Question 2: Requirements Engineering

Part a.

Answer:

Requirements Engineering Process

The requirements engineering process varies from domain to domain. But the general activities involved are:

- **Elicitation**
- **Analysis & modelling**
- **Feasibility**
- **Document**
- **Validation:**

Sommerville has given the depiction of this as below:

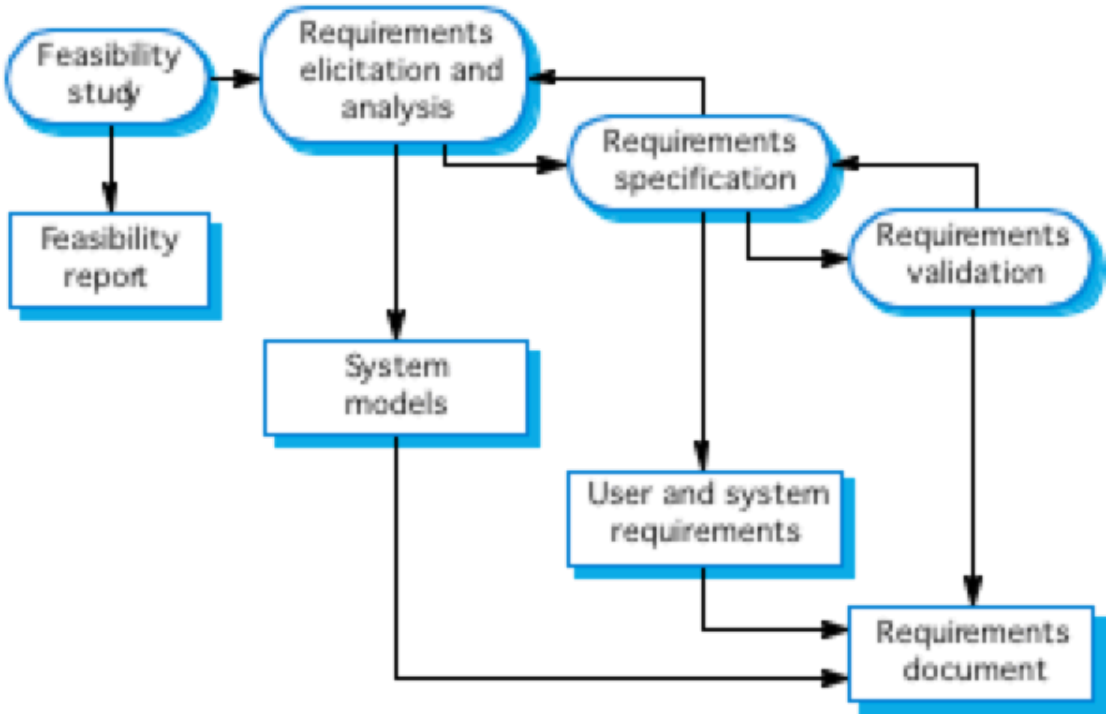

Figure 2 - Requirements Engineering Process

Importance of requirements engineering process

The front end activity of any software development process is requirements engineering. To handle changes in requirements, requirements engineering plays a crucial role for software development.

Most of the requirements engineering takes place in the initial stage of the project. The later you change or modify requirements in the life cycle, the more costly it will be.

