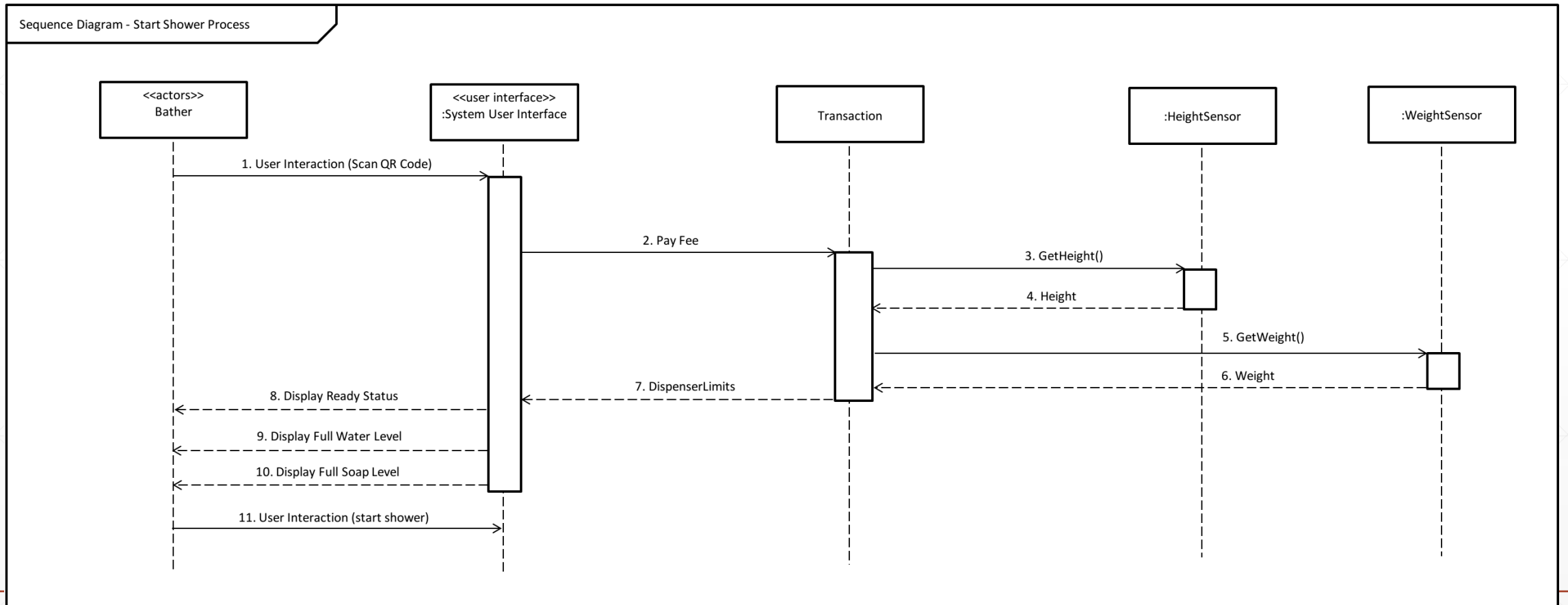
# Dynamic Modelling

## Sequence Diagram Textual Description

- The sequence diagram above the sequence of events involved in the payment process. Using the AliPay API the Pay to Go Shower System is able to process payment without any physical components. Once the payment has successfully been verified the shower will await the user's initiation.

