Software Requirements Specification
Restaurant
Menu & Ordering System

CSSE3002 The Software Process

• Report On •

Software Requirements Specification
Restaurant
Menu & Ordering System

• Submitted To •

Dr David Carrington
School of Information Technology & Electrical Engineering University of Queensland

• Prepared By •

Team Danger Tutorial One
Mr Tyson Henning (41213250)
Mr Daryl Keehn (40766357)
Mr Jonathan Thompson (40525460)
Mr Michael Wildermoth (40092560)

April 2008

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

REVISION HISTORY

Name: Jonathan Thompson
Date: 24/04/08
Reason: Document Draft
Version: 1.0

Name: Jonathan Thompson, Tyson Henning
Date: 27/04/08
Reason: Document Final Draft
Version: 1.1

Name: Tyson Henning
Date: 28/04/08
Reason: Document Final
Version: 1.2

1 INTRODUCTION

The following section provides an overview of the derived Software Requirements Specification (SRS) for the subject Restaurant Menu and Ordering System (RMOS). To begin with, the purpose of the document is presented and its intended audience outlined. Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant software will do and the relevant benefits associated with it. The nomenclature used throughout the SRS is also offered. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

1.1 Purpose

The purpose of this SRS is to outline both the functional and non-functional requirements of the subject RMOS. In addition to said requirements, the document also provides a detailed profile of the external interfaces, performance considerations and design constraints imposed on the subsequent implementation. It is the intention that the presented set of requirements possesses the following qualities; correctness, unambiguousness, completeness, consistency, verifiability, modifiability and traceability. Consequently, the document should act as a foundation for efficient and well-managed project completion and further serve as an accurate reference in the future. The primary audience of this SRS document will be the development team employed to implement the specified RMOS. It will not only provide an extensive capacity for project planning and progress assessment but it will further assist with developer/stakeholder interactions. The secondary document audience comprises the stakeholders of the project, that is, restaurateurs and associated staff. To this audience group, this SRS should convey and confirm the required functionality and represent a contractual agreement between the involved parties.

1.2 Scope

In current formal dining environments, some form of physical static menu is utilised to convey the available food and beverage choices to customers. Said menus are generally paper based and hence impose restrictions on the textual real estate available and the ability a restaurateur has to update them. This document specifies the requirements for a restaurant paper menu and ordering replacement strategy to alleviate the problems associated with the current archaic method. Three related concepts are encompassed by the general scope of the Restaurant Menu and Ordering System. The first pertains to the replacement of paper-based menus using an electronic format, the second relates to a complementary electronic strategy for the front of house handling of a customer's order and the third surrounds the process of transferring said electronic orders to the kitchen for preparation. It should be noted that while the suggested strategy incorporates the use of various hardware components, the primary focus of the presented SRS relates to the constituent software elements.

1.2.1 Overview

The Restaurant Menu and Ordering System is a software package to facilitate ordering within a traditional restaurant. The customer is able to view the menu, place orders, call the waiter, and organise the final bill through the surface computer interface built into their table. Waiters are able to initialise a table for customers, control table functions remotely to assist customers, confirm

orders, send orders to food preparation staff and finalise the customer's bill – all through their wireless tablet PC. The food staff, with their touch-display interfaces to the system, are able to view orders sent to the kitchen by waiters. During preparation, they are able to let the waiter know the status of each item, and can send notifications when items are completed, again through the touch-display.

The system contains full accountability and logging systems, and supports supervisor actions to account for exceptional circumstances, such as a meal being refunded or walked out on.

Customers are presented with an attractive and easy-to-use surface computer GUI with a drag-and-drop 'object' metaphor in their menus. Waiters are able to perform all actions that the table system normally handles via their tablet PCs, so in the event of a customer being unable to operate the surface computer, the waiter can handle orders traditionally while using retaining the accountability and logging functions of the system, and retaining the same channel of communication with food staff.

1.2.2 Benefits

Greater flexibility in menus, an increase in restaurant productivity and capacity for extensive business auditing are the primary benefits associated with the RMOS. Menu updates can be rolled out at any time with no extra labour from printing and distributing new menus, allowing for more dynamic pricing and content changes. With the underlying software system taking responsibility for a customer's order throughout its lifecycle, not only is accuracy ensured, but all actions are logged in a database for analysis and accountability of staff. This allows the company to audit and account staff time, materials and restaurant efficiency, as well as to review exceptional circumstances for future handling.

1.3 Nomenclature

This subsection presents definitions for the terms and acronyms used throughout this SRS as they relate to the subject RMOS.

Term	**Description**
Item	Single serving of food/beverage
Order	Comprises one or more items
Meal	Comprises one or more orders (associated with one customer)
Customer	Restaurant patron that orders/pays for a meal
Staff	General restaurant employee
Waiter	Staff member whose primary job is to take orders/serve meals to customers
Chef	Staff member whose primary job is to prepare items
Supervisor	Staff member whose primary job is to manage restaurant operations
Table	Comprises one or more seats at which customers sit and place orders from
Account	Comprises all the meals from a table
Payment	Comprises the total cost of zero or more meals and zero or more tips

Server	Backend computer that hosts the restaurant menu and ordering system
Surface Computer	Built into tables to provide customers with menu/ordering functionality
Tablet	Wireless mobile computer to provide staff with customer serving functionality
Display	Touch screen to provide a means for chefs to interact with the system
Register System	Point of sale terminal for handling bill payments
Bankcard	Customer debit/credit card
Menu	Surface computer representation of the available items and other options

Table 1.3.1 System Terminology

Acronym	**Description**
SRS	Software Requirement Specification
RMOS	Restaurant Menu and Ordering System
DBMS	Database Management System
LAN	Local Area Network
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IEEE 802.11	Wireless Local Area Network Standard
SSLv3	Secure Sockets Layer Version 3
WPA2-PSK	Wi-Fi Protected Access 2 with Pre-Shared Key

Table 1.3.2 System Acronyms

1.4 Structure

The structure of this Software Requirements Specification is as follows. Section 2 presents an overall description of the subject RMOS. Attention is given to putting the resultant software product into perspective and further outlining end-user characteristics, system constraints and assumptions. Section 3 is devoted to the explicit specification of software requirements both functional and non-functional in nature. The functional requirements listed have been demarcated according to the categories of system users. For completeness, Section 4 extends upon Section 3 through the inclusion of UML analysis models and diagrams. To begin with, the identified RMOS use cases are given. In supplement, activity diagrams for each use case are presented along with an overall class diagram and relevant statechart diagrams.

2 OVERALL DESCRIPTION

The following section presents an overall description of the subject RMOS. In particular, the product has been put into perspective through a detailed assessment of the system, user, hardware, software and communication interfaces, memory considerations, operational modes and site adaptation requirements. Further, characteristics of the system's end-users are discussed along with the identified system constraints and assumptions. To conclude the section, an apportioning of requirements has been outlined.

2.1 Product Perspective

The software described in this SRS is the software for a complete RMOS system. The system merges various hardware and software elements and further interfaces with external systems. Thus, while the software covers the majority of the system's functionality, it relies on a number of external interfaces for persistence and unhandled tasks, as well as physically interfacing with humans.

2.1.1 System interfaces

The RMOS interfaces with an existing payment system, including a cash register and software-accessible credit/EFTPOS system, in order to quickly and easily handle customer billing. The payment system should be operable such that it can return information to the RMOS system as to whether payment was successful or failed.

2.1.2 User interfaces

There are three separate user interfaces used by the RMOS software, each related to an interfaced physical hardware device (see Section 2.1.3). These three user interfaces are the Surface Computer UI, Tablet UI and Display UI.

Surface computer UI

The Surface Computer UI is the interface used by restaurant customers. This interface uses the surface computer paradigm - users interact with the system by dragging 'objects' around on the flat-screen touch-sensitive display. For the RMOS, users can manipulate objects such as items of food, dietary requirements, tips and menus on the surface of their table. Such objects can be moved into static objects such as meals and payments to perform various functions. In addition to this object manipulation paradigm, a limited system menu is necessary. Users will summon their restaurant menu, which is combined with a system/command menu, using an easy touch gesture, a double-tap on the touch surface, and dismiss it with a similar gesture or by tapping a close button GUI element. The GUI will take a small percentage of the table's screen, so the UI will be clear and uncluttered.

Tablet UI

The Tablet UI is designed to run on a small, wireless-enabled touch-screen tablet PC, to be used by waiters to accommodate customer needs. This UI will be designed for use with a stylus input into the touch-screen. Because the number of operations the UI needs to support is relatively limited, there will be no nested menu structure. The UI shall provide simple graphical interfaces, similar to a map, to allow the user to select tables/customers as the target of operations.

Display UI

The Display UI provides kitchen staff with simple functionality related to ordered items. The UI will display the list of items in large, easy-to-read text, sorted by time of submission with additional information (such as dietary requirements and the destination table) displayed in tabulated format. Input is provided by fingertips, as opposed to a stylus

2.1.3 Hardware interfaces

There are three external hardware devices used by the RMOS, each related to a user interface (see Section 2.1.2). These devices are the surface computers, the wireless tablets and the touch displays. All three devices must be physically robust and immune to liquid damage and stains. The devices (with the possible exception of displays) must also have good industrial design aesthetics, as they are to be used in place of normal restaurant tables and notepads and will be in direct contact with customers. The devices behave as 'terminals' in the sense that they never have a full system image, do not store data and are not used for the core logic of the system. However, they should be fully capable computers that can use textual data from the server along with local UI/interpretation code to display UI elements and take input. All order and transaction records should be stored on the server, not these computers. The performance of dumb terminals over an area the size of a restaurant is likely to be unacceptable. In all three cases, the hardware device takes information from the RMOS and processes the information to display. It also provides user input information to the RMOS.

2.1.4 Software interfaces

The RMOS will interface with a Database Management System (DBMS) that stores the information necessary for the RMOS to operate. The DBMS must be able to provide, on request and with low latency, data concerning the restaurant's menu, employees (and their passwords) and available dietary requirements. Additionally, it should take and archive data provided to it by the RMOS. This data will include records of all orders and transactions (system states and state changes) executed by the RMOS. The DBMS must store all data such that it can be used for accounting, as well as accountability.

2.1.5 Communications interfaces

The RMOS will interface with a Local Area Network (LAN) to maintain communication with all its devices. It should use a reliable-type IP protocol such as TCP/IP or reliable-UDP/IP for maximum compatibility and stability. All devices it will interface with should contain standard Ethernet compatible, software accessible LAN cards to maintain communication between the server and the surface computers, tablets, displays and the external payment system. Devices that are wireless should also use Ethernet compatible cards, using the IEEE 802.11b/g standard and having support for WPA2-PSK encryption. The use of IEEE 802.11n transmission standard hardware is also acceptable if all other local hardware is conformant to the same standard.

2.1.6 Memory

The memory usage of the RMOS will obviously have to be constrained by the devices it is intended to run on. In particular, the wireless tablets, as mobile devices, may have limited memory - this should be taken into account when writing the tablet software. Memory constraints upon the server,

surface computers and displays are not likely to be an issue as each will likely have at least a gigabyte of primary memory and hundreds of gigabytes or more of secondary memory.

2.1.7 Operations

The RMOS has only one mode of operation. However, because of the restaurant environment it is used in, it must be able to operate for long periods, without error. The server must be able to operate unattended indefinitely. It should not need physical interaction except for upgrades and failure of hardware elements. Backup and recovery should be handled by the DBMS and operating system, or external software running on a timed backup system. Interaction from the RMOS should not be required. Since stateful data should not be stored on any of the devices other than the server, keeping a system image on the server for each device may be a sufficient operational method to facilitate restoration should a device become corrupted.

2.1.8 Site adaptation requirements

Site configuration for the RMOS is expected to encompass the following steps:

- Install the server, surface computers and displays
- Acquire sufficient tablets for all staff that need to use them
- Network all devices, install operating systems, server software and DBMS
- Secure network, distribute initial passkeys
- Install RMOS software
- Configure server RMOS software

Some customisation of RMOS software elements may be required, including:

- Table layout maps
- GUI elements, especially for customer-facing UIs

2.2 User Characteristics

The end-users of the RMOS fall into three primary categories, unskilled, partly skilled and highly skilled.

Unskilled user

The users of the surface computers are walk-in customers and should therefore be assumed to have no relevant prior skills or education other than basic abilities to operate an automated system; no more complex than a parking meter or vending machine.

Partly skilled user

The users of the tablets and displays are waiters and chefs respectively and they should be able to use the system and further be able to train others with minimal training themselves. They must be able to explain all elements of the user interfaces except the server. Supervisors also fall into the same category, though they will have to learn other sections of the system (refunds etc); these should not be of notably greater complexity than the standard functions. This class of user would be expected to have a junior high-school certificate education or equivalent.

Highly skilled user

The initial installation and configuration of hardware and the constituent RMOS system components (especially the server) is guaranteed to require someone with notable computer experience, including extensive experience with network and operating systems to complete it. The software should not be needlessly complex, but it is still expected not to be entirely 'plug and play'. This class of user is expected to have a high-school certificate or equivalent, as well as extensive computer experience.

2.3 Constraints

The RMOS should be written in an object-oriented language with strong GUI links and a simple, accessible network API. The primary candidate tool chains are Java/Swing, C++/Qt and Python/Qt. The system must provide a capacity for parallel operation and system design should not introduce scalability issues with regard to the number of surface computers, tablets or displays connected at any one time. The end system should also allow for seamless recovery, without data loss, from individual device failure. There must be a strong audit chain with all system actions logged. While interfaces are covered in Section 2.1, it is worth noting that this system is likely to conform to what is available. With that in mind, the most adaptable and portable technologies should be used for the implementation. The system has criticality insofar as it is a live system. If the system is down, then customers must not notice, or notice that the system recovers quickly (seconds). The system must be reliable enough to run crash and glitch free more or less indefinitely, or facilitate error recovery strong enough such that glitches are never revealed to its end-users.

