# 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, and 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. (5) |
| Transfer funds |  |  |  | 2 |
| Check balance |  |  |  | 2 |
| Withdraw cash |  |  |  | 5 |
| Deposit funds |  |  |  | 5 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 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

Withdraw Cash

### 2nd

Deposit Funds

### 3rd

Transfer

Check balance

# 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.

* **Withdraw Cash**

A customer swipes bank card, asks for cash, gets approved, receives and receipt.

* **Deposit Funds**
* **Transfer**
* **Check balance**

#### Business use cases

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

### GUC1 – Do Transaction

Information about the use case, metadata

**Author** – BA class of 6/16/2017

**Date created** – 6/16/17

**Date revised** -

**Actors** – Customer

**Other systems** – Bank

Sometimes called (supporting actors)

**Level** – Group of goals

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

**Type** – System

Options are: system, business

**Design constraints** – SjfhDJFyuY\*88558j ATM chassis

**Priority** - 5

#### Value to sponsor (goal)

Reduce overhead costs by offering a similar automated process.

#### Pre-conditions

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

* Cash in machine must be equal to or greater than **Max amount of withdrawal (7c)**.

#### Course of Events

(Current state of machine is showing a greeting on the idle screen.)

1. The use case starts when the actor inserts their bank card.
2. The system prompts for PIN (ET#1). The actor enters PIN.
3. The system requests bank to validate user and retrieves all account information for that user.
4. The system makes a bank communication log entry.
5. The system prompts the actor with a main menu screen (SD#1).
6. <<include>> GP1 **Do Secure Transaction**
7. The system prompts user for another transaction (ET#?). The actor declines.
8. The system prints receipt.
9. The system ejects the card. The system prompts the user to take card, receipt and thank-you (ET#3). The actor takes the receipt. The actor takes the card.
10. The system returns to idle state screen after a transaction end delay
    1. RULE – Transaction end delay – 15 seconds.

### Error flows

### Alternate flows

(#7) Do another transaction – any of GP1 Do Secure Transaction

### GP1 - Do Secure Transaction

**PUC1 Withdraw Cash**, PUC2 Transfer Funds, PUC3 Inquire …, PUC4 Deposit Funds..

These use cases are secure transactions and show on the main menu.

### PUC1 – Withdraw Cash

Information about the use case, metadata

**Author** – BA class of 6/16/2017

**Date created** – 6/16/17

**Date revised** -

**Actors** – Customer

**Other systems** – Bank

Sometimes called (supporting actors)

**Level** – Partial

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

**Type** – System

Options are: system, business

**Design constraints** – SjfhDJFyuY\*88558j ATM chassis

**Priority** - 5

#### Course of Events

1. The actor selects Withdraw Cash option.
2. The system prompts for account to withdraw from (ET#2). The actor select account.
3. The system prompts for withdrawal amount. (SD#2) The actor enters amount to withdraw.
4. The system validates the amount.
   1. RULE – Available funds – Requested amount is less than or equal to accounts’ available funds.
   2. RULE – Daily withdrawal time– Checking account transactions are limited to the **Max amount of withdrawal** per 24 hour period.
   3. RULE – Max amount of withdrawal - $500 per account.
5. The system requests the bank to debit account and get receipt information (R#1). The system makes a bank communication log entry.
6. The system debits its Cash in machine amount by the withdrawal amount.
7. The system dispenses the cash. The system prompts the actor to take cash(ET#?). The actor takes the cash.
8. The system makes a cash dispensing log entry.

