

## Tutorial 04: Identifying Use Cases

1. For the Ticket Machine scenario below, determine the functional requirements, and list any assumptions you make.
2. Apply the User Goal Technique to identify use cases.
3. Apply Event Decomposition Technique and refine the list of use cases found.

The system to be developed is a GUI that allows a passenger to purchase a train ticket. Before boarding a train, a passenger must have a valid ticket. At the ticket machine, a potential passenger can select the number of zones s/he wishes to travel. The passenger can insert cash and/or coins to at least the value of the journey. The machine returns the ticket and any change, unless the ticket machine is out of change. A passenger can cancel a transaction at any time prior to a ticket being issued. If there is no passenger input after 30 seconds, the ticket machine cancels the transaction. If a ticket machine is out of order it will not accept input. A potential passenger can purchase a single journey ticket, a return ticket or a multi-rider ticket (which is an advance purchase of the equivalent of ten tickets for the same number of zones that carries a 10% discount).

### 1. Functional Requirements

1	The passenger should be able to select the number of zones they want to travel.
2	The passenger should be able to select the type of ticket (e.g., one-way, return).
3	The passenger should be able to purchase a multi-rider ticket at a discounted price.
4	The passenger should be able to cancel a transaction at any time prior to a ticket being issued.
5	The passenger should be able to pay cash for the ticket.
6	The machine should dispense any change for overpayment.
7	The machine should print a ticket for the zones and type selected.
8	The machine should automatically cancel a transaction after 30 seconds inactivity.
9	The machine should not accept any input if out of order.

### 2. User goal technique

	User	User goal	Pertinent Use Cases
1	Passenger	Select number of zones to travel	Select zones, Calculate payment
2		Pay cash for ticket	Pay for ticket, Calculate payment, Dispense change
3		Collect ticket	Collect ticket, Print ticket
4		Collect change	Collect change, Dispense change

	User	User goal	Pertinent Use Cases
5		Cancel ticket purchase	Cancel transaction, Check inactivity
6		Select type of ticket (e.g., single, return)	Select ticket, Calculate payment
7	<b>Maintenance</b>	Check serviceability of system	Check system status, Check remaining money, Display out of order

### 3. Event decomposition technique

	Event type	Event	Pertinent Use Cases
1	<b>External</b>	Selecting zones	Select zones, Calculate payment
2		Paying correct cash for ticket	Pay for ticket, Calculate payment, Dispense change, Print ticket
3		Overpayment of ticket price	Dispense change
4		Ticket purchase cancellation	Cancel transaction
5		Selecting ticket type (e.g., single, return)	Select ticket type, Calculate payment
6		Checking serviceability of system	Check system status, Check remaining money, Display out of order
7	<b>Temporal</b>	Transaction cancellation after 30 seconds of inactivity	Check inactivity, Cancel transaction
8	<b>State Event</b>	Calculating cash payment received	Calculate payment, Apply discount
9		Applying discount when bulk purchase threshold reached	Apply discount, Calculate payment
10		Payment received	Make payment, Calculate payment, Dispense change, Print ticket
11		Overpayment of ticket price	Calculate change, Dispense change
12		After every change dispensation	Check remaining money
13		Machine is unserviceable	Check system status, Display out of order