# Dynamic Modelling

## Sequence Diagram



# Dynamic Modelling

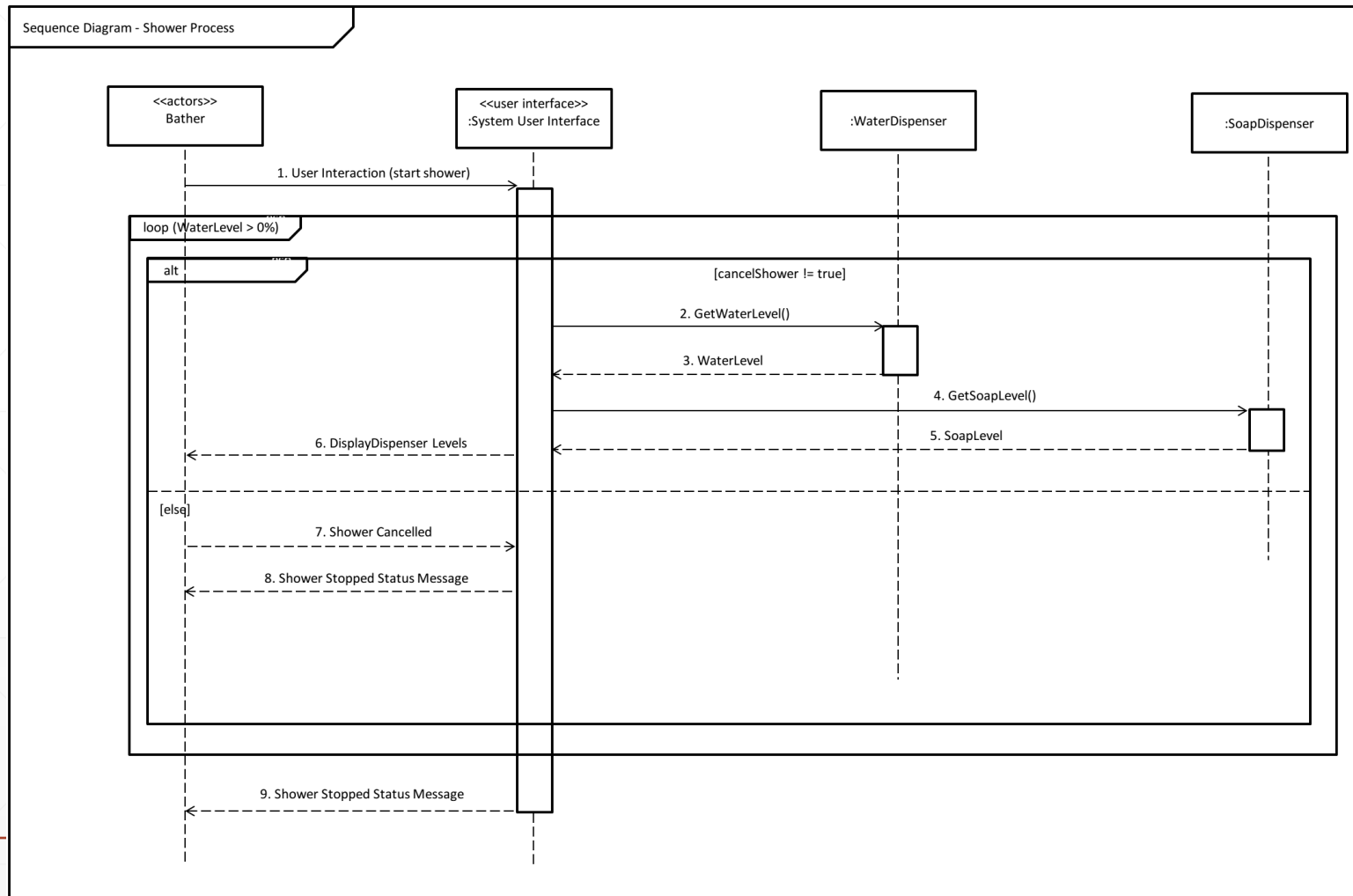
## Sequence Diagram Textual Description

- The Sequence Diagram on the previous slide shows the start shower process. Once the user has scanned the QR Code and payed the flat fee, the API associated with the QR Code will return a true boolean value to the control unit of the shower system which will trigger the Shower Ready event where the dispensers are filled based on the user's dimensions. The user can then start the shower process when ready.