2.4 Assumptions

The SRS assumes that none of the constituent system components will be implemented as embedded applications. The implication is that the target hardware will provide a capacity for standalone program/application deployment and not require customised embedded firmware to be written. It is further assumed that tablet PCs of sufficient processing capability and battery life will be utilised. The surface computers employed by the system should facilitate being utilised/left on for extended periods (sufficient for daily use) and that they are programmable in the same fashion as x86 architecture computers. Finally, it is further assumed that the deployment environment is capable of supporting an IEEE 802.11 wireless network for system communication.

2.5 Apportioning of Requirements

This subsection pertains to both the functional and non-functional requirements omitted unintentionally from this SRS document.

3 REQUIREMENTS

The following section presents the complete set of functional and non-functional requirements identified for the subject RMOS. Functional requirements are listed first, according to their relationship to the overall system, customers, waiters, chefs and supervisors. The non-functional requirements that pertain to safety, security, the interface, human engineering, qualification, operation, maintenance and performance are subsequently presented. The functional requirements have been specified using a natural language description and as such, the reader is directed to Section 4 (UML Analysis Models) for further detail.

3.1 Functional Requirements

This subsection presents the identified functional requirements for the subject RMOS. Initially, general requirements that pertain to the whole system are given. Where possible, subsequent requirements have been demarcated based on their relevance to the users of the system, that is, customers, waiters, chefs and supervisors.

3.1.1 General

Table 3.1.1 presents the identified functional general requirements that directly relate to the entire subject RMOS.

Requirement	Description
G01	A server shall host the RMOS and provide system data processing and storage capability.
G02	A surface computer shall provide a customer with all customer system functionality.
G03	A tablet shall provide a waiter/supervisor with all waiter/supervisor system functionality (according to access control).
G04	A display shall provide a chef with all chef system functionality.
G05	All system functionality shall be accessible through touch sensitive surface computers, tablets and displays via simple touch gestures.
G06	A tablet shall be capable of interfacing with a register to facilitate the accurate processing of a payment.

Table 3.1.1 Functional General Requirements

3.1.2 Customer

Table 3.1.2 presents the identified functional customer requirements that directly relate to the customers of the subject RMOS.

Requirement	Description
C01	A customer shall be able to engage their menu by double tapping the activated surface computer in their table.
C02	A customer shall be able to dismiss their menu by double tapping its dismiss option.

C03	A customer shall be able to create an empty pending order through their engaged menu.
C04	A customer shall not be able to dismiss their engaged menu while there is a non-empty pending order associated with the engaged menu.
C05	A customer shall be able to navigate through the available items in their engaged menu.
C06	A customer shall be able to add an item to a pending order by dragging the item from the engaged menu onto the order.
C07	A customer shall be able to remove an item from a pending order by dragging the item off the order.
C08	A customer shall be able to add a special dietary requirement to an order by dragging the requirement from the engaged menu onto the order.
C09	A customer shall be able to add a wildcard special dietary requirement to an order in the case that their requirement is not represented by the system.
C10	A customer shall be able to place an order through their engaged menu if it is pending and not empty.
C11	A customer shall be able to cancel an order through their engaged menu if it is pending and not yet placed.
C12	A customer shall be able to call for waiter assistance through their engaged menu.
C13	A customer shall be able to engage bill mode to finalise payment through their engaged menu.
C14	A customer shall be able to disengage bill mode to cancel the billing process through their engaged menu.
C15	When in billing mode, a surface computer shall display a representation of a cash payment for the whole table.
C16	When in billing mode, a surface computer shall display a representation of a bankcard payment for each customer.
C17	When in billing mode, a surface computer shall display a representation of every meal ordered that may each be dragged into a payment.
C18	When in billing mode, a surface computer shall display a representation of tip denominations that may be dragged into a payment.
C19	A customer shall be able to drag a meal into a bankcard payment or a cash payment.
C20	A customer shall be able to drag a meal out of a bankcard payment or a cash payment.
C21	A customer shall be able to drag a tip denomination into a bankcard payment or a cash payment.
C22	A customer shall be able to drag a tip denomination out of a bankcard payment or a cash payment.

Table 3.1.2 Functional Customer Requirements

3.1.3 Waiter

Table 3.1.3 presents the identified functional waiter requirements that directly relate to the waiters (and supervisors) of the subject RMOS.

Requirement	Description
W01	A waiter shall be able to log into a tablet using their assigned username and password.
W02	A waiter shall be able to log out of a tablet.
W03	A waiter shall be able to activate a surface computer and consequently open its associated account through a tablet.
W04	A waiter shall be able to deactivate a surface computer and consequently close its associated account through a tablet.
W05	A waiter who activates a surface computer shall be assigned to the table that contains it.
W06	A waiter assigned to a table shall be alerted via their wireless tablet when: • An order is placed from that table • An item ordered by that table is rejected by the kitchen • An item ordered by that table is ready to be served • The table has requested waiter assistance
W07	A tablet shall allow a waiter to accept an order placed by a customer through a surface computer.
W08	A tablet shall allow a waiter to reject an order placed by a customer through a surface computer.
W09	A tablet shall allow a waiter to indicate the delivery of an item to its customer.
W10	A tablet shall allow a waiter to process a payment using cash.
W11	A tablet shall allow a waiter to process a payment using a bankcard.

Table 3.1.3 Functional Waiter Requirements

3.1.4 Chef

Table 3.1.4 presents the identified functional chef requirements that directly relate to the chefs (and supervisors) of the subject RMOS.

Requirement	Description
K01	A chef shall be able to accept a customer's order item through a display.
K02	A chef shall be able to reject a customer's order item through a display.
K03	A chef shall be able to indicate that a customer's order item is ready to be served through a display.

Table 3.1.4 Functional Chef Requirements

3.1.5 Supervisor

Table 3.1.5 presents the identified functional supervisor requirements that directly relate to the supervisors of the subject RMOS.

Requirement	Description
S01	A supervisor shall be able to do everything a waiter can.
S02	A supervisor shall be able to do everything a chef can.
S03	A supervisor shall be able to abort/purge a customer's meal from the active system with no expectation of payment.

S04	A supervisor shall be able to abort/purge a table's account/meals from the active system with no expectation of payment.
S05	A supervisor shall be able to issue a refund for one or more items to a customer.

Table 3.1.5 Functional Supervisor Requirements

3.2 Non-Functional Requirements

This subsection presents the identified non-functional requirements for the subject RMOS. The subcategories of non-functional requirements given are safety, security, interface, human engineering, qualification, operational and maintenance.

3.2.1 Safety

Table 3.2.1 presents the identified non-functional safety requirements that directly relate to the entire subject RMOS.

Requirement	Description
F01	The system shall log every state and state change of every surface computer, tablet and display to provision recovery from system failure.
F02	The system shall be capable of restoring itself to its previous state in the event of failure (e.g. a system crash or power loss).
F03	The system shall be able to display a menu at all times to facilitate manual order taking should the need arise.
F04	The system shall utilise periodic 30-second keep-alive messages between tablets and the server to monitor tablet operational status.
F05	The system shall flag tablets that fail to send timely keep-alive messages as non-operational and disassociate the assigned waiter from the tablet.

Table 3.2.1 Non-Functional Safety Requirements

3.2.2 Security

Table 3.2.2 presents the identified non-functional security requirements that directly relate to the entire subject RMOS.

Requirement	Description
Y01	Wireless communication throughout the system will be encrypted using SSLv3 at the application layer and WPA2-PSK at the data link layer.
Y02	The WPA2-PSK password used for wireless communication must have a bit-strength of at least 80 bits.
Y03	The WPA2-PSK password used for wireless communication must be changed every three months.
Y04	A waiter password used for tablet login must have a bit-strength of at least 64 bits.
Y05	A waiter password used for tablet login must be changed every three months.
Y06	A waiter shall only be able to log into one tablet at any given instance of time.
Y07	A waiter that attempts to log into a second tablet while already logged into

	another tablet shall be rejected and notified through both tablets.
Y08	The system shall provide two levels of access: • A supervisor level for unrestricted access to system functionality • A waiter level for access to waiter functionality
Y09	A surface computer shall not require a user to log in.
Y10	A tablet shall require a user to log in using a username and password.
Y11	A display shall not require a user to log in.

Table 3.2.2 Non-Functional Security Requirements

3.2.3 Human engineering

Table 3.2.3 presents the identified non-functional human engineering requirements that directly relate to the entire subject RMOS.

Requirement	Description
H01	Any element of the system will take no longer than 10-seconds to restart.
H02	A surface computer must not dismiss an engaged menu unless the customer requests it.

Table 3.2.3 Non-Functional Human Engineering Requirements

3.2.4 Performance requirements

Table 3.2.4 presents the identified non-functional performance requirements that directly relate to the entire subject RMOS.

Requirement	Description
P01	The server shall be capable of supporting no less than 200 concurrent connections from any combination of surface computers, tablets and displays.
P02	The server shall be capable of supporting an arbitrary number of surface computers, tablets and displays, that is, it shall provide no limit on how many devices are in the system.
P03	The server shall be capable of supporting an arbitrary number of active meals/orders, that is, no meals/orders shall be lost under any circumstances.
P04	The server shall be capable of supporting an arbitrary number of active customer payments, that is, no payments shall be lost under any circumstances.

Table 3.2.4 Non-Functional Performance Requirements

4 UML ANALYSIS MODELS

4.1 Use Cases

This subsection extends upon the functional requirements given in Section 3.1 through the presentation of detailed use cases. To facilitate an unambiguous and clear view of how the end-users interact with the subject RMOS, the actors (end-users) involved in the use cases, a use case diagram and detailed use case descriptions are provided. The use cases that find representation are Log In, Log Out, Activate Table, Deactivate Table, Accept Order, Deliver Item, Process Bankcard Payment, Process Cash Payment, Abort Meal, Abort Account, Issue Refund, Place Order, Call Waiter, Pay Bill, Accept/Reject Item and Indicate Item Ready.

4.1.1 Actors

There are four actors in the RMOS, waiter, chef, supervisor and customer. While the waiter, chef and customer actors are base specialisations, the supervisor class inherits from both the waiter and chef actors as a generalisation.

4.1.2 Use case diagram

Figure 4.1.1 presents a use case diagram for the subject RMOS. The key interactions between the end-users of the system and the system itself are depicted.

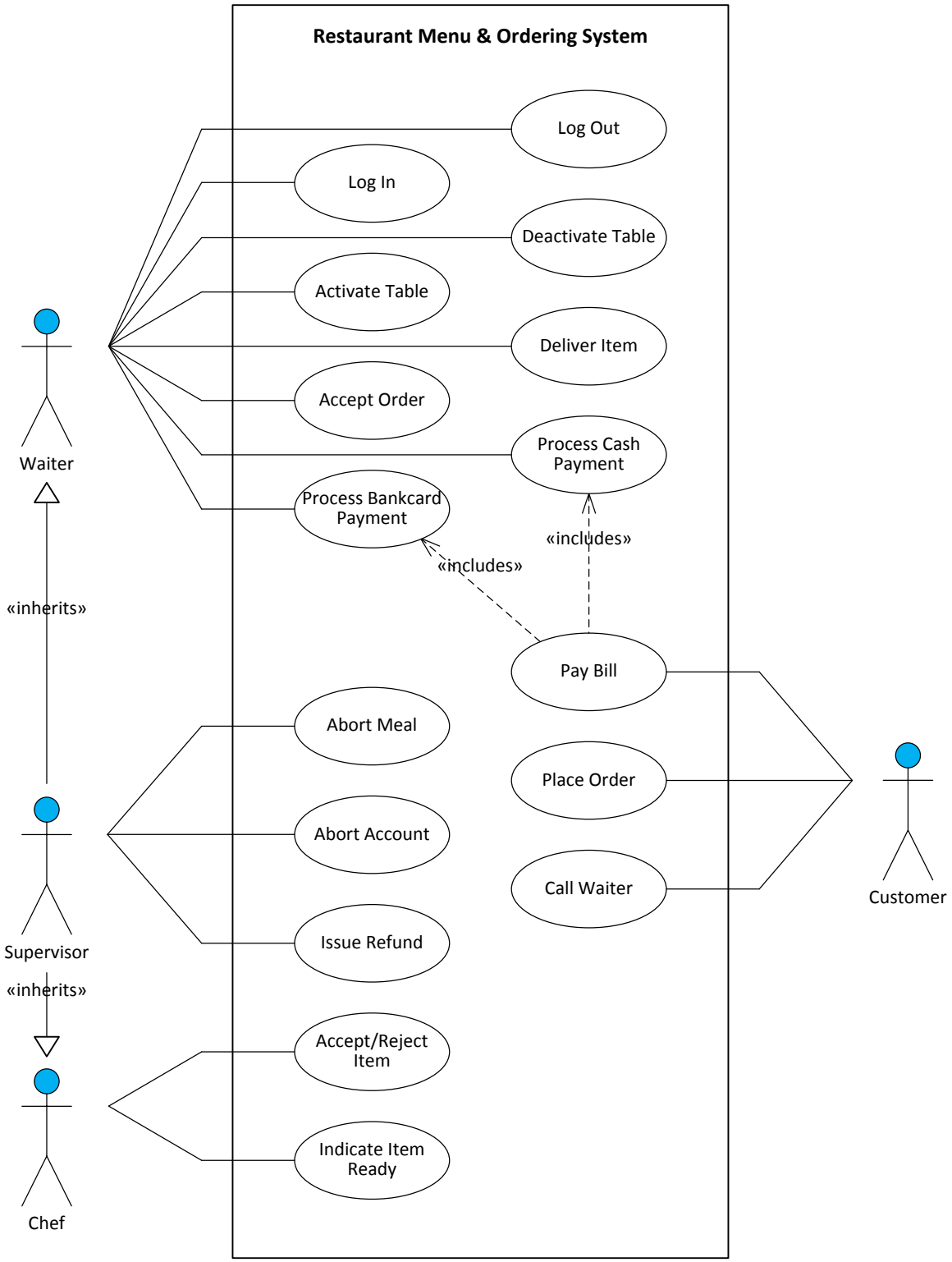

Figure 4.1.1 Restaurant Menu & Ordering System Use Case Diagram

4.1.3 Use case descriptions

Table 4.1.1 presents the Log In use case description to show the interaction between a waiter and a tablet when logging into the system.

