**Task 1**

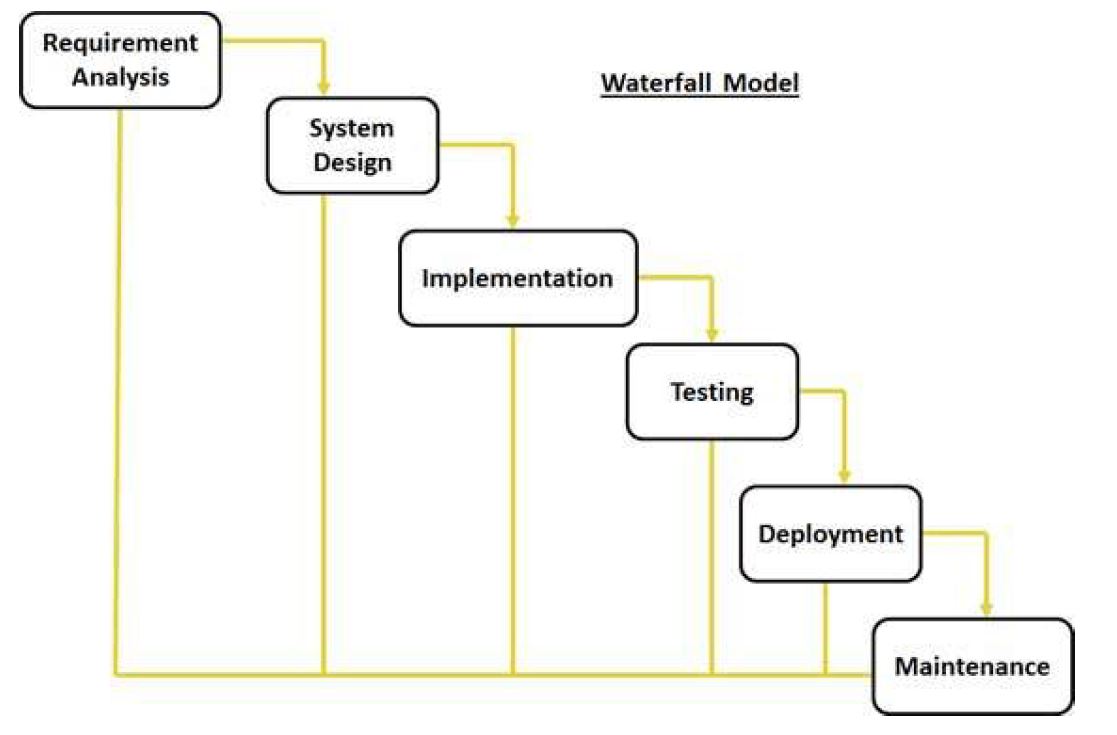
1.1 Describe the different traditional and agile methodologies available and their

features.

Life Cycle Models

* Traditional Approaches

1. Waterfall Model.



1. **Requirement Gathering and analysis:**

All possible requirements of the system to be

developed are captured in this phase and documented in a requirement specification doc.

1. **System Design:**

The requirement specifications from first phase are studied in this phase

and system design is prepared. System Design helps in specifying hardware and system

requirements and also helps in defining overall system architecture.

1. **Implementation:**

With inputs from system design, the system is first developed in small

programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

1. **Integration and Testing:**

All the units developed in the implementation phase are

integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

1. **Deployment of system:**

Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

1. **Maintenance:**

There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released.

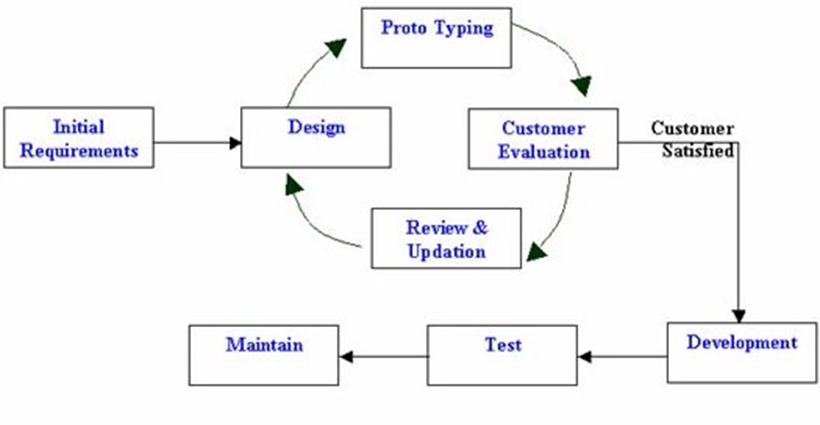
Maintenance is done to deliver these changes in the customer environment.

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most

appropriate are:

1. Requirements are very well documented, clear and fixed.
2. Product definition is stable.
3. Technology is understood and is not dynamic.
4. There are no ambiguous requirements.
5. Ample resources with required expertise are available to support the product.
6. The project is short.
7. Prototyping Model.

* The prototype model is used to overcome the limitations of waterfall model.
  + In this model, instead of freezing the requirements before coding or design, a prototype is built to clearly understand the requirements.
  + This prototype is built based on the current requirements.
  + Through examining this prototype, the client gets a better understanding of the features of the final product.



Following is the stepwise approach to design a software prototype:

* **Basic Requirement Identification:**

This step involves understanding the very basics product requirements especially in terms of user interface. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.

* **Developing the initial Prototype:**

The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed and the workarounds are used to give the same look and feel to the customer in the prototype developed.

* **Review of the Prototype:**

The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an organized manner and used for further enhancements in the product under development.

* **Revise and enhance the Prototype:**

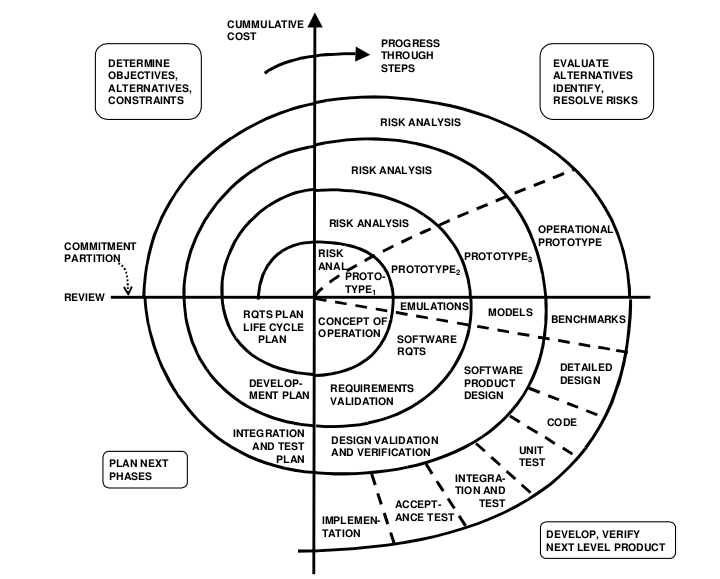
The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like, time and budget constraints and technical feasibility of actual implementation.

The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until customer expectations are met.

Prototypes can have horizontal or vertical dimensions. A horizontal prototype displays the user interface for the product and provides a broader view of the entire system without focusing on internal features. A vertical prototype, on the other hand, is a detailed elaboration of a function or subsystem in the product.

The purpose of both the horizontal and vertical prototypes is different. Horizontal prototypes are used to get more information about user interface levels and business requirements. It can even be presented in sales demos to get business into the market. Vertical prototypes are technical in nature and are used to get details on the exact functioning of the subsystems. For example: database requirements, interaction and computational load in each subsystem.

1. Spiral Model.

The spiral model combines the idea of iterative and prototype development with the systematic, controlled aspects of the waterfall model. It allows for incremental releases of the product, or incremental refinement each time around the spiral. The spiral lifecycle model allows for elements of the product to be added in when they become available or known. This assures that there is no conflict with previous requirements and design.

Documents are produced when they are required, and the content reflects the information necessary at that point in the process. Like the product they define, the documents are works in progress. The idea is to have a continuous stream of products produced and available for user review.

The development spiral consists of four quadrants:

* Quadrant 1: Determine objectives, alternatives, and constraints.
* Quadrant 2: Evaluate alternatives, identify, resolve risks.
* Quadrant 3: Develop, verify, next-level product.
* Quadrant 4: Plan next phases.
* Agile Methods

1. Kanban.

In modern software development, Kanban (or the Kanban methodology) expands on those same just-in-time concepts by ensuring that the level of required work at any given time roughly equates to the work capabilities of the team. Thus, development work is performed in a just-in-time fashion, with similarly minimal waste.

