Appendix B - **Example Of The Process**

In this section, I give an example of the process from the point of view of the business analyst. It shows the work performed by the business analyst and their observations of work performed by other roles.

The Automated Menu Ordering System has been updated to include the new pay bill functionality (exactly as described in the process activity examples). This functionality has been deployed to the customer. As a result of the pay bill functionality being made available to the restaurant's customers, the restaurant owner has identified a new high priority business need. The customer has to be able to pay their bill using several payment methods.

In this example, the AMOS is released to the customer on a quarterly basis. Software builds are developed on a monthly sprint cycle. A product backlog for the AMOS contains enough user stories for the next release.

Elicit Business Needs

The product owner captures the new high-priority business need with an epic in the product backlog.

The business analyst interviews the stakeholders in order to identify the cause of this new business need. They discover that when groups of people are seated at the same table, individuals only want to pay for the items that they ordered.

A user story for this epic could be the following.

Epic User Story

Split Bill Feature	
The bill for a single order can be split over several customers. Each customer may use a different payment method.	
For	restaurant owner
who wants	customers to be able to be able to pay for their bill using several different payment methods
so that	each person pays for the items that they ordered
unlike	wait staff understanding the way the bill is to be split, taking the bill to the cashier to have it separated into several bills according to the customer's instructions and returning those bills to the customer table
Additional information	
Wait staff	Split bills include a tip field.
Cashier	Split bills include a tax field
Restaurant Owner	Up to 8 customers may be seated at a regular table

Figure 85 – Split Bill Epic

The epic describes how split bills are handled by the restaurant today. This process is modeled using a business use case.

Business Use Case

Figure 86 shows the Split Bill business use case with 2 actors.

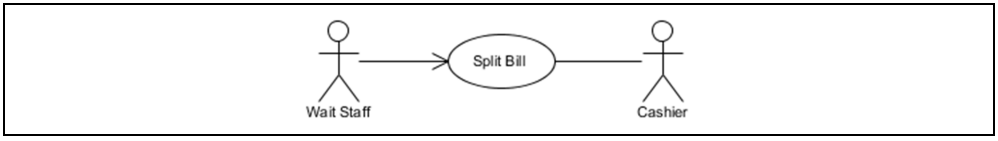

Figure 86- Split Bill Business Use Case

The diagram shows wait staff as the primary actor who gains benefit from the business use case. The cashier is a secondary actor, who contributes to this process.

Business Use Case Activity Diagram

The Split Bill feature expands on the Pay Bill activity diagram to add alternate flows for handling a customer request to split the bill over several payment methods. (See Figure 23 for the Pay Bill activity diagram.)

Figure 87 shows how the split bill feature impacts the Pay Bill business activity diagram.

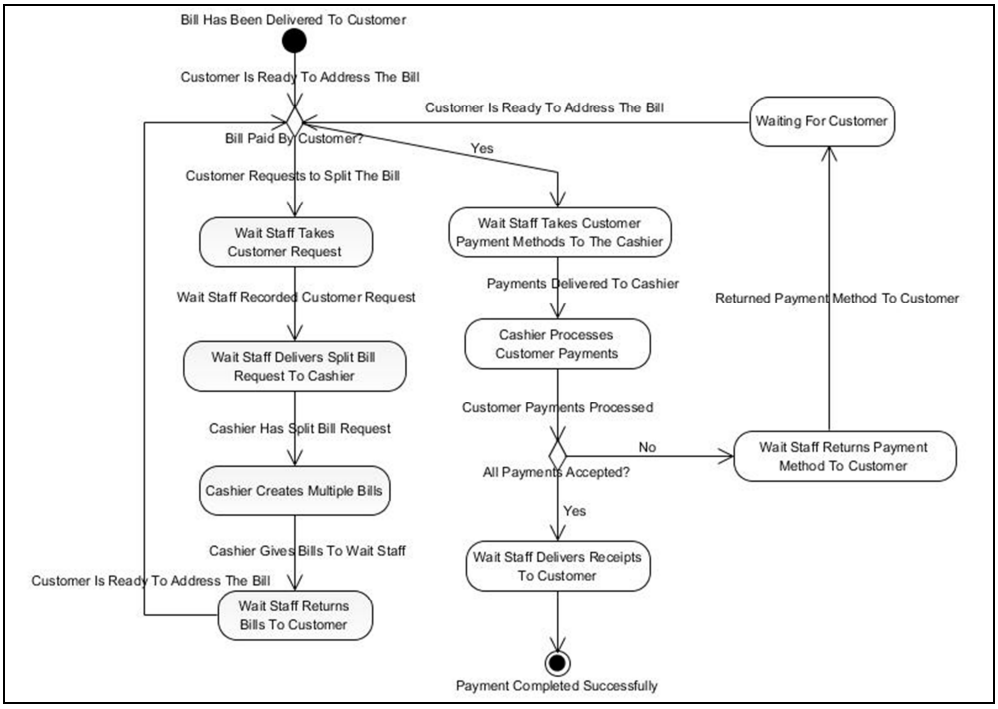

Figure 87 – Split Bill Activity Diagram

- **Unlike the system use case, the primary actor does not need to initiate the process.**

The activity diagram shows a subset of the Pay Bill business use case. The split bill activity is shown as an alternate flow (to the left of the diagram) that branches when the customer is ready to address their bill.

