**MINISTRY OF EDUCATION AND TRAINING**



ORDER SUPPORTING SYSTEM

F-MENU

**CAPSTONE PROJECT**

Specialty: Software Engineering

Project members:

|  |  |  |
| --- | --- | --- |
| Nguyen Hoang Duc Hieu | - | HieuNHD01435 |
| Hoang Xuan Thanh | - | ThanhHX00719 |
| Pham Van Lam | - | LamPV01203 |
| Luong Trong Nghia | - | NghiaLT01146 |

Supervisor: Mr. Phan Truong Lam

Hanoi – 2013

RECORD OF CHANGE

\*A – Added M – Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Author | A/M/D | Change Description | Version |
| 09/03/2013 | HieuNHD | A | Newly created | 1.0 |
| 20/03/2013 | HieuNHD | M | Added report 2 | 1.1 |
| 03/04/2013 | HieuNHD | M | Added report 3, 4 | 1.2 |
| 15/04/2013 | HieuNHD | D | Delete “Context” of report 1 | 1.3 |
| 15/04/2013 | HieuNHD | M | Added report 5 | 1.4 |
| 16/04/2013 | HieuNHD | M | Added report 6 | 1.5 |
| 16/04/2013 | HieuNHD | M | Spellcheck | 1.6 |

TABLE OF CONTENTS

[A. INTRODUCTION 6](#_Toc353874495)

[I. Problem Abstract 6](#_Toc353874496)

[II. Existing Products 7](#_Toc353874497)

[1. AdeptPros GeniusMenu: 7](#_Toc353874498)

[2. iPOS: 8](#_Toc353874499)

[3. The limitations of the existing products 8](#_Toc353874500)

[III. Our proposal to the problems 9](#_Toc353874501)

[IV. The Project team 10](#_Toc353874502)

[1. General information of the Project 10](#_Toc353874503)

[2. Members and roles 10](#_Toc353874504)

[3. Planned Timeline 11](#_Toc353874505)

[4. Advantages on work 11](#_Toc353874506)

[5. Disadvantages 11](#_Toc353874507)

[B. PROJECT MANAGEMENT PLAN 12](#_Toc353874508)

[I. Project Overview 12](#_Toc353874509)

[1. Project Name 12](#_Toc353874510)

[2. Project Objectives 12](#_Toc353874511)

[3. Project Scope 12](#_Toc353874512)

[4. Development Environment 13](#_Toc353874513)

[5. Tools and Techniques 13](#_Toc353874514)

[II. Analysis and selection of tools 14](#_Toc353874515)

[1. Tools & hardware requirements 14](#_Toc353874516)

[2. IDE requirements 16](#_Toc353874517)

[III. Project Organization 17](#_Toc353874518)

[1. Software Process Model 17](#_Toc353874519)

[2. Roles and responsibilities 18](#_Toc353874520)

[IV. Project Lifecycle 19](#_Toc353874521)

[3. Project stage summary 19](#_Toc353874522)

[4. Project stage and milestone 19](#_Toc353874523)

[V. Project Management Plan 20](#_Toc353874524)

[1. Tasks 20](#_Toc353874525)

[2. Task Sheet: Assignment and Timetable 23](#_Toc353874526)

[C. SOFTWARE REQUIREMENTS SPECIFICATION 24](#_Toc353874527)

[I. Introduction 24](#_Toc353874528)

[1. Objectives 24](#_Toc353874529)

[2. Scope 24](#_Toc353874530)

[3. Acronym and Definitions 24](#_Toc353874531)

[II. Overall Description 25](#_Toc353874532)

[4. Product Functions 25](#_Toc353874533)

[5. Interfaces 25](#_Toc353874534)

[III. Requirements 26](#_Toc353874535)

[1. Customer Requirements 26](#_Toc353874536)

[2. External Interface Requirements 27](#_Toc353874537)

[3. Communication Protocol 29](#_Toc353874538)

[4. Functional Requirements 30](#_Toc353874539)

[5. Function Descriptions 31](#_Toc353874540)

[6. Use cases 34](#_Toc353874541)

[D. DESIGN & IMPLEMENTATION 55](#_Toc353874542)

[I. Overview 55](#_Toc353874543)

[II. Proposed System Architecture 55](#_Toc353874544)

[III. Database Design 57](#_Toc353874545)

[1. Entity Relationship Diagram (ERD) 57](#_Toc353874546)

[2. Table descriptions 57](#_Toc353874547)

[IV. Detailed Design 65](#_Toc353874548)

[1. Server 65](#_Toc353874549)

[2. Waiter 76](#_Toc353874550)

[3. Chef 92](#_Toc353874551)

[E. TEST DOCUMENTATION 99](#_Toc353874552)

[F. USER’S MANUAL 100](#_Toc353874553)

[I. Waiter 100](#_Toc353874554)

[1. Home screen 100](#_Toc353874555)

[2. Add more order detail in list table serving 106](#_Toc353874556)

[3. Update an order detail 107](#_Toc353874557)

[4. Bill an order detail 108](#_Toc353874558)

[5. Logout the app 109](#_Toc353874559)

[II. CHEF 110](#_Toc353874560)

[1. Home screen 110](#_Toc353874561)

[G. APPENDIX 113](#_Toc353874562)

[I. References 113](#_Toc353874563)

[II. Meeting Minutes 114](#_Toc353874564)

[1. Meeting with customer 114](#_Toc353874565)

[2. Kick off meeting 114](#_Toc353874566)

[3. Project meeting week 2 115](#_Toc353874567)

[4. Project meeting week 3 115](#_Toc353874568)

[5. Project meeting week 5 116](#_Toc353874569)

[6. Project meeting week 7 116](#_Toc353874570)

[7. Project meeting week 8 117](#_Toc353874571)

[8. Project meeting week 9 117](#_Toc353874572)

[9. Project meeting week 10 118](#_Toc353874573)

# INTRODUCTION

## Problem Abstract

Currently, in the traditional restaurant/ café operations, there are some known problems. Most operations still require manual works from the employees: In order to serve a table of several customers, a waiter has to bring out the paper menu for the customers to choose dishes, wait to receive the orders from the customers, and then bring the order into the kitchen for the chefs; Similarly, if a customer wants to check out, the manager has to see the list of dishes they had in order to calculate the total price for the customer. If the chefs want to notify the waiter to come in the kitchen to bring out the foods, they often have to place one person to do that job, running in and out of the kitchen to notify the waiters. Low accuracy, time and effort consuming, relies too much on manual works are the problems of the traditional restaurant operations.

In order to overcome those problems, some tools and technologies were applied. Now every waiter in the world uses pens and papers to write down customers’ orders and notes. Some restaurants have built a bell ringing notification system for the chefs and waiters. At the receptionist’s desk there usually is a computer to do the storing and calculating jobs, with applications like Microsoft Excel, Word and some POS (point-of-sale) software. But overall, technology just plays a small part and everything is not connected; there is no integrated system that supports the work of the waiters, chefs and managers of a restaurant at the same time.

## Existing Products

Currently there are several similar products:

### AdeptPros GeniusMenu:



Figure A-II-1-1: AdeptPros GeniusMenu

GeniusMenu is a software product from AdeptPros, USA. GeniusMenu is introduced to be able to run one every environments (Android, iOS …) and all kind of devices (smart phones and tablets). This software has its main function as an electronic menu, which shows customers the menu of the restaurant, helps them in making their decisions. GeniusMenu uses a label printer in the Chef’s side. It also has billing and checkout functions.

### iPOS:

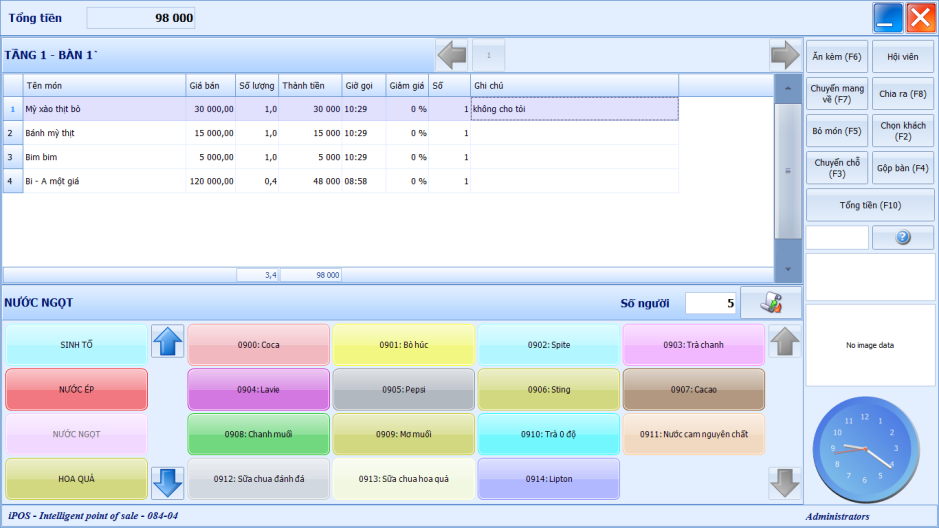


Figure A-II-2-1: iPOS

iPOS is a software product from Vietnam. This essentially is a software system that supports a restaurant’s manager, with the main functions of remote management, billing and revenue reports management, menu management …

There are also several other products like alacarte Presto, Microsoft Surface, Frame … But these products are all customer-oriented, not made for restaurant employees’ use like F-Menu.

### The limitations of the existing products

* Does not support Vietnamese language
* Lack of integration to other restaurant functions (like billing)
* Does not have a user-friendly interface
* Still requires lots of manual works

## Our proposal to the problems

The F-Menu is a software system that consists of one server application (F-Menu Server) and two Android tablet applications (F-Menu Waiter and F-Menu Chef). It provides a restaurant/ café’s employees a more effective, easier and simpler way to get orders from the customers and send them to the bar/ kitchen. The F-Menu Waiter (run on Android tablets) will support the waiter in arranging, changing the tables and ordering dishes for customers, updating the restaurant/ café’s table status. A waiter can use his tablet to send requests to the F-Menu Server, the server then processes and prioritizes the orders, and send them to the bar/ kitchen’s tablet (F-Menu Chef). The F-Menu Server application will also allow managers to manage the orders and waiter’s accounts, update the menu and manage table reservations. Specific functions will be listed in details in the Functional Requirements.