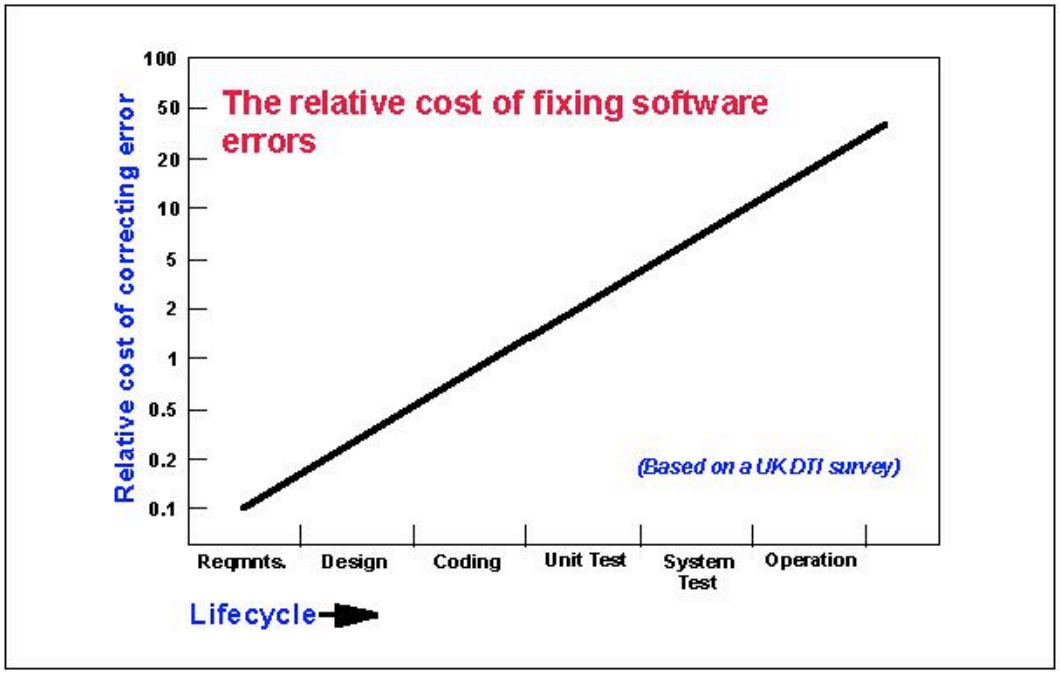

Figure 3 - Cost of fixing errors in SDLC

As we have studied in Section 1.4.5 of Unit 1, that the cost of fixing errors rises with each stage. So if the requirements have been well-defined, the cost of fixing any ambiguities or incomplete requirements will decrease.

A good amount of preparation is needed before starting off with a project. Requirements engineering is different for different domains or systems. As a base, the feasibility of the project needs to be assessed. Along with it, the risks associated with it should be identified as well. The estimation of project budget, timeline and feasibility can be calculated from the requirements. Any risk should be re-evaluated regularly throughout the development lifetime of a system because changes in the environment can change the associated development risks.

Today, many people have a good understanding of software but not all are well aware of the need for requirements engineering, they are not aware of the fact that how critical requirements are for each life cycle stage, namely; planning, designing, testing and operating.

The automated systems are not only based on software containing database and applications but they also include networking, people, procedures etc. These kinds of systems totally transform the way what an organization and the process for doing it. Any system that is being introduced in to an organization should fit in the working environment. So to better define the business needs, requirements engineering is an important issue. One needs to know about the users of the system, any other interface that the system would be interacting with, any constraints, etc. Therefore it is requirements engineer's task to get the complete and right picture of the system to be developed.

Martin & Leffinwell analyzed and described the importance of requirements management.

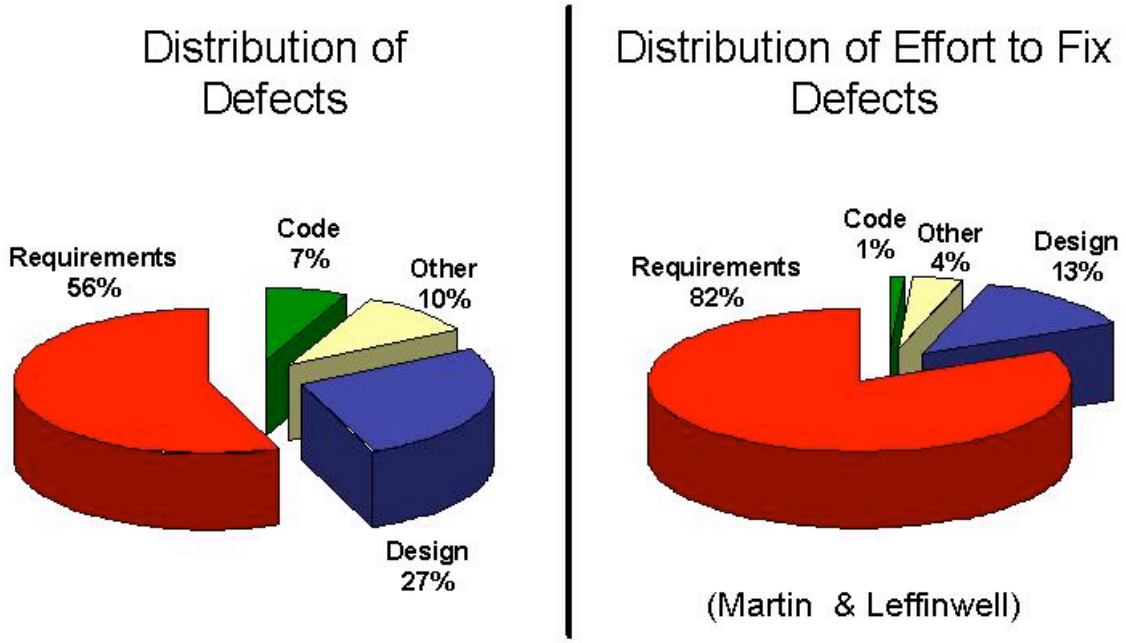

Figure 4 - Distribution of defects

As we see in the figure above, the more the defects and errors are figured out in the requirements phase, it decreases the efforts of fixing it in latter stages of development life cycle.