The Kanban method was originally developed inside Toyota Motor Corporation, by an industrial engineer named Taiichi Ohno. The goal was to drastically reduce waste across the manufacturing floor at Toyota, and Kanban methods proved to do exactly that. Toyota’s Six Rules, as described by Taiichi Ohno in his 1998 book Toyota Production System.

Moving into the realm of modern software development, kanban has been adapted to fit the needs of most development life cycles. As previously discussed, the kanban methodology aims to balance the level of active or in progress work with the capabilities of the team to handle said work. If your team is handling all current work in progress with ease, more tasks can be added to that in progress list, to hopefully ensure there’s a constant balance of efficiency.

It includes 7 steps

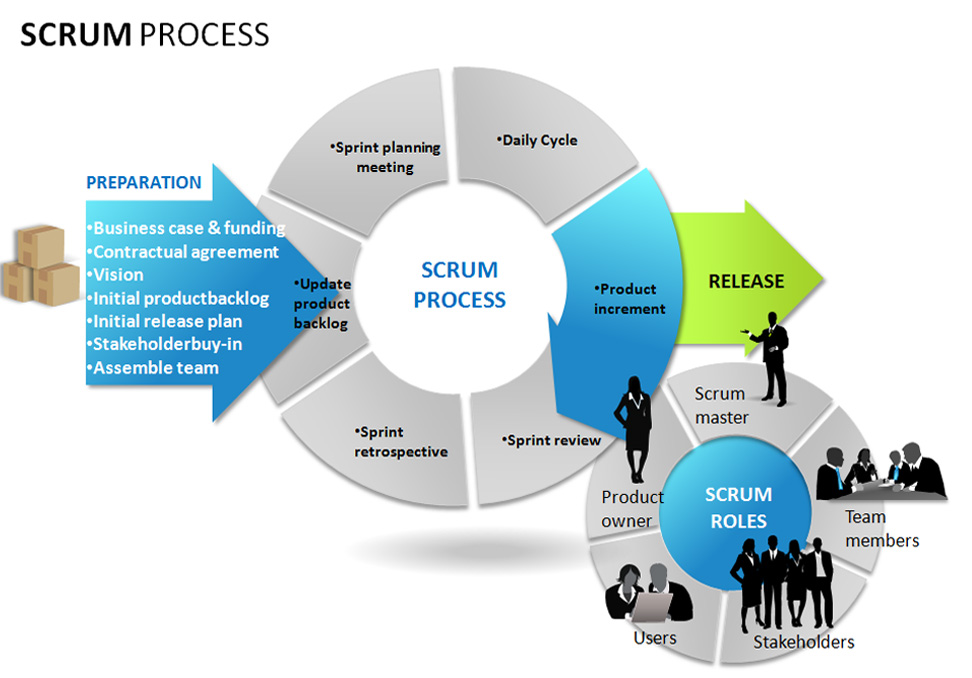
* Product backlog
* Requirements
* Design
* Development
* Testing
* Deployment
* Done

Creating the Kanban Board

A **Kanban board** is a work and workflow visualization tool that enables you to optimize the flow of your work. Physical **Kanban boards**, like the one pictured below, typically use sticky notes on a whiteboard to communicate status, progress, and issues

1. Agile Scrum.

Agile Software Development is an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto. Solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context.



1.2 Discuss the strengths and weaknesses of each methodology.

* Waterfall Model

|  |  |
| --- | --- |
| Pros | Cons |
| Simple and easy to understand and use | No working software is produced until  late during the life cycle. |
| Easy to manage due to the rigidity of the  model. each phase has specific deliverables and a review process. | High amounts of risk and uncertainty. |
| Phases are processed and completed  one at a time. | Not a good model for complex and  object-oriented projects. |
| Works well for smaller projects where requirements are very well understood. | Poor model for long and ongoing  projects. |
| Clearly defined stages. | Not suitable for the projects where  requirements are at a moderate to high  risk of changing. So risk and uncertainty  is high with this process model. |
| Well understood milestones. | It is difficult to measure progress within  stages. |
| Easy to arrange tasks. | Cannot accommodate changing  requirements. |
| Process and results are well documented. | No working software is produced until  late in the life cycle. |

* Prototyping Model.

|  |  |
| --- | --- |
| Pros | Cons |
| Increased user involvement in the  product even before implementation | Risk of insufficient requirement analysis  owing to too much dependency on  prototype |
| Since a working model of the system is  displayed, the users get a better  understanding of the system being  developed. | Users may get confused in the  prototypes and actual systems. |
| Reduces time and cost as the defects can  be detected much earlier. | Practically, this methodology may  increase the complexity of the system as  scope of the system may expand beyond  original plans. |
| Quicker user feedback is available  leading to better solutions. | Developers may try to reuse the existing  prototypes to build the actual system,  even when its not technically feasible |
| Missing functionality can be identified  Easily | The effort invested in building prototypes  may be too much if not monitored  properly |
| Confusing or difficult functions can be  Identified |  |

* Spiral Modal

|  |  |
| --- | --- |
| Pros | Cons |
| Changing requirements can be  accommodated. | Management is more complex. |
| Allows for extensive use of prototypes | End of project may not be known early. |
| Requirements can be captured more  Accurately | Not suitable for small or low risk projects  and could be expensive for small  projects. |
| Users see the system early. | Process is complex |
| Development can be divided into smaller  parts and more risky parts can be  developed earlier which helps better risk  management. | Spiral may go indefinitely. |
|  | Large number of intermediate stages  requires excessive documentation. |

* Kanban

|  |  |
| --- | --- |
| **STRENGTHS** | **WEAKNESS** |
| Projection about failover, redundancy, disaster recovery, data centre locations, and storage requirements are accurate due to the inputs from the developers | cost |
| Before go-live, the load testing makes the application a bit slow. The development team quickly fixes the bottlenecks. The application is released on time |  |
| Fast |  |

* Creating Kanban Board

|  |  |
| --- | --- |
| **STRENGTHS** | **WEAKNESS** |
| **Reduces Waste** | **Required Constant Board Monitoring** |
| **Forces Event-Driven Workflow** | **Potential for Complexity** |
| **Allows for Flexibility** | **Possible Bottlenecks** |

* AGILE Scrum Methodology

|  |  |
| --- | --- |
| **STRENGTHS** | **WEAKNESS** |
| Progress can be measured. | High dependency on modelling skill |
| Reduced development time | Management complexity is more |
| Quick initial reviews occur | Requires highly skilled developers/designers |
| Increases reusability of components |  |

1.3 Compare and contrast the strengths and weaknesses of these traditional and agile systems analysis methodologies that you identified.

|  |  |
| --- | --- |
| **Traditional model** | **AGILE** |
| Following a top-down approach and making changes are not easy, since ending one phase leads to the next | The team performs experiments with different techniques and gradually comes up with the best solution |
| It has a leadership style to work | In agile communication the communication flow is free, everyone can present their ideas in a team |
| The project plan is created before the system development starts | The project work is provided to the customer at a low level, that is, as soon as a module is prepared, a demonstration is given to the customer to confirm the work progress in the right direction |
| Believes in a one-time delivery of the product | Agile is based on the gradual delivery of the product |

1.4 Critically evaluate (give your own opinion on) the strengths and weaknesses of

the traditional and agile systems analysis methodologies

In traditional work load is high so if we get one error we must start again, and we must do it again and again when expected outcome is come. And if one error comes, we must start over again but agile methodology the best part is the work load is separated and work load is very low, so we can do it easy and its time consuming is also low. If we get one error, we can do it in that part very easily we don’t have to take the process from start, So agile methodology is best model in my opinion.

1.5 Discuss typical transition problems faced by organizations that move from the

traditional to the agile approach.