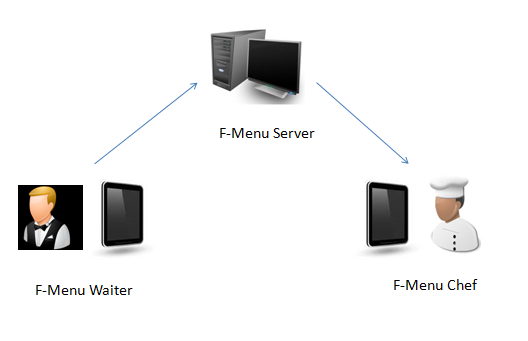
****

Figure A-III-1: F-Menu System

## The Project team

### General information of the Project

Project name in Vietnamese: Hệ thống hỗ trợ phục vụ bàn F-Menu

Project name in English: F-Menu Order Supporting System

Project name abbreviation: F-Menu

Outcome product: Software application system

Implementation: 2013

### Members and roles

Supervisor: Mr. Phan Trường Lâm

Team members:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Full Name** | **Student number** | **Tel. number** | **Email** | **Role** |
| 1 | Nguyễn Hoàng Đức Hiếu | 01435 | 0123 525 8904 | [Hieunhd01435@fpt.edu.vn](mailto:Hieunhd01435@fpt.edu.vn) | Project Manager |
| 2 | Phạm Văn Lâm | 01203 | 097 386 7701 | [Lampv01203@fpt.edu.vn](mailto:Lampv01203@fpt.edu.vn) | Member, Server developer |
| 3 | Hoàng Xuân Thành | 00719 | 098 207 0536 | [Thanhhx00719@fpt.edu.vn](mailto:Thanhhx00719@fpt.edu.vn) | Member, Tester |
| 4 | Lương Trọng Nghĩa | 01146 | 0168 990 4162 | [Nghialt01146@fpt.edu.vn](mailto:Nghialt01146@fpt.edu.vn) | Member, Client developer |

Table A-IV-2-1: Members and roles

### Planned Timeline

15 weeks to build the whole system:

* Initial Stage (Study and define problems): 2 weeks
* Definition Stage (Requirements Elicitation): 2 weeks
* Development Stage (Design & Development): 8 weeks
* Transition Stage (Testing & QA Phase): 2 weeks
* Termination Stage (Close out): 1 weeks

### Advantages on work

* The team receives many supports and assistance from supervisor
* The team already has the first customer – and also the one who brings up the idea - HieuHT00817
* The fact that Android tablets are relatively cheap also helps the team in working and testing with real devices.
* The team consists of 4 members, which have been working with each other for a few months, so understanding between members is not a problem.
* All members have a certain level of technical and theoretical skills.
* All members are responsible with assigned work and willing to work with high intensity.
* All members are equal; all opinions are respected and considered.

### Disadvantages

* Programming with Android is relatively new with the team.
* In the progress, there are a number of techniques that none of the members understand well like Web service, Server push …
* The team does not have any experience in managing project and building project management system.
* Some members still have their own work and studies.

# PROJECT MANAGEMENT PLAN

## Project Overview

### Project Name

Currently, the project’s name is Order Supporting System F-Menu (abbr. F-Menu). The F stands for FPT University. The name F-Menu means that this is an application comes from FPT University that assists the works of a restaurant/café’s waiters, chefs, and managers. The project’s name may be changed due to specific situations, or due to customers’ will.

### Project Objectives

The project’s main objective is to build a system consists of 1 Manager application and 2 Client application, 1 for Chef and 1 for Waiter (that can be implemented on many Tablets) that are all connected through Web service, to support the work of the staff in a restaurant/ café. The system, when implemented will help the waiters save time and efforts, make their actions more accurate, and create a professional working environment. The system must be reliable, fast, friendly, and easy to use.

### Project Scope

* The scope of the system:
  + Help the waiters in suggesting and making orders for customers.
  + Help the waiters in managing the restaurant’s tables and their currently serving tables.
  + Help the managers in managing menus and list of dishes, materials and financial reports.
  + Help the Chefs in managing all coming orders, material requests.
* Target users of the system:
  + All restaurants, restaurant chains, café.

### Development Environment

#### Hardware Requirements

* Personal computers for developing/testing with the minimum configuration: 2GB of RAM, 100 GB of hard disk, Core 2 Duo 2.0 GHz.
* A server computer with the minimum configuration: 4GB of RAM, 100GB of hard disk, Core 2 Duo 2.0 GHz.
* Android tablets with 3G and Wi-Fi powered for testing and deploying purposes

#### Software Requirements

* Operating System: Windows 7, Mac OSX Lion 10.7.5
* Framework: .NET Framework 4.0, ASP .NET Web Services
* IDE: Visual Studio 2010, Eclipse 3.6.2 with ADT Plugin
* UML Tool: VP UML 10.0
* Others: MS Office, MS Project, Adobe Photoshop CS5

### Tools and Techniques

#### Tools

|  |  |  |
| --- | --- | --- |
| **Tools** | **Description** | **Version** |
| Microsoft Project | Project management software, support for planning, assigning task, control progress… | 2010 |
| Microsoft Office | Use Word, Excel, PowerPoint to make reports, create charts, and make presentations | 2010 |
| VP UML | Design various UML diagrams | 10.0 |
| Microsoft Visual Studio | Development purpose | 2010 |
| Eclipse | Development purpose | 3.6.2 |
| Microsoft SQL Server | Design and store database | 2008 |

Table B-I-5-1: Tools

#### Techniques

|  |  |
| --- | --- |
| **Technique** | **Description** |
| ASP.NET Web service application | Framework for developing web service |
| Windowns Form Application .net Framework 4 | Framework for developing Management Console application |
| TCP Socket | Developing TCP server |
| Android | Developing tablet clients for Waiters and Chefs |

Table B-I-5-2: Techniques

## Analysis and selection of tools

### Tools & hardware requirements

* For the clients, we choose to develop on Android, because the Android tablets are the cheapest in the market, in compare to other OS like iOS or Windows 8.

|  |  |  |
| --- | --- | --- |
| iOS’s cheapest tablet | Windows 8’s cheapest tablet | Android’s cheapest tablet |
| Apple iPad Mini (16GB) with Wi-Fi | Microsoft Surface (32GB) | Teclast P76e (8GB) |
| $375 | $559.90 | $70 |
| http://allphone.vn/uploads/gallerys/IPad_Mini_Black.jpg | http://compass.surface.com/assets/aa/11/aa111ac0-84fa-4ebd-b54c-f977f0da449f.jpg | http://www.tabletpcphones.com/image/cache/data/Products/A0062/Teclast%20P76E%20Tablet%20PC-500x500.jpg |

Figure B-II-1-1: Various kinds of tablet

* In the Android market only, there are already a lot of models to choose from. In order to pick out the right one, we will have to look at the requirements for the hardware:
  + It has to be light (to be comfortable for the holder)
  + It must have a user-friendly interface with multitouch support
  + It must support Vietnamese language
  + It must have at least 512MB of RAM and 1.0GHz CPU (to work smoothly)
  + It must be cheap (to be affordable)
  + It must support Wi-Fi connection (at least 802.11 b/g/n)
  + It must have a screen of 7” (the perfect size to be hold on one hand)
  + It must have a long battery life (to serve a whole session without charging)
* After a careful research, we have chosen the AINOL Novo7 Mars as the main client for the system.

AINOL Novo7 Mars specifications:



Figure B-II-1-2: AINOL Novo7 Mars tablet

|  |  |
| --- | --- |
| Display Size | 7.0 inch |
| Display Pixels | 1024 x 600 |
| Display Type | TFT LCD |
| Multitouch | Yes |
| CPU | 1 GHz, AML8726-M3 Cortex A9 ARM-based |
| GPU | MALI-400 3D |
| RAM | 1GB DDR3 |
| Internal Storage | 8GB Flash |
| External Storage | microSDHC Card |
| Camera | 0.3 Megapixel (front) |
| Wireless | Wi-Fi 802.11b/g/n |
| Wired | miniUSB, 3.5mm audio |
| Battery | 3700 mAh lithium-ion polymer |
| Size | 191 x 125 x 10.9 mm |
| Weight | 360g |
| Price | $80 |

Table B-II-1-1: AINOL Novo7 Mars specifications:

This tablet is light (only 360g), fit to the hand of the user (7.0” display screen), has 1GB of RAM and 1GHz CPU, good battery capacity (3700 mAh) and come with a good price ($80).

### IDE requirements

* For the development purpose of the Management Console, we have decided to use Microsoft Visual Studio 2010, because VS supports C#, many tools and plugins, has useful extensions, and our developers are familiar with developing in this IDE.
* For the development purpose of the Clients, we have decided to use Eclipse 3.6.2 with ADT (Android Developers Tools) plugin version 21.1.0. This IDE comes as a bundle with SDK, the IDE itself, Android platform, latest Android platform tools, and the Android simulator. Those are all we need to use to develop Android applications.

## Project Organization

### Software Process Model

Because this system is developed from scratch, and requires many changes and updates, we have chosen the Iterative Development Model to base on and develop the system.

This approach interleaves the activities of specification, development and validation. The initial system is rapidly developed from very abstract specifications which are provided from the initial customer. Then this is refined with customer input to produce a system that satisfies the customer’s needs.

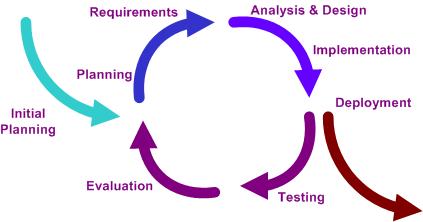


Figure B-III-1: Iterative development model

In term of project management process, we have to perform the following phases:

* Initial planning: Use UML tool like VP UML to model and design the abstract system, helping the stakeholders to understand the whole system and each other.
* Requirements: Elicit functional and non-functional requirements.
* Analysis & Design: Analyze requirements, design model and functions for the system.
* Implementation: Develop the system based on what was designed and analyzed.
* Testing: Test the functions of the system, also test the non-functional requirements.
* Evaluation: Evaluate all the works that are done to the point, then decide what to do next
* Deployment: The final stage: Deploying and packaging the system**.**
* Termination: Close out the project.

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Full Name** | **Role** | **Works** |
| 1 | Nguyễn Hoàng Đức Hiếu | Project Manager | Process Managing  Designing  Collecting and Developing Documents |
| 2 | Phạm Văn Lâm | Member, Developer | Developing Server side  Database Designing |
| 3 | Hoàng Xuân Thành | Member, Tester | Testing  Designing |
| 4 | Lương Trọng Nghĩa | Member, Developer | Developing Client side  Client Designing |

Table B-III-2: Roles and responsibilities

## Project Lifecycle

Because of the time limit and characteristics of a Capstone project, Requirements Stage and Analysis&Design Stage are merged into one Analysis&Design Stage. Testing and Evaluation Stage is merged into one Testing&Evaluation Stage.

