# 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

System

Operator

## Use case names

### System use cases

Withdraw cash

Request balance

Transfer funds

Make a deposit

Display ads and info

### Business use cases

Collect deposits

Load cash in machine

Start machine

Shut off machine

Stock envelopes

Replace paper

Replace ink

## Use case groups found

* Perform maintenance
* Do transaction

## Partial goal names found

* Start session
* End session

# 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. |
| Withdraw cash |  |  |  | 7 |
| Request balance |  |  |  | 3 |
| Transfer funds |  |  |  | 4 |
| Make deposit |  |  |  | 7 |
| Display ads, info | 1 | 3 | 3 | 3 |

## 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

Transfer Funds

Make deposit

### 3rd

Request balance

Display ads

### 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.

* **Withdraw Cash** – customer starts session, requests cash, gets cash and receipt.

#### Business use cases

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

### UC1-S – Do secure session

Information about the use case, metadata

**Author** – BA class of 10/5/17

**Date created** - 10/5/17

**Date revised** -

**Actors** – Customer

**Other systems** – Bank

Sometimes called (supporting actors)

**Level** – Goal (includes group)

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

**Includes –** UC2-S Do Secure Transaction

**Extends -** UC2-S Do Secure Transaction

**Type** – System

Options are: system, business

**Design constraints** – ATM chassis NexGen S9484848, ATMNet

**Priority** – (could be prioritized based on need for security had we known about it before)

**State** - future

options are baseline/current , future, Phase 1, 2, 3, …, proposed future state/in negotiation

#### Value to sponsor (goal)

Provide security

#### Pre-conditions

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

#### Course of Events

Bold words indicate data dictionary items.

1. The use case starts when the actor inserts card.
2. The system reads the card. The system prompts actor for **PIN** (ET#1). The actor enters PIN and confirms.
3. The system requests bank to validate PIN with card data. The bank approves and sends **account** info (and maybe **customer** (DD) info?). The system records **communication entry** (DD).
4. The system displays main menu (SD#1).
5. **<<include>> UC2-S Do Secure Transaction**
6. The system prompts the actor for another transaction (SD#6). The actor declines.
7. The system prompts the actor if they want a receipt (SD#5). The actor accepts.
8. The system prints the receipt.
9. The system prompts the actor to take receipt (SD#7). The actor takes receipt.
10. The system ejects the card. The system prompts the actor to take the card (SD#8). The actor takes the card.
11. The system displays a thank you screen (SD#9). The system waits the amount of time before idle screen is shown (DD). The system displays idle screen (SD#10).

#### Extension points – optional

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

**Secure transaction** (6) – see **UC2-S – Do secure transaction**

#### Alternative flows – errors, exceptions

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

Errors occur at communication to other systems.

* **Cancel button** (2, 4) – The system displays cancel screen (SD#111). The actor cancels. The use continues at step #10.

#### Notes/ Special Requirements

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

### UC2-S – Do secure transaction

Information about the use case, metadata

**Author** – BA class of 10/5/17

**Date created** - 10/5/17

**Date revised** -

**Actors** – Customer

**Other systems** – Bank

Sometimes called (supporting actors)

**Level** – Group

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

**Included by:** UC1-S – Do secure session

**Extended from:** UC1-S – Do secure session

**Special use cases** – UC2-S1 Withdraw Cash, UC2-S2 Request Balance, UC2-S3 Transfer Funds, UC2-S4 Make Deposit

**Type** – System

Options are: system, business

**Design constraints** – ATM chassis NexGen S9484848, ATMNet

**Priority** – n/a

**State** - future

options are baseline/current , future, Phase 1, 2, 3, …, proposed future state/in negotiation

#### Value to sponsor (goal)

Reduce cost of transactions, customer experience, resource management, improved availability

#### Course of Events

None

### UC2-S1 – Withdraw Cash

Information about the use case, metadata

**Author** – BA class of 10/5/17

**Date created** - 10/5/17

**Date revised** -

**Actors** – Customer

**Other systems** – Bank (see SSD Withdraw Cash)

Sometimes called (supporting actors)

**Level** – Partial

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

**Grouped by:** UC2-S – Do secure transaction

**Type** – System

Options are: system, business

**Design constraints** – ATM chassis NexGen S9484848, ATMNet

**Priority** – 7

**State** - future

options are baseline/current , future, Phase 1, 2, 3, …, proposed future state/in negotiation

#### Pre-conditions

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

System must have at least the **daily withdrawal limit** (DD) in cash on hand (DD).

#### Course of Events

1. The actor selects Withdraw Cash.
2. The system prompts actor for account to use (SD#2). The actor selects account to use.
3. The system prompts actor for amount to withdraw (SD#3). The actor enters amount and confirms.
4. The system validates the amount of withdrawal.
   * RULE – **Sufficient available funds** - Amount to withdrawal is equal to or less than account balance.
   * RULE – **Daily withdrawal limit** – Amount to withdraw is
     + less than **daily withdrawal limit** (DD)
   * RULE - **$20 increment** – Amount must be evenly divisible by 20.
5. The system requests the bank to record the **transaction** (DD). The bank acknowledges the message. The system logs the communication.
6. The system dispenses cash. The system logs the cash dispensing. The cash on hand is decremented by amount dispensed.
7. The system prompts the actor to take cash (SD#4). The actor takes the cash.