Use Case	Log In
Primary Actor	Waiter
Goal In Context	Enable waiter access to the system through a tablet
Preconditions	The waiter has a valid username and password and is not already logged in
Trigger	The waiter requires access to the system to perform their job
Scenario	1) The waiter selects 'Log In' from the tablet menu 2) The tablet prompts the user for their username and password 3) The user enters their username and password 4) The tablet enables access to the system according to access control
Exceptions	The waiter enters an invalid username or password

Table 4.1.1 Log In Use Case Description

Table 4.1.2 presents the Log Out use case description to show the interaction between a waiter and a tablet when logging out of the system.

Use Case	Log Out
Primary Actor	Waiter
Goal In Context	Disable waiter access to the system through a tablet
Preconditions	The waiter is already logged in
Trigger	The waiter no longer requires access to the system to perform their job
Scenario	1) The waiter selects 'Log Out' from the tablet menu 2) The tablet disables access to the system
Exceptions	

Table 4.1.2 Log Out Use Case Description

Table 4.1.3 presents the Activate Table use case description to show the interaction between a waiter and a tablet when activating a table for use by a customer.

Use Case	Activate Table
Primary Actor	Waiter
Goal In Context	Activate the surface computer in a table to enable customer functionality
Preconditions	The waiter is already logged in
Trigger	A new group of customers have been seated at the table
Scenario	1) The waiter selects 'Activate Table' from the tablet menu 2) The waiter selects a free deactivated table from those available 3) The waiter selects the seats to be occupied by customers 4) The surface computer in the table is activated and an account is created 5) An empty order is automatically created for each customer at the table 6) The waiter is assigned to take care of the table's account
Exceptions	There are no free deactivated tables available

Table 4.1.3 Activate Table Use Case Description

Table 4.1.4 presents the Deactivate Table use case description to show the interaction between a waiter and a tablet when deactivating a table once its customers leave.

Use Case	Deactivate Table
Primary Actor	Waiter
Goal In Context	Deactivate the surface computer in a table once its customers have left
Preconditions	The account (every meal) associated with the table has been finalised/paid
Trigger	A table of customers finalise their account and leave the restaurant
Scenario	1) The waiter selects 'Deactivate Table' from the tablet menu 2) The waiter selects the table to be deactivated 3) The table is deactivated and its account details are archived 4) The account is deleted from the active system
Exceptions	

Table 4.1.4 Deactivate Table Use Case Description

Table 4.1.5 presents the Accept Order use case description to show the interaction between a waiter and a tablet when accepting a new order placed by a customer.

Use Case	Accept Order
Primary Actor	Waiter
Goal In Context	Accept an order that has been placed by a customer
Preconditions	A customer has placed an order
Trigger	The waiter chooses to serve the customer
Scenario	1) The waiter selects 'Take Order' from the tablet menu 2) The waiter selects a table with pending orders from those assigned to them 3) The waiter selects a pending order from the selected table 4) The waiter confirms the order and selects ‘Accept’ from the tablet menu 5) The items in the order are sent to the kitchen for preparation 6) The order is added to the customer’s meal (and the table’s account)
Exceptions	The waiter rejects the order by selecting 'Reject' from the tablet menu

Table 4.1.5 Accept Order Use Case Description

Table 4.1.6 presents the Deliver Item use case description to show the interaction between a waiter and a tablet when delivering an order item to a customer.

Use Case	Deliver Item
Primary Actor	Waiter
Goal In Context	Deliver a ready item to its customer
Preconditions	An item for a table assigned to the waiter is prepared and ready
Trigger	The system sends an alert to the waiter notifying them the item is ready
Scenario	1) The waiter reads the alert, noting the item and its table/seat number 2) The waiter delivers the item to the customer who ordered it 3) The waiter marks the item as delivered through the tablet
Exceptions	

Table 4.1.6 Deliver Item Use Case Description

Table 4.1.7 presents the Process Bankcard Payment use case description to show the interaction between a waiter and a tablet when processing the bankcard payment of a customer bill.

Use Case	Process Bankcard Payment
Primary Actor	Waiter
Goal In Context	Charge a customer for their meals taking a bankcard payment
Preconditions	Meals have been assigned to the customer's bankcard payment bill
Trigger	The waiter chooses to finalise a customer's bill
Scenario	1) The waiter selects 'Bill Table' from the tablet menu 2) The waiter selects a table with outstanding bills from which the customer's have asked to finalise their account 3) The waiter selects a customer to process payment for, from a list of customers with meals assigned to their bankcard payment bill 4) The waiter swipes the customer's bankcard through the tablet 5) The system interfaces with the register system to process the payment 6) The meals are marked as paid and disappear from the surface computer
Exceptions	The transaction is rejected by the register system

Table 4.1.7 Process Bankcard Payment Use Case Description

Table 4.1.8 presents the Process Cash Payment use case description to show the interaction between a waiter and a tablet when processing the cash payment of a customer bill.

Use Case	Process Cash Payment
Primary Actor	Waiter
Goal In Context	Charge a table of customers for their meals taking a collective cash payment
Preconditions	Meals have been assigned to the table's cash payment bill
Trigger	The waiter chooses to finalise a customer's bill
Scenario	1) The waiter selects 'Bill Table' from the tablet menu 2) The waiter selects a table with outstanding bills from which the customer's have asked to finalise their account 3) The waiter selects 'Cash Payment' from the tablet menu 4) The waiter takes cash from customers and moves to the payment system 5) The tablet automatically connects with the payment system when in proximity 6) The waiter processes the payment using the payment system 7) The meals are marked as paid and disappear from the surface computer 8) The waiter returns to the customers with their change
Exceptions	Insufficient cash is provided by the customers

Table 4.1.8 Process Cash Payment Use Case Description

Table 4.1.9 presents the Pay Bill use case description to show the interaction between a customer and a surface computer when paying for their meals.

Use Case	Pay Bill
Primary Actor	Customer
Goal In Context	A customer decides to pay for their meals when they are ready to leave
Preconditions	Every item ordered by the customer has been delivered or cancelled
Trigger	The customer asks to finalise their bill
Scenario	1) The customer double-taps the table to bring up a menu 2) The customer selects 'Ask For Bill' from the menu 3) The surface computer enters billing mode and displays a representation of each customer's bankcard payment, a table cash payment and each customer's meal 4) The customer drags/assigns their meals into the appropriate payment 5) The customer drags/assigns any tip denominations into the appropriate payment 6) The customer finalises the assignment process and an alert is sent to the waiter assigned to the table 7) The waiter processes the payment 8) The customer double-taps the dismiss menu option on the table to close the menu
Exceptions	The customer cancels billing mode

Table 4.1.9 Pay Bills Use Case Description

Table 4.1.10 presents the Place Order use case description to show the interaction between a customer and a surface computer when placing an order.

Use Case	Place Order
Primary Actor	Customer
Goal In Context	Place an order for menu items from the restaurant
Preconditions	The customer has been seated at an activated table
Trigger	The customer wants to order one or more menu items
Scenario	1) The customer double-taps the table to bring up a menu 2) The customer selects 'Order' from the menu 3) The surface computer in the table displays an empty order and the available food/beverage/dietary requirement items 4) The customer navigates through the available items and adds the ones they want by dragging them from the menu onto their order (menu items can be removes by dragging them out of the order) 5) The customer selects 'Place Order' from the menu 6) The waiter assigned to the table's account is notified that the order has been placed 7) The customer double-taps the dismiss menu option on the table to close the menu
Exceptions	The customer cancels the order by selecting 'Cancel' from the menu

Table 4.1.10 Place Order Use Case Description

Table 4.1.11 presents the Call Waiter use case description to show the interaction between a customer and a surface computer when calling for waiter assistance.

Use Case	Call Waiter
Primary Actor	Customer
Goal In Context	Get assistance from the waiter assigned to the customer's table
Preconditions	The customer has been seated at an activated table
Trigger	The customer decides they require waiter assistance
Scenario	1) The customer double-taps the table to bring up a menu 2) The customer selects 'Call Waiter' from the menu 3) An alert is sent to the waiter assigned to the table
Exceptions	

Table 4.1.11 Call Waiter Use Case Description

Table 4.1.12 presents the Abort Meal use case description to show the interaction between a supervisor and a tablet when aborting a customer's meal.

Use Case	Abort Meal
Primary Actor	Supervisor
Goal In Context	Abort a customer's meal with no expectation of payment
Preconditions	The meal exists and is associated with the customer
Trigger	Exceptional circumstances arise and the supervisor must abort the meal
Scenario	1) The supervisor selects 'Abort Meal' from the tablet menu 2) The supervisor selects a table from those that have active meals 3) The supervisor selects an active meal from the selected table 1) The details of the meal, the abortion and the supervisor are archived 4) The active meal is deleted from the active system
Exceptions	

Table 4.1.12 Abort Meal Use Case Description

Table 4.1.13 presents the Abort Account use case description to show the interaction between a supervisor and a tablet when aborting a table's account.

Use Case	Abort Account
Primary Actor	Supervisor
Goal In Context	Abort a table's account with no expectation of payment
Preconditions	The account exists and is associated with the table
Trigger	Exceptional circumstances arise and the supervisor must abort the account
Scenario	2) The supervisor selects 'Abort Account' from the tablet menu 3) The supervisor selects a table from those that have active accounts 4) The details of the account, the abortion and the supervisor are archived 5) The active account is deleted from the active system
Exceptions	

Table 4.1.13 Abort Account Use Case Description

Table 4.1.14 presents the Issue Refund use case description to show the interaction between a supervisor and a tablet when aborting a table's account.

Use Case	Issue Refund
Primary Actor	Supervisor
Goal In Context	Refund a customer's payment for a meal
Preconditions	The customer paid for a meal in the restaurant
Trigger	Exceptional circumstances arise and the supervisor must refund the payment
Scenario	1) The supervisor confirms that refund should be given 2) The supervisor selects 'Issue Refund' from the tablet menu 3) The supervisor selects the meal from a list for the given date 4) The supervisor selects 'Confirm Refund' from the tablet menu 5) The details of the meal, the refund and the supervisor are archived 6) The supervisor returns the cost of the meal to the customer
Exceptions	

Table 4.1.14 Issue Refund Use Case Description

Table 4.1.15 presents the Accept/Reject Item use case description to show the interaction between a chef and a display when accepting or rejecting an item requested by a customer.

Use Case	Accept/Reject Item
Primary Actor	Chef
Goal In Context	Notify waiters if an item cannot be prepared
Preconditions	The item has not already been accepted or rejected
Trigger	Chef proceeds to action the pending items on a kitchen display
Scenario	1) The chef checks if the pending item can be fulfilled and rejects or accepts it 2) An accepted pending item is added to a list of items to prepare 3) A rejected pending item is removed from the kitchen display and an alert is sent to the waiter assigned to the pending item's account
Exceptions	

Table 4.1.15 Accept/Reject Item Use Case Description

Table 4.1.16 presents the Indicate Item Ready use case description to show the interaction between a chef and a display when indicating that an accepted item is ready to be delivered to its customer.

Use Case	Indicate Item Ready
Primary Actor	Chef
Goal In Context	Alert the appropriate waiter that an item is ready to be delivered
Preconditions	The item has been accepted by a chef
Trigger	Item preparation completes
Scenario	1) The chef selects the item from the list of items to prepare 2) The chef selects 'Ready' from the display menu 3) An alert is sent to the waiter assigned to the prepared item's account
Exceptions	

Table 4.1.16 Indicate Item Ready Use Case Description

4.2 Activity Diagrams

Figure 4.2.1 presents the Log In activity diagram to provide a graphical representation of a waiter logging into the system.