### Project stage summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Stage** | **Start** | **End** | **Elapsed time** |
| Initiation Planning Stage | 7/1/2013 | 11/1/2013 | 1 week |
| Analysis & Design Stage | 14/1/2013 | 25/1/2013 | 2 weeks |
| Development Stage | 28/1/2013 | 1/3/2013 | 3 weeks |
| Testing&Evaluation Stage | 4/3/2013 | 8/3/2013 | 1 week |
| Analysis & Design Stage 2 | 11/3/2013 | 15/3/2013 | 1 week |
| Development Stage 2 | 18/3/2013 | 12/4/2013 | 4 weeks |
| Testing&Evaluation Stage 2 | 15/4/2013 | 19/4/2013 | 1 week |
| Deployment Stage | 22/4/2013 | 26/4/2013 | 1 week |

Table B-IV-3-1: Project stage summary

### Project stage and milestone

|  |  |  |
| --- | --- | --- |
| **Stage** | **Description** | **Milestone** |
| Initiation Planning Stage | The initiation planning stage will intend to get project requirements and review those, estimate project size and effort, create the internal work order and team building | Report 1  (introduction)  Report 2  (Project Plan) |
| Analysis & Design Stage | The analysis and design stage will develop the product requirements and establish the business case for the F-Menu system. The major use cases will be developed as well as the high level Project Plan. Prepare tools and infrastructure. | Report 3  (SRS)  Report 4  (Design  and Implementation) |
| Development Stage | The development stage will analyze the requirements and will develop the architectural prototypes, create test cases, develop product modules, integrate modules, system test, FI … | Report 4  (Design  and Implementation) |
| Testing & Evaluation Stage | The testing and evaluation stage will execute the testing process and performing QA operations to detect system bugs, flaws, leakages… and evaluate the quality of the whole system. | Report 5  (Test Documentation) |
| Deployment Stage | Boxing software package, complete additional document (References, User Guide…) | Report 6  (Conclusion)  Final Report |
| Termination Stage | Present Project | Project Presentation |

Table B-IV-3-2: Project stage and milestone

## Project Management Plan

### Tasks

#### Planning

|  |  |
| --- | --- |
| Description | Plan to manage human resources, define the project goals and objectives, identify tasks and how goals will be achieved, quantify the resources needed, and determine budgets and timelines for completion. |
| Output | Planning document, schedule |
| Deliverables | Report 1, MPP file |
| Dependencies and Constraints | N/A |
| Risks | Underschedule or overschedule |

Table B-V-1-1: Task 1– Planning

#### Training

|  |  |
| --- | --- |
| Description | Study the basics of developing in Android and other technologies like Web service, TCP servers… |
| Output | N/A |
| Deliverables | N/A |
| Dependencies and Constraints | N/A |
| Risks | The technologies may be new and difficult to learn |

Table B-V-1-2: Task 2 - Training

#### Software Requirements Analysis

|  |  |
| --- | --- |
| Description | Analyze software requirements to create software requirements specification document |
| Output | Software Requirement Specification document |
| Deliverables | SRS Document file |
| Dependencies and Constraints | N/A |
| Risks | N/A |

Table B-V-1-3: Task 3 – Analysis

#### Develop Server side application

|  |  |
| --- | --- |
| Description | Develop the server side application for the manager |
| Output | Source code |
| Deliverables | Executable program and source code |
| Dependencies and Constraints | N/A |
| Risks | There are many functions in the server side application (booking, change table, managing accounts…) so this might costs a long time to develop. |

Table B-V-1-4: Task 4 – Develop server side application

#### Develop Notify server

|  |  |
| --- | --- |
| Description | Develop the Web service and TCP server used for connecting clients |
| Output | Source code |
| Deliverables | Executable program and source code |
| Dependencies and Constraints | Clients interface development is done |
| Risks | Members are not familiar with working with Web service and TCP server. |

Table B-V-1-5: Task 5 – Develop Notify server

#### Develop Client side application

|  |  |
| --- | --- |
| Description | Develop 2 Android applications, 1 for Chef and 1 for Waiter |
| Output | Source code |
| Deliverables | Executable program and source code |
| Dependencies and Constraints | N/A |
| Risks | Members are not familiar with developing in Android environment. |

Table B-V-1-6: Task 6 – Develop Client side application

#### Testing

|  |  |
| --- | --- |
| Description | Creating test case and execute test |
| Output | Test plan, test case document, test report, all tested modules and tested system |
| Deliverables | Test documentations (Test plans, test reports) |
| Dependencies and Constraints | Developments are done |
| Risks | Unit test may not be performed thoroughly causing spending many efforts in system test phase. |

Table B-V-1-7: Task 7 - Testing

### Task Sheet: Assignment and Timetable

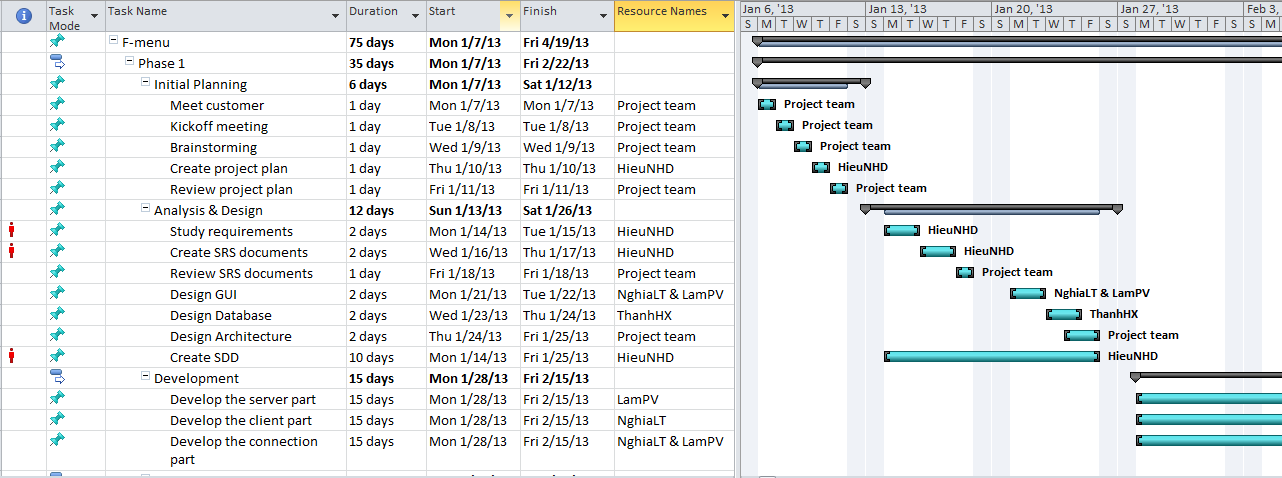


Figure B-V-2-1: Master Plan excerpt 1

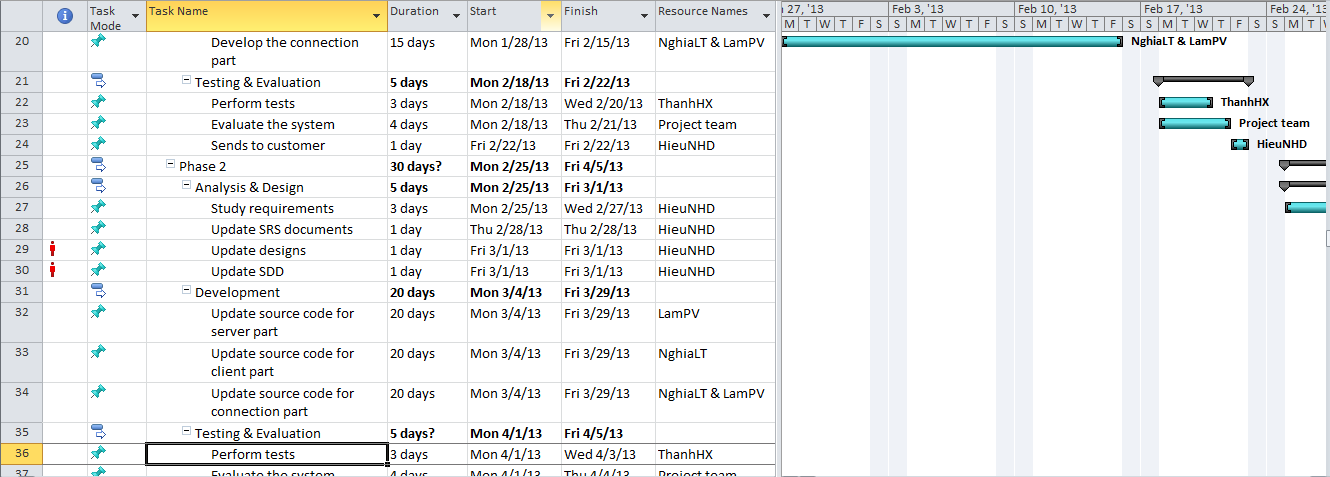


Figure B-V-2-2: Master Plan excerpt 2

*Please refer to the attached Master Schedule document: F-Menu Master Plan.mpp*

# SOFTWARE REQUIREMENTS SPECIFICATION

## Introduction

### Objectives

This is the Software Requirements Specification for the F-Menu system. This SRS details the capabilities and functions that the F-Menu must be capable of performing. The F-Menu is designed to be a system that assists the work of the waiters, chefs and managers in a restaurant/ café. These requirements will assure that the system will correctly and reliably perform its intended functionality. This specification will provide general, as well as specific requirements to be used in the design, testing and validation of the system.

### Scope

This Software Requirements Specification applies to the F-Menu system, including the F-Menu Android applications (F-Menu Waiter and F-Menu Chef) and the F-Menu Server. The system will act as a communication mediator between the waiters and chefs and as a management tool for managers of a restaurant.

### Acronym and Definitions

|  |  |
| --- | --- |
| **Acronym and Definitions** | **Meaning** |
| App | Short for Application |
| Client | Android application that runs on Android tablet PC |
| Customer | The restaurant’s customer |
| Dish | The product that the restaurant provides: foods, drinks … |
| F-Menu Chef | The Android application for Chef |
| F-Menu Server | The Windows form application for Manager |
| F-Menu Waiter | The Android application for Waiter |
| FR | Functional Requirements |
| GUI | Graphic User Interface |
| Menu | “Menu” with a capitalized M means the restaurant’s menu, the other is the software specified menu |
| SRS | Software Requirements Specification |
| Tablet | Tablet PCs using Android OS, main client devices |
| TCP server | A notify server that is used to transfer messages between clients |
| Waiter | A person who serve in a restaurant |
| Web service | Used to connect everything in the system |

Table C-I-3-1: Acronym and Definitions

## Overall Description

### Product Functions

The F-Menu is a software system that consists of one server application (F-Menu Server) and two Android tablet applications (F-Menu Waiter and F-Menu Chef). It provides a restaurant/ café’s employees a more effective, easier and simpler way to get orders from the customers and send them to the bar/ kitchen. The F-Menu Waiter (run on Android tablets) will support the waiter in arranging, changing the tables and ordering dishes for customers, updating the restaurant/ café’s table status. A waiter can use his tablet to send requests to the F-Menu Server, the server then processes and prioritizes the orders, and send them to the bar/ kitchen’s tablet (F-Menu Chef). The F-Menu Server application will also allow managers to manage the orders and waiter’s accounts, update the menu and manage table reservations. Specific functions will be listed in details in the Functional Requirements.