# Dynamic Modelling

## Sequence Diagram



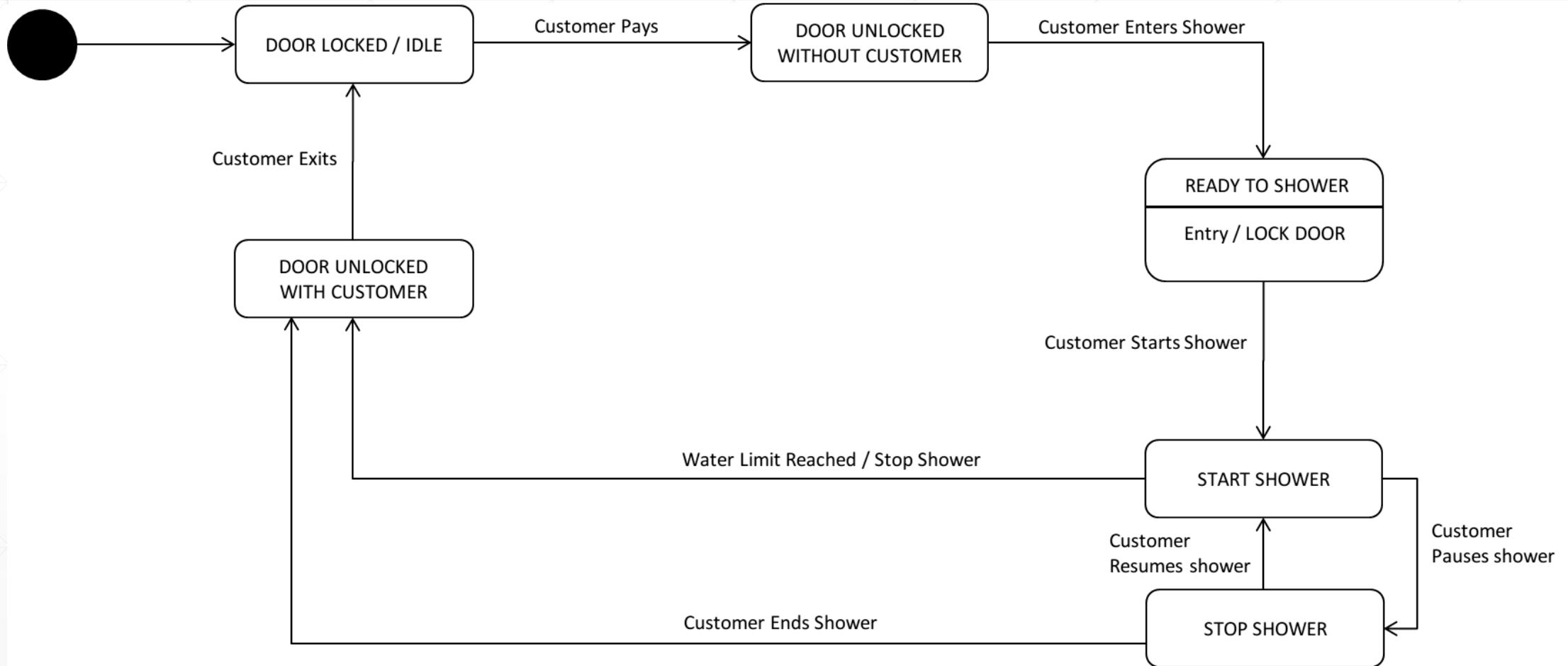
# Dynamic Modelling

## Sequence Diagram Textual Description

- The Sequence Diagram above shows the processes involved in the actual shower. The shower process will continue as long as the water dispenser contains water and if the user has not stopped the process. However the user may turn on and off the water during the process.