The transition from waterfall to agile software development requires careful planning, collaboration and change management. When implemented correctly, the agile model can result in your project team accelerating production efforts and working more cohesively. Below are the vital points to consider for smooth transition to agile:

a.Training Required:

Get all the team members to undergo training on Agile (XP, Scrum, etc). Entire team has to attend the agile training. I would suggest starting the training from the top. As per my understanding, this initiative will be a successful one if the top management understands the agile. Middle managers should go for agile certification. Developers and testers can attend at least 8 hours of agile training to start with. Otherwise, agile is not that easy to learn in 8 or 16 hours. Every person will need to read a lot before practicing at the floor. Give them their own time to learn the agile. They will take time to wear the new skin of the agile development.

b.Agile Coach Needed:

Even when the team is starting with just a few practices, handholding/mentoring from an agile coach is needed. A seemingly simple practice like standup meeting also needs guidance in the initial stages. Make a group of expert people who will provide the rules and regulations, policies, guidelines for the project. You can call this group anything.Traditional companies call them Agile PMO or Agile COE etc. Remember these people need to be expert. If any company does not have these people then hire them. These people will be pillars in this initiative.

c.Initial longer iteration period:

Start with 4-6 weeks iteration rather than 1 week iteration. Many new comers to Agile feel suffocated with 1 week iterations. Earlier the waterfall teams would have delivered softwares once in 6 months, and suddenly asking them to deliver in a short period makes them resist to Agile. Start with the 4 weeks sprint cycle. As per my experience,developers and testers prefer the longer duration. They would prefer 4 weeks sprint, which is also quite good.

d.Initial dismantling of current team is not favourable :

Scrum and XP advocates specific team structure like having Product Owner, Team, cross functional teams, no hierarchy, a team coach, scrum master, etc.This is a very sensitive issue. Suddenly informing theteam that all of you are same, might heart the ego of senior people. So, better not to worry about dismantling the current team structure. Let the team learn slowly the importance of the values and decide what is best for them.

e.Different financial reconciliation practices required:

Agile methodology also requires different financial reconciliation practices. Because teams will go through more iteration, and potentially even change some of the original project specifications, the Project Manager will have to implement more check-points to assess budget. To exercise budgetary control, the Project Manager must assess and dole out project hours to team members in smaller chunks – hours should be assigned to each team member weekly.

**Task 2**

2.1 Explain the background to your business problem and propose a potential solution for it.

This catering service management had few typical problems.

* Defects in the application leading to rework;
* Unable to meet deadlines (this led to financial crunch);
* Unhappy development & Management teams.

In this management, they did not follow any process; even the requirements/ defects/ enhancements were not tracked. This did not help their purpose, as revenue of the company started to show downward trend.

I worked with all stakeholders & identified adopting Agile-SCRUM would be good for the organization for the below reasons:

* It was easy & quick to adapt;
* There were frequent changes to requirements from their client;
* They had no tools to track the requirements.

2.2 Carryout a feasibility study for your proposed solution to the business problem.

Assess at least the Technical Feasibility, Economic Feasibility (Cost-Benefit analysis) and Operational Feasibility.

*Note: You may perform Legal & Schedule feasibility as well if applicable.*

In your feasibility study, critically evaluate how the strengths & weaknesses

of traditional and agile methodologies apply to your business problem and

any transitional problems that could arise if the organization (i.e. the

organization you have selected) is currently using the traditional approach.

* Technical Feasibility

To build this system we need computers and internet connection. Select computers with recommended specifications because we need to install softwares. Purchase softwares to design system also.

* Economic Feasibitity

ROI =

ROI =

ROI = 190,138.00/478,552.00= 40%

* Operational Feasibility

After the implentation of the employees need to knowledge to handle the system so we need to train employees after implentation. First we need to measure every employees knowledge to use system. We can ask questions to employees and ask some tasks to do in the system .It can be useful to measure employees knowledge. After we need to provide special training to employees. In this situation we need to cosider employees knowledge wise(specially low knowledge employees).

Feasibility Study for Solution

Create a desktop application or web-based application Successfully without any running errors. Customer and Management System Both can use this system. In that case Customer can also View the orders after that Coordinator gets the order from customer sent to other members. If the stock is unavailable the message should have sent to vendors. So, we can manage separate each team to do their jobs and it will finish faster and they can maintain the time management get good customer feedback.

If Customer feedback is good, they get more customer and get more money than other competitors. They make their progression very easy manner.

So, we want to discuss with Agile team what kind of application they want done by the team how to improve that how to do that in very fast and very easy with agile methodologies other traditional methodologies may consume more time but in Agile all are done with a same time get what client want very easy reduce the time. And how to design the user interface all are done by Agile methodologies very easy.

And know how to design all kind of team members (Client Side-Catering Management System). Share with Agile team. In Agile Methodologies we can get higher quality product. It was better in terms of overall business value and it was more productive.

2.3 Justify the importance and relevance of the feasibility criteria you assessed above, in terms of your business problem.

Feasibility studies give you the opportunity to be a step ahead of any issues. Solving an issue it even becomes a problem. A catering services feasibility study requires a thorough market analysis involving a study of the economic,technical and operational feasibility. The purpose of the feasibility study is to provide honest and reliable information to the client to help them decide whether to develop their proposed catering project.

**Task 3**

3.1 Perform Systems Analysis for your proposed solution (system) using a suitable methodology.

Use suitable fact-finding techniques to investigate and identify user and

system requirements.

*Note: You may draw Use Case diagrams if required.*

Identify any constraints/limitations including any security issues.

In this catering service system I used these are main techniques

* Interviews

In this method I met many employees in the shop and ask many questions about their daily tasks. I got ideas from employees to develop a system. Following details are got it from each employees

1. Chefs

Chefs get the task from customer services. After they will prepare to cook. They have problems like getting materials quickly because they are getting orders continuously. Cheffs took the materials from the store.

1. Manager

Manger need to mange employees and mark attendance of the employees. Sometimes manager need to motivate work effectively

1. Customer services

In this department employees handling phone calls(getting orders from customers and get feedback from the customer.

1. Delivery boy

Get the delivery details from the customer services and to deliver food on time.

1. Cashier

Mange accounts and pay bills for materials.

* Observation

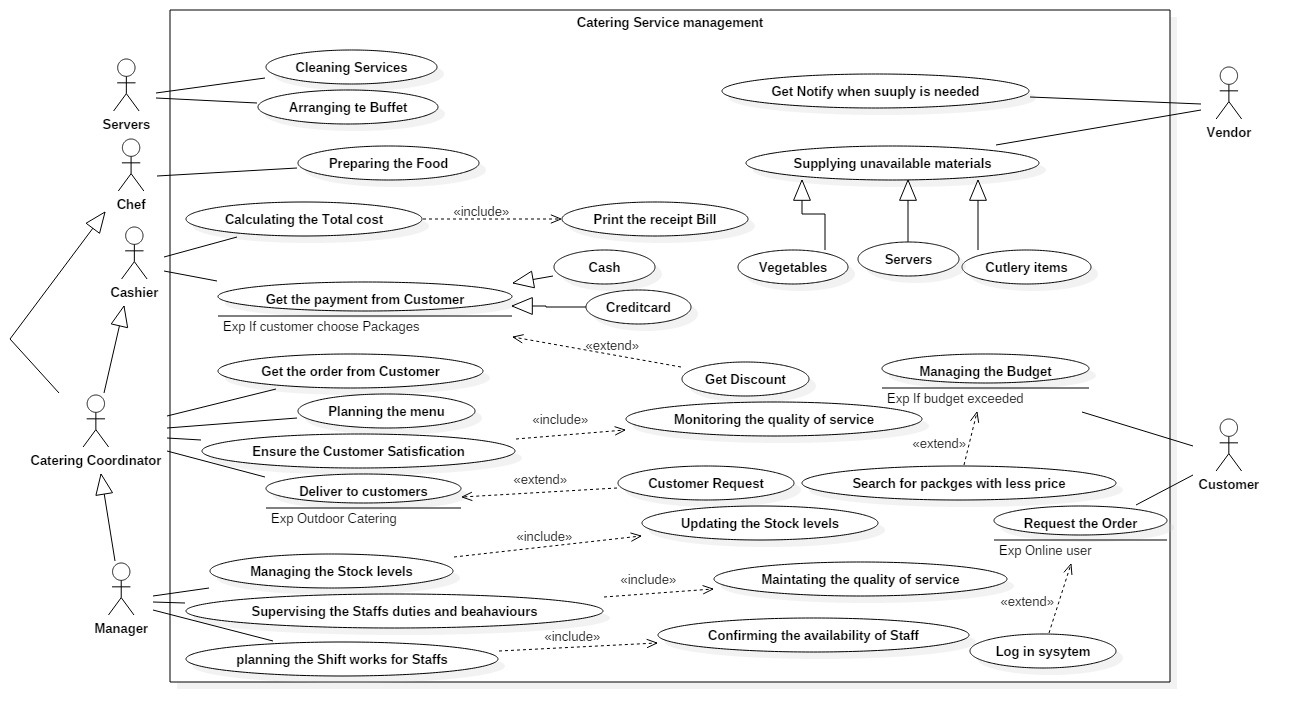
When I went to catering shop I spend several hours to note activities in catering shop. I found some problems in catering shop.