### Interfaces

#### User Interfaces

The F-Menu Clients (Waiter and Chef) will be accessed through an Android tablet either by a waiter or a chef. The F-Menu Server will be accessed through a PC, with a Windows form User Interface.

#### User Characteristics

The F-Menu is designed with a user friendly GUI, anybody who has use a tablet or smart phone before should be able to use the application.

## Requirements

### Customer Requirements

The requirements below are the requirements collected from customer prior to the elicitation of any other requirements.

#### Operational distribution or deployment: Where will the system be used?

* + - * The system will be used for a restaurant/ café environment.
      * The system will assist the place’s employees (waiters, managers, chefs) in serving customer orders professionally, accurately and fast.
      * Synchronizing and digitizing data for management.

#### Mission profile or scenario: How will the system accomplish its mission objective?

* + - * The system will use Tablets and computers as main hardware; Wi-Fi is the main network infrastructure.
      * The software is a Tablet application, designed to support waiters/ waitresses, chefs, and managers.
      * Placing orders fast and accurately (Names, quantities …)
      * Minimize the time used for transferring orders (from customers to the bar), so that a waiter can serve continuously, improve productivity by 3-4 times at peak hour.
      * Every data transmission will use Wi-Fi as connection (fast and low cost)
      * The system helps the process of customers’ payments.
      * It can be programmed to automatically process all orders, synchronizing all tasks.
      * Managing Orders: Pending – Served – Paid

#### Performance and related parameters: What are the critical system parameters to accomplish the mission?

* + - * Fast
      * Accurate
      * Easy to integrate and extend with other modules.

#### Effectiveness requirements: How effective or efficient must the system be in performing its mission?

* + - * Delay from the ordering side to the serving side must be less than 5 seconds.
      * The server and the tablet’s application must not be frozen during performance.
      * The data must be backed up.
      * In case Tablets or server crash, there must be a plan to fix.

### External Interface Requirements

#### User Interface

The user interfaces have to satisfy the principles below:

* **Learnable**: The user interface must be easy to learn
* **Efficiency**: The user interface must minimize the number of steps the system takes to complete its task.
* **Memorable**: Interface should become easier to use each time the user interacts with it again.
* **Visibility**: Important information (example: detected information, captured image) should be clearly visible.
* **Consistency**: Like-items should always be displayed and act the same way across the entire application.

#### Hardware Interface

At the client side, Android application needs an Android tablet with Wi-Fi capability to run and connect with the other parts of the system.

At the server side, the application needs a server PC with minimum requirements:

* CPU: Core 2 Duo 2.0GHz
* RAM: 2GB
* HDD: 100GB
* Internet connection

#### Software Interface

|  |  |  |
| --- | --- | --- |
| **Category** | **Software Name** | **Version** |
| Operating system | Windows | 7 |
| Android | 4.0.1 |
| MacOS | 10.7.5 |
| Framework | .NET Framework | 4.0 |
| ASP .NET Web Service | 2.0 |
| Data management system | SQL Server | 2008 R2 SP1 |
| IDE | Visual Studio | 2010 |
| Eclipse | 3.6.2 |
| Others | Newtonsoft.Json extension |  |
| Eneter.Messaging.Framework library |  |
| UML Tools | VP UML | 10.0 |

Table C-III-2-1: Software Interface

### Communication Protocol

F-Menu system uses standard protocols (SOAP messages) and TCP/IP to communicate between clients and server.

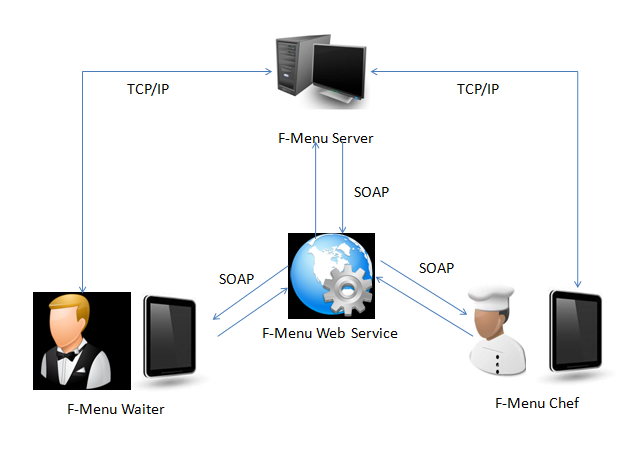


Figure C-III-3-1: Communication Protocol

### Functional Requirements

#### Function List

Below is the function list for the F-Menu system.

##### Functions of F-Menu Waiter:

|  |  |
| --- | --- |
| **Code** | **Description** |
| FR01 | Login/ logout |
| FR02 | Arrange tables for customer |
| FR03 | Create orders |
| FR04 | Manage orders |
| FR05 | Billing |
| FR06 | View paid cash |
| FR07 | Save data locally |

Table C-III-4-1: Functions of F-Menu Waiter

##### Functions of F-Menu Chef

|  |  |
| --- | --- |
| **Code** | **Description** |
| FR08 | Login/ logout |
| FR09 | List pending dish orders |
| FR10 | Send notify messages to waiters when dishes are finished or not available .etc… |
| FR11 | Materials request |

Table C-III-4-2: Functions of F-Menu Chef

##### Functions of F-Menu Server

|  |  |
| --- | --- |
| **Code** | **Description** |
| FR12 | Login/logout |
| FR13 | Manage Menu |
| FR14 | Manage accounts |
| FR15 | Financial reports |
| FR16 | Manage tables |

Table C-III-4-3: Functions of F-Menu Server

### Function Descriptions

#### FR01: Login/ logout

* Access right: Users
* Description: Users use this function to manage their sessions by logging in or out of the system, which grant them the rest of the system’s functions.

#### FR02: Arrange table for customer

* Access right: Waiters
* Description: Waiters use this function to arrange customers(s) to their seats. This will mark the status of the table as “Using”.

#### FR03: Create orders

* Access right: Waiters
* Description: Waiters use this function to create a new order request for customer to send to chef. Waiters select dish from the Menu, add notes if special requests are made, then send them to chef.

#### FR04: Manage orders

* Access right: Waiters
* Description: Waiters use this function to edit the details of the orders made, like undo an order (that has not been done by the chef), decrease the price for the food left (like when the customer order 10 beers but only consume 5, the waiter will decrease 5 from the bill).

#### FR05: Billing

* Access right: Waiters
* Description: Waiters use this function to show the bill to the customer.

#### FR06: View paid cash

* Access right: Waiters
* Description: After show the bill to the customer, the waiter may take cash directly from the customer. This function can show the total amount of cash a waiter has to return to the restaurant at the end of the day.

#### FR07: Save data locally

* Access right: Waiters
* Description: This function allows the tablet to download all data and save them locally in case of lost connection happens.

#### FR08: Login/ logout (for chefs)

* Access right: Users
* Description: Users use this function to manage their sessions by logging in or out of the system, which grant them the rest of the system’s functions.

#### FR09: List pending dish orders

* Access right: Chefs
* Description: This function displays all orders automatically for chefs to fulfill.

#### FR10: Send notify messages to waiters

* Access right: Chefs
* Description: This function allows chefs to send back notifications to waiters in case a dish is done, or cannot be served, etc…

#### FR11: Materials request

* Access right: Chefs
* Description: This function allows chefs to send notification to manager to buy new materials when materials are running out.

#### FR12: Login/ logout (for managers)

* Access right: Users
* Description: Users use this function to manage their sessions by logging in or out of the system, which grant them the rest of the system’s functions.

#### FR13: Manage Menu

* Access right: Managers
* Description: Managers use this function to manage the list of materials, list of dishes and use them to create the daily Menu.

#### FR14: Manage accounts

* Access right: Managers
* Description: Managers use this function to manage the accounts of all users in the system, including viewing, changing their status, reset their passwords…

#### FR15: Financial reports

* Access right: Managers
* Description: Managers use this function to view the financial reports or print them out.

#### FR16: Manage table

* Access right: Managers
* Description: Managers use this function to book tables, view table status, change table status…

### Use cases

#### Main use case:

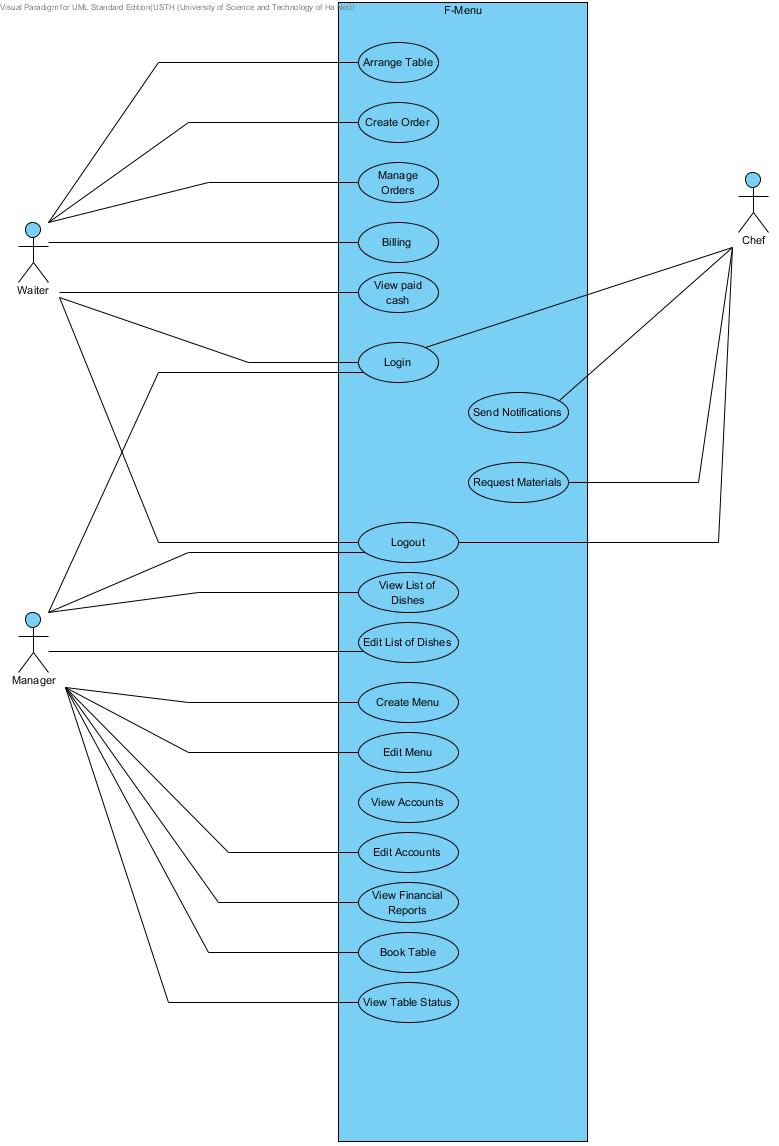


Figure C-III-6-1: Main use case

#### Functional specifications

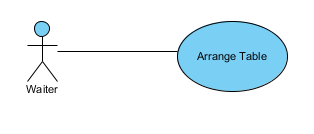


Figure C-III-6-2: Arrange table

|  |  |
| --- | --- |
| **Use Case Name** | **1 - Arrange table** |
| Author | HieuNHD |
| Brief Description | This function allows waiter to arrange a table for the customer. |
| Actors | Waiter |
| Preconditions | Waiter has already logged in the system. |
| Post-conditions | Customer is arranged a table. |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Waiter opens the "Table" menu item from the tablet, and then chooses a table from a table list. |  |  |  | | 2 |  | The application asks for confirmation from waiter. |  |  | | 3 | Waiter confirms choosing the tables. |  |  |  | | 4 |  | The table's status is changed from "Available" to "Using".  The table is set to the waiter that currently serving it. |  |  | |