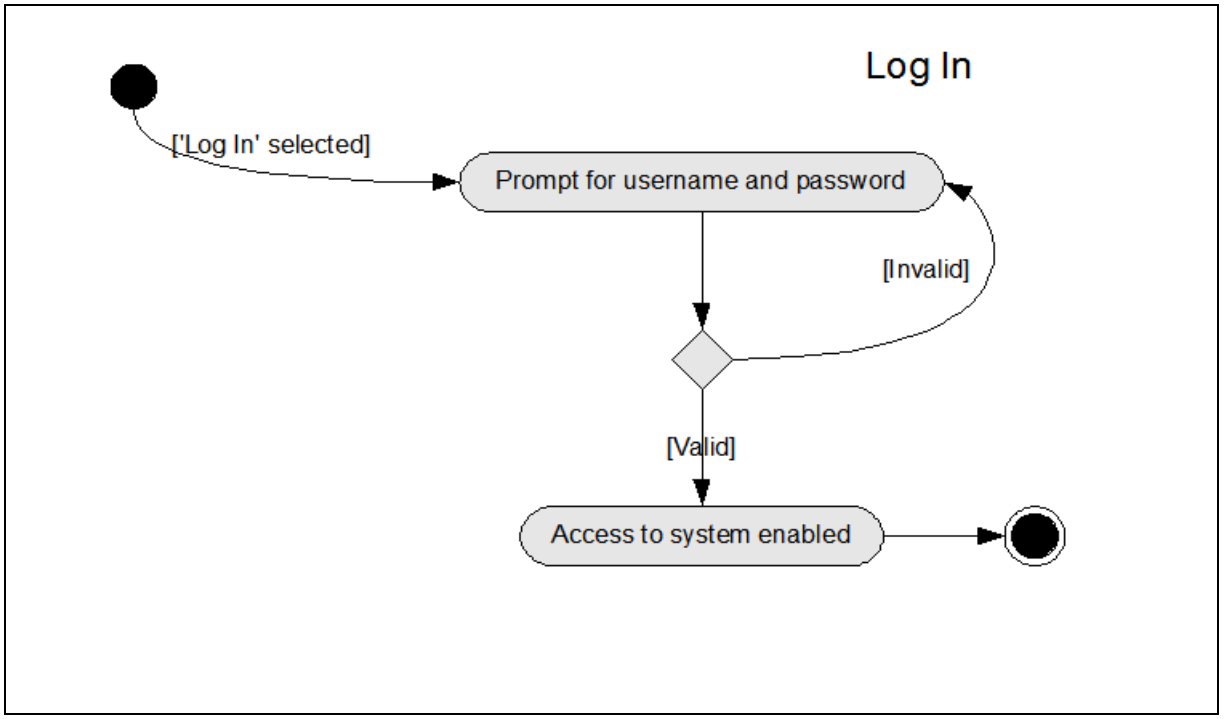

Figure 4.2.1 Log In Activity Diagram

Figure 4.2.2 presents the Log Out activity diagram to provide a graphical representation of a waiter logging out of the system.

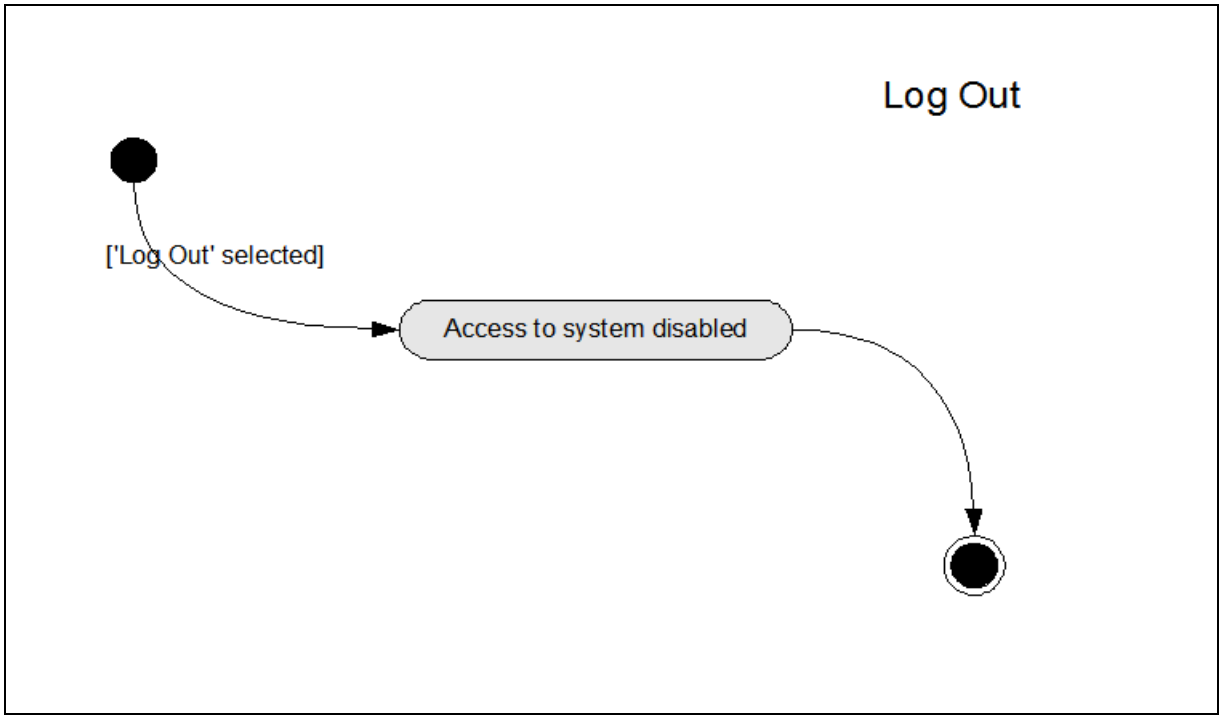

Figure 4.2.2 Log Out Activity Diagram

Figure 4.2.3 presents the Activate Table activity diagram to provide a graphical representation of a waiter activating the surface computer in a table.

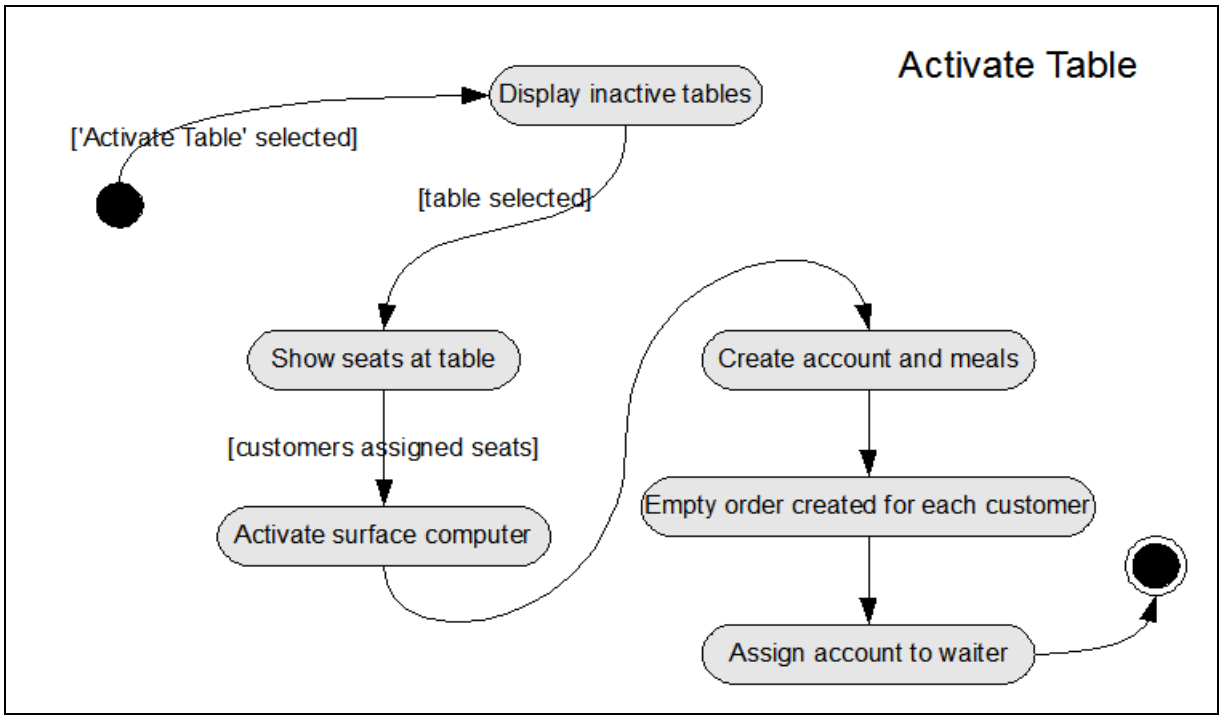

Figure 4.2.3 Activate Table Activity Diagram

Figure 4.2.4 presents the Deactivate Table activity diagram to provide a graphical representation of a waiter deactivating the surface computer in a table.

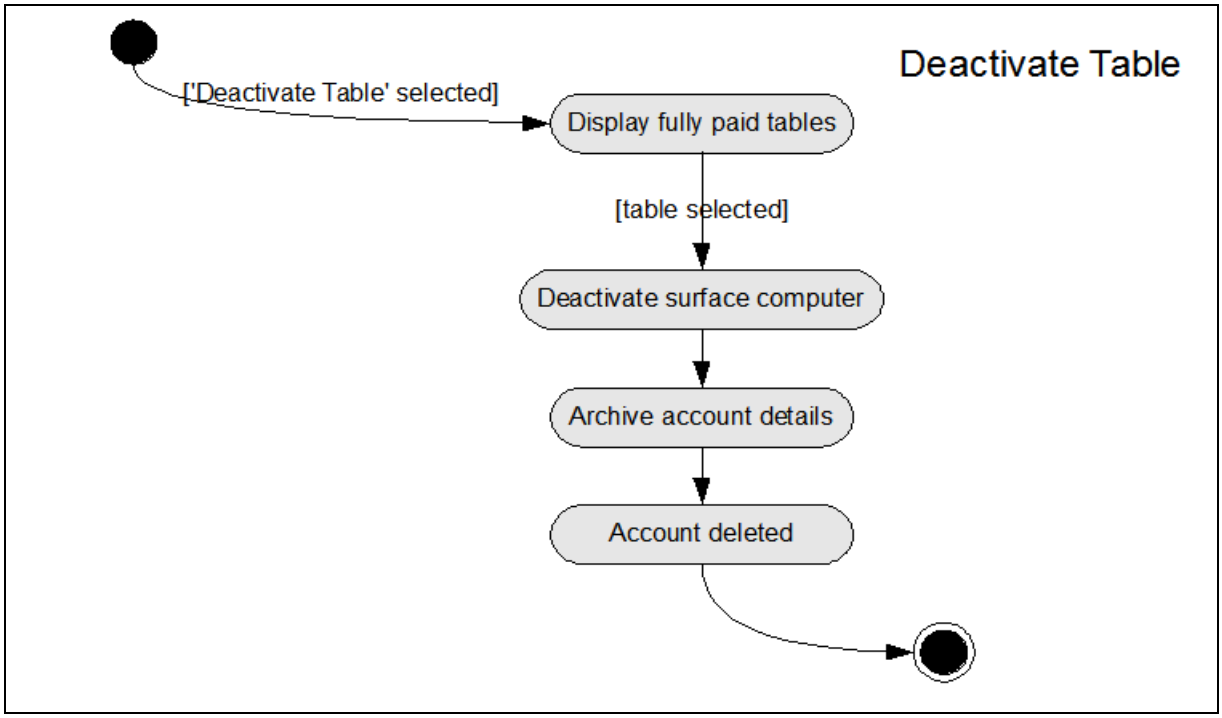

Figure 4.2.4 Deactivate Table Activity Diagram

Figure 4.2.5 presents the Accept Order activity diagram to provide a graphical representation of a waiter accepting a customer's order.

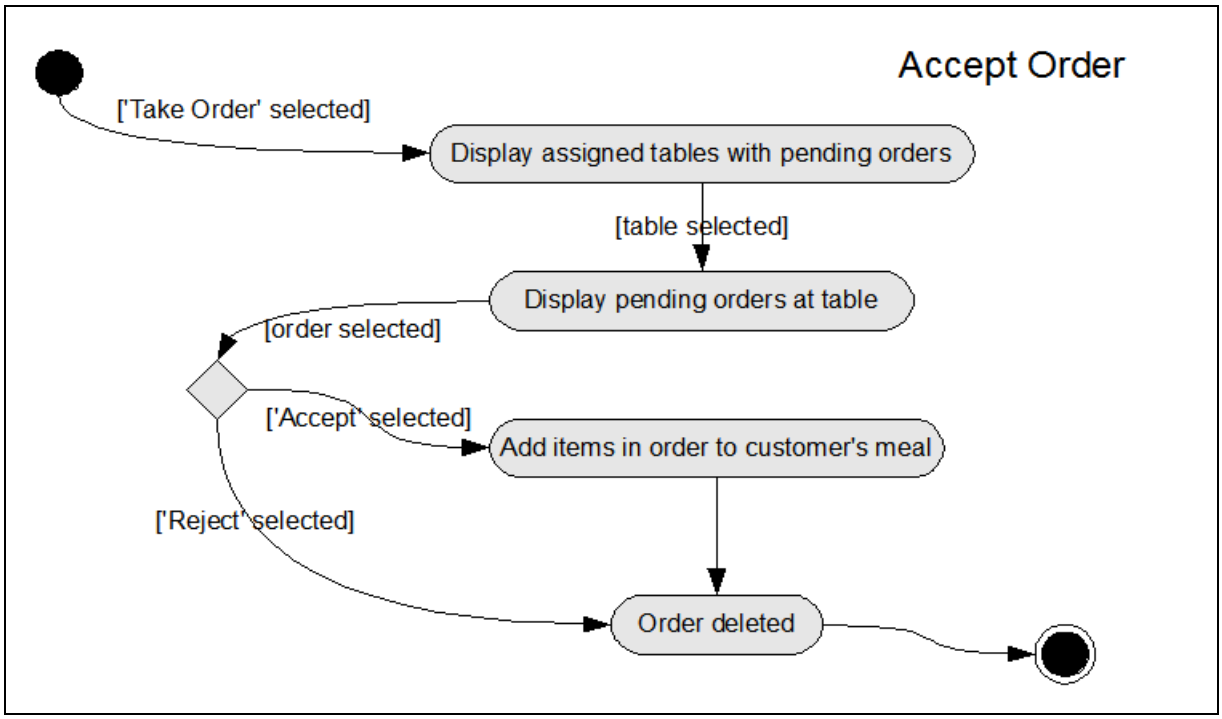

Figure 4.2.5 Accept Order Activity Diagram

Figure 4.2.6 presents the Deliver Item activity diagram to provide a graphical representation of a waiter delivering an item to a customer.