To summarize, the ***need and importance of the requirements engineering process*** is marked by the following factors:

- Requirements are unclear, ambiguous and have several sources to come from.
- Requirements are not always easy to define in words.
- There are many distinctive kinds of requirements having different levels of detail.
- It might become difficult to manage if the number of requirements keeps on increasing.
- Requirements are interconnected and are contingent upon one another.
- It's not an easy task to satisfy the requirements.
- There are many stakeholders involved meaning requirements should be approved by all, thus management and agreement to a common set of requirements by the cross functional group is very important.
- Requirements most likely change.
- The relative importance of some requirements depends on the availability of time and budget.
- Most importantly, the latter stages of the life cycle are dependent on the accuracy of requirements.

Part b.

Answer

Feasibility Assessment

Once the feasibility assessment of the project is done and the system is found to be feasible, then the development process can start.

The feasibility assessment would be done considering the following factors:

Developer Side Issue:

1. Has the problem or a one like it, been solved before? **Yes**
2. Is it a data processing problem? **Yes, it includes this.**
3. Is the required input data available? **Yes**
4. Will the technology performance be adequate? **Yes**
5. Will the proposed solution be affordable and delivered on time? **Yes, following the best development life cycle.**
6. Are there circumstances in the system's intended operational environment which may jeopardise its successful use? **Yes, there might be factors like end users find it difficult to operate, essential hardware might not be delivered on time.**

Output of the feasibility assessment: System is feasible but its success depends on certain features.

Stakeholders

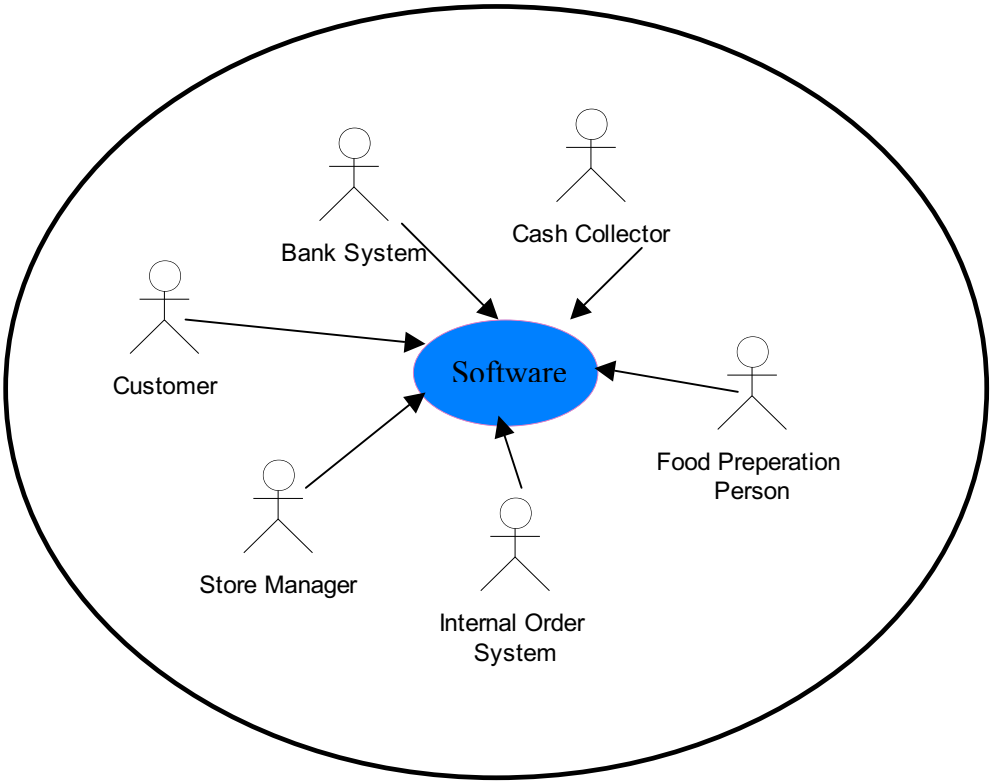

Figure 5 – End users - I

The High Tech Restaurant and Bar system has several stakeholders. The customers, store manager, internal ordering system, food preparation person (cook and bartender), Bank system and cash collector are end users of the system use the system as shown in Figure 5, above.