Table C-III-6-2: Use case 1

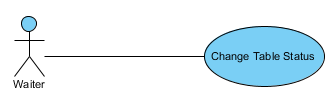


Figure C-III-6-3: Change Table Status

|  |  |
| --- | --- |
| **Use Case Name** | **2 - Change Table Status** |
| Author | HieuNHD |
| Brief Description | This function allows waiter to change the table status for a customer. |
| Actors | Waiter |
| Preconditions | Customer is arranged a table. |
| Post-conditions |  |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Waiter opens the "Serving Tables" menu item from the tablet, chooses a table from a table list, then chooses to change its status. |  |  |  | | 2 |  | The application asks for confirmation from waiter. |  |  | | 3 | Waiter confirms choosing the tables. |  |  |  | | 4 |  | The table's status is changed. |  |  | |

Table C-III-6-3: Use case 2

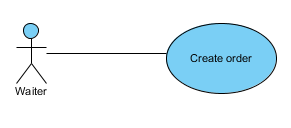


Figure C-III-6-4: Create order

|  |  |
| --- | --- |
| **Use Case Name** | **3 – Create order** |
| Author | HieuNHD |
| Brief Description | This function allows waiter to make an order for the customer. |
| Actors | Waiter |
| Preconditions | The customer is already arranged a table. |
| Post-conditions | The order is sent to chef. |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Waiter enters orders into tablet (attaches notes if customer makes special order). |  |  |  | | 2 |  | The application asks for confirmation. |  |  | | 3 | Waiter confirms the order. |  |  |  | | 4 |  | The order is sent to chef side. |  |  | | 5 |  | The order's status is changed to "Pending", color is turned to Yellow (Orange if special). |  |  | |

Table C-III-6-4: Use case 3

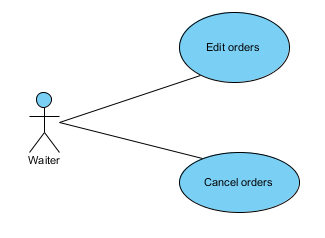


Figure C-III-6-5: Manage order

|  |  |
| --- | --- |
| **Use Case Name** | **4 – Manage order** |
| Author | HieuNHD |
| Brief Description | This function allows waiter to change an order for the customer. |
| Actors | Waiter |
| Preconditions | A previous order must exist. |
| Post-conditions | The order is updated and sent to chef. |
| Flow of Events  (Edit Order) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Waiter updates (adds new dish into) an existing order. |  |  |  | | 2 |  | The application asks for confirmation. |  |  | | 3 | Waiter confirms updates. |  |  |  | | 4 |  | The order is updated. |  |  | | 5 |  | The updated order's color turns to Red in the Chef's side. |  |  | |
| Flow of Events  Exception  (Cancel Orders) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Waiter chooses to cancel order from Table details. |  |  |  | | 2 |  | The application asks for confirmation. |  |  | | 3 | Waiter confirms cancellation. |  |  |  | | 4 |  | The order is canceled and is removed from Chef's queue. |  |  | |

Table C-III-6-5: Use case 4

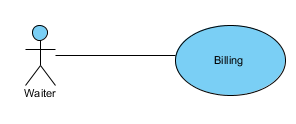


Figure C-III-6-6: Billing

|  |  |
| --- | --- |
| **Use Case Name** | **5 – Billing** |
| Author | HieuNHD |
| Brief Description | This function allows waiter to show the bill to the customer with the tablet. |
| Actors | Waiter |
| Preconditions | The "Create order" step is already done. |
| Post-conditions | Able to proceed to accept customer's payment. |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Waiter chooses to view the total bill. |  |  |  | | 2 |  | The application automatically calculates the total amount of payment then displays the bill. |  |  | |

Table C-III-6-6: Use case 5

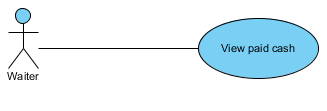


Figure C-III-6-7: View paid cash

|  |  |
| --- | --- |
| **Use Case Name** | **6 – View paid cash** |
| Author | HieuNHD |
| Brief Description | This function allows waiter to receive cash directly from the customer then save information in the application. |
| Actors | Waiter |
| Preconditions | Waiter has already show customer the bill. |
| Post-conditions |  |
| Flow of Events  (View paid cash) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Waiter chooses to view the total amount of cash he has to return to the restaurant. |  |  |  | | 2 |  | The total accumulated amount of cash to that point is displayed. |  |  | |

Table C-III-6-7: Use case 6

#### Login/ logout functions

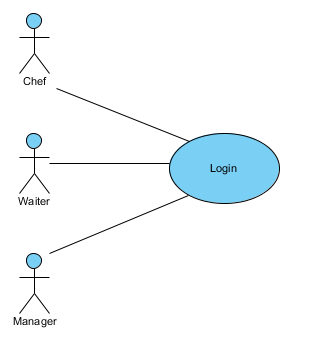


Figure C-III-6-8: Login

|  |  |
| --- | --- |
| **Use Case Name** | **7 – Login** |
| Author | HieuNHD |
| Brief Description | This function allows employee to login the system. |
| Actors | Waiter, Manager, Chef |
| Preconditions |  |
| Post-conditions | The employee is logged in the system. |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Employee enters username and password. |  |  |  | | 2 |  | The system notifies that the employee is successfully logged in the system. |  |  | |
| Flow of Events (Exception) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Employee enters wrong username or password. |  |  |  | | 2 |  | The system notifies that the log in process is failed. |  |  | |

Table C-III-6-8: Use case 7

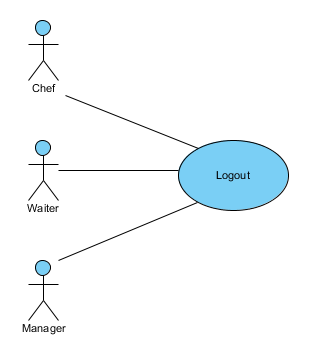


Figure C-III-6-9: Logout

|  |  |
| --- | --- |
| **Use Case Name** | **8 - Logout** |
| Author | HieuNHD |
| Brief Description | This function allows employee to logout the system. |
| Actors | Waiter, Manager, Chef |
| Preconditions | Employee has already logged in the system. |
| Post-conditions | The employee is logged out of the system. |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Employee clicks on Logout. |  |  |  | | 2 |  | The application asks for confirmation. |  |  | | 3 | Employee confirms. |  |  |  | | 4 |  | Employee is logged out of the system. |  |  | |

Table C-III-6-9: Use case 8

#### Chef functions

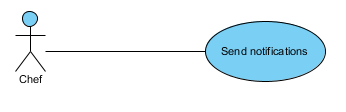


Figure C-III-6-10: Send notifications

|  |  |
| --- | --- |
| **Use Case Name** | **9 – Send notifications** |
| Author | HieuNHD |
| Brief Description | This function allows chef to send notifications back to the waiters. |
| Actor | Chef |
| Preconditions | Chef already receives an order. |
| Post-conditions | Finished order's status is changed to "Done", color turned to Green.  Not available order's status is changed to "N/A", color turned to Gray. |
| Flow of Events  (Notify Complete order) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Chef finishes a dish. |  |  |  | | 2 | Chef double taps on the finished dish's name on the tablet. |  |  |  | | 3 |  | The dish's status is changed to "Done", color turned to Green. |  |  | |
| Flow of Events  Exception  (Notify cancel order) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | If a dish cannot be done, Chef does a long press on the dish's name on the tablet. |  |  |  | | 2 |  | The unavailable dish's status is changed to "N/A", color turned to Gray. |  |  | |

Table C-III-6-10: Use case 9

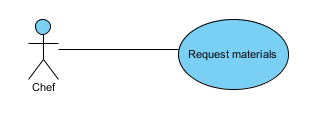


Figure C-III-6-11: Request materials

|  |  |
| --- | --- |
| **Use Case Name** | **10 – Request materials** |
| Author | HieuNHD |
| Brief Description | This function allows chef to send a notification to the manager to request materials. |
| Actor | Chef |
| Preconditions | Materials are running out. |
| Post-conditions | Notes are sent to manager. |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Chef sends note to manager to buy more materials. |  |  |  | | 2 |  | A message is displayed shows that Notifications are successfully sent to the F-Menu Server. |  |  | |

Table C-III-6-11: Use case 10

#### Manager functions

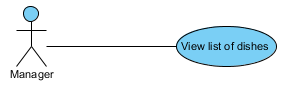


Figure C-III-6-12: View list of dishes

|  |  |
| --- | --- |
| **Use Case Name** | **11 – View list of dishes** |
| Author | HieuNHD |
| Brief Description | This function allows manager to view the list of dishes. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions |  |
| Flow of Events  (View list of Dishes) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager chooses to view List of Dishes. |  |  |  | | 2 |  | List of Dishes is displayed. |  |  | |

Table C-III-6-12: Use case 11

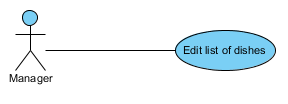


Figure C-III-6-13: Edit list of dishes

|  |  |
| --- | --- |
| **Use Case Name** | **12 – Edit list of dishes** |
| Author | HieuNHD |
| Brief Description | This function allows manager to edit the list of dishes. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions | List of dishes is updated. |
| Flow of Events  (Edit list of Dishes) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager chooses to edit List of Dishes. |  |  |  | | 2 |  | List of Dishes is updated. |  |  | |

Table C-III-6-13: Use case 12

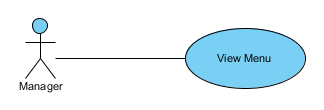


Figure C-III-6-14: View Menu

|  |  |
| --- | --- |
| **Use Case Name** | **13 – View Menu** |
| Author | HieuNHD |
| Brief Description | This function allows manager to view the Menu. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions |  |
| Flow of Events  (View Menu) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager chooses to view an existed Menu. |  |  |  | | 2 |  | The chosen Menu is displayed. |  |  | |