# Dynamic Modelling

## State Chart Diagram



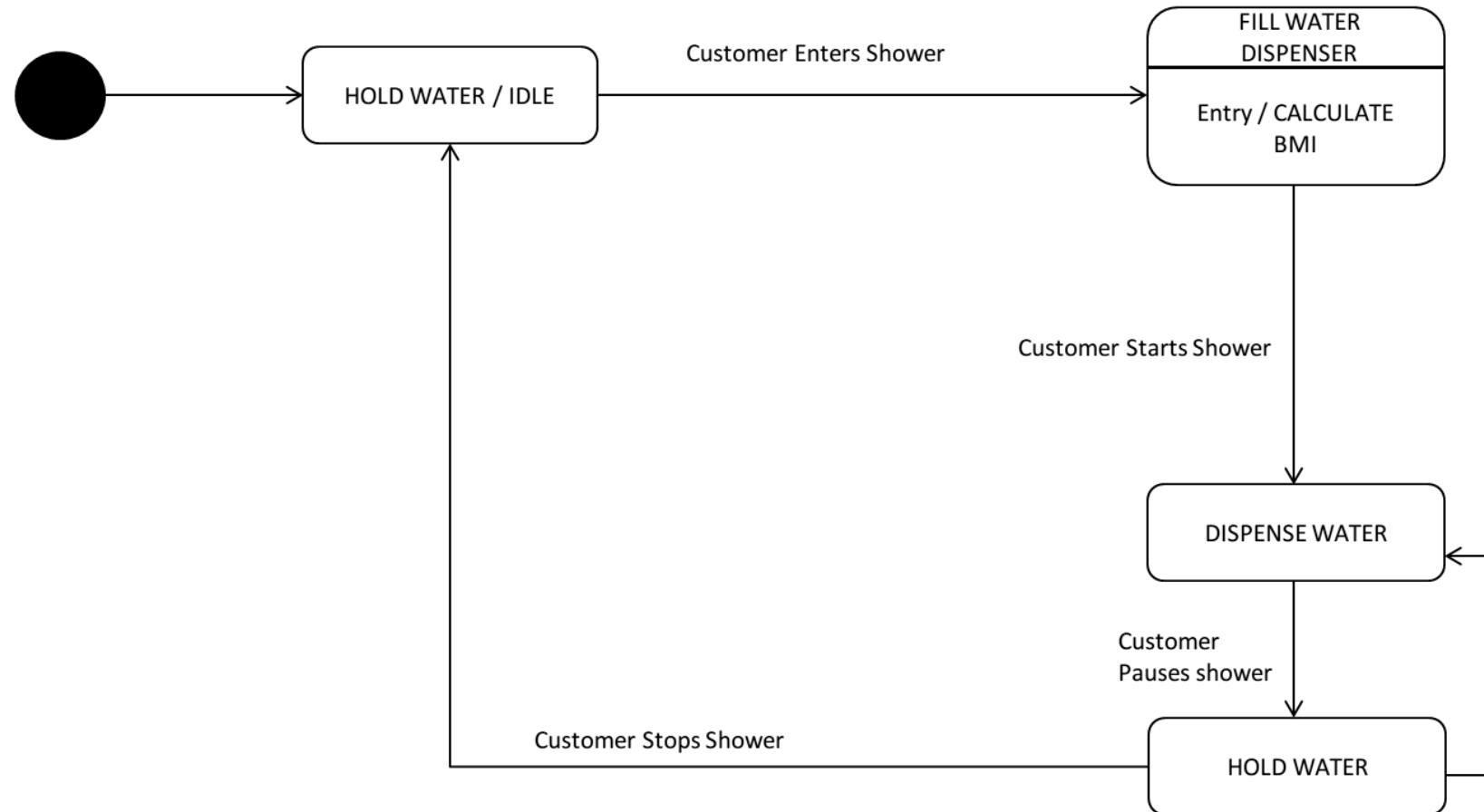
# Dynamic Modelling

## State Chart Diagram Textual Description

- The State Chart Diagram in the previous slide indicates the states of the entire Pay to Go Shower system. As shown during any stage of the shower the user is allowed to cancel the shower.

# Dynamic Modelling

## State Chart Diagram



# Dynamic Modelling

## State Chart Diagram Textual Description

- The State Chart Diagram in the previous slide indicates the states of the water dispenser for the Pay to Go Shower system.