The existing Pay Bill steps have been modified so that they refer to multiple payments.

The activity starts when the customer has their bill. The customer may now request to split the bill. The split bill steps are described in the Pay Bill use case as an alternate flow. The Pay Bill main flow steps have been modified to accommodate multiple payments.

Preconditions:

- The customer has their bill.

Main Flow

1. Customer is ready to pay their bill
2. Wait staff takes the customer payments
3. Wait staff delivers the customer payments to the cashier
4. Cashier processes the customer payments

5. Cashier gives receipts for customer payments, to the wait staff
6. Wait staff delivers the customer payment methods, receipts and any monetary change to the customer

The use case ends.

Postcondition:

- The bill was paid.

A.1 Alternate Flow - Split Bill

7. At step 1, Customer makes a request from the wait staff to split the bill
8. Wait staff understands the details of the customer request
9. Wait staff takes the customer request to the cashier
10. Cashier creates multiple bills according to the customer request
11. Cashier gives the bills to the wait staff
12. Wait staff returns the bills to the customer
13. Customer indicates that they are ready to address their bill
14. The use case continues from 1

A.2 Alternate Flow– (The) A customer payment is rejected

15. At step 4, 1 or more of customer payments cannot be processed
16. The wait staff returns the unprocessed customer payment methods to the customer
17. Customer indicates that they are ready to address their bill
18. The use case continues from step 1

In the use case steps, a *split bill* alternate flow has been added and the *customer payment is rejected* alternate flow has been modified to show that any payment method that is rejected is returned to the customer.

Logical Model

Once the product owner agrees with the business analysts' understanding of the need, a good place to start looking for impacts to the system is the logical model.

Figure 88 is shows the current relationships between the data in the product. The current system only allows 1 payment per customer order at a table.

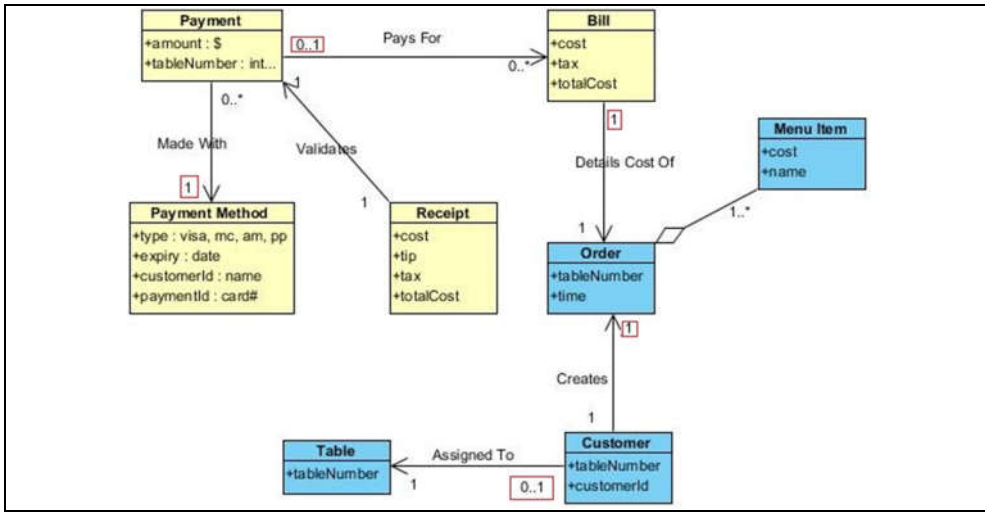

Figure 88- Split Bill Options Class Diagram

By analyzing the logical model, the business analyst identifies a number of different options that satisfy multiple payments for a single customer order.

I have highlighted five relationship cardinalities in the diagram with a red box. These highlights identify alternate changes to a relationship that allow a customer to make several payments. These 5 options are:

1. A payment may be Made with more than 1 payment method:

By changing the *1* on the Made With relationship between the Payment and the Payment Method, to a *many* relationship.

2. A bill may be paid using several payments:

By changing the *0..1* on the Pays For relationship between the Payment and the Bill, to a *0..many* relationship.

3. An order may be split into several bills:

By changing the *1* on the Details Cost Of relationship between the Bill and the Order, to a *many* relationship.

4. A customer can be allowed to place several orders:

By changing the *1* on the Creates relationship between Order and Customer, to a *many* relationship.

5. A table can contain several customers at the same time:

By changing the *0..1* relationship between Table and Customer to a *0..many* relationship.

Implementation Options

The business analyst documents and presents the implications of each of these options to the stakeholders. For the sake of the following examples, it is assumed that up to 8 payment methods may be used to pay for a single order. (I.e. the word *many* is replaced with the number 8, where it used in a relationship.)

- **I will often write up the results of an analysis in a set of PowerPoint slides. The slides detail each option with its pros and its cons.**

For each of these 5 options, I provide a sequence diagram and its description. The example sequence diagrams show the normal sequence of events when a bill split into 2 bills. Alternate and error paths are not considered.

- **Showing all alternate paths for 5 options is beyond the scope of this example and left as an exercise for the reader.**

Option 1: A payment may be made with more than 1 payment method

Figure 89 contains a sequence diagram that shows how this option affects interfaces to the user and to backend systems.