They are

* Distance between customer services and kitchen too far so if customer services staff want to inform the order he need to walk and he will miss some calls also.
* In delivery some motor bike doesn’t have fuel so drivers need to sped additional time to take motor bike to fuel station.
* Employees wrote their user name and password on the hanging board.

User stories

1. As a customer, I want to know the quality of materials so that I can get good and healthy food.
2. As a customer, I want to choose the convenient payment method so that I can make the payment according to my convenience.
3. As a customer, I want to know the details of the package in full so that I can choose the appropriate one.
4. As a customer, I want to check my orders so that I can confirm whether the ordered items and delivered items are the same.
5. As a customer, I want to calculate the budget amount and decide whether it is should be economical so that I can save money.
6. As a customer, I want to know about the services and loyalty of the business so that I can verify whether it can satisfy my desires.
7. As an online customer, I want to choose the delivery method so that I can get my order delivered to the wanted place when I am not eligible to go and collect the order.
8. As a customer, I want to get my order delivered on time so that I can schedule my event.
9. As an online customer, I want to verify the authority, ratings and reviews so that I can choose the correct catering services.
10. As a customer, I want to confirm that servers behave in a polite manner.
11. As a customer, I want to confirm that services are neat and tidy and the food is healthy so that I can serve healthy food for my guests.
12. As a customer, I want to have the menu which meet my requirements and beneficiary for me so that I can choose the menu easily.
13. As a customer, I want to make sure that payment is flexible so that I can pay the advance first and balance later or on delivery time.
14. As a customer, I want to have the menu card so that I can correctly select the beverages, starter, dessert and main dish as according my wish.
15. As a chef, I want to confirm the tools are easy to handle so that I can complete the order as soon as possible.
16. As a chef, I want to finalize and confirmed the order from coordinator so that I can get the needed things.
17. As a chef, I want to get the materials and things on time so that I can make the foods on time.
18. As a coordinator, I want to have the exact menu items or packages so that I can pass the order to manager.
19. As a vendor, I want to get the notification when the stock is out of level so that I can refill.
20. As a cashier, I want to get the fixed cost of the orders from coordinator so that I can get the correct payment.
21. As a cashier, I want to get the payment either from card or cash so that I can summarize the total as card and cash easily.
22. As a cashier, I want to know whether the customer choose package so that I can give discount.
23. As a manager, I want to the stock level updated so that I can give notification to vendors.
24. As a manager, I want to the order details from coordinator so that I can take over if the coordinator not available.
25. As a manager, I want to generate the reports so that I can calculate the performance.
26. As a manager, I want to confirm that staffs are working according to their schedule so that I can make sure that staffs are available.
27. As a manager, I want to expect the quality of materials and low prices from vendors so that I can maintain the goodwill in society and I can reduce the cost for customers.
28. As a server, I want to get the schedule from manager so that I can aware of my duties.
29. As an online customer, I want to order through online so that I can get loyalty points.
30. As a manager, I want to get the feedbacks from customer so that I can improve the services.
31. As a manager, I want to put the notices so that I can inform immediately to staffs.



Use case Scenario

I am explaining this system from use case diagram. Then you can understand very clearly, and I am explained two important use cases.

|  |  |
| --- | --- |
| **Use case id** | **1** |
| **Use case name** | **Managing budgets** |
| **Primary actor** | **Customer** |
| **Main Success Scenario** | **1.Customer will request /ask about the details of the meals.**  **2.Coordinator will give the needed details.**  **3.If satisfied customer will ask about the cost**  **4.Coordinator will estimate the cost.**  **5.Customer will be satisfied with the details and cost.**  **6.Coordinator will confirm the order.** |
| **Extensions** | **3.a If the customer is not satisfied with the details given, coordinator will give more details.**  **5.a if the customer is not satisfied with the cost.**  **5.b.1 Change the meals and begins the scenario again.**  **5.b.2 Coordinator will explain about the available packages**  **5.b.3 Customer will choose the package** |

|  |  |
| --- | --- |
| **Use case id** | **2** |
| **Use case name** | **Get the payment from customer** |
| **Primary Actor** | **Cashier** |
| **Pre-conditions** | **1.Customer must confirm the order**  **2.Coordinators must forward the estimate to cashier.** |
| **Main success Scenario** | **1.Cashier will type the details according to the estimate.**  **2.Cashier will calculate the total.**  **3.Customer will give the cash.**  **4.Cashier will print the bill.** |
| **Extensions** | **1.If the customer chooses the packages the customer will get the discount.**  **3. If the payment through the credit card**  **3.b customer will submit the card**  **3.c customer must acknowledge the signature and bill.** |

After these two-use case scenarios you will understand how Agile methodology effect this System. Agile is an iterative approach to project management that allows you to manage a project by breaking it into several stages. So, that will appear in this Catering Service Management System. All jobs divided into some parts and every part will get by from one team, so the Agile methodologies effect this System. Now This System same as Agile methodologies.

Agile methodologies effect System and Encourage customer involvement.

**1) Confirming Order**

A customer can order through online or directly. Online customer needs to login to the system to order. An order may include meals or packages. A meal or packages may include menu items like starter, main dish, dessert, or beverages**.** If customer is satisfied with the selected meals or packages and cost customer may confirm the order. If customer is not satisfied with the selected meals, they can go for packages or can select the meals again according to their wish.

**2) Confirming the delivery**

If customer request to deliver the order, coordinator will collect the details of delivery place, date, time etc. coordinator will check the availability of the staffs and vehicles. If the vehicles and staffs are available on that date or time coordinator will confirm the delivery for that order else, he will check again and again. Also, coordinator will update the schedule according to the time.

**3) Getting supply from vendors**

Once customer confirm the order, coordinator will pass the order to manager. Manager will update the stock level at the same time manager will check the stock level. If the stocks available chef will process the order. Else manager will send the notification to vendors about the necessary items. Once the vendors receive the notification they will collect the necessary items and send the items to manager.

**4) Preparing food/meals**

Once customer confirm the order, coordinator will check whether chef is available if not he will check again and again. If chef is available coordinator will pass the order to chef. Once the chef is get the order on that date chef will check the availability of the stock, if stock is not available chef will send the message to manager. Manager will notify about that to vendors. Once the items are ready chef will ready to prepare the food.

**5) Payment from customer**

Once the customer confirms the order coordinator will pass the order to cashier. If customer choose the packages, cashier will give the discount if not customer is not entitled for discount, so the total amount will differ, so cashier will recalculate and will display the net amount. Customers can make their payment according to their convenience as cash or credit card. If customers are paying through the card the bank must authorize the transaction.

Functional Requirement

* Managing order
* Input: Customer details, order details
* Process: get order; add order, update order, cancel order, get customer details, preparing order.
* Output: order successfully prepared.
* Managing Discount
* Input: Total amount, percentage.
* Process: calculate the discount.
* Output: print bill.
* Managing delivery
* Input: customer details, order details.
* Process: delivering the order.
* Output: “successfully delivered” message to customer mobile or other devices.

Non-Functional Requirement

* **Maintainability** - Should be able to evolve the software to support new requirements.
* **Portability** - Should be able to use the system in a different configuration.
* **Reliability** - The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day.
* **Performance.**

3.2 Explain how effective your analysis was in terms of the methodology that you

used.

Agile is an iterative, team-based approach to development. This approach emphasizes the rapid delivery of an application in complete functional components. Rather than creating tasks and schedules, all time is “time-boxed” into phases called “sprints.” Each sprint has a defined duration (usually in weeks) with a running list of deliverables, planned at the start of the sprint. Deliverables are prioritized by business value as determined by the customer. If all planned work for the sprint cannot be completed, work is reprioritized and the information is used for future sprint planning.