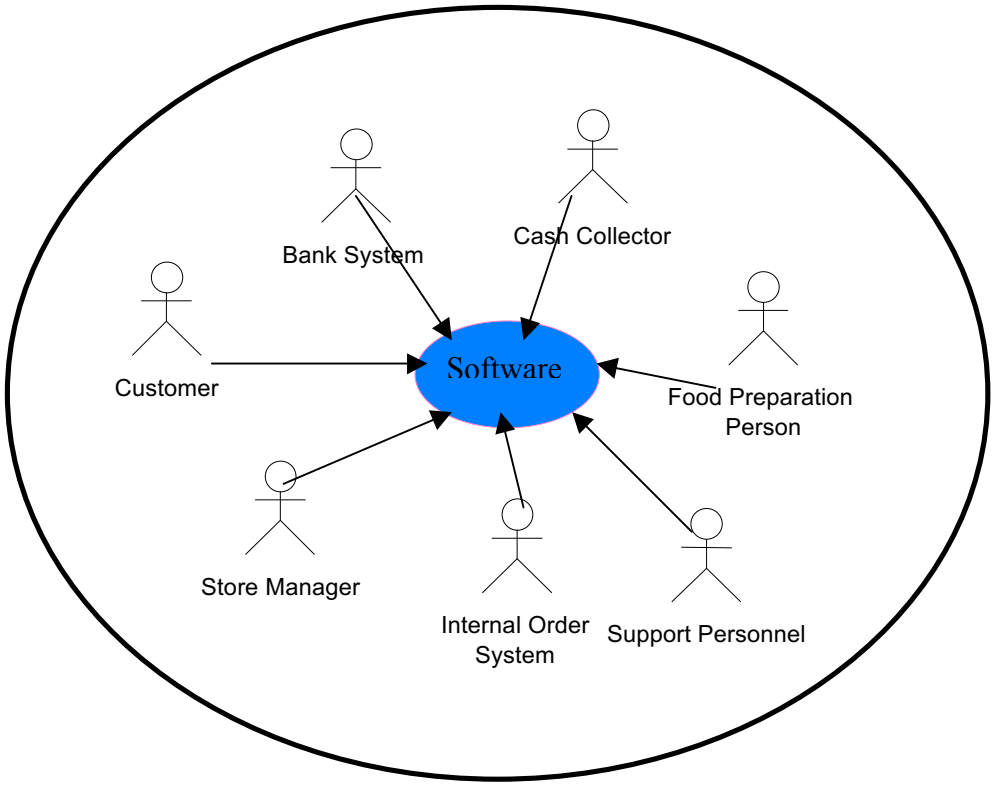

Figure 6: End Users – II

Generally all software requires maintenance and support over a period of time, so technically the end users will be given support by the maintenance and support people. So as you can see in Figure 6, these immediate stakeholders have been added.

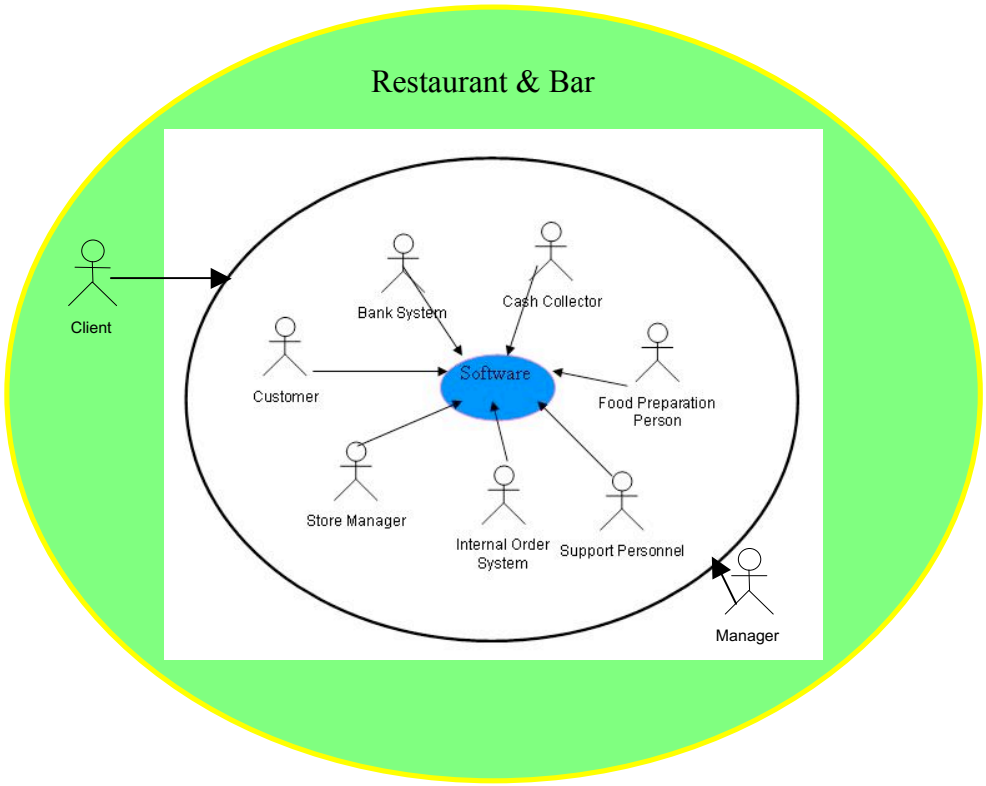

Figure 7 – Indirect Users

Next in figure 7, we see the stakeholders that might not have a direct contact with the system but still the system has an impact on them. The owner of the Restaurant and Bar will be affected by the fact that how well the system that he has introduced in his restaurant fares. We may refer to him as the functional beneficiary or the client who is commissioning the system. The manager responsible for end users also falls in to the same category of stakeholders.