#### Alternative flows – errors, exceptions

* **Cancel button** (?) – The system displays cancel screen (SD#111). The actor cancels. The use continues at step #10.
* **Insufficient funds** (4) - The system displays insufficient funds screen (SD#112). The actor confirms. The use continues at step #4 (main menu).

## Data Dictionary (separate file)

* **Constants**
  + **daily withdrawal limit** = $500
    - measured per 24 hour period starting at midnight based on last transaction location time zone and DST…. You work it out…
  + seconds before idle screen shows after transaction end
    - 5 seconds
* Account
  + PIN
  + number
  + type
  + balance
  + daily amount withdrawn
* **ATM**
  + cash on hand
* Card data
  + account number
  + expiration date
* Bank message
  + Transaction
* Log entry
  + Transaction
  + date time
* Transaction
  + Account
  + amount
  + date/time

## Rules (separate file)

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

1 –

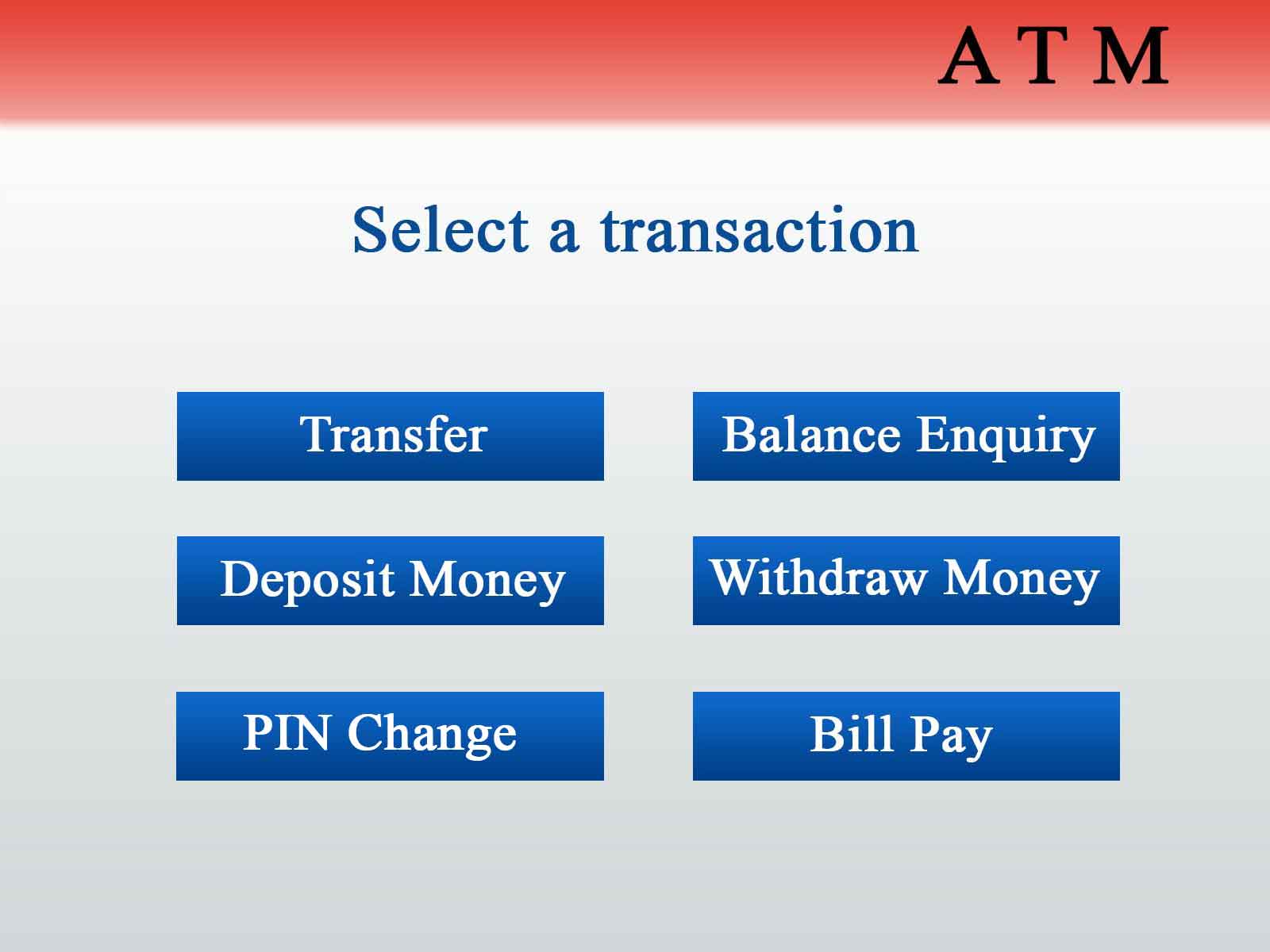
## Glossary (separate file)

## Text messages (separate file)

ET#1 Enter PIN – Please enter your PIN

## Screens/images (separate file)

SD#1 – Main menu



SD#2 – Select account

## Reports (separate file)

R#1 -