As work is completed, it can be reviewed and evaluated by the project team and customer, through daily builds and end-of-sprint demos. Agile relies on a very high level of customer involvement throughout the project, but especially during these reviews.

* The customer has frequent and early opportunities to see the work being delivered, and to make decisions and changes throughout the development project.
* The customer gains a strong sense of ownership by working extensively and directly with the project team throughout the project.
* If time to market for a specific application is a greater concern than releasing a full feature set at initial launch, Agile can more quickly produce a basic version of working software which can be built upon in successive iterations.
* Development is often more user-focused, likely a result of more and frequent direction from the customer.

3.3 Explain what criteria you used to select the methodology and justify your choice

in terms of your business problem.

In this agile scrum technologies, we can separate the works, so we get the work very easier. If everyone gets their jobs and understand their works and do it well. In traditional we have a problem that is we can’t share the work and it takes more time so we can’t satisfied the customer in this agile scrum technologies have an easy way they have separate their works, and if customer order an item, the order should be send to chef coordinator cashier also so they each know what happen then they will satisfies customer. If customer satisfies the company get grow, then automatically they will get success.

Test Automation starts at the unit level. Unit tests should be written by developers for any new feature that is developed. These Unit Tests form the foundation of a larger automation practice that spans all the way up to the System GUI Tests.

It is the responsibility of the developers to ensure that for every new feature that is developed, a set of coherent and solid Unit Tests are written to prove that the code works as intended and meets the requirements.

Unit Tests provide the most ROI to the team as they are very quick to run, easy to maintain and modify (as there are no dependencies) and when there are errors in code, it is quickly fed back to the developer.

**Task 4**

4.1 Design the system to meet user and system requirements that you identified in

the previous task.

Draw Data Flow Diagrams, Flow Charts, UML models as appropriate for the system you are designing.

Depending on the methodology that you are using (Traditional or Agile),

describe design aspects specific to your selected methodology. (i.e. any

special aspects with regard to database designing, UI designing,

architectural designing)

Depending on the nature of the system that you are designing (i.e. database application, web or mobile application or any other type of application)

describe specific tools & techniques that you use for designing it.

**Catering Service Management System**

Introduction

A customer can order through online or directly. An order may include meals or packages. A meal or packages may include menu items like starter, main dish, dessert, or beverages.

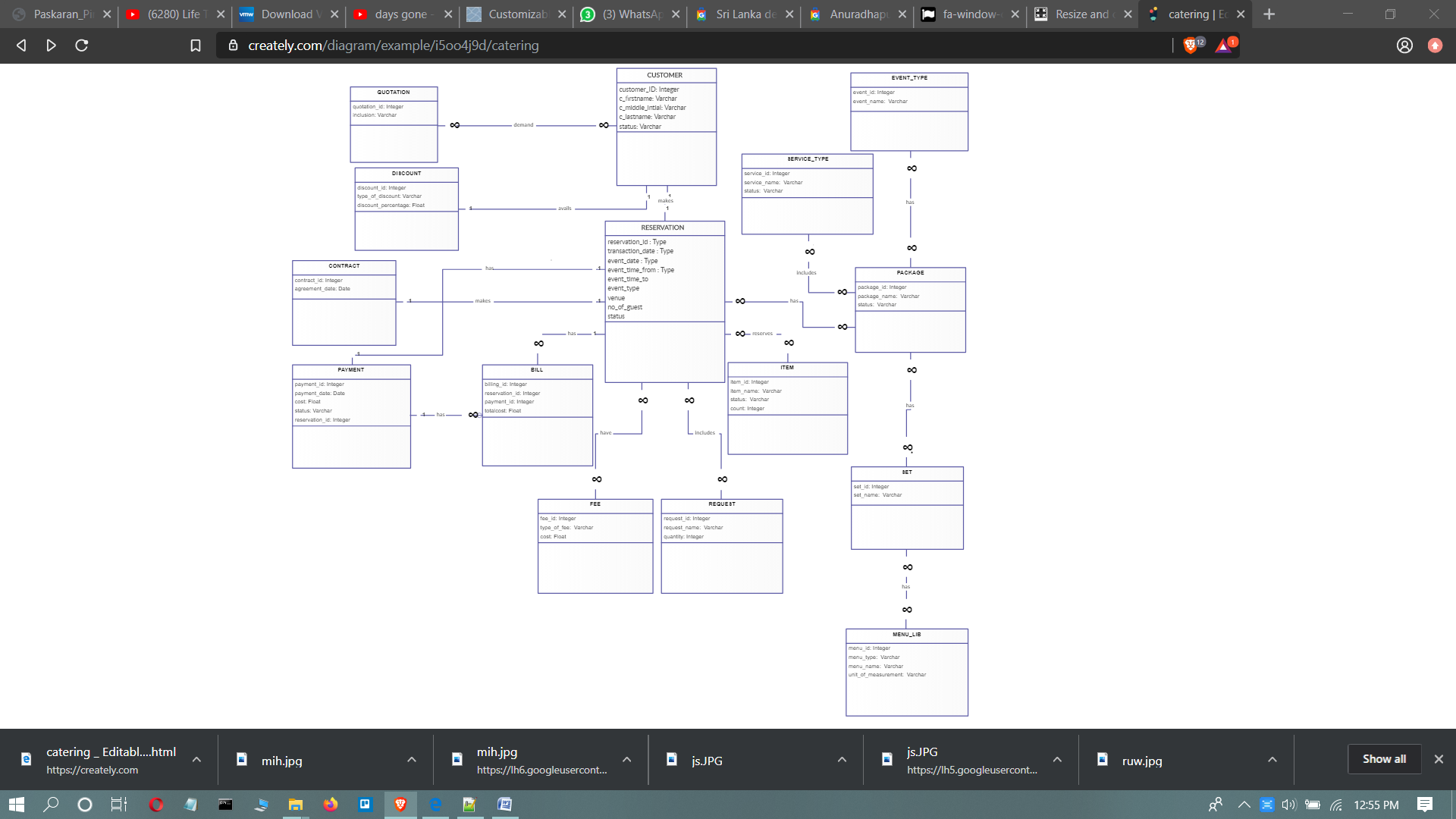
If customer visiting directly, coordinator will meet the customer and helps to identify the suitable package or meals. Customer can choose the meals according to their needs or they can choose packages as they can receive discounts for package selection. Once the customer selected the meals coordinator can calculated the total cost. If customer is satisfied with the selected meals or packages and cost customer may confirm the order. If customer is not satisfied with the selected meals, they can go for packages or can select the meals again according to their wish.

Once the customer confirms the order, coordinator will pass the order to cashier and manager for further process. Cashier will prepare the invoice for the order. Customer can pay either by credit card or cash. Customer must pay the 35% of the cost as initial amount when confirming the order rest of the amount can be paid later or on delivery time. Cashier will aware of the packages and the discount.