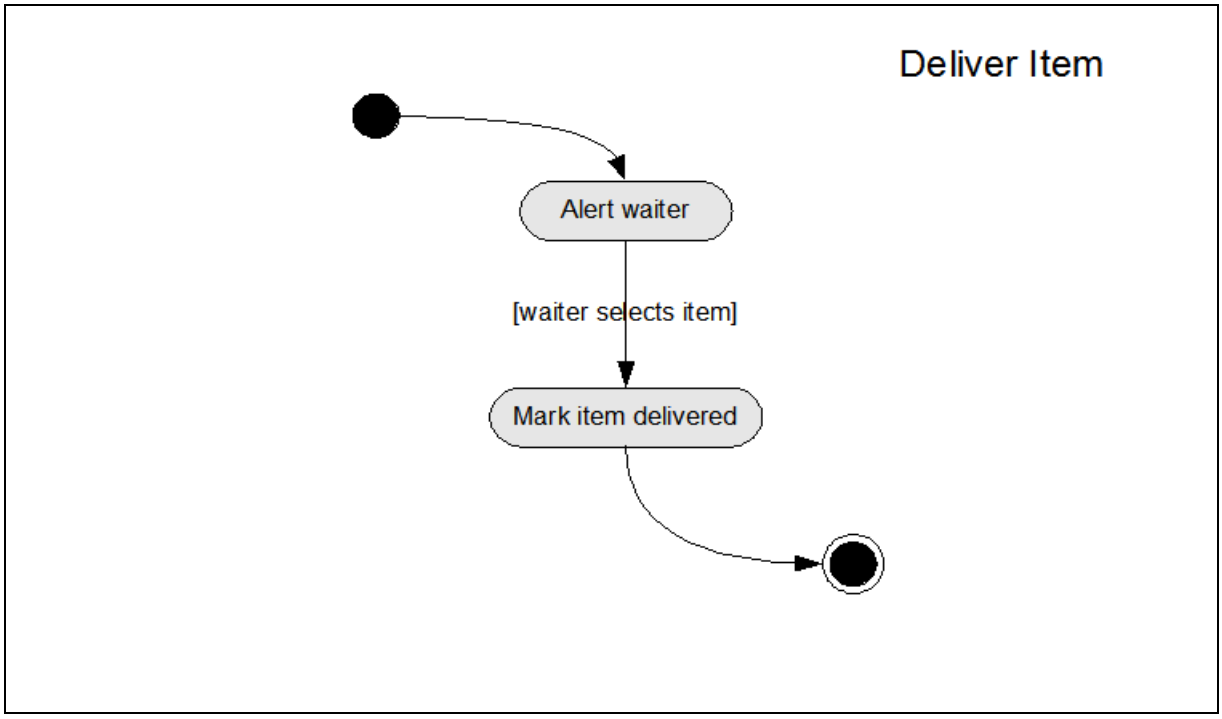

Figure 4.2.6 Deliver Item Activity Diagram

Figure 4.2.7 presents the Process Bankcard Payment activity diagram to provide a graphical representation of a waiter processing a bankcard payment.

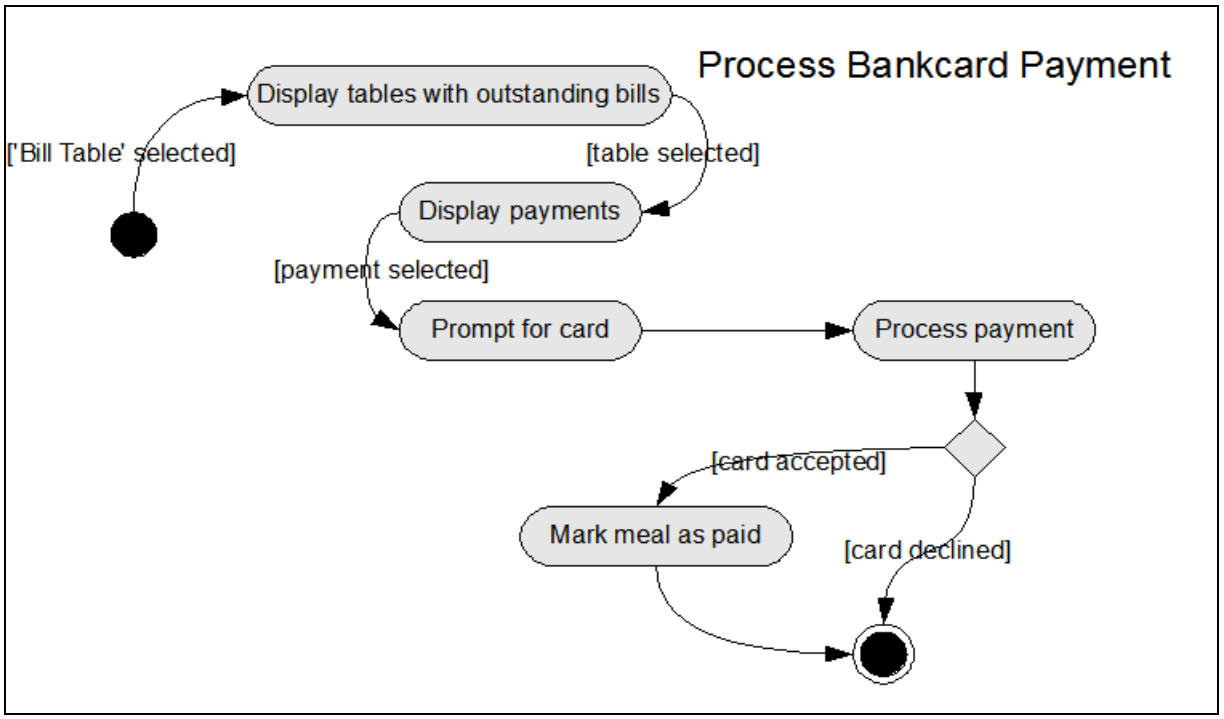

Figure 4.2.7 Process Bankcard Payment Activity Diagram

Figure 4.2.8 presents the Process Cash Payment activity diagram to provide a graphical representation of a waiter processing a cash payment.

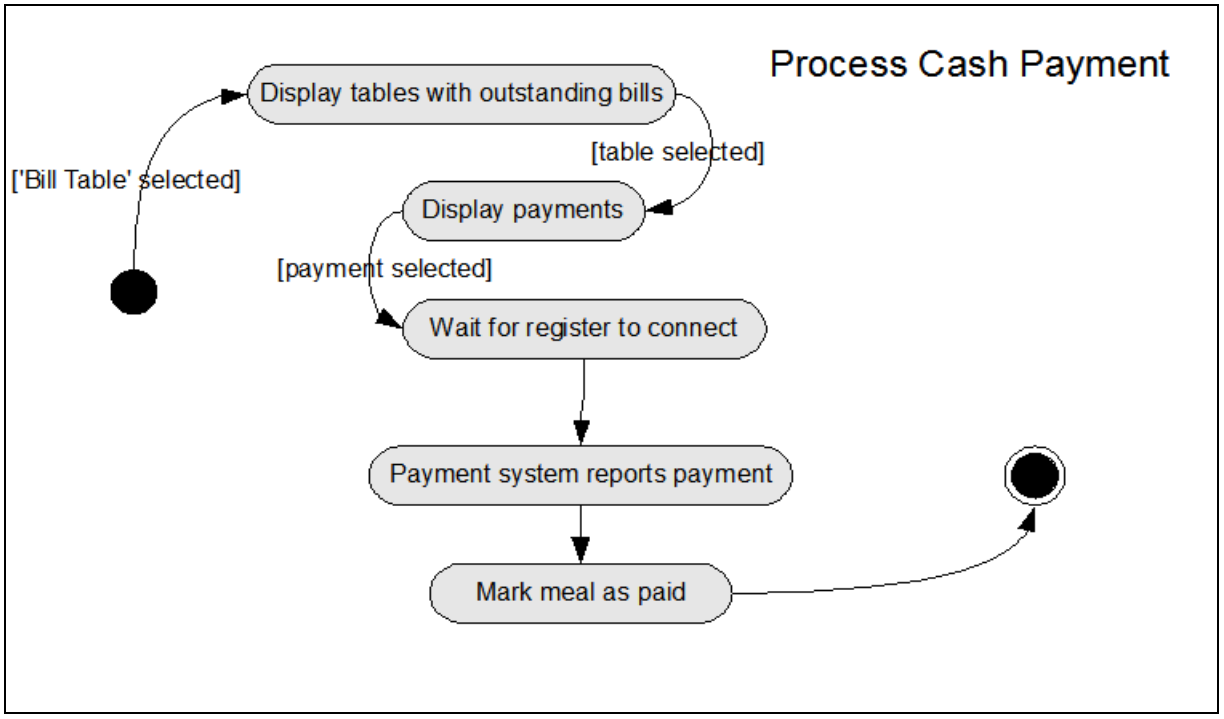

Figure 4.2.8 Process Cash Payment Activity Diagram

Figure 4.2.9 presents the Pay Bill activity diagram to provide a graphical representation of a customer paying for their meal.

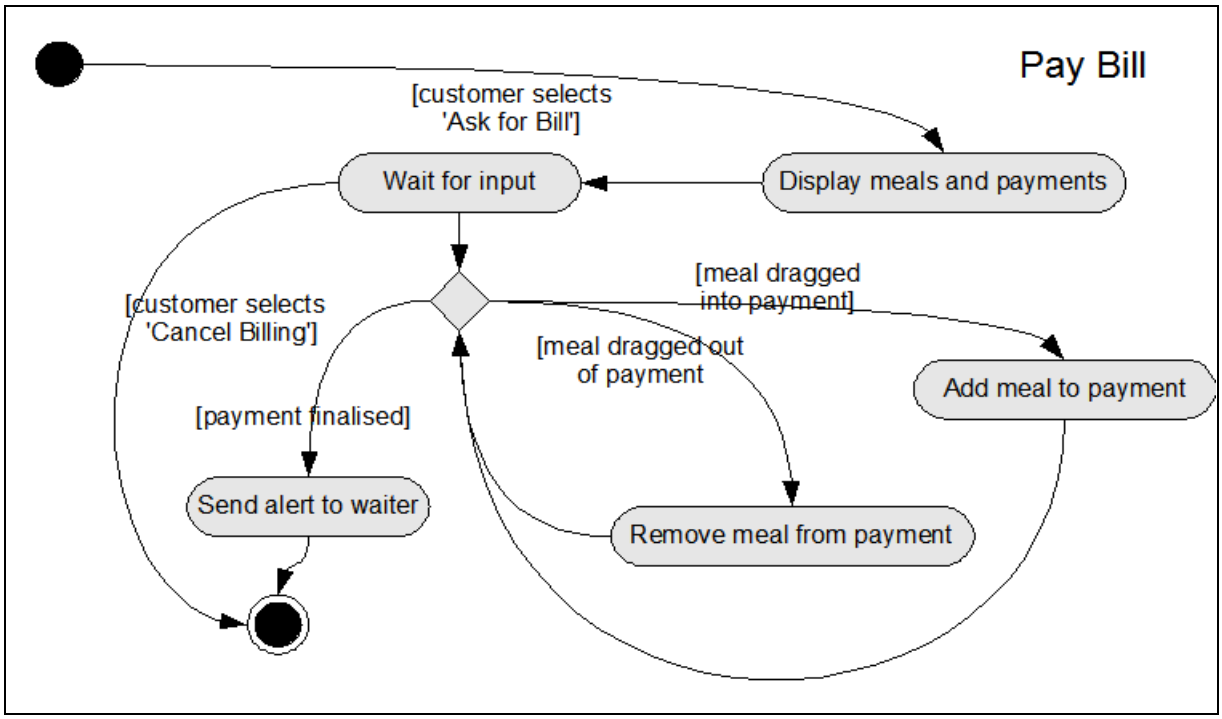

Figure 4.2.9 Pay Bill Activity Diagram

Figure 4.2.10 presents the Place Order activity diagram to provide a graphical representation of a customer placing an order.

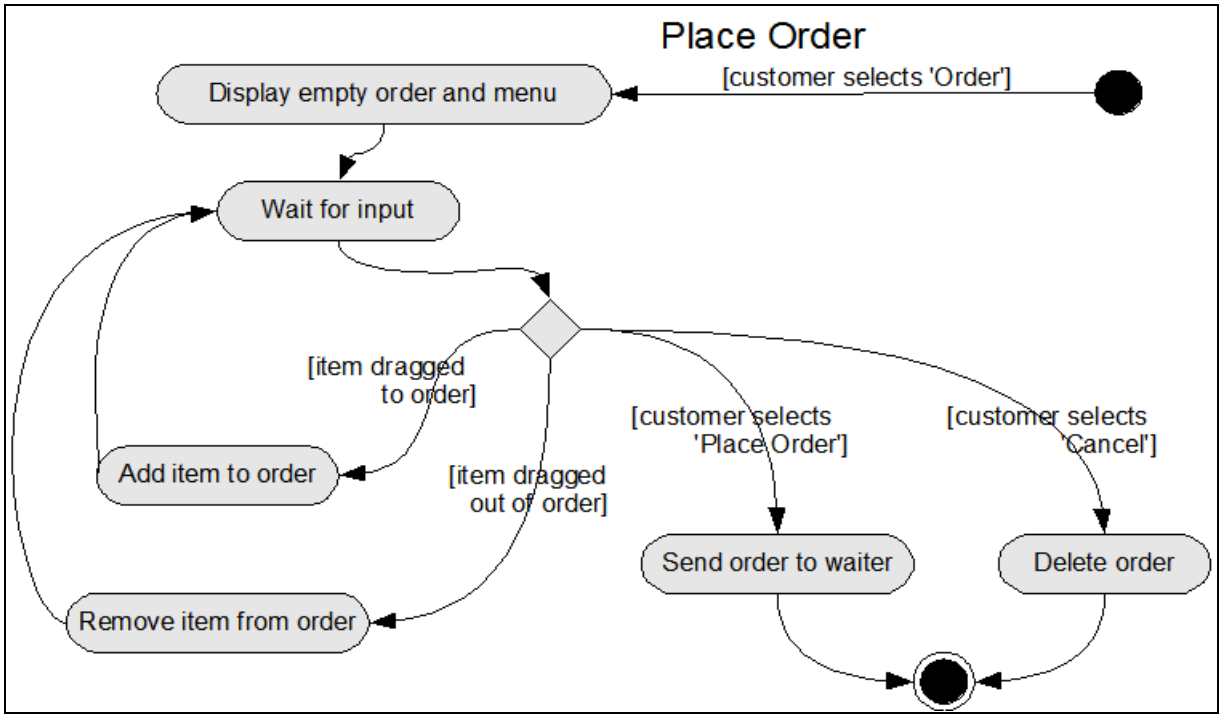

Figure 4.2.10 Place Order Activity Diagram

Figure 4.2.11 presents the Call Waiter activity diagram to provide a graphical representation of a customer calling for waiter assistance.

