# Strategy

## Problem statement

The access to banking services is restricted to bank hours affecting the majority of our bank customers by having to wait to the next business day, adding more staff or hours, or finding another solution. Our solution will allow the bank to provide services at any hour and in locations where a bank branch is not located reducing overhead costs. Replacing teller and not having bank open with automation of transaction external to bank reduces overhead and improves customer experience.

## Constraints

Infrastructure, technology, laws, etc.

## Stakeholders

I. Hafmunee – Bank president

Gee Kee Tekhed – CIO of bank

# Analysis – high level

Post-elicitation stage done with all analysts after the initial interviews and document reviews. Do this first.

## Actors

* Customer
* Operator

## Use case names

### System use cases

* Transfer funds
* Check balance
* Withdraw cash
* Deposit funds

## Business use case

* Start Stop machine with activity in middle

### Partial Business use cases

* Reload machine cash
* Reload receipt paper
* Retrieve cards
* Reset machine
* Do routine maintenance on machine

## Group names - system

* Do account transaction – transfer, balance, withdraw and deposit

## Grouping of partials found

* Service machine – reload cash, paper, get cards, reset, maintenance.

# Project management

## Prioritization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use case name | Business /  Market  usage %  (3=100-67  2= 66-34, 1=0-33%) | Business / pricing value ( 3=exec/high,  2=mgmt./med.  1=staff/low) | Result = usage \* value | In a group, ask for the top 3 and  count the  total. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Project Iterations

Move the use cases with priorities down to a schedule so that you can see what kind of sprints (Agile) you will have.

### set up

There needs to be a software stage to set up and get all the pieces initially running as a framework. It’s called stubbing out the system or mocking the system depending on what you need.

### 1st

### 2nd

### 3rd

### 4th

# Analysis documents

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

## 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.

#### System use cases

Only those use cases which start after the software is running and an actor directly interacts with the system.

Grouped use cases are either system or business use cases and can be groups of value/goal level use cases or groups of partial use cases like used in this example. Also common in this section are the Manage Entity type of use case which expands into the CRUD use cases. e.g. Manage Account = Create Account, Read Account (Display details of account), Update Account, Delete Account.

#### Business use cases

Any use case that does not completely exist as an interaction between software and the actor.

### UC1 - Withdraw Cash

Information about the use case, metadata

**Author** – BA class of

**Date created** -

**Date revised** -

**Actors** –

**Other systems** –

Sometimes called (supporting actors)

**Level** –

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

**Type** –

Options are: system, business

**Design constraints** –

**Priority** -

#### Value to sponsor (goal)

#### Pre-conditions

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

#### Course of Events

#### Extension points – optional

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

#### Alternative flows – errors, exceptions

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

Errors occur at communication to other systems.

#### Notes/ Special Requirements

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

## Data Dictionary (separate file)

## Rules (separate file)

Reusable rules referenced in multiple use cases. Referenced by R#1, R#2, etc.

1 –

## Glossary (separate file)

## Text messages (separate file)

ET#1 –

## Screens (separate file)

SD#1 -