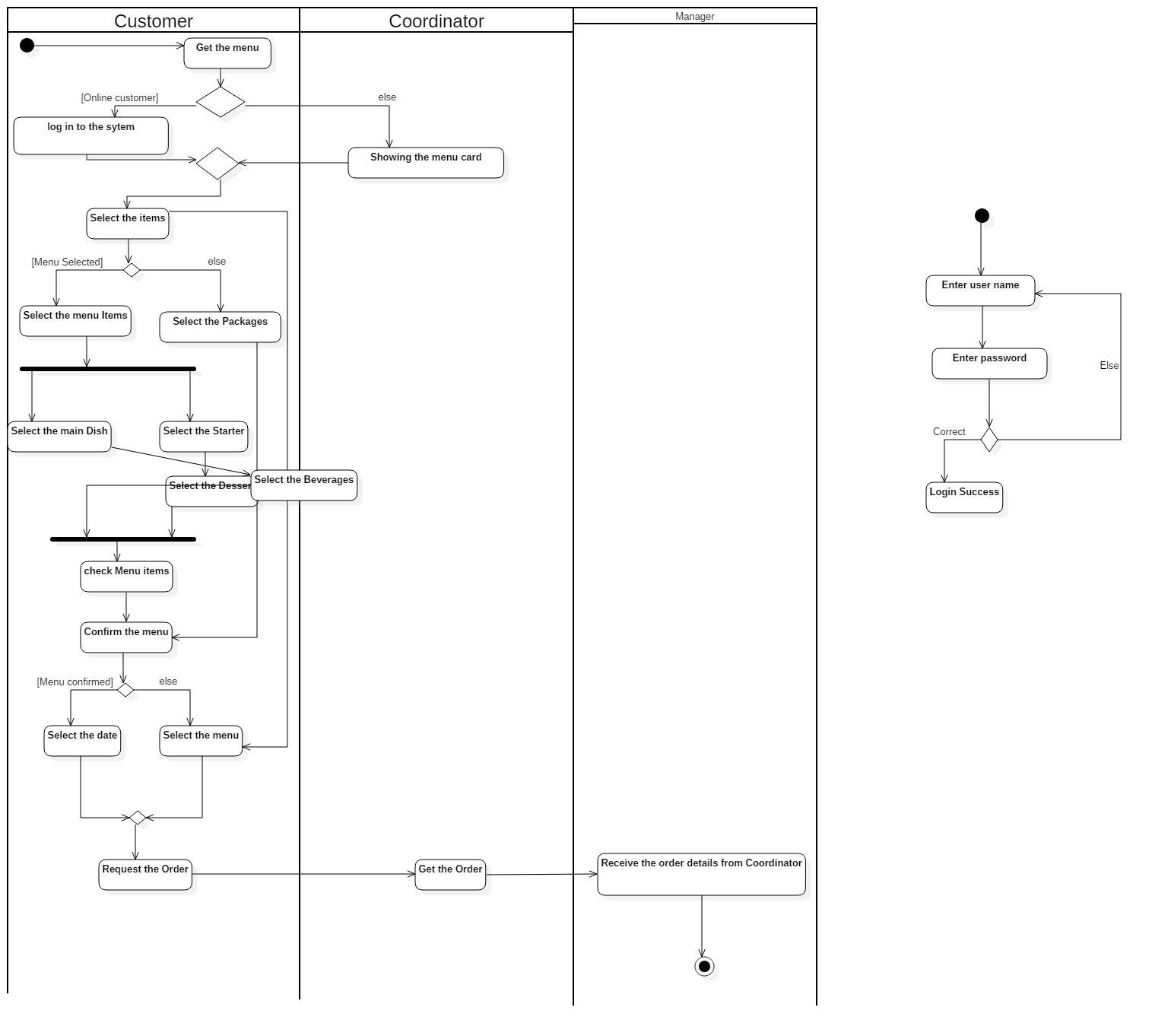
Customers who are ordering through online they can get loyalty points. When they are selecting meals or packages, at that moment they can get the sum. Customer can view the status of the order and the previous order they placed. Customers can give their feedback about the catering services so that manager can improve the services of the catering.

Manager will receive the orders from coordinator. Manager will generate all the reports like sales, orders, payment etc. Manager will prepare the schedule and will confirm the staffs are working according to their schedule. Manager will expect to update the stock levels immediately. When stock is out of level manager will send the notification to vendor. Manager will expect the quality of materials and low prices from vendors.

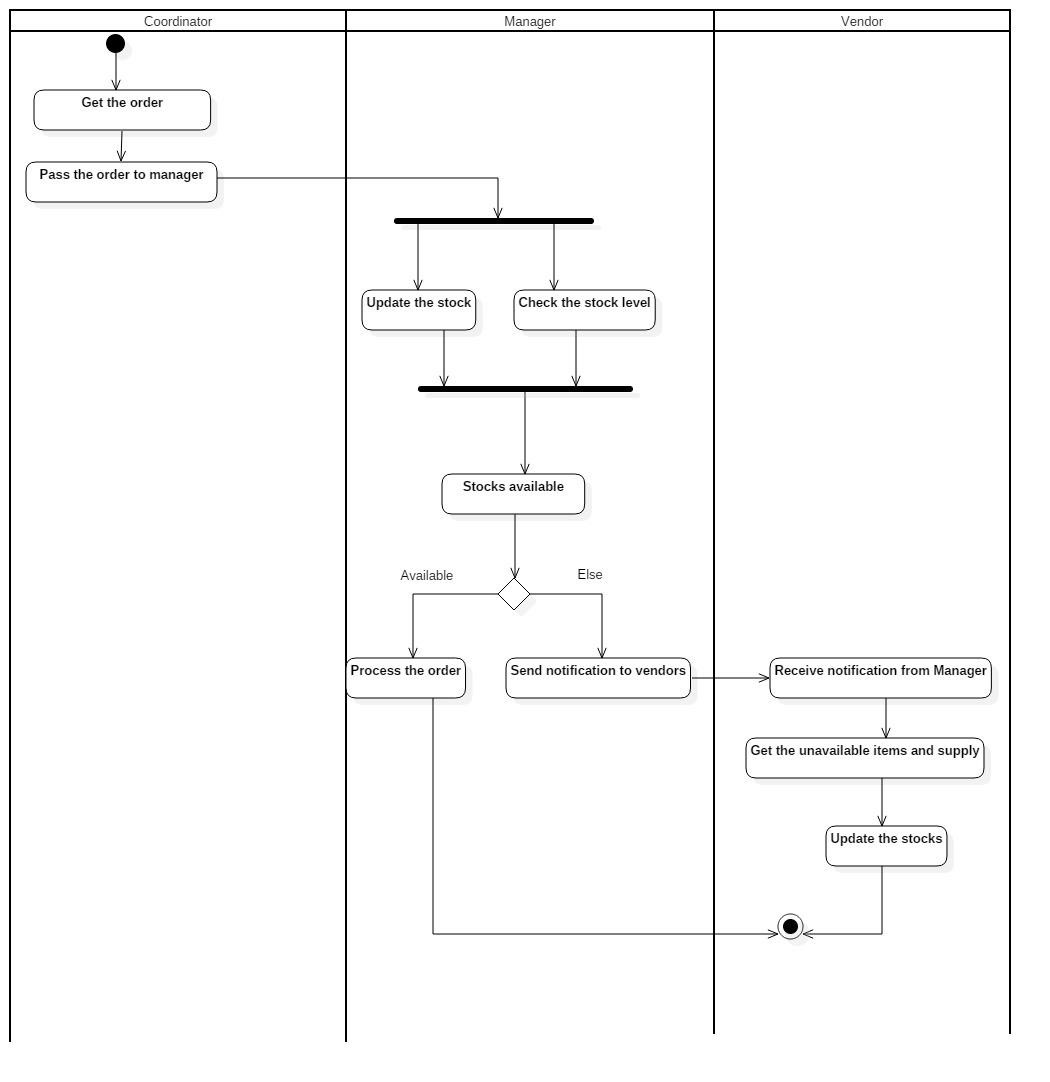
Coordinator should pass the finalized and confirmed order to chef, as chef need to confirm the needed tools for that order. Chef expects the materials, things should be on time.