# System Architecture

---

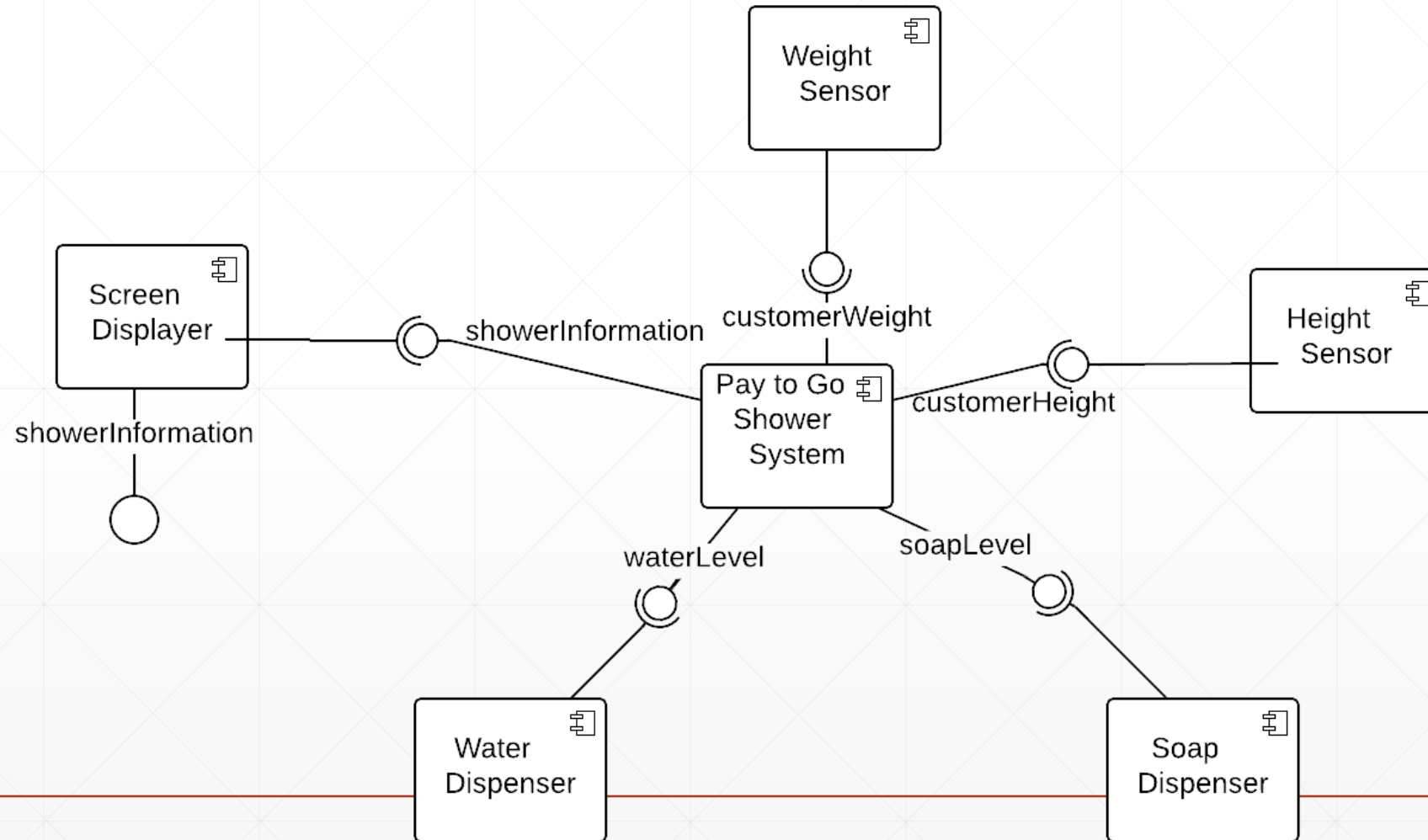
## What is System Architecture ?

- Software Architecture partitions a system structure into various subsystems and their related interfaces, where each subsystem is independent of other subsystems.