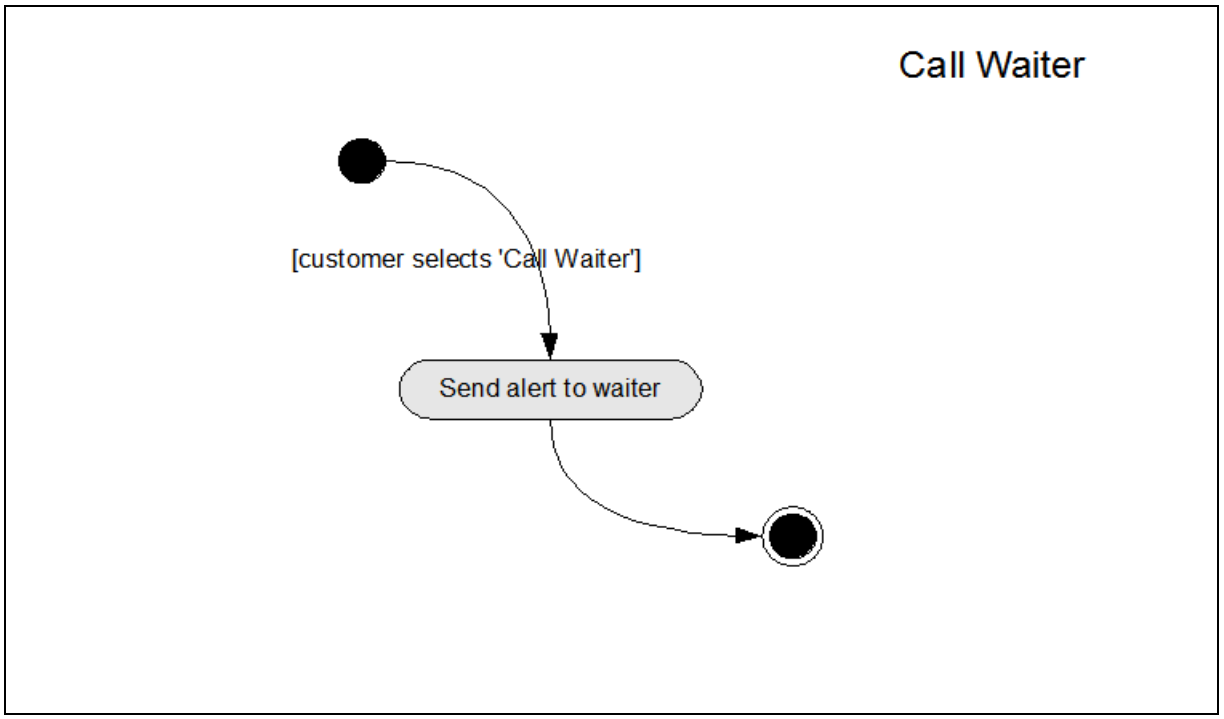

Figure 4.2.11 Call Waiter Activity Diagram

Figure 4.2.12 presents the Abort Meal activity diagram to provide a graphical representation of a supervisor aborting a customer's meal.

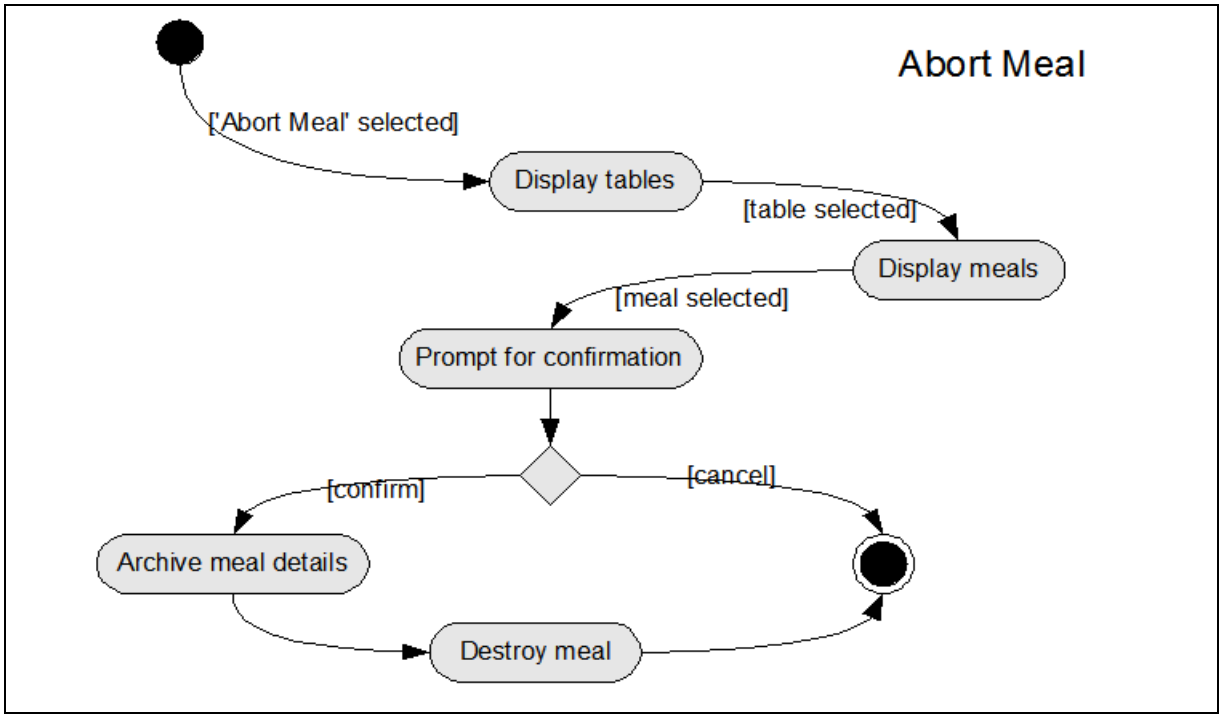

Figure 4.2.12 Abort Meal Activity Diagram

Figure 4.2.13 presents the Abort Account activity diagram to provide a graphical representation of a supervisor aborting a table's account.

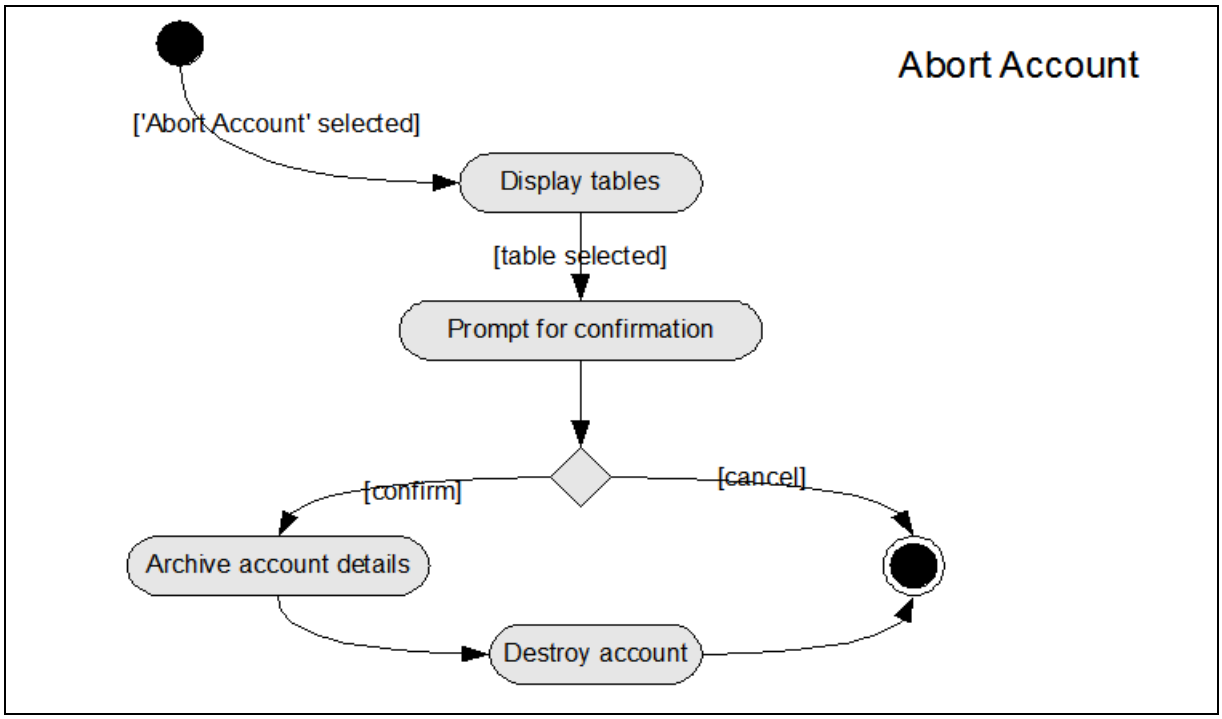

Figure 4.2.13 Abort Account Activity Diagram

Figure 4.2.14 presents the Issue Refund activity diagram to provide a graphical representation of a supervisor issuing a customer with a refund.

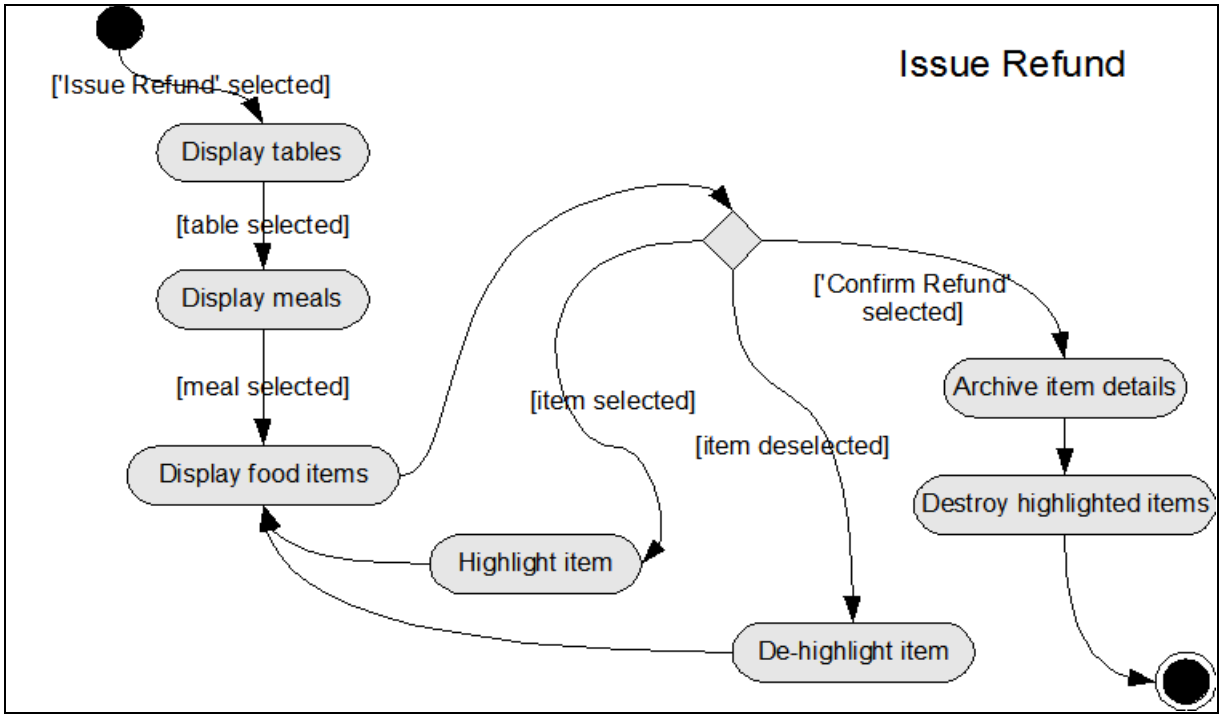

Figure 4.2.14 Issue Refund Activity Diagram

Figure 4.2.15 presents the Accept/Reject Item activity diagram to provide a graphical representation of a chef accepting or rejecting an item from a customer's order.