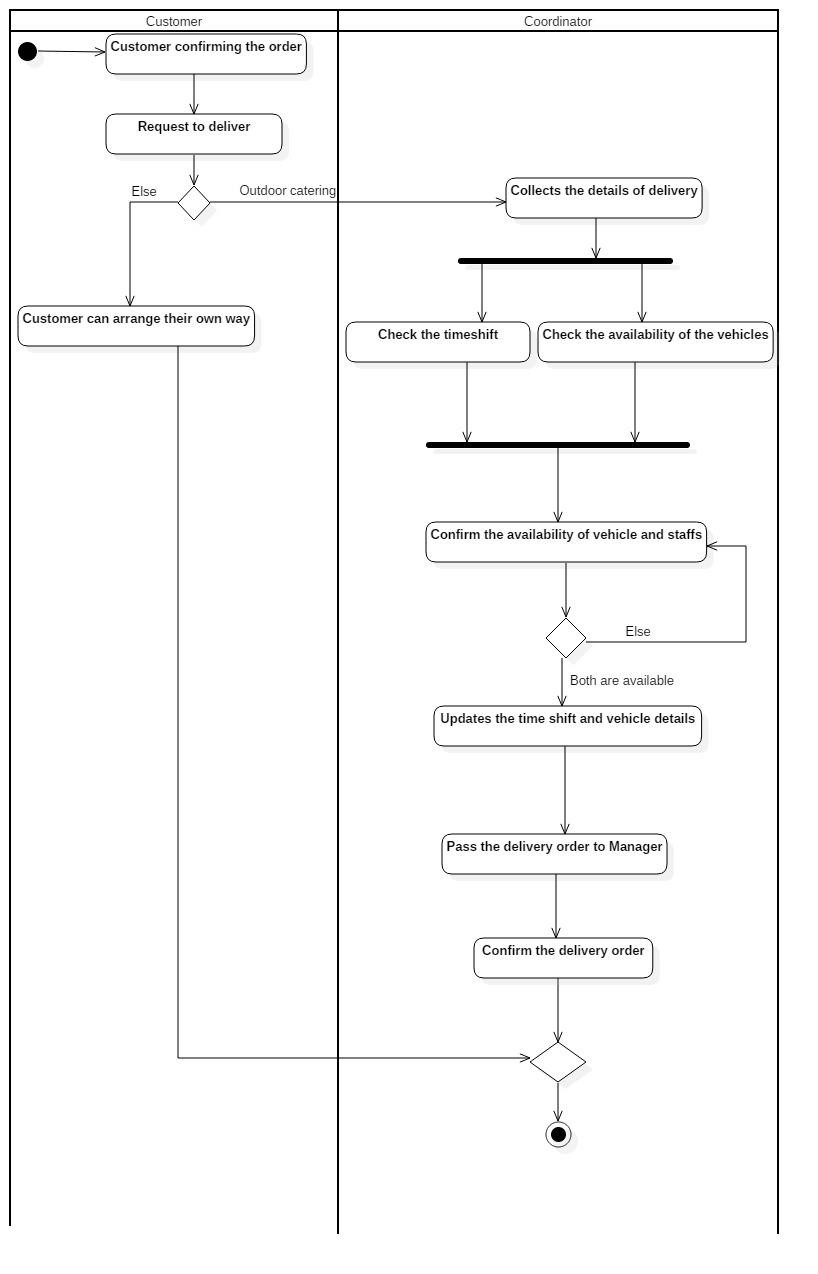


**Confirm the Order**

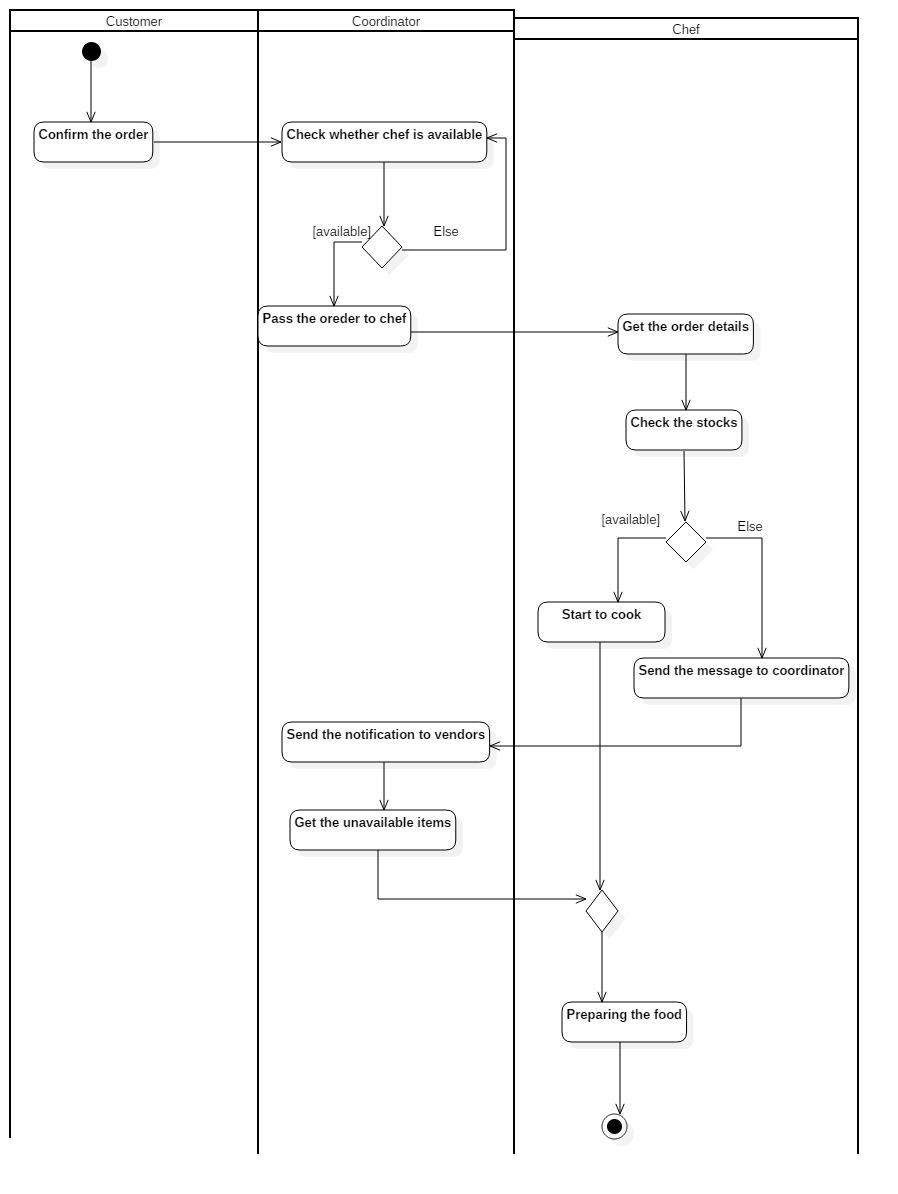


**Getting supply from vendors**

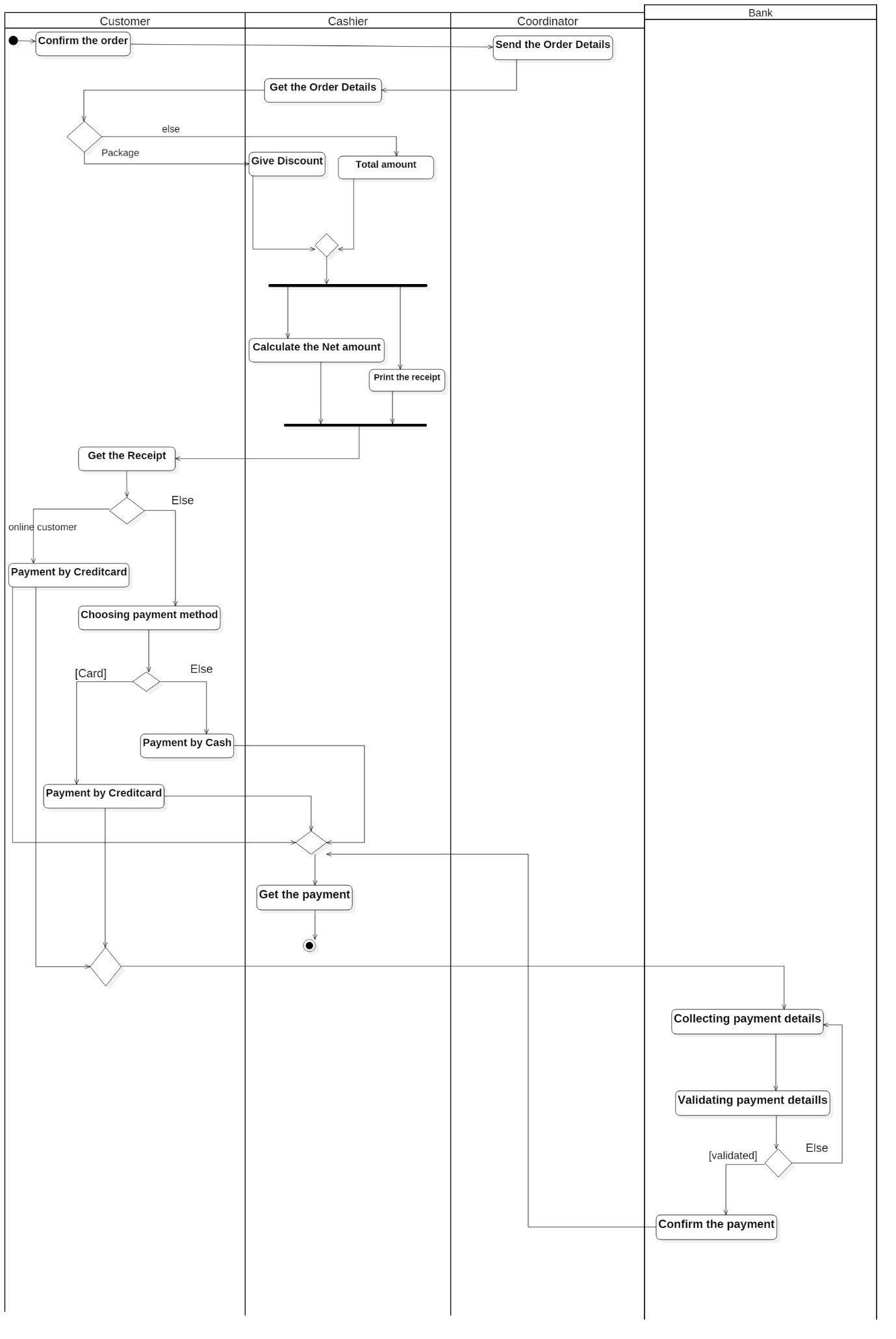
****

  **Confirming the delivery**

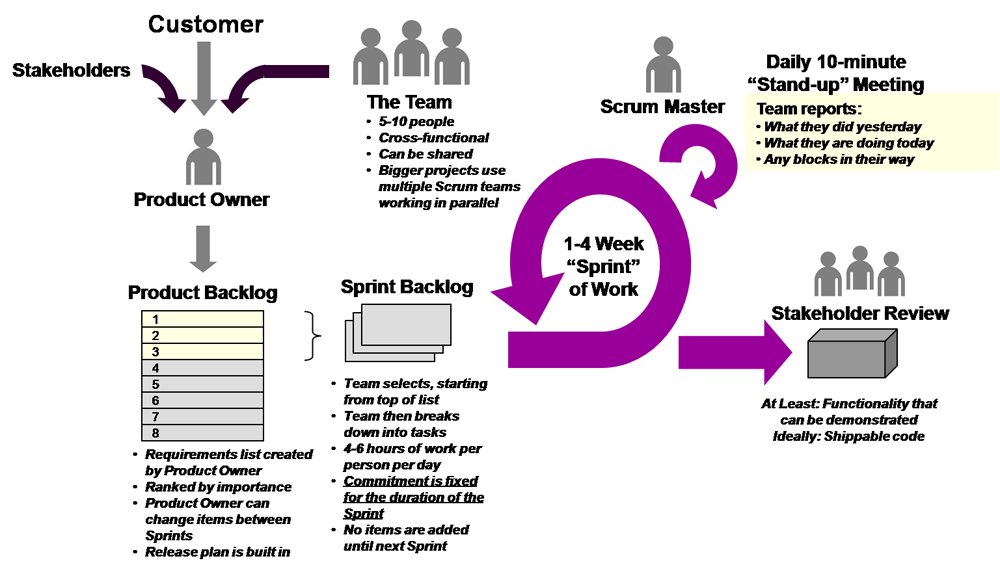
**Preparing food/meals**



**Payment from customer**



Agile methodology



When we use agile methodology,

* It was easy & quick to adapt
* Management needs frequent changes to requirements from their client.
* Management had no tools to track the requirement.
* Product Owner: This person coordinates with various stakeholders, creates and maintains product backlog, and works to maximise the ROI on the product.
* Scrum Master: Arole orthogonal to Product Owner, this person helps the team achieve tasks and build long term sustainability. He is a coach for the team.
* Developers: Everyone else on the team is called a developer, irrespective of their qualifications, expertise and work.
* Product Backlog: This is an artifact created by the Product Owner. It contains all the requirements in the form of user stories. These stories will someday become a part of the working software.
* Sprint**:**This is a time boxed period in which the team agrees to develop specific user stories given by the Product Owner and works to convert them to a working software by the end of it.
* Sprint Backlog**:**Top priority items from the Product Backlog that the team will work on during the sprint are put in the Sprint Backlog. These stories will ideally become the working software at the end of the sprint.

Using the Agile methodology and Scrum framework, let’s look into developing the following features for our web application:

* Rate and Review Restaurants
* Filters for searching restaurant and dishes
* Online ordering

Developing these features involve following steps:

1. Ideation
2. Design
3. Implementation
4. Testing
5. Deployment

**Rate and Review Restaurants:**In the ideation phase, the entire scrum team discusses how to build a simple working prototype which involves left swipe-right swipe for like-dislike thus making it possible for users to quickly rate the catering service management.  
In the design phase, after getting feedback from the users/ customer, team designs the features with additional interactions.  
After design comes the implementation phase where the developers code this simple rating feature for the testing phase, a test version is shared internally to be tested and made bug-free by the developers.  
Post testing, the feature is made live for the users to use.  
This marks the end of the sprint.

Now the product owner performs various users testing sessions and captures user needs. These requirements are then worked upon as part of the next iteration.  
In future iterations, the team implements features like adding reviews, photos etc.

**Filters for searching restaurant and dishes:**Team creates a persona for the target user based on user research and concluded that having a search feature will solve a certain pain-point . A working prototype is created for the search bar.  
It is then tested for usability. Based on feedback the team designs and implements the search feature. Then it is hosted as a beta version and distributed internally for testing and simultaneously developers fix the bugs. Finally the feature is deployed.  
In the future iterations, features like filters and location based search are added.

**Online Ordering:**The team created an inventory service management portal for the management owners where the menu and pricing can be updated. This provided a nice base to start an online catering service.  
Post initial ideation, user research is conducted and personas are created based on which the team designed and implemented the online order feature. The beta version was rigorously tested and all bugs were fixed. Initial hypotheses were validated by performing contextual enquiry sessions. After iterating over this, a refined version was deployed with only credit cards as the payment method.  