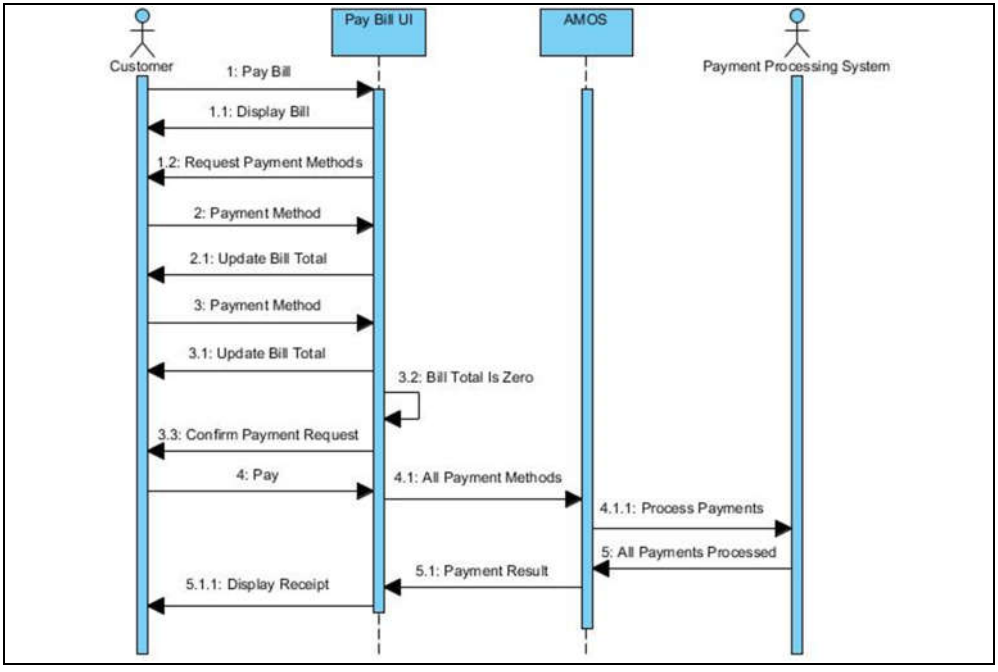

Figure 89 – Split Bill Option 1 Sequence Diagram

When the customer makes a request to pay their bill, the system displays the bill and requests that the customer enter up to 8 payment methods. The customer completes from 1 up to 8 of these payment options by entering a payment method and the amount to apply to each. As each payment method is entered, the bill total is updated to reflect the amount left to pay. When the

total payment amount from entered payment methods is the same as that on the bill (Bill Total Is Zero), the customer is prompted to confirm payment. The customer confirms payment and a request to process the payments is sent to the payment processing system.

Note that if the payment processing system does not accept multiple payment methods in a single request, then up to 8 payment requests must be sent to the system. Only when all payments have processed successfully is the payment complete and the customer informed that their bill is paid.

Notes:

- The system will need to handle the situation where a payment fails to process.
- The system could validate payment methods prior to processing a payment.

Option 2: A bill may be paid using several payments

Figure 90 contains a sequence diagram that shows how this option affects interfaces to the user and to backend systems.

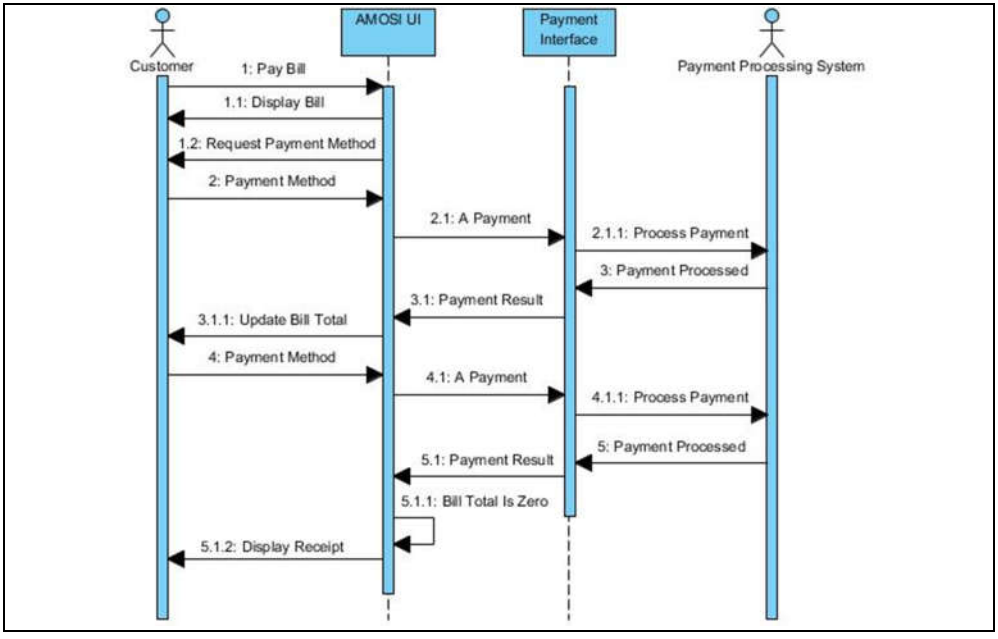

Figure 90 - Split Bill Option 2 Sequence Diagram

When the customer makes a request to pay their bill, the system displays the bill and requests that the customer enter a payment method and an amount to pay that is less than or equal to the bill. The payment information is sent to the payment processing system. Once this payment is processed and there is an outstanding balance, the customer is requested to enter another payment

method and the amount to pay. This continues until the full balance of the bill is paid (Bill Total Is Zero).