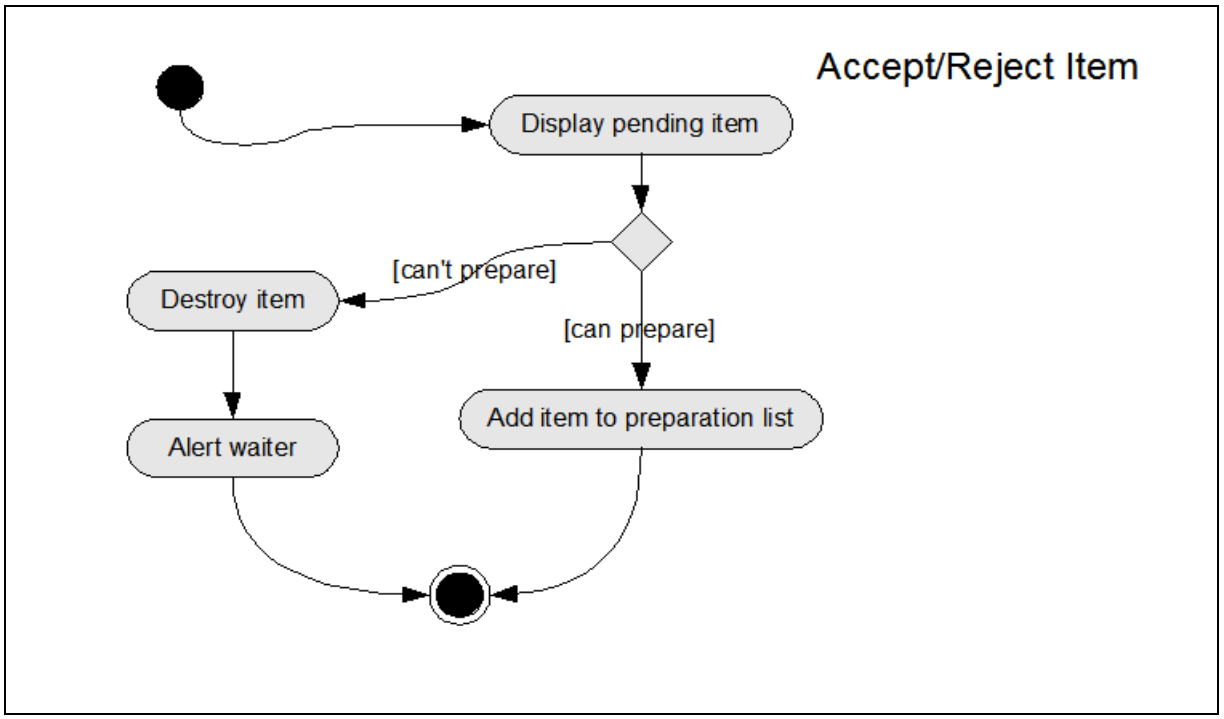

Figure 4.2.15 Accept/Reject Item Activity Diagram

Figure 4.2.16 presents the Indicate Item Ready diagram to provide a graphical representation of a chef indicating that a customer's item is ready for delivery.

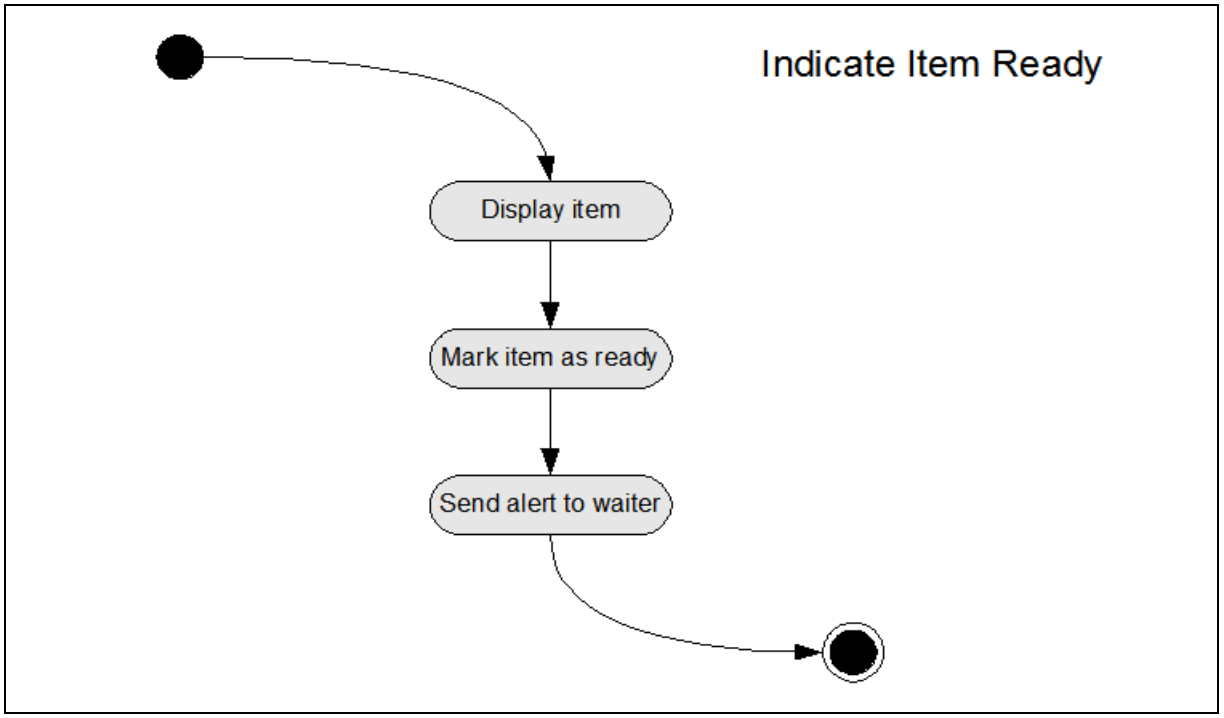

Figure 4.2.16 Indicate Item Ready Activity Diagram

4.3 Class Diagram

4.3.1 Class descriptions

The following subsection presents descriptions for the classes identified for the subject RMOS.

Item

This class represents an item of food or beverage from the restaurant's menu. It exists as a part of either a single order or a single meal, but not both at the same time (an order gets decomposed into the constituent Items and they are added to a meal). An Item that is part of an order may contain a dietary requirement. An Item contains the menu item's name, price, description, status and an image or model to depict the Item on a surface computer or display.

DietaryRequirement

This class represents a customer's dietary requirement (e.g. vegetarian, celiac). It exists as part of either a single Order or a single Item, but not both at the same time. A DietaryRequirement contains both the name and a description of the particular requirement it represents.

Order

This class represents a collection of Items and DietaryRequirements. It is part of a single Meal and maintains information about whether it has been placed or approved. An Order can be in either the placed, cancelled or the approved state. An Order will typically be deleted shortly after it has been approved and its Items added to the customer's Meal, tying the Items in the Order to an Account.

Meal

This class represents a collection of all items ordered by a single customer. It is part of a single Account and maintains the seat number related to the Meal. The total price of a Meal is also maintained.

Account

This class represents the associated Meals and Payments of a Table of customers. It consists of one or more Meals and one or more Payments. An account is related to one Table and is associated with one Tablet.

Table

This class represents a physical table at which a customer may be seated and its integrated surface computer. If the represented table is activated, an Account will be associated with the Table. It contains information about its current mode and its unique identification number. Further, it may be woken (activated), put to sleep (deactivated), placed in ordering mode or placed in billing mode.

Tablet

This class represents a wireless tablet used by staff; it essentially represents a waiter/supervisor logged into the tablet and consequently the system. The class contains information about who is

logged into the Tablet. It may also be associated with any number of Alerts that can be sent to it by the system.

Alert

This class represents a message sent to a Tablet to alert the waiter/supervisor of an event related to one of the Tables they have been assigned. It maintains a category (e.g. 'item rejected', 'item ready') and a description. An Alert belongs to exactly one Tablet.

Payment

This class represents a payment to be made to the restaurant. It may contain any number of Meals and TipDenominations and is related to exactly one Account. A Payment maintains its total value and the seat number of the paying customer.

BankcardPayment

This class represents an extension of the Payment class for customer payments that are to be paid using a bankcard.

CashPayment

This class represents an extension of the Payment class for customer payments that are to be paid using cash.

TipDenomination

This class represents a tip denomination to be given to a waiter. A TipDenomination belongs to a single Payment and contains information about the denomination value.