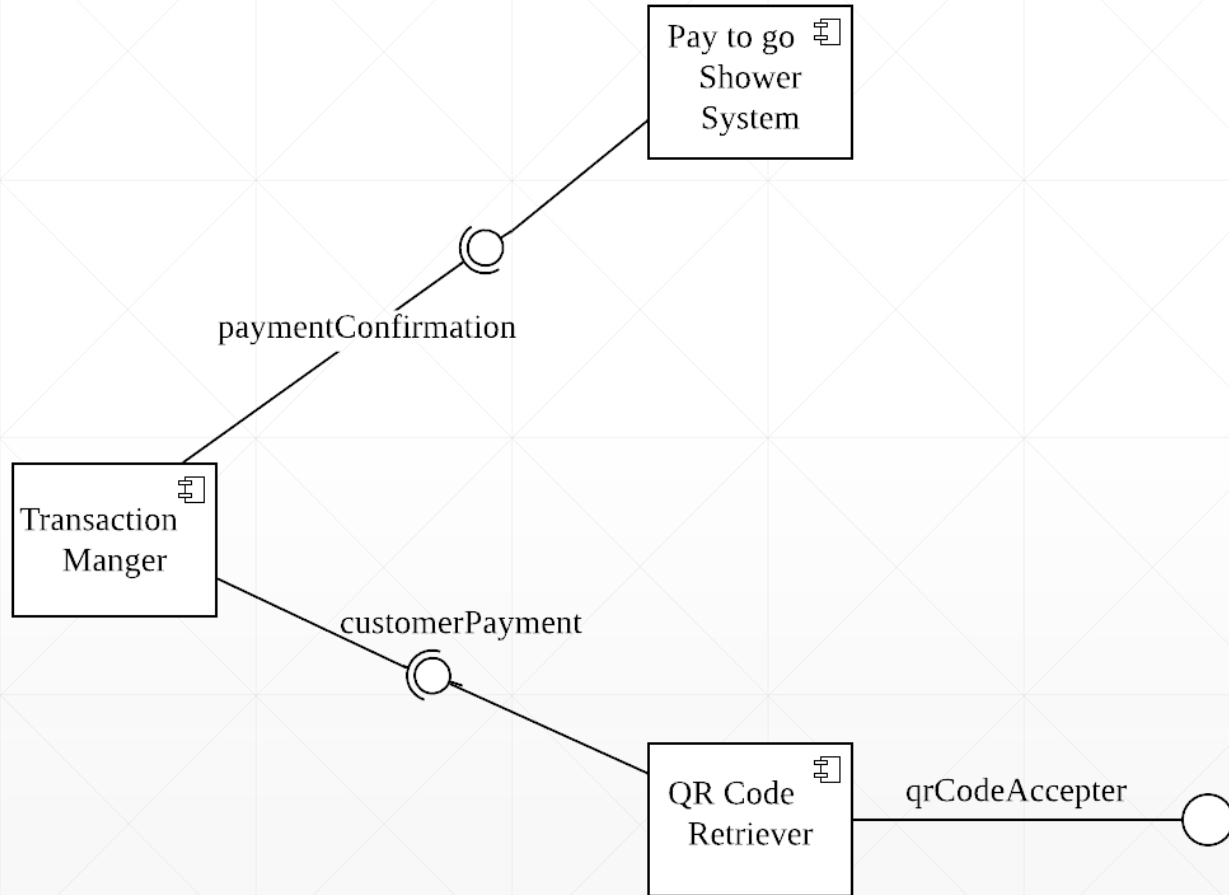
# Software Architecture

## Component Diagram



# Software Architecture

## Component Diagram



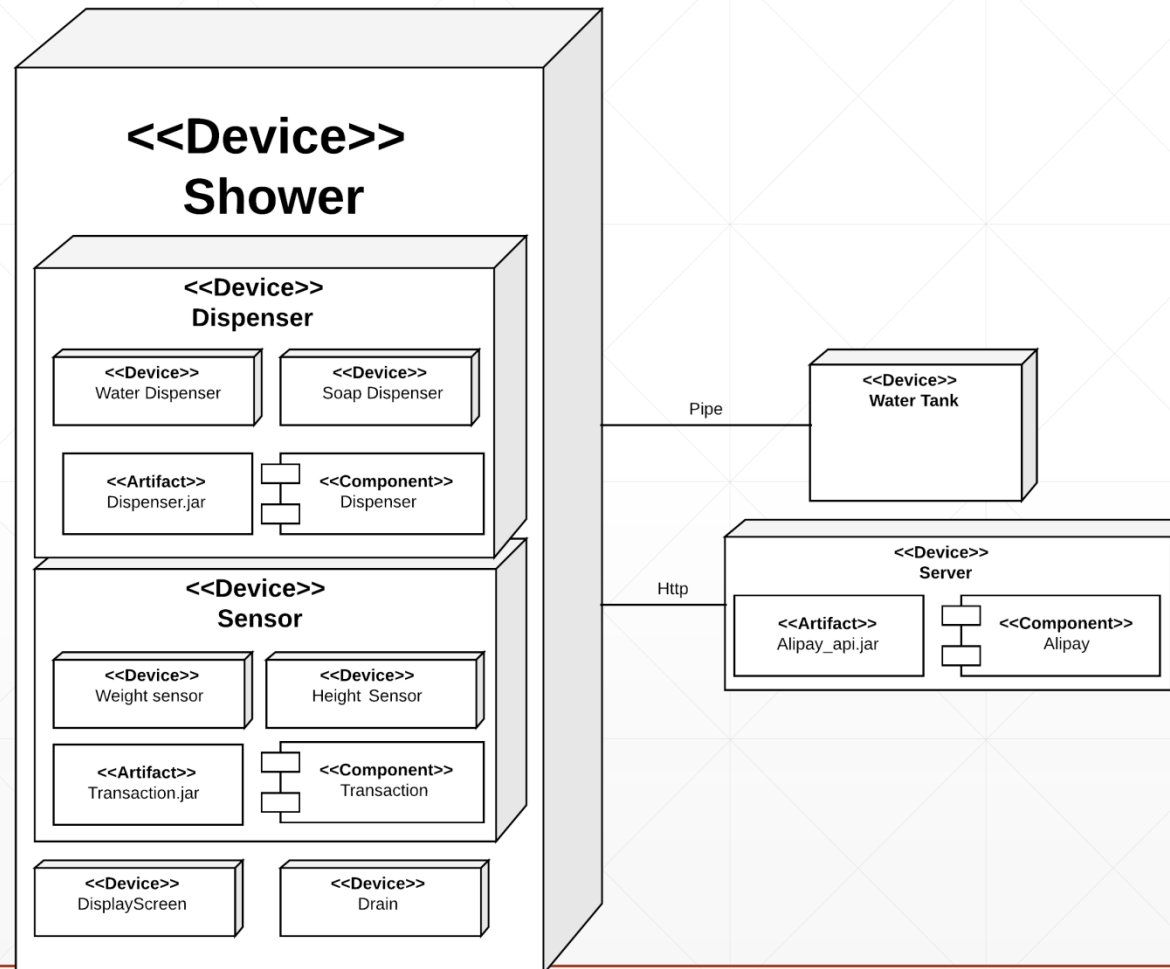
# Software Architecture

## Component Diagram Textual Description

- The component diagrams on the previous slides show all the components that is included in the shower system along with the required and provided interface for each component. The first component diagram shows the components involved in the showering process of the system whilst the second diagram shows the components involved in payment process of the system.

# Software Architecture

## Deployment Diagram



# Conclusion

---

# Conclusion

- Sometimes individuals do not have immediate access to showers in public. The Pay To Go Shower System enables users to freshen up regardless of their location. Users simply scan the QR Code and pay for the service. The AliPay API will then send a confirmation signal to the shower system which will cause it to start a new transaction. This is what makes the system unique as another system is currently not well known.

# Conclusion

## Contributions

Ottor	Sequence Diagrams (revision), Use Case Diagrams (revision), Use Case Narratives, Class Diagram (revision), State Chart Diagram (revision),
Nicoy	Use Case Diagrams, Class Diagram, State Chart Diagram, Component Diagram (revision), Sequence Diagrams (draft)
Kenneth	Deployment Diagram, Use Case Diagrams, Class Diagram, State Chart Diagram, Component Diagram (revision)
David	Component Diagram, Use Case Diagrams, Class Diagram, State Chart Diagram, Component Diagram (revision)

## System Deployment

The Pay to Go shower systems would be positioned in public places where individuals are more likely to want to take showers. Examples include the beach, or near a gym or courts (Individuals who take public transportation would want to freshen up before departing).



# Thank You.

---

# Content

- Introduction
- Requirement Analysis
- Class Design
- Dynamic Model Design
- Layout of the System
- Design of the Tables(If you use DB)
- Conclusion
- Q&A