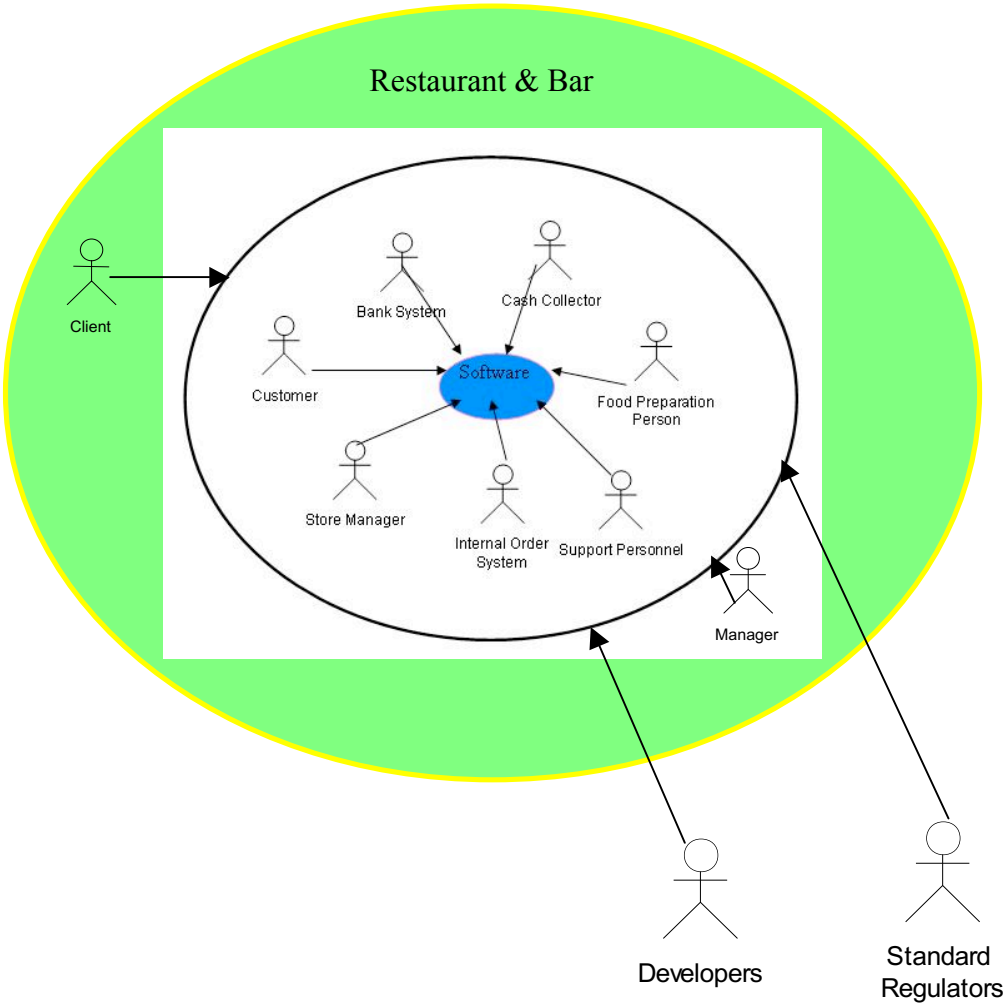

Figure 8 - Developers & Regulators

The other stakeholders out of the restaurant and bar are the developers and the standard regulators as shown in Figure 8, above. Developers include the analysts, designers, programmers, testers, quality assurance representatives, trainers, project managers etc. But developers play the role of stakeholders during the process of requirements engineering. The Standard regulators are the personnel from the organization or auditors from external agency who ensure that the system conforms to certain standards and other constraints.

Techniques for Information Gathering

The information gathering techniques give the project team members a way to extract the client needs or requirements from stakeholders and then helps in validating these requirements with stakeholders. To get the complete picture of the system to be developed, several techniques for information-gathering would be employed. The techniques to be used for high tech restaurant and bar are as follows:

1. Interview technique

Benefits
The interviewing technique has many benefits:

- Requires least preparation.
- Less planning & scheduling effort required if compared with large workshops.
- Lesser stakeholder commitment of the individuals and small groups than large workshops.
- An opportunity to find and clarify any topic in detail.

2. Joint Application Development (JAD) technique

Benefits

- Concurrent assembling and merging of huge amount of information is taken into account using this technique
- An enormous quantity of fine information is gathered in shorter time period.
- Any differences are determined without any delay by the expeditor
- This technique is provision of a discussion where compound viewpoints relating to a topic or subject can be examined.

3. Survey method

Benefits

1. It can be rotated to several stakeholders or any other origin of information.
2. In a short time period, a meaningful amount of data can be gathered.
3. After interviewing technique, survey method can be used to verify the assumptions made by the system analysts.