Notes:

- This option is simple to implement because it repeats existing functionality, with the addition of a running total.
- It also has the benefit that when a payment method is rejected, the customer is simply requested to pay the same total with a different payment method.
- The downside is that the customer has to calculate the cost of the menu items that they want to pay for.

Option 3: An order may be split into several bills

Figure 91 contains a sequence diagram that shows how this option affects interfaces to the user and to backend systems.

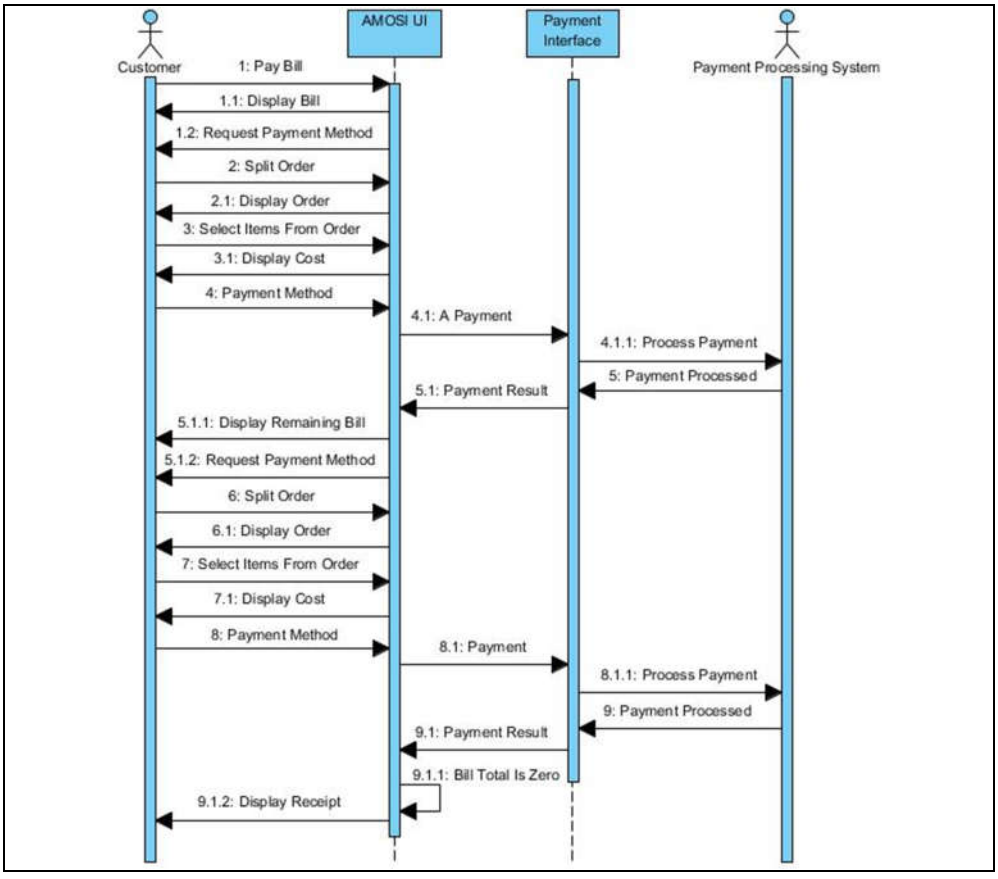

Figure 91 – Split Bill Option 3 Sequence Diagram

When the customer requests to pay their bill, the bill is displayed and a payment option is requested. The customer may opt to split the order. The system displays the customer order and the customer selects menu items from the order that will be applied to the current bill. The bill is updated to reflect the selected items. The customer enters and confirms a payment method for the selected items. The payment method is sent to the payment processing system. The payment is processed and the bill is redisplayed with the remaining amount to pay.

The customer may enter a payment method and pay the remaining amount on the bill or (as shown in the diagram) the customer requests to split the order again. The order is redisplayed with unpaid items shown as selectable. The customer selects items to pay and enters a payment method. The payment is processed and the remaining balance is displayed.

This continues up to 8 times or until the bill is paid (Bill Total Is Zero). (After 7 payments, the customer is no longer given the option to split the bill.)

Notes:

- This option satisfies the need for parties to pay for only items that they ordered.
- However, because bills are processed in a serial manner, it is time consuming and more work for the customer than other options.

Option 4: A customer is allowed to place several orders

Figure 92 contains a sequence diagram that shows how this option affects interfaces to the user and to backend systems. This option allows a customer to create many orders. Each order is processed as a separate entity.