Later iterations would include multiple payment methods and features like delivery scheduling.

4.2 Assess the effectiveness of your design, paying attention to the methodology.

In here user asks how to be successful and he asks he want to do separate system for everyone. He wants to give the work in contributory to everyone, so he wants to give a work to coordinator, Cashier, chef and etc. so in here I am selecting spiral agile methodology, in agile methodology we can give the project to everyone under the scrum master so that is the reason for I am selecting agile and it will helps to user.

4.3 Justify your design decisions in terms of your business problem.

I suggest mobile application because it has many advantages

### Customer Experience

A mobile app helps restaurant owners establish a fast and convenient customer experience, which means reduced wait time and increased sales. Ordering is the first feature that comes to mind when you’re thinking about a catering app. First, they don’t guarantee any ROI for catering services owners. Second, for big and crowded catering or chains online orders, if not organized right, can lead to a real logistical hell at the catering services.

Meal ordering is a more valuable feature, that, unfortunately, is less utilized by catering app developers. However, allowing app users to choose and order their meals in advance, thus significantly reducing visitor’s wait time, will help not only establish better customer experience at your restaurant but also make the staff’s work easier and faster. You can add even more value to this feature by letting app users browse their order history and store their favorite orders. This will make the ordering process even faster and more convenient.

### Delivery

Today, extending a restaurant service with the food delivery feature becomes a reasonable option for restaurant mobile app developers. And while it may not be the primary feature to include in a new app, it’s still a prospective direction for catering services.

### More payment options

Online payments are a big thing now. All types of digital solutions dealing with commerce have adopted it, so have restaurant apps. There are two types of online payments you can adopt as a restaurant app developer to make restaurant services more effective. The first is mobile payments processed via payment gateways such as Stripe or Braintree (a PayPal subsidiary).

However, restaurant owners may dislike this idea if mobile payments service is provided by a booking platform since it would mean more additional fees that may have a negative impact on the ROI.

### Sufficient feedback

A business’s success depends on gathering visitor’s feedback. Most of the restaurant apps and food discovery platforms provide users with the possibility to leave a review on their visit. User reviews play two important roles at the same time. First, as a business owner or/and service provider, you can get feedback on your services to understand how to improve them, thus satisfying your customers even more. But – not less importantly – your visitors’ reviews act as additional motivators for other people to visit a certain restaurant. Thus, it may be a logical decision to place user reviews on the apps/restaurant’s website and allow people to share their reviews.

Reason for using uml

With the use of UML, an appropriate UML development tool, and an application process or methodology, the design and refining of the application is shifted from the development phase to an analysis and design phase. This reduces risk and provides a vehicle for testing the architecture of the system before coding begins. The analysis and design overhead will eventually pay dividends as the system has been user driven, documented and when it’s time to start developing, many UML tools will generate skeleton code that will be efficient, object oriented and promote re-use.