Sequence of Stages & Activities & People involved in each activity

ID		Task Name	Resource Names
1		**High Tech Restaurant and Bar**	
2		**Initial Investigation**	**System Analysts,Client**
3		Current Manual System	
4		Basics of automated applications	
5		Tools and techniques	
6		Existing Data	
7		**Submission of Project Proposal**	**System Analysts,Project Manager,Client**
8		Domain Understanding	
9		Information gathering	
10		Initial Draft	
11		Project Manager review	
12		Review Rework	
13		Client Review	
14		Review Rework	
15		Project manager review and approval	
16		**Approval of PPF by client**	
17		**Submission of Project Specification Form**	**Project Manager,System Analysts,Client**
18		Project History Investigation	
19		Budgeting	
20		Research Planning	
21		Development Planning	
22		Evaluation & Test Planning	
23		Project Manager Review	
24		Review Rework	
25		Client Review	
26		**Approval of PSF by client**	
27		**Cycle 1: Inventory Module**	
28		**Requirements level**	**System Analysts**
29		Requirements definition	
30		Validation of requirements against client's requirements	
31		Sub-System specification	
32		Verification	
33		**Design level**	**Designer**
34		Sub-System Design	
35		Sub-System test	
36		**Implementation level**	**Programmers,Testers**
37		Module Specification	
38		Module Test	
39		**User Manual**	**Technical Writer**
40		Documenting User guide for Inventory Module	
41		**Client Review**	**Client,Team Lead**
42		Review Rework	
43		**Approval**	Project Manager,Client

ID		Task Name	Resource Names
44		**Cycle 2: Menu Module**	
45		**Requirements level**	**System Analysts**
46		Requirements definition	
47		Validation of requirements against client's requirements	
48		Sub-System specification	
49		Verification	
50		**Design level**	**Designer**
51		Sub-System Design	
52		Sub-System test	
53		**Implementation level**	**Programmers,Testers**
54		Module Specification	
55		Module Test	
56		Integration & Testing	
57		**User Manual**	**Technical Writer**
58		Documenting User guide for Menu Module	
59		**Client Review**	**Client,Team Lead**
60		Review Rework	
61		**Approval**	Project Manager,Client
62		**Cycle 3: Customer Module**	
63		**Requirements level**	**System Analysts**
64		Requirements definition	
65		Validation of requirements against client's requirements	
66		Sub-System specification	
67		Verification	
68		**Design level**	**Designer**
69		Sub-System Design	
70		Sub-System test	
71		**Implementation level**	**Programmers,Testers**
72		Module Specification	
73		Module Test	
74		Integration & Testing	
75		**Client Review**	**Client,Team Lead**
76		Review Rework	
77		**User Manual**	**Technical Writer**
78		Documenting User guide for customer module	
79		**Approval**	Project Manager & Client

ID		Task Name	Resource Names
80		**Cycle 4: Order Module**	
81		**Requirements level**	**System Analysts**
82		Requirements definition	
83		Validation of requirements against client's requirements	
84		Sub-System specification	
85		Verification	
86		**Design level**	**Designer**
87		Sub-System Design	
88		Sub-System test	
89		**Implementation level**	**Programmers,Testers**
90		Module Specification	
91		Module Test	
92		Integration & testing	
93		**User Manual**	**Technical Writer**
94		Documenting User guide for Order Module	
95		**Client Review**	**Client,Team Lead**
96		Review Rework	
97		**Approval**	Project Manager & Client
98		**Cycle 5: Payment Module**	
99		**Requirements level**	**System Analysts**
100		Requirements definition	
101		Validation of requirements against client's requirements	
102		Sub-System specification	
103		Verification	
104		**Design level**	**Designer**
105		Sub-System Design	
106		Sub-System test	
107		**Implementation level**	**Programmers,Testers**
108		Module Specification	
109		Module Test	
110		Integration & testing	
111		**User Manual**	**Technical Writer**
112		Documenting User guide for Payment Module	
113		**Client Review**	**Team Lead,Client**
114		Review Rework	
115		**System Testing**	**Team Lead,Testers,analysts,Project Manager**
116		Approval	
117		**Deliver**	
118		Presentation/Demonstration	Team Lead
119		**Support**	Support Personnel

Note: a Gif image of project plan has been zipped with the assignment for better viewing.

Outputs of the project

Introduction

Software scope

An abstract description of the proposed system is as follows:

- Privileged customers have their logins
- Ordering of food and drinks is automated
- Graphical displaying of menus to customers with options like chef's recommendation and special deals

- Orders are automatically forwarded to the kitchen and bar
- Online payments
- Discreet recall of waiter
- Customer can access web and mails
- Synchronization with customer's own technology

Expected ***benefits and goals*** of the proposed system can be:

- It will help in avoiding and eliminating the problem of rush hours or lengthy lines of customers thus maximizing the number of customers
- Reliable and secure method of payment
- Making ordering and payments method easier for customers
- Customers have the added utility of browsing the web and checking mails instead of waiting for their order
- Privileged customers have added options like their own customer accounts
- There will be a reduction in the number of employees working at the counter, thus saving labor costs.
- As the system will be computerized so it would be less prone to errors.
- Reliable and efficient management of stock, menu and reports.