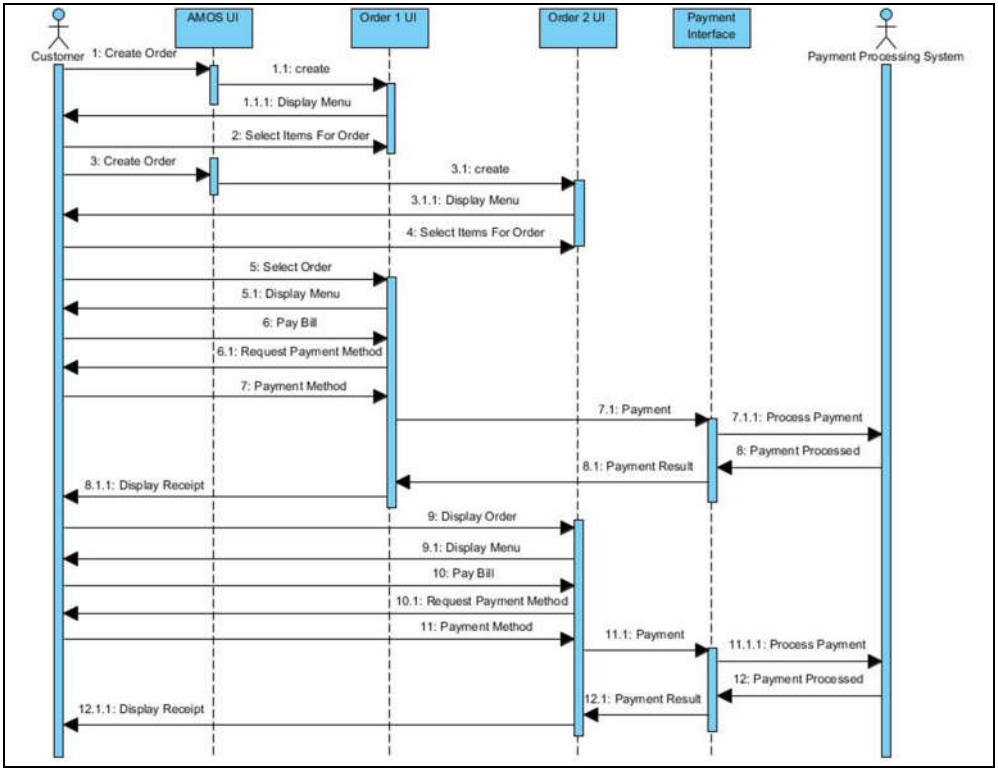

Figure 92 – Split Bill Option 4 Sequence Diagram

A customer who is assigned to a table creates an order. The menu is displayed and the customer selects items for that order.

The same customer creates another order and selects menu items for that order.

(The customer may create up to 8 orders.)

The customer selects the first order and requests to pay the bill. The system requests and processes a payment method for the first order as normal. The receipt for that order is displayed to the customer.

The customer selects the second order and requests to pay the bill. The system requests and processes a payment method for the second order as normal. The receipt for that order is displayed to the customer.

The customer may select menu items and make payments on an order without having any impact on other orders that they have created.

When all orders have been paid, the customer makes copies of their receipts and leaves the table.

Notes:

- This option re-uses the existing ordering process.
- However, it is difficult for the customer to understand and navigate the user interface.
- It is also a complex user interface to implement.

Option 5: A table can contain several customers at the same time

Figure 93 contains a sequence diagram that shows how this option affects interfaces to the user and to backend systems. In this option, additional customers are created and processed in parallel.

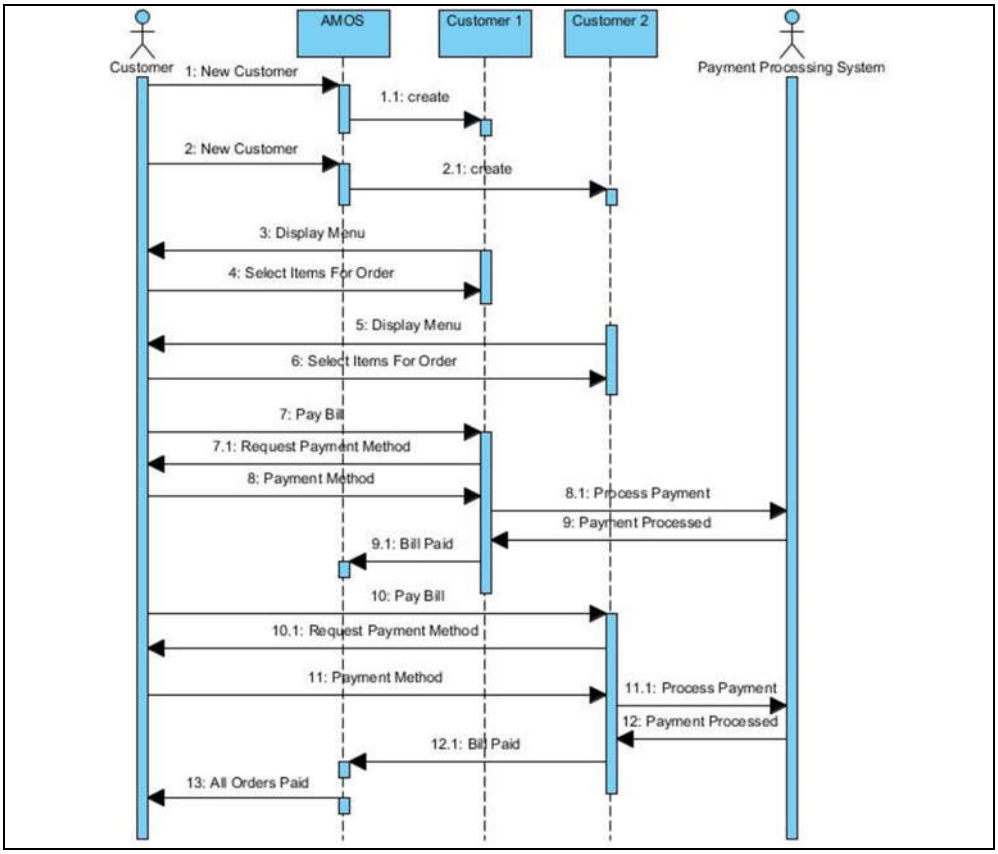

Figure 93 – Split Bill Option 5 Sequence Diagram

When a customer is assigned to a table, the system creates a customer user interface.

