# Analysis - detail

Final versions of the requirements documents used for walk-throughs with the main stakeholders.

## System

**Design constraints** – Vendormatic #SJDY8383838

Useful when you don’t use just business terms e.g. web site is required, SQL Server is required, location must be…, hardware must be…

## Use Cases

### Summary of all use cases

A summary of the use cases in bullets or numbers serving like a table of contents. Summary focuses on inputs and outputs.

* UC1 – Purchase product with cash
* UC2 – Purchase product with credit card
* UC3 – Purchase product with Apple Pay

### UC1 – Purchase product with cash

Information about the use case, metadata

**Author** – Agile use case class 2/20/18

**Date written** – 2/20/18

**Actors** – Customer

The only role that initiates/triggers this flow of events. It does not include participants.

**Related systems** – HQ system

Sometimes called (supporting actors)

**System** – embedded

Would not be used if this were a business use case

**Priority** –

Get this from the priority chart.

**Type** – Goal

Options are: goal | partial goal | group of goals | group of partial goals

#### Pre-conditions

Rules for beginning this use case: state of system prevents usage, must be testable

* Machine must have at least one product in inventory
  + Inventory level = 0.
* Machine must not have jam sensor activated
* Machine must have cash to make change
  + RULE: Change making cash = 2 nickels, 3 dimes, 3 quarters

#### Course of Events

The sequence of tasks in conversation format between actor and system. Start each number with the system except the trigger and combine actor responses to system events. Rules are placed under the task unless they can be reused and then they are referenced and placed in a separate file.

1. The use case starts when the actor makes a selection by bin number (SD#1)
2. The system prompts (M#1) you for payment of transaction amount. The actor inserts cash.
3. The system counts and validates cash
   1. RULE: Cash is USD
   2. RULE: Maximum bill is $5
   3. RULE: follows Level 2 counterfeit detection <https://en.wikipedia.org/wiki/Currency_detector>
4. The system dispenses food item. The system checks jam.
5. The system records transaction. The system removes inventory.
6. The system asks HQ to save transaction.
7. The system displays thank-you message (M#2).
8. The actor takes item.
9. System starts time and after time-out shows welcome message (M#4).
   1. Time-out period – 10 seconds.

#### Extension points – optional

Sequences that return control back to the course of events after finished.

* Make change (#6) – System calculates amount to return and uses largest increments of bills and coins to return to customer.
* Display ad (#8) - .System shows ad (M#3)

#### Extension points – errors, exceptions

Errors occur at any point where there is a validation of a rule.

Errors occur at communication to other systems.

* Money not USD (#3a) – system prompts for USD (#ERR1), returns cash.
* System tilt indicator (any)

#### Post-conditions

Is it really important to review the necessary outcomes of this use case? If so, summarize here.

* Inventory has been updated.
* Money received is correct.

#### Notes/ Special Requirements

Any kind of quality, capacity, security, availability, disaster recovery information.

* Machine should be locked at all times.
* Inventory should sufficient light for selecting.
* Machine should be insulated for optimum temperature of
  + Candy - 40-75
  + Beverage – 30 – 35
  + Food can spoil – 33-41
* Power outlet less than 10 feet away.

### UC2 – Use case name

Information about the use case, metadata

**Author** –

**Date written** –

**Actors** –

The only role that initiates/triggers this flow of events. It does not include participants.

**Related systems** –

Sometimes called (supporting actors)

**System** –

Would not be used if this were a business use case

**Priority** –

Get this from the priority chart.

**Type** –

Options are: goal | partial goal | group of goals | group of partial goals

#### Pre-conditions

Rules for beginning this use case: state of system prevents usage, must be testable

#### Course of Events

The sequence of tasks in conversation format between actor and system. Start each number with the system except the trigger and combine actor responses to system events. Rules are placed under the task unless they can be reused and then they are referenced and placed in a separate file.

1. The use case starts when the actor …
2. The system responds by … The actor does something else.
3. The system validates something… The actor responds
   1. **RULE – Name:** The conditions that must be met for the validation.
   2. **RULE – Name:** The conditions that must be met for another validation.
4. The system responds by … The actor does something else.
5. The system does something…

#### Extension points – optional

Sequences that return control back to the course of events after finished.

* Sequence name (#) –

#### Extension points – errors, exceptions

Errors occur at any point where there is a validation of a rule.

Errors occur at communication to other systems.

* Sequence name (#) –

#### Post-conditions

Is it really important to review the necessary outcomes of this use case? If so, summarize here.

#### Notes/ Special Requirements

Any kind of quality, capacity, security, availability, disaster recovery information.