Overall Description

External Interface Requirements

Hardware Interface

1. Touch screen pads
2. Customer end display
3. Card reader
4. Modem
5. Kitchen Display
6. Bar display

Software Interface

1. Menu Module
2. Payment module
3. Ordering module
4. Inventory Module
5. Customer Module

User Interface

General

1. User should be able to navigate the system without any difficulty.

2. System supports native language of the country and other commonly spoken languages.

3. User should be able to place order according to his choices

4. User should be able to make payment using cash/credit/debit card.

5. User should get a bill or receipt after making the payment.

6. The system should be designed keeping in mind its security and safety. It should be robust enough to deal with any hackers or severe handling.

Use Case Model

A use case represents a discrete unit of transaction between a user and the system. The use cases identified after the client's interview are depicted in the use case diagram.

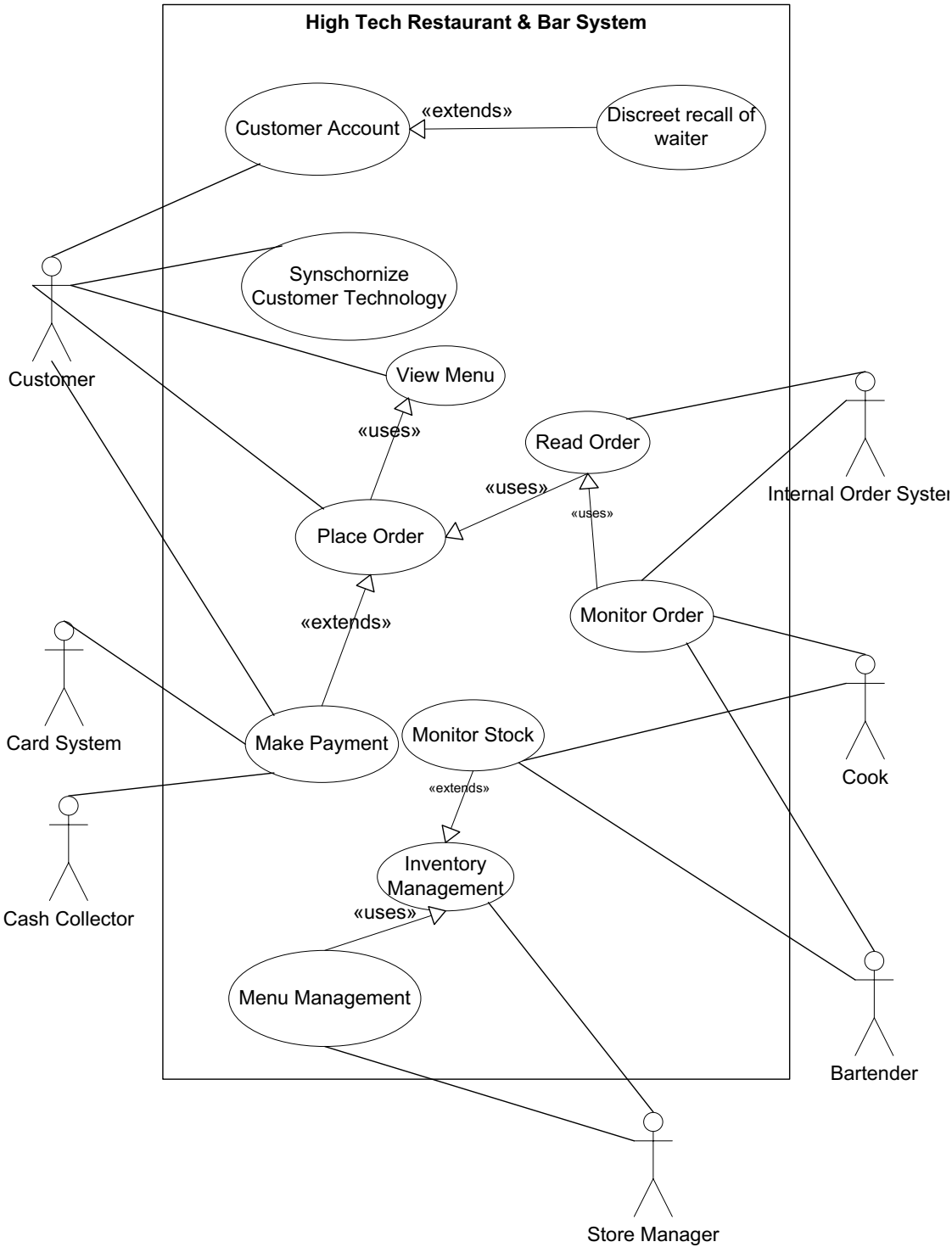

Figure 9 - Use Case Model

As you can see in the above use case model that a customer can login, synchronize his technology with the system, view menu, place order and make payment. The order placed by the customer is read by the internal ordering system and is passed onto the cook and the bartender. The cook and bartender monitor stock and order and prepare the order for the customer. The inventory and menu are managed and updated by the store manager. When a customer makes a payment these are handled by the cash collector and the debit/credit card system.