A new customer interface is created each time a new customer user interface is assigned to the table.

(The user interface could look something like a browser with tabs, where each tab is for a specific customer. With up to 8 tabs available to a table.)

Each customer can order from the menu, request assistance and pay for their bill as normal. Customers may be added to the table or removed from the table (after their bill has been paid), at any time.

Notes:

- The system processing remains the same. Only the user interface needs to change to support multiple customers at the same table.
- The downside to this option is that the customer has to know how they want to split the bill before they order.

- **Like me, you can possibly find variations of these options that may improve the customer experience by reordering some of the steps or optimizing the user interface. The purpose of this exercise is to show that there are many options and not to document every possible solution.**

Assessing The Options

Input is solicited from the product owner, solution architects, UI designers and the development team for each of these options.

Selected options are then presented to the customer (and other stakeholders who are impacted by implementation of the split order feature).

The customer decides to pursue option 2. Option 2 allows the customer to pay the bill incrementally with several payments. Each payment has its own payment method and amount paid. The customer is repeatedly requested to pay their bill until the bill total is zero.

Business To System Mapping

With an option selected, the business process can be mapped to system functionality.

Returning to the business activity diagram, the new and adjusted business steps have been included in a split bill business to system mapping diagram. Figure 94 shows the mapping of each of the split bill business steps to the existing Process Customer Payment system use case.

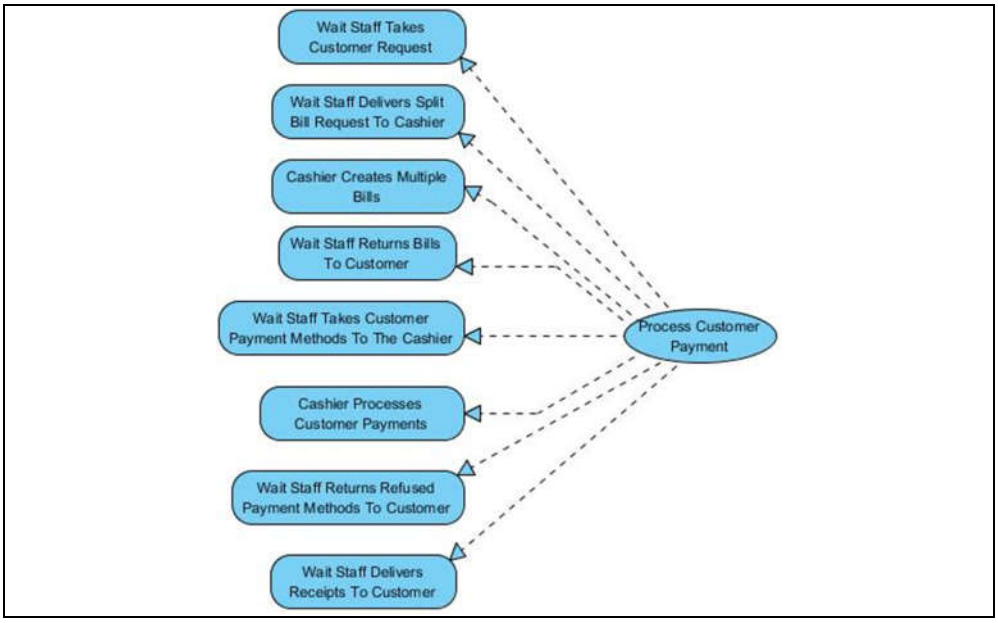

Figure 94 – Split Bill Business To System Mapping Diagram

Wait Staff Takes Customer Request – The system provides the option to make partial payments on a bill.

Wait Staff Delivers Split Bill Request To cashier – This step is no longer necessary.

Cashier Creates Multiple Bills – The system will make an additional request to pay the bill if there is an outstanding balance on the bill.

Wait Staff Returns Bills To Customer – This step is not implemented. The customer does not see partial bills, only partial payments.

Wait Staff Takes Customer Payment Methods To The Cashier – The system sends partial payments to the payment processing system.

Cashier Processes Customer Payments – Many payments are processed by the payment processing system for a single bill.

Wait Staff Returns Refused Payment Methods To Customer – If a payment is refused the outstanding balance on the bill remains the same...

Wait Staff Delivers Receipts To Customer – When all payments have processed successfully, the receipt is displayed to the customer. (The customer does not see multiple receipts in this solution.)

System Use Case

The system use case diagram is the same as before, because the Process Customer Payment system use case actors have not changed.

In Figure 95 an alternate path has been added to the Process Customer Payment system case activity diagram, shown in Figure 27.