Table C-III-6-14: Use case 13

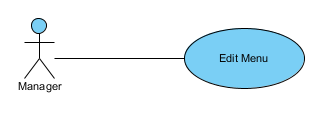


Figure C-III-6-15: Edit Menu

|  |  |
| --- | --- |
| **Use Case Name** | **14 – Edit Menu** |
| Author | HieuNHD |
| Brief Description | This function allows manager to edit the Menu. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions | Menu is updated. |
| Flow of Events  (Edit Menu) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager uses existing dishes from the List of Dishes to create a Menu for every day. |  |  |  | | 2 |  | The Menu on the server is updated. |  |  | | 3 |  | The Menu on the tablets is updated. |  |  | |

Table C-III-6-15: Use case 14

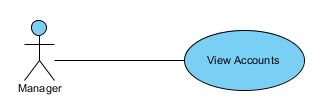


Figure C-III-6-16: View accounts

|  |  |
| --- | --- |
| **Use Case Name** | **15 – View accounts** |
| Author | HieuNHD |
| Brief Description | This function allows manager to view the employees’ accounts. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions |  |
| Flow of Events  (View Accounts) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager chooses to view the status of the waiter's accounts. |  |  |  | | 2 |  | The status of every waiter's account is displayed. |  |  | |

Table C-III-6-16: Use case 15

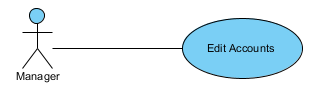


Figure C-III-6-17: Edit Accounts

|  |  |
| --- | --- |
| **Use Case Name** | **16 – Edit accounts** |
| Author | HieuNHD |
| Brief Description | This function allows manager to edit the employees’ accounts. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions |  |
| Flow of Events  (Edit Accounts) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager chooses to edit a waiter's account. |  |  |  | | 2 |  | The waiter's account is updated. |  |  | |

Table C-III-6-17: Use case 16

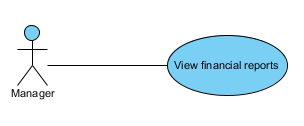


Figure C-III-6-18: View financial reports

|  |  |
| --- | --- |
| **Use Case Name** | **17 – View financial reports** |
| Author | HieuNHD |
| Brief Description | This function allows manager to view the financial reports. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions |  |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager chooses to view financial report of any day. |  |  |  | | 2 |  | The chosen financial report is displayed. |  |  | |

Table C-III-6-18: Use case 17

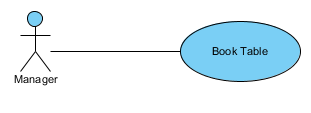


Figure C-III-6-19: Book table

|  |  |
| --- | --- |
| **Use Case Name** | **18 – Book table** |
| Author | HieuNHD |
| Brief Description | This function allows manager to book the table. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions |  |
| Flow of Events (Book Table) | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager receives table book request from customer A. (via telephone, online ...)  Manager marks table as "Booked". |  |  |  | | 2 |  | Status of the table is changed to "Booked to A". |  |  | |

Table C-III-6-19: Use case 18

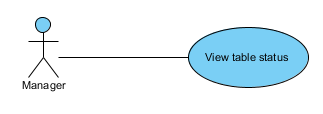


Figure C-III-6-20: View table status

|  |  |
| --- | --- |
| **Use Case Name** | **19 – View table status** |
| Author | HieuNHD |
| Brief Description | This function allows manager to view the table status. |
| Actor | Manager |
| Preconditions |  |
| Post-conditions |  |
| Flow of Events | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Actor Input | System Response |  |  | | 1 | Manager chooses to view a table’s status. |  |  |  | | 2 |  | Status of the table is displayed. |  |  | |

Table C-III-6-20: Use case 19

# DESIGN & IMPLEMENTATION

## Overview

This document provides a high-level description of the F-Menu system architecture, the architectural styles and components that have been selected to best achieve the use cases. It also gives a better understanding of the system’s design, using a number of different architectural views to depict different aspects of the system.

## Proposed System Architecture

* System Architecture Diagram:

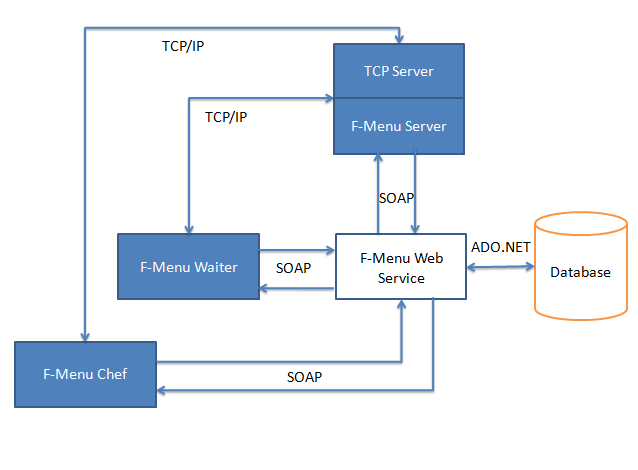


Figure D-II-1: System Architecture Diagram

* System Architecture Description:

F-Menu system consist of 2 types of client applications (1 for waiter, 1 for chef) that run on Android OS tablet, 1 Windows form application for manager, 1 Web service and 1 TCP server (which is a module in the Windows form application).

The F-Menu Waiter sends order information to the F-Menu Chef, F-Menu Server through the Web service and vice versa. The Clients (F-Menu Chef and F-Menu Waiter) use the TCP Server to send notifications between them.

## Database Design

### Entity Relationship Diagram (ERD)

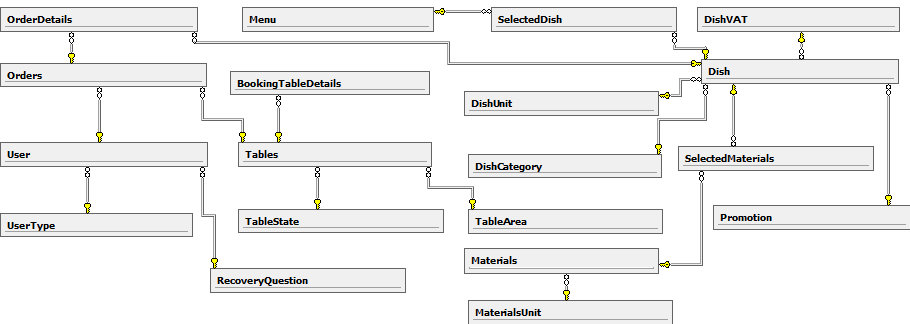


Figure D-III-1-1: Entity Relationship Diagram

### Table descriptions

* Table OrderDetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | orderID | int |  |  |
| 3 | Time | datetime |  |  |
| 4 | Quantity | int |  |  |
| 5 | TotalPrice | Int |  |  |
| 6 | Note | nvarchar(255) |  |  |
| 7 | dishName | nvarchar(50) |  |  |
| 8 | VAT | decimal (18, 2) |  |  |
| 9 | Promotion | decimal (18, 2) |  |  |
| 10 | Recipes | nvarchar(MAX) |  |  |
| 11 | Description | nvarchar(MAX) |  |  |
| 12 | Unit | nvarchar(50) |  |  |

Table D-III-2-1: OrderDetails

* Table Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | OrderID | int |  |  |
| 2 | UserName | nvarchar(50) |  |  |
| 3 | Size | int |  |  |
| 4 | CheckInTime | datetime |  |  |
| 5 | CheckOutTime | datetime |  |  |
| 6 | TotalCost | int |  |  |
| 7 | Deposit | int |  |  |
| 8 | CustomerMoney | int |  |  |
| 9 | isPaid | bit |  |  |
| 10 | tableName | nvarchar(50) |  |  |

Table D-III-2-2: Orders

* Table Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | Name | nvarchar(50) |  |  |
| 2 | typeID | int |  |  |
| 3 | Image | nvarchar(MAX) |  |  |
| 4 | Password | nvarchar(MAX) |  |  |
| 5 | rQuestionID | int |  |  |
| 6 | RecoveryAnswer | nvarchar(255) |  |  |
| 7 | isConnect | bit |  |  |

Table D-III-2-3: Users

* Table UserType

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Name | nvarchar(50) |  |  |
| 3 | Description | nvarchar(255) |  |  |

Table D-III-2-4: UserType

* Table RecoveryQuestion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Name | nvarchar(50) |  |  |

Table D-III-2-5: Recovery Question

* Table Menu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Name | nvarchar(50) |  |  |
| 3 | StartDate | date |  |  |
| 4 | Description | nvarchar(255) |  |  |

Table D-III-2-6: Menu

* Table SelectedDish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | dishID | int |  |  |
| 3 | menuID | int |  |  |

Table D-III-2-7: Selected Dish

* Table Tables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Name | nvarchar(50) |  |  |
| 3 | Description | nvarchar(255) |  |  |
| 4 | areaID | int |  |  |
| 5 | stateID | int |  |  |
| 6 | Size | int |  |  |
| 7 | Note | nvarchar(255) |  |  |

Table D-III-2-8: Tables

* Table TableState

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Name | nvarchar(50) |  |  |
| 3 | Description | nvarchar(255) |  |  |

Table D-III-2-9: TableState

* Table TableArea

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Name | nvarchar(50) |  |  |
| 3 | Description | nvarchar(255) |  |  |

Table D-III-2-10: Table Area

* Table Dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Image | nvarchar(MAX) |  |  |
| 3 | Name | nvarchar(50) |  |  |
| 4 | categoryID | int |  |  |
| 5 | Size | int |  |  |
| 6 | unitID | int |  |  |
| 7 | Price | int |  |  |
| 8 | VATID | int |  |  |
| 9 | promotionID | int |  |  |
| 10 | Recipes | nvarchar(MAX) |  |  |
| 11 | Description | nvarchar(MAX) |  |  |

Table D-III-2-11: Dish

* Table: DishVAT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Attribute | Type | Constraints | Description |
| 1 | ID | int |  |  |
| 2 | Value | decimal (18, 2) |  |  |
| 3 | Description | nvarchar(MAX) |  |  |

* Table: Promotion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Value | decimal (18, 2) |  |  |
| 3 | Description | nvarchar(MAX) |  |  |
| 4 | StartTime | date |  |  |
| 5 | EndTime | date |  |  |

Table D-III-2-12: Dish VAT

* Table: Materials

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | dishID | int |  |  |
| 3 | Name | nvarchar(50) |  |  |
| 4 | unitID | int |  |  |
| 5 | Quantity | int |  |  |
| 6 | Description | nvarchar(MAX) |  |  |

Table D-III-2-13: Materials

* Table: MaterialsUnit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Attribute** | **Type** | **Constraints** | **Description** |
| 1 | ID | int |  |  |
| 2 | Name | nvarchar(50) |  |  |
| 3 | Description | nvarchar(255) |  |  |