#### 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)

* **Account information** –
  + **Card #**
  + **Balance**
  + **Available funds**
  + **Transactions within last 24 hours**
* **Log entry**
  + PIN info – never recorded.
* PIN – digits to authorize users’ bank cards stored by bank system
  + **Validation rule –** max 8 digits

## 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)

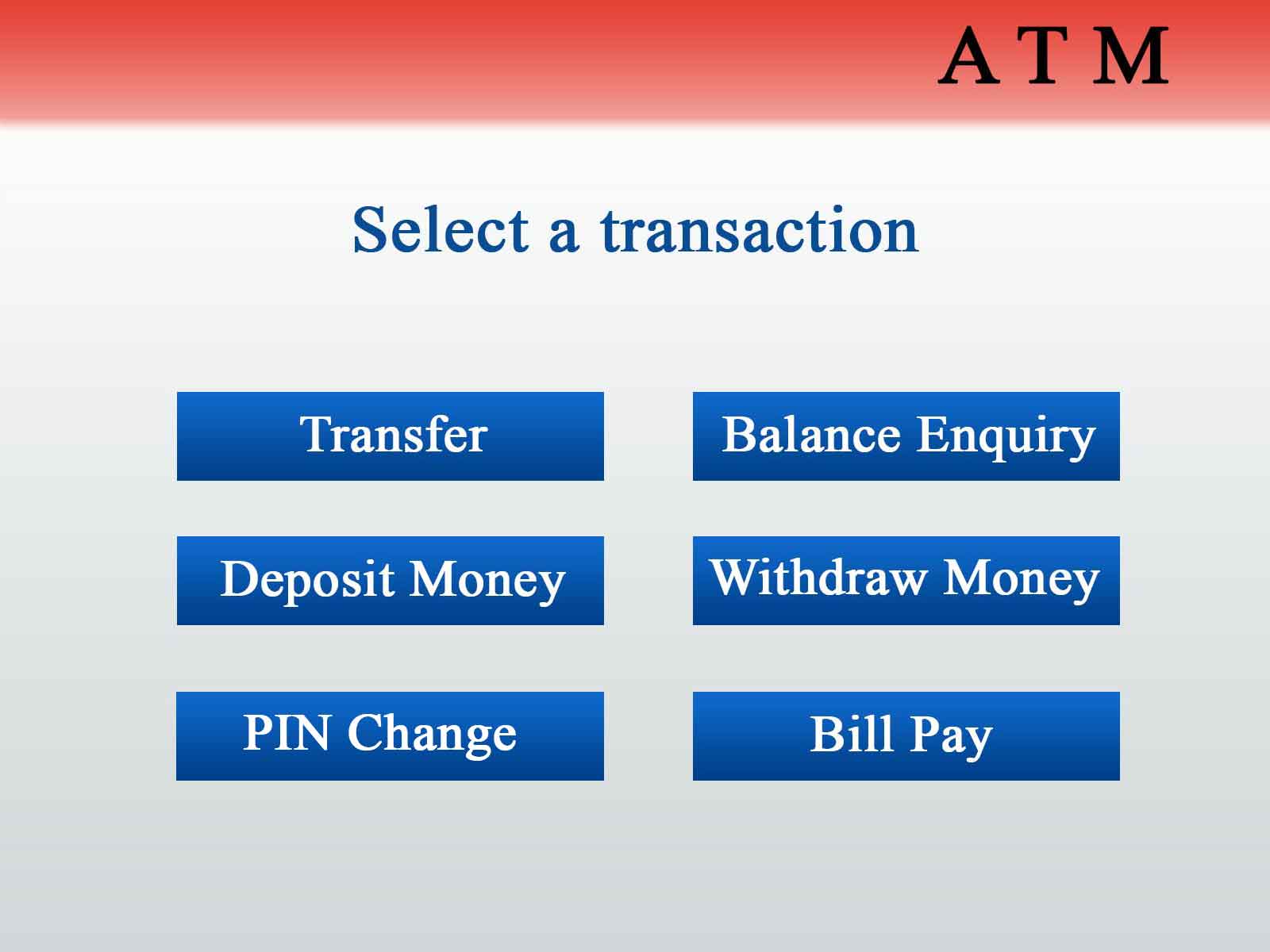
ET#1 – **PIN prompt**: Please enter your PIN.

ET#3 - User take card and thank-you – Please take card and thanks for stopping by.

## Reports

R#1 - receipt

## Screens (separate file)

SD#1 –

**SD#2 – Prompt for amount.**