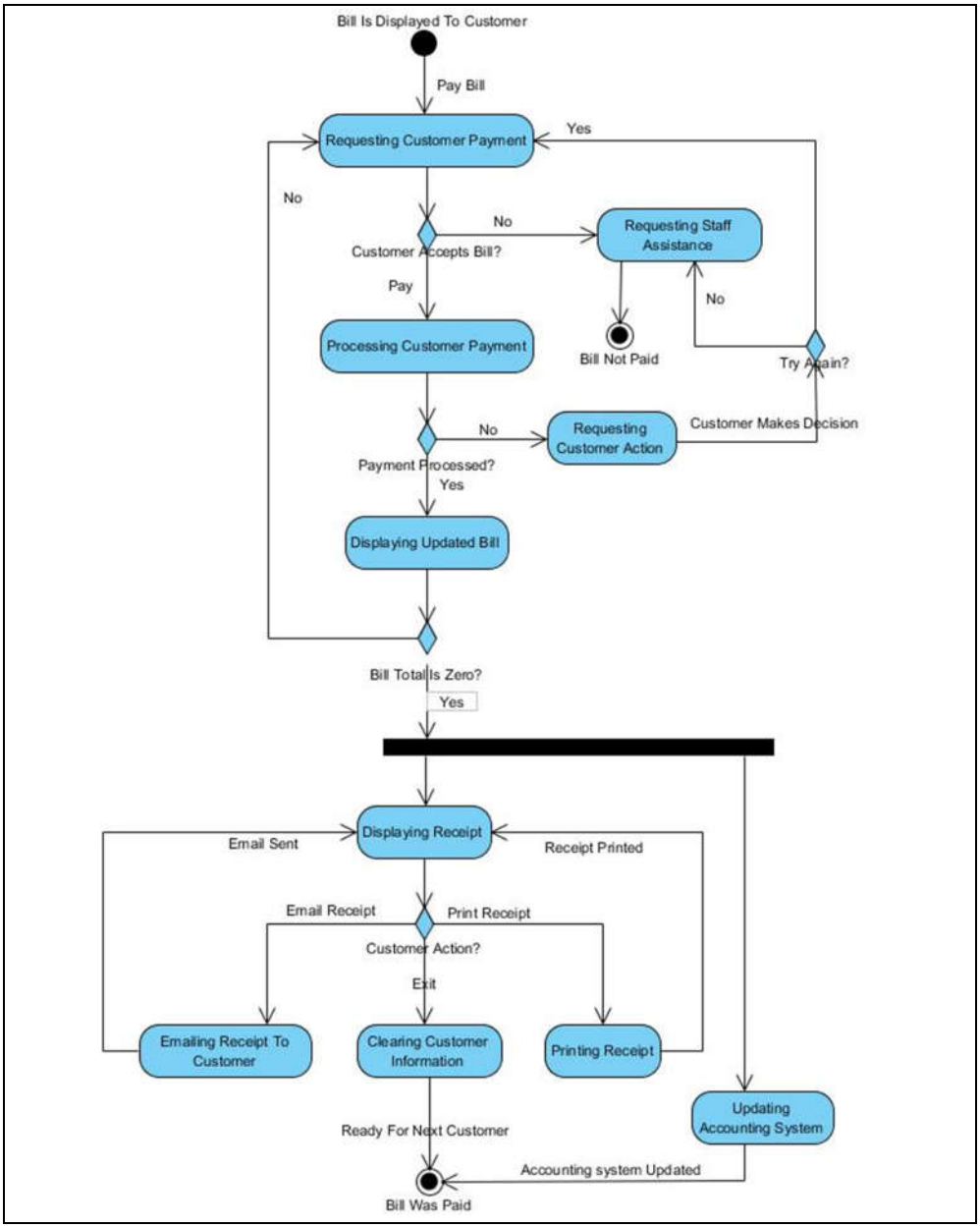

Figure 95 – Process Customer Payment(Split Bill) Activity Diagram

When a customer payment is requested, the system will now allow an amount that is less than the total bill to be entered. (On the 8th payment, the full bill total is displayed and this cannot be changed.)

After a payment is processed, the system calculates the remaining bill total. If the total is greater than zero, the customer is requested to re-enter a payment method. If the total is zero then the receipt is displayed to the customer and the accounting system is updated.

If a customer payment method fails to process then the customer is still prompted for assistance. The customer may continue using a different payment method and the bill total remains unchanged.

- **The above diagram shows the simplest implementation of the business need. I would suggest that each time a payment is processed, a receipt is created and the accounting system is updated. This change would require approval from the stakeholders and input from the project team.**

User Stories

I derived the following user stories from the Split Bill system activity diagram.

Parent	Pay Bill Epic
Overview	Process Customer Payment (Split Bill Feature)
This user story describes how a customer can make several partial payments towards paying their bill.	
User Story	
As a I want so that	customer to be able to use several partial payments to pay the total owed several people are able to contribute towards the bill.
Acceptance Criteria	
Given when then	that a customer bill is displayed a payment method is requested the customer is able to enter an amount to pay using that payment method that is less than or equal to the bill total. (the minimum amount that can be entered is $5 and the system will prevent the amount owed from being less than $1.)
Given when then	that a customer bill is displayed 7 payments have been made towards the bill the customer is prompted to pay the remaining balance. (the customer is not able to change the amount to pay.)
Given when then	that a customer payment is processed successfully the customer is requested to enter another payment total amount owed will be updated to reflect the payment that was successfully processed

User Interface Design

The UI designer creates mockups for the Split Bill sequence diagram shown in Figure 96.