Table D-III-2-14: Materials Unit

## Software Design

### Component Diagram

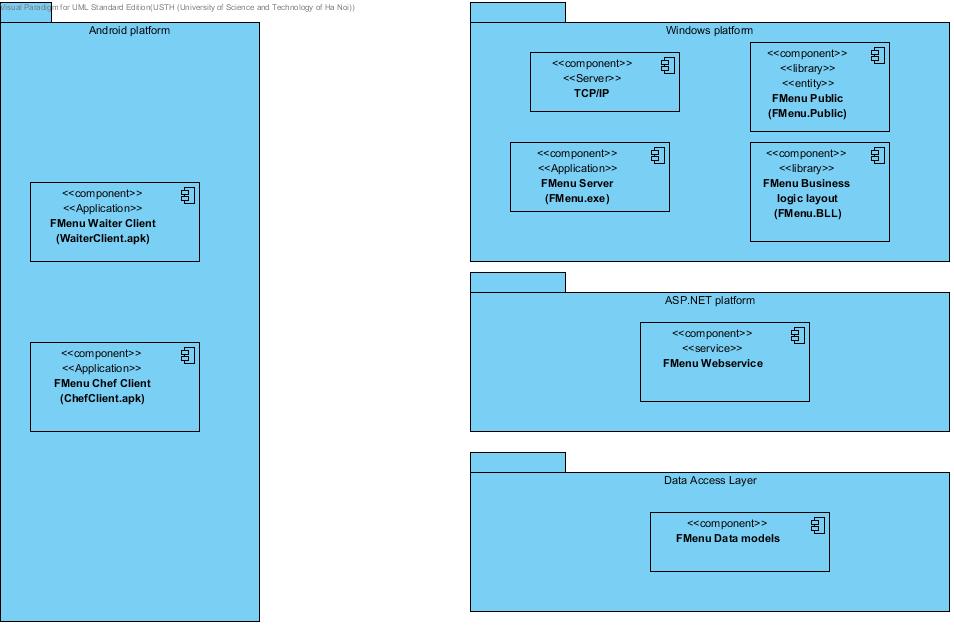


Figure D-IV-1-1: System General Component Diagram

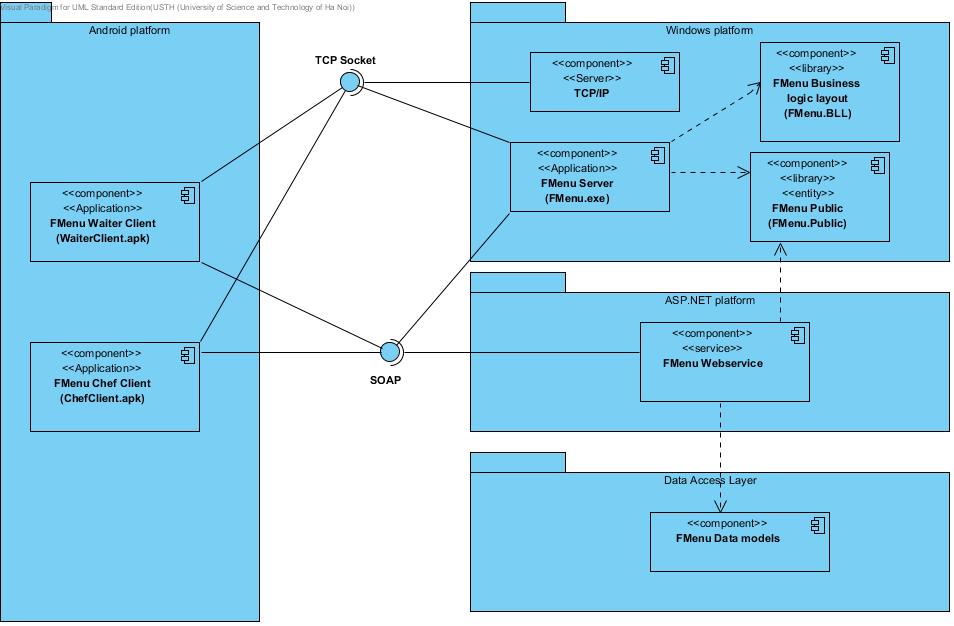


Figure D-IV-1-2: System Detailed Component Diagram

### F-Menu Server

#### Class Diagram

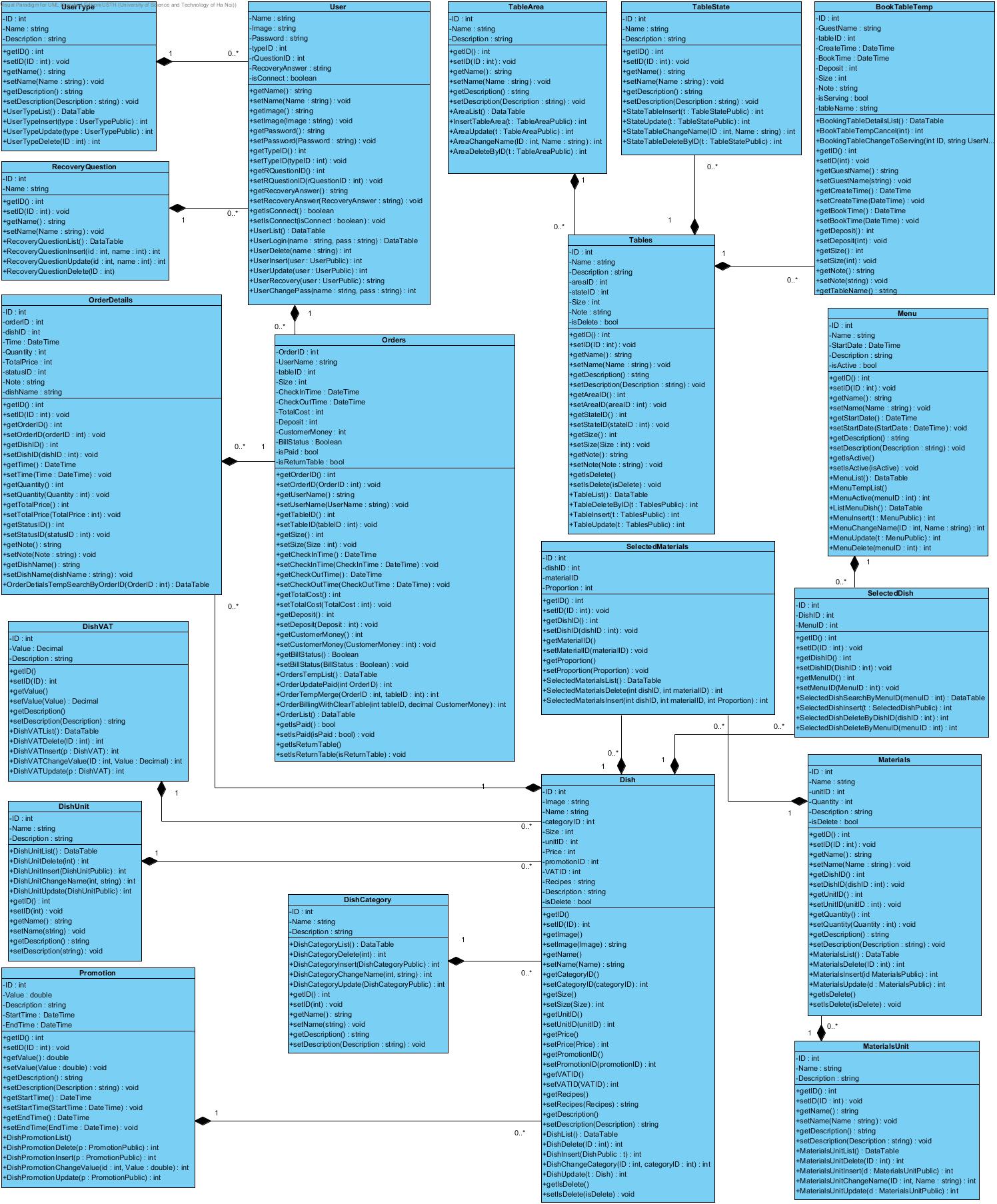


Figure D-IV-2-1: F-Menu Server Class Diagram

*(Refer to Attached Files for higher resolution image)*