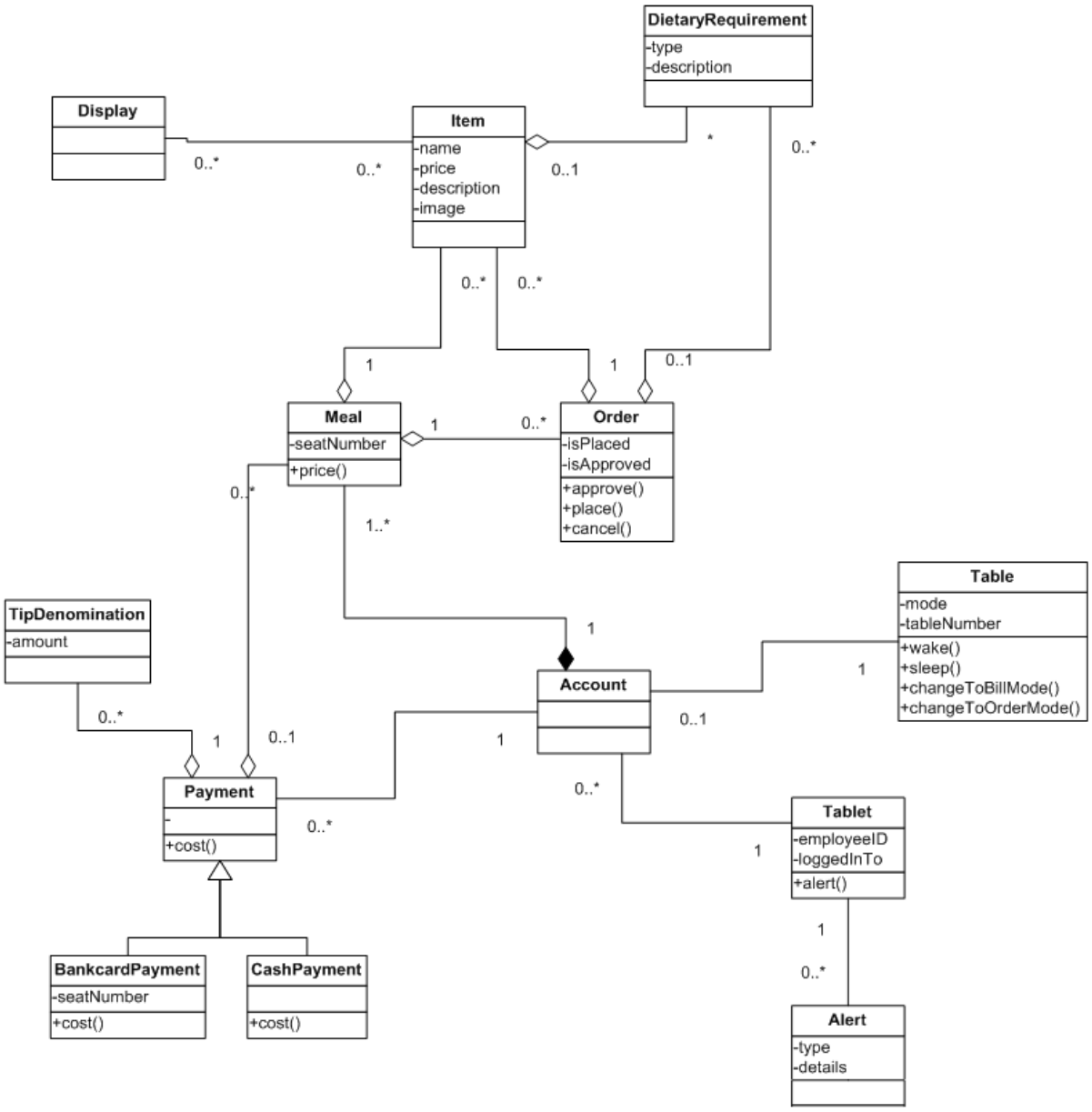

Figure 4.3.1 Restaurant Menu & Ordering System Class Diagram

4.4 Statechart Diagrams

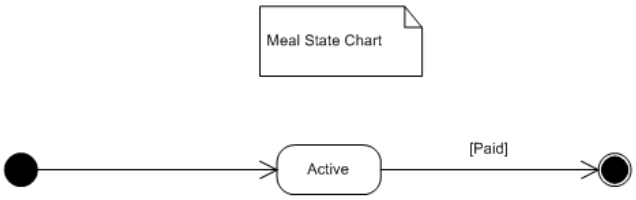

Figure 4.4.1 Meal Class Statechart Diagram

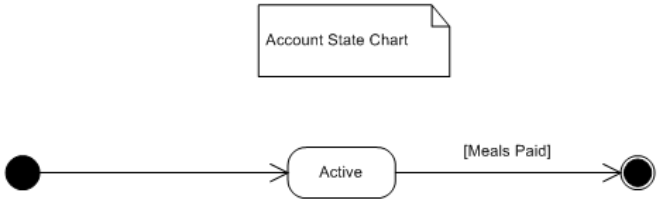

Figure 4.4.2 Account Class Statechart Diagram

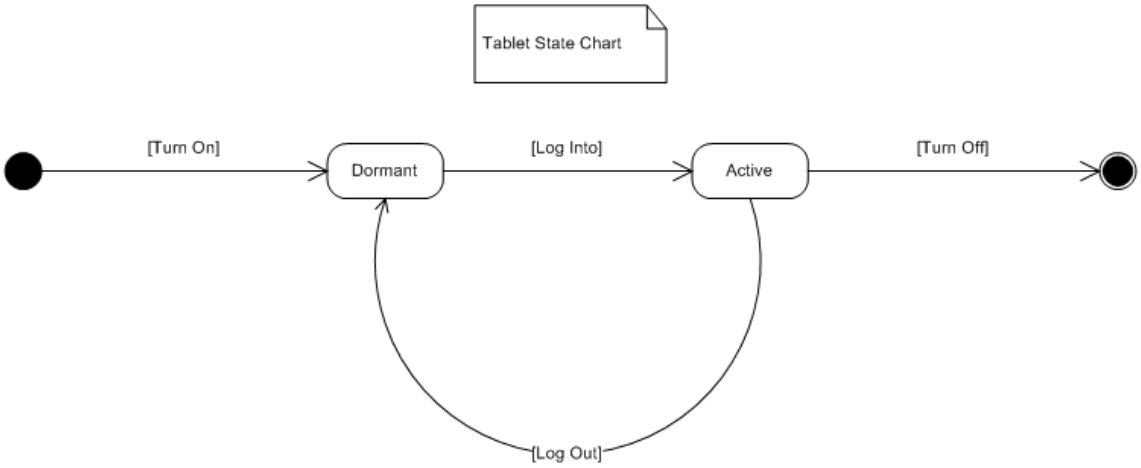

Figure 4.4.3 Tablet Class Statechart Diagram

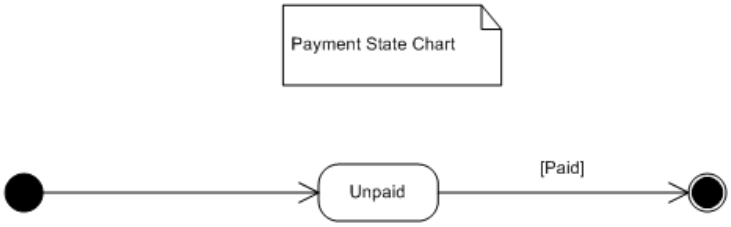

Figure 4.4.4 Payment Class Statechart Diagram

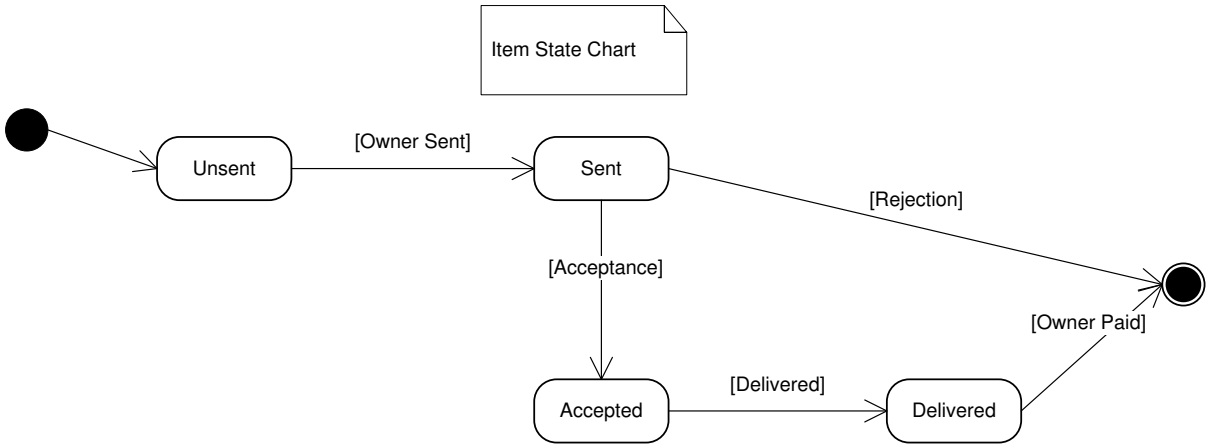

Figure 4.4.5 Item Class Statechart Diagram

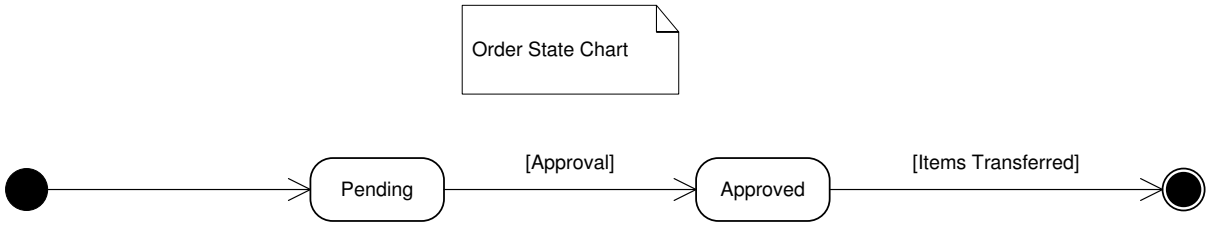

Figure 4.4.6 Order Class Statechart Diagram

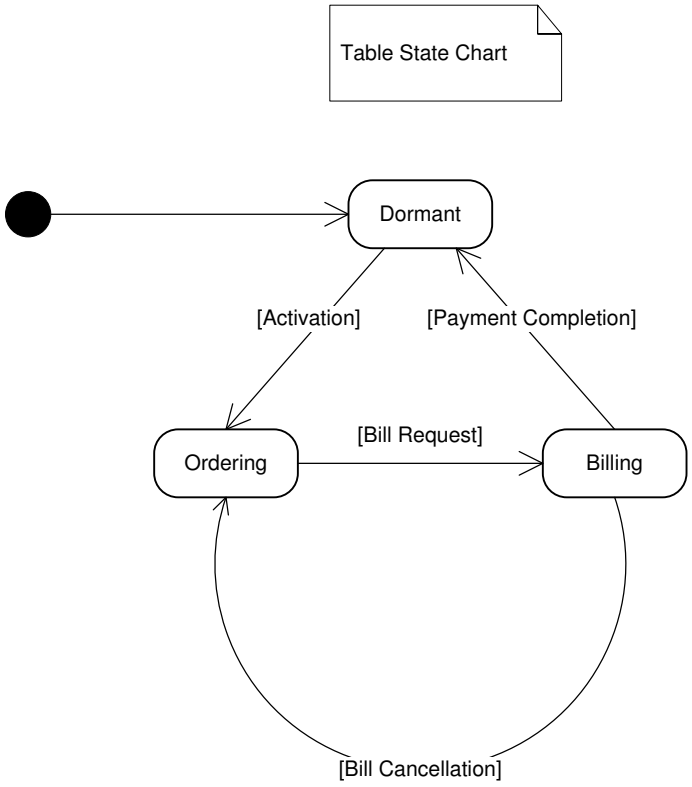

Figure 4.4.7 Table Class Statechart Diagram

APPENDIX A – REFERENCES

C. Larman, *APPLYING UML AND PATTERNS An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3rd ed., Massachusetts: Pearson Education, 2005.

D. Carrington, CSSE3002 Course Notes, School of ITEE University of Queensland, 2008

IEEE Recommended Practice for Software Requirements Specifications, IEEE Standard 830, 1998.

APPENDIX B – PROCESS EVALUATION

In order to improve an engineering process, it must be recorded and analysed. This appendix seeks to collate time data based on both the planned schedule and the actual time spent during the process. Furthermore, it discusses the problems encountered during the process of developing the subject SRS and suggests remedies to prevent these problems from arising in future iterations.

Development Process

The team's original plan for developing the Software Requirements Specification (SRS) was straightforward. Each team member was to review lecture notes and related textbook chapters to ensure familiarity with the process (background research). They were to also research existing computerised restaurant menu/ordering systems to gather an idea of what already currently exists and what the requirements of the system would be (requirements elicitation).
The list of requirements gathered from this were then to be refined in a meeting with all team members, which would result in a consistent and complete set of requirements defining the system. This would include a use case model, use case descriptions, sequence diagrams, activity diagrams, UML class analysis diagram and state chart diagrams.

From this set of requirements, the final SRS was to be developed written as a formal SRS as per the IEEE template. The requirements gathered in previous phases were collated and each section was to be written in turn making sure that later sections were consistent with previously written sections and then reviewed by all team members to check for any errors or inconsistencies. The completed SRS once agreed to by all team members was then to be submitted.

Problems Encountered

The most serious problem encountered was that of finding the time in which everyone could meet outside of university, work hours and other commitments. With all of the team members having busy timetables, as well as other commitments, we were limited in the times that we were able to meet and therefore had to spend a lot of time working via email.
Second was the inaccurate estimation of time required to complete the writing of the SRS. The time required for requirements elicitation was underestimated, delaying the schedule and leaving less time available for diagram production and report writing. This miscalculation occurred because the team had no previous metrics by which to estimate the time needed to complete the requirements elicitation phase.

Suggested Process Changes

The primary change to the SRS development process that this report suggests pertains to time estimation. It is not ideal to plan the development of a SRS with no previous metrics on which to base the time estimations. It is recommended that a longer window of time be allocated to requirements elicitation, diagram production and the writing of the final report.

Time Estimation/Expenditure Comparison

The differences between the submitted SRS plan and the actual process undertaken have been summarised below.

a) Preparation of SRS Plan
b) Proof reading of the SRS Plan
c) Research
d) 'Requirements Elicitation' meeting
e) Complete analysis/use case modelling
f) Complete sequence, activity, UML class analysis, and statechart diagrams
g) Review requirements
h) Complete first draft
i) Proof read first draft
j) Complete final report draft

Tables B 1.1-3 presented below highlight the aforementioned time discrepancies. Early completion dates and shorter time expenditures are highlighted in green; late completion dates and extra time expenditures are highlighted in red.

Plan Writing				
Task	Predicted Completion	Actual Completion	Predicted Time Expended	Actual Time Expended
SRS plan first draft	08/04/08	08/04/08	1.5 hours (meeting)	1.5 hours (meeting)
SRS plan final draft	09/04/08	09/04/08	1.5 hours (MW)	1 hour (MW)
SRS plan proof read	10/04/08	11/04/08	0.5 hours (all)	0.5 hours (TH) 0 hours (DK) 0.5 hours (JT) 0 hours (MW)
SRS plan edited and submitted	12/04/08	13/04/08	0.5 hours (MW)	0.5 hours (MW)

Table B 1.1 Plan Writing Time Data Listing

Preliminary Documentation				
Task	Predicted Completion	Actual Completion	Predicted Time Expended	Actual Time Expended
Background Research	14/04/08	14/04/08	6 hours (all)	2 hours (TH) 1 hour (DK) 3 hours (JT) 2 hours (MW)
Requirements Elicitation	15/04/08	22/04/08	5 hours (meeting)	6 hours (meeting)
Complete analysis/use case modelling	17/04/08	20/04/08	4 hours (all)	1 hour (TH) 0 hours (DK) 4 hours (JT) 12 hours (MW)
Complete other diagrams	21/04/08	26/04/08	10 hours (all)	2 hours (TH) 0 hours (DK) 2 hours (JT) 8 hours (MW)
Review requirements	22/04/08	22/04/08	3 hours (meeting)	3 hours (TH) 3 hours (DK) 5 hours (JT) 5 hours (MW)

Table B 1.2 Preliminary Documentation Time Data Listing

Final Documentation				
Task	Predicted Completion	Actual Completion	Predicted Time Expended	Actual Time Expended
Report first draft	25/04/08	24/04/08	4 hours (all)	6 hours (TH) 4 hours (DK) 15 hours (JT) 15 hours (MW)
Report proof read	26/04/08	26/04/08	2 hours (all)	10 hours (TH) 0 hours (DK) 4 hours (JT) 0 hours (MW)
Report final draft	27/04/08	28/04/08	2 hours (MW)	10 hours (TH) 0 hours (DK) 12 hours (JT) 0 hours (MW)
Report submitted	28/04/08	28/04/08	0.5 hours (MW)	0.5 hour (TH) 0.25 hours (JT)

Table B 1.3 Final Documentation Time Data Listing

APPENDIX C – PROCESS PLAN

Requirements Engineering Process

Research:

Review lecture notes and related textbook chapters to ensure familiarity with the requirements engineering process and the essential components of a Software Requirements Specification. This includes gaining an understanding of the different categories of requirements, the various methods and diagrams used to describe them and an understanding of why requirements engineering is important. Additionally, research existing computerised restaurant menu/ordering systems and document this research for inclusion in the report.

Requirements Elicitation:

Because there is not an actual client for this assignment, the elicitation phase will consist of the team brainstorming possible functionality from the "client's" point of view. The elicited requirements will be based primarily on the research previously conducted into existing computerised restaurant systems.

Requirements Analysis:

During the analysis phase, team members will take the derived requirements and transform them into a constant, thorough and complete set of requirements. This will include formulating a use case model, use case descriptions, sequence diagrams, activity diagrams, a UML analysis class diagram and statechart diagrams.

Requirements Specification/Documentation:

The outcome of this final phase will be the production of the final report, including the full Software Requirements Specification and an appendix concerning process reflection. Details of the specification will be finalised and all requirements and diagrams will be collated and organised into a single coherent Software Requirements Specification document.

Team Members

Tyson Henning (TH) (41213250)
Jonathan Thompson (JT) (40525460)
Daryl Keehn (DK) (40766357)
Michael Wildermoth (MW) (40092560)

Team Schedule

Week 6 08/04 - Prepare plan first draft
Week 6 10/04 - Review and proofread plan first draft
Week 6 12/04 - Amend review changes and submit inspection plan
Week 7 14/04 - Complete research
Week 7 15/04 - 'Requirements Elicitation' brainstorming meeting
Week 7 17/04 - Complete analysis/use case modelling
Week 8 21/04 - Complete sequence, activity, UML analysis class, and statechart diagrams
Week 8 22/04 - Review requirements

Week 8 25/04 - Prepare inspection report draft
Week 8 26/04 - Review and proofread report first draft
Week 8 27/04 - Complete report final draft and submit

Time Estimations

1.5 hours (meeting) - Prepare plan first draft
0.5 hours (all) - Review and proof read plan first draft
0.5 hours (MW) - Amend review changes and submit inspection plan
6.0 hours (all) - Complete research
5.0 hours (meeting) - 'Requirements Elicitation' brainstorming meeting
4.0 hours (all) - Complete analysis/use case modelling
10.0 hours (all) - Complete sequence, activity, UML analysis class, and statechart diagrams
3.0 hours (meeting) - Review requirements
4.0 hours (all) - Complete report first draft
2.0 hours (all) - Review and proofread report first draft
2.0 hours (MW) - Complete report final draft and submit