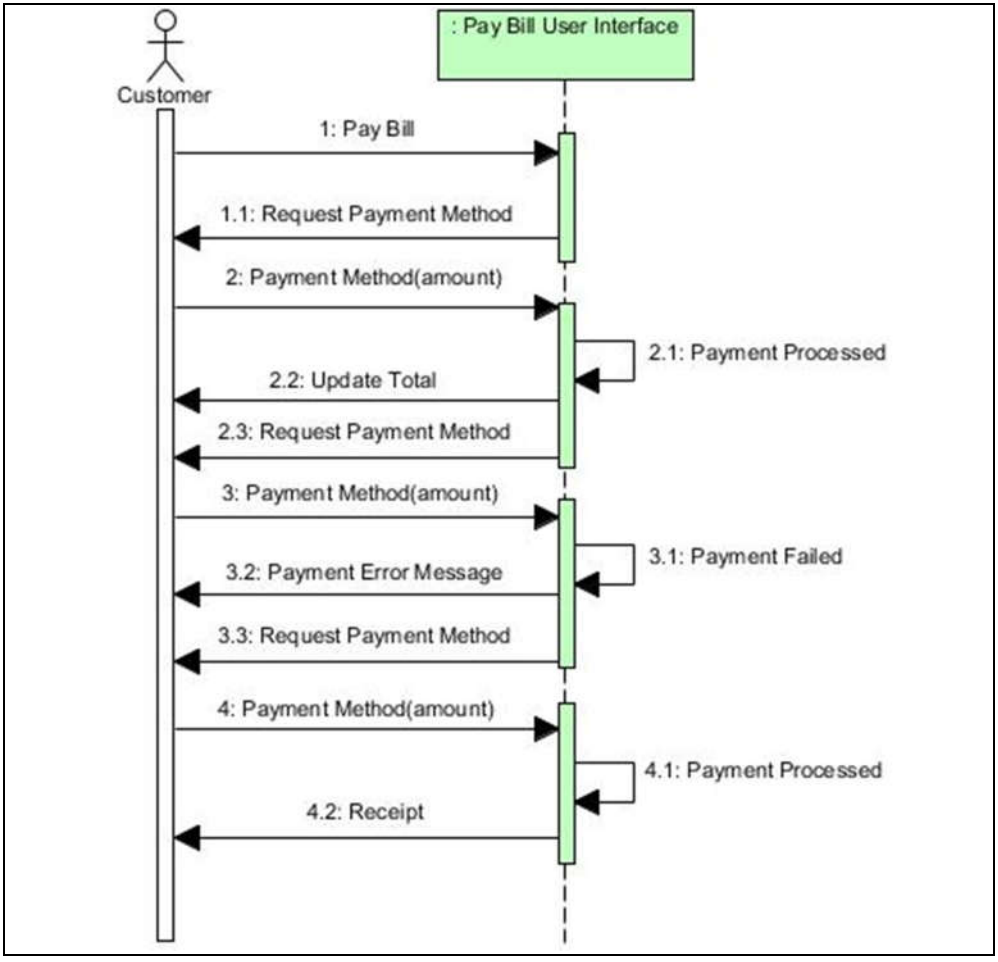

Figure 96 – Split Bill User Interface Sequence Diagram

The customer requests to pay their bill. The system requests a payment method.

The customer enters a payment method (and an amount to pay that is less than the bill total) and the payment is processed successfully. The system updates the bill total to reflect the amount paid and requests another payment.

The customer enters a payment method (and an amount to pay) and the payment fails to process. The system informs the user of the error and requests another payment.

The customer enters a payment method (and an amount to pay that is equal to the remaining bill total) and the payment is processed successfully. The system displays a receipt for the bill.

The user interface mockups are referenced by the user story.

Development

The user story is reviewed an approved by the product owner.

The user story is reviewed and estimated by the development team.

The acceptance criteria are reviewed and approved by quality assurance.

The user story is moved from the product backlog to the sprint backlog.

The sprint starts and the user story is developed and released in an incremental build.

The incremental build is reviewed by all stakeholders.

Test Cases

When the user story is pulled into the next sprint cycle, quality assurance write test cases that validate the user story.

An example test case might be:

Step #	Action	Inputs	Expected Outputs
1	Pay Bill selected	N/A	Bill payment request screen is displayed with a total of $10.99 Amount displayed can be edited to a value between $5 and the $10.99
2	Make Payment is selected	Visa credit card (number, expiry date, name, code) A payment amount of $5 is entered	Payment is processed successfully and a payment completed message is displayed Bill payment request screen is displayed with a total of $5.99 Amount displayed cannot be edited.
3			

The incremental build is validated against the test case. If the incremental build passes validation testing, then the split bill feature may be released to the customer.

Release

The deployment manager prepares the software build for deployment to the customer site.

The writer creates release notes for the split bill feature and updates the user manuals.

The deployment manger installs the updated software into the customer environment and distributes user manuals and release notes to the customer.

The customer uses the split bill feature in their restaurants.

The End.