Use Case Description

***Use Case 1*.** Customer Account

Primary Actor: Customer

Flow of Events:

1. User selects his language preference for the session.
2. User enters customer ID and password
3. User clicks on submit
4. User is logged in

***Use Case 2*.** Discreet recall of waiter

Primary Actor: Customer

Flow of Events:

1. User views 'discreet recall of waiter'

***Use Case 3*.** Synchronize Customer Device

Primary Actor: Customer

Flow of Events:

1. User selects his language preference for the session.
2. User Selects 'Customer' menu.
3. User selects 'synchronize customer device' option.

***Use Case 4*.** View Menu

Primary Actor: Customer

Flow of Events:

1. User selects his language preference for the session.
2. User Selects 'View menu'.

3. User selects one of the following options:
    - Chef's recommendation
    - Special/Combination Deals
    - Starters
    - Main Meal
    - Drinks
    - Desserts

***Use Case 5*.** Place Order

Primary Actor: Customer

Flow of Events:

1. User selects his language preference for the session.
2. User selects from the menu.
3. User selects from the drinks menu
4. User selects from the chef's recommendation, special deals, starters or desserts
5. User confirms the order

***Use Case 6*.** Make Payment

Primary Actors: Customer, Credit/Debit system, cash collector.

Flow of Events:

1. User enters the mode of payment. (Credit/Debit/Cash)
2. User selects to make the payment in cash
3. Cash collector collects the money and gives back the change if required.
4. User selects to make the payment by credit/debit card
5. User enters required information and gets the final bill.

***Use Case 7*.** Menu Management.

Primary Actor: Store Manager.

Flow of Events:

1. The Store manager enters the system with his login and password.
2. The store manager selects:
    - Add Menu
    - Update Menu
    - Delete Menu
    - Update Price
3. The Store manager updates or adds the required information.
4. The Store manager saves the changes.

5. The Store manager logs out.

***Use Case 8*.** Inventory Management

Primary Actor: Store Manager

Flow of Events:
1. The store manager enters the system
2. Store manager selects:
    - Add inventory
    - Update Inventory
    - Delete Inventory

3. Store manager updates the inventory accordingly.
4. Place a demand for the required inventory.

***Use Case 9*.** Monitor Stock

Primary Actor: Cook, Bartender

Flow of Events:
1. The cook or bartender enters the system
2. Updates the menu accordingly.
3. Place a demand for the required inventory.

***Use Case 10*.** Monitor Order

Primary Actor: Cook, Bartender

Flow of Events:
1. The cook or bartender enters the system
2. View the orders placed by the customer
3. Take the order and mark them as taken by them
4. Updates the order status accordingly.
5. Assign a waiter for order delivery

***Use Case 11*.** Read Order.

Primary Actor: Internal Order system.

Flow of Events:
1. Internal order system reads the order
2. Communicates the order to the cook and bartender

APPENDIX A

Key for probability and impact risk management

KEY	
PROBABILITY	
Very likely (H)	High chance of this risk occurring, thus becoming a problem > 70%
Probable (M)	Risk like this may turn into a problem once in a while {30% < x < 70%}
Improbable (L)	Not much chance this will become a problem {0% < x < 30%}
IMPACT	
Catastrophic (H)	Loss of system; unrecoverable failure of system operations; major damage to system; schedule slip causing launch date to be missed; cost overrun greater than 50% of budget
Critical (M)	Minor system damage to system with recoverable operational capacity; cost overrun exceeding 10% (but less than 50% of planned cost
Marginal (L)	Minor system damage to project; recoverable loss of operational capacity; internal schedule slip that does not impact launch date cost overrun less than 10% of planned cost

References

[B]

[Boehm, 1988]
Boehm, B. W. "*A Spiral Model of Software Development and Enhancement."* TRW Defense Systems Group. [May, 1988], pp. P 61-72

[P]

[Parekh, 2005]
Parekh, N. "*Spiral Model - A New Approach Towards Software Development*". [1/13/2005] Available on World Wide Web at: <http://www.buzzle.com/editorials/1-13-2005-64082.asp>

[S]

[Sharp et al, 1999]
Sharp, H; Finkelstein, A & Galal, G. *"Stakeholder Identification in the Requirements Engineering Process".* [11/12/99]

[Sommerville, 2004]
Sommerville, I
Software Engineering
Chapter 7