#### Sequence Diagrams

##### Add Dish

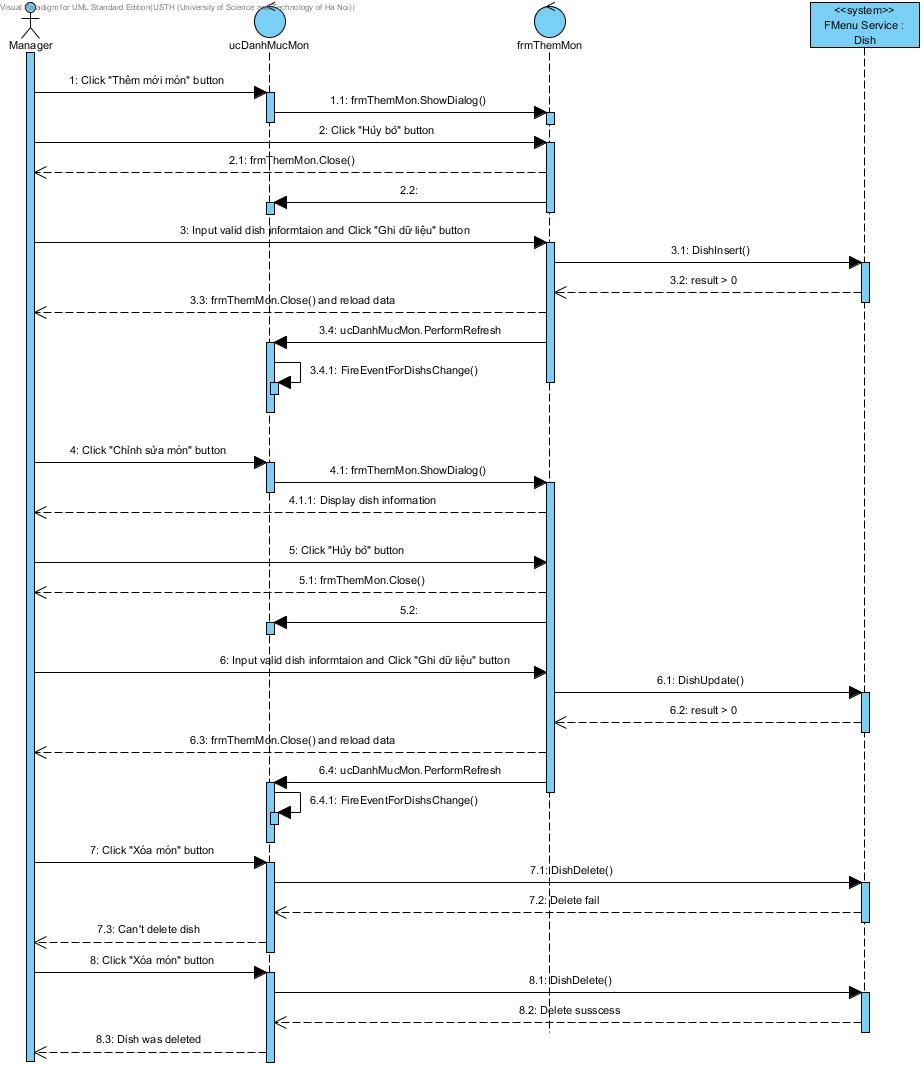


Figure D-IV-2-2: Add dish sequence diagram

##### Add Dish Material

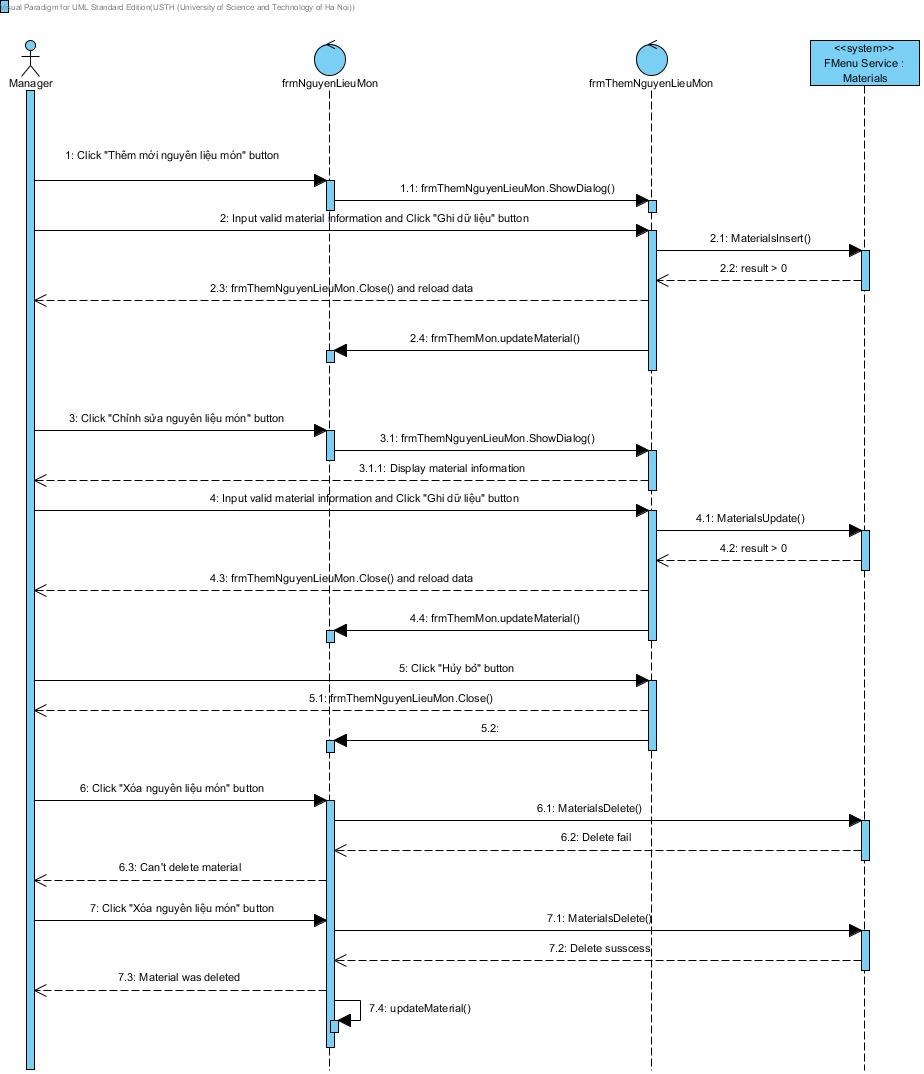


Figure D-IV-2-3: Add dish material sequence diagram

##### Dish Promotion

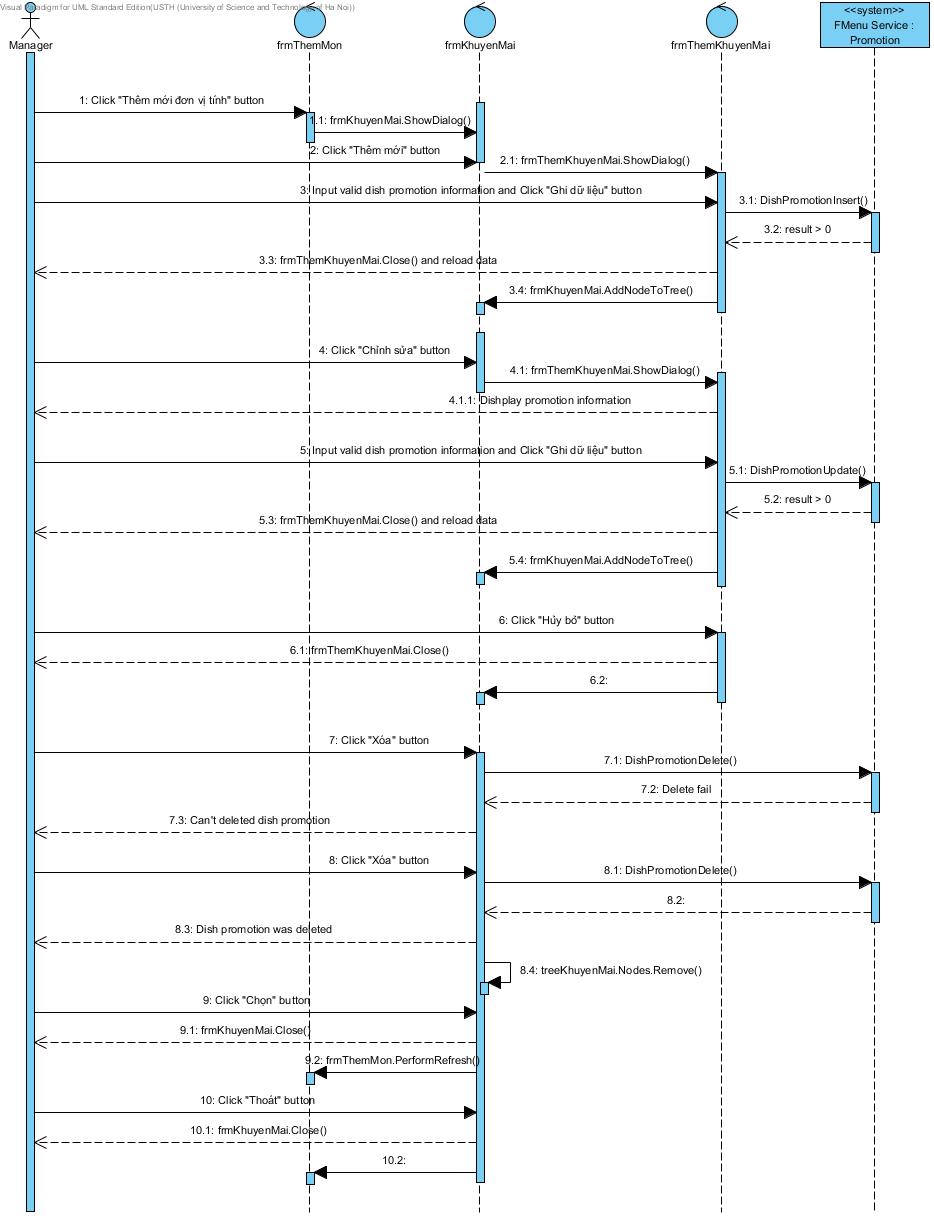


Figure D-IV-2-4: Dish promotion sequence diagram

##### Dish Unit

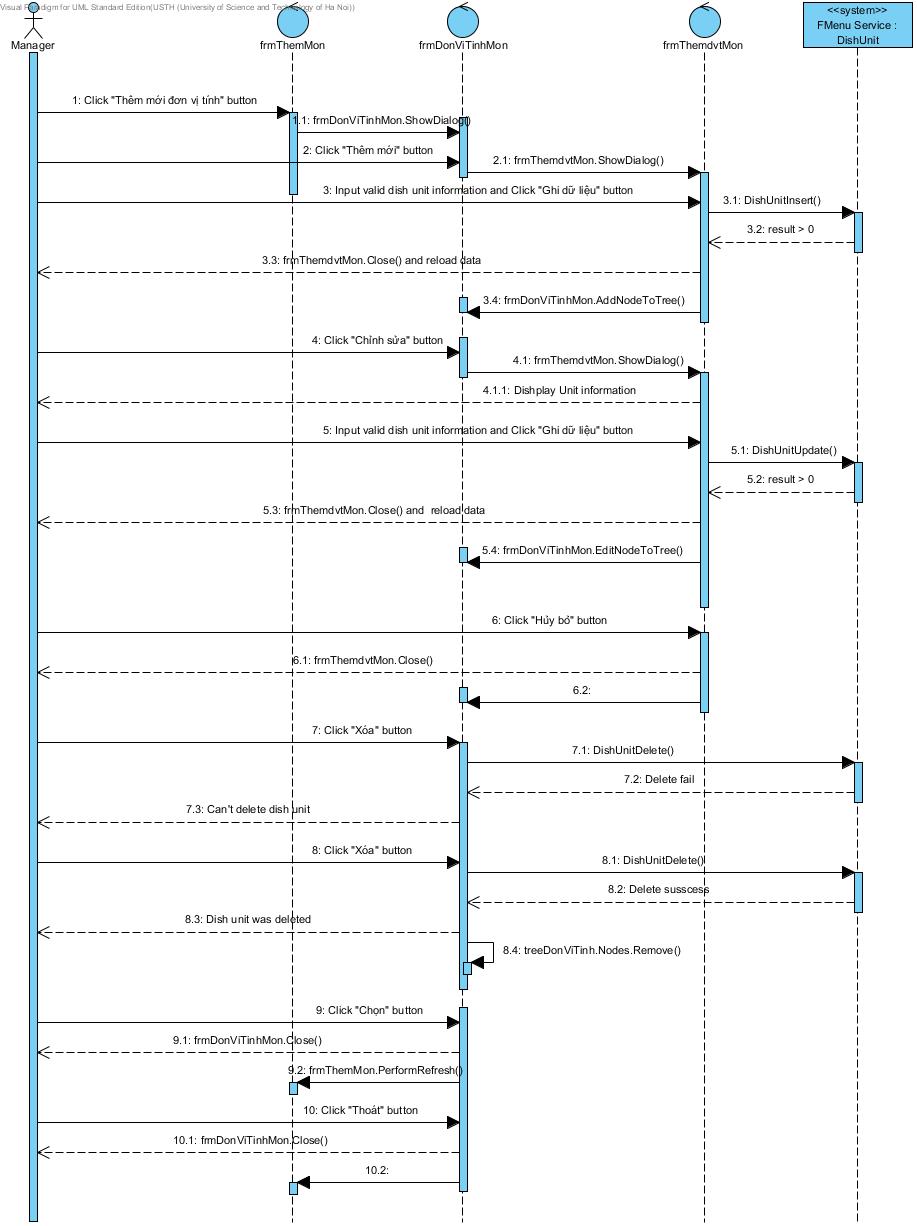


Figure D-IV-2-5: Dish unit sequence diagram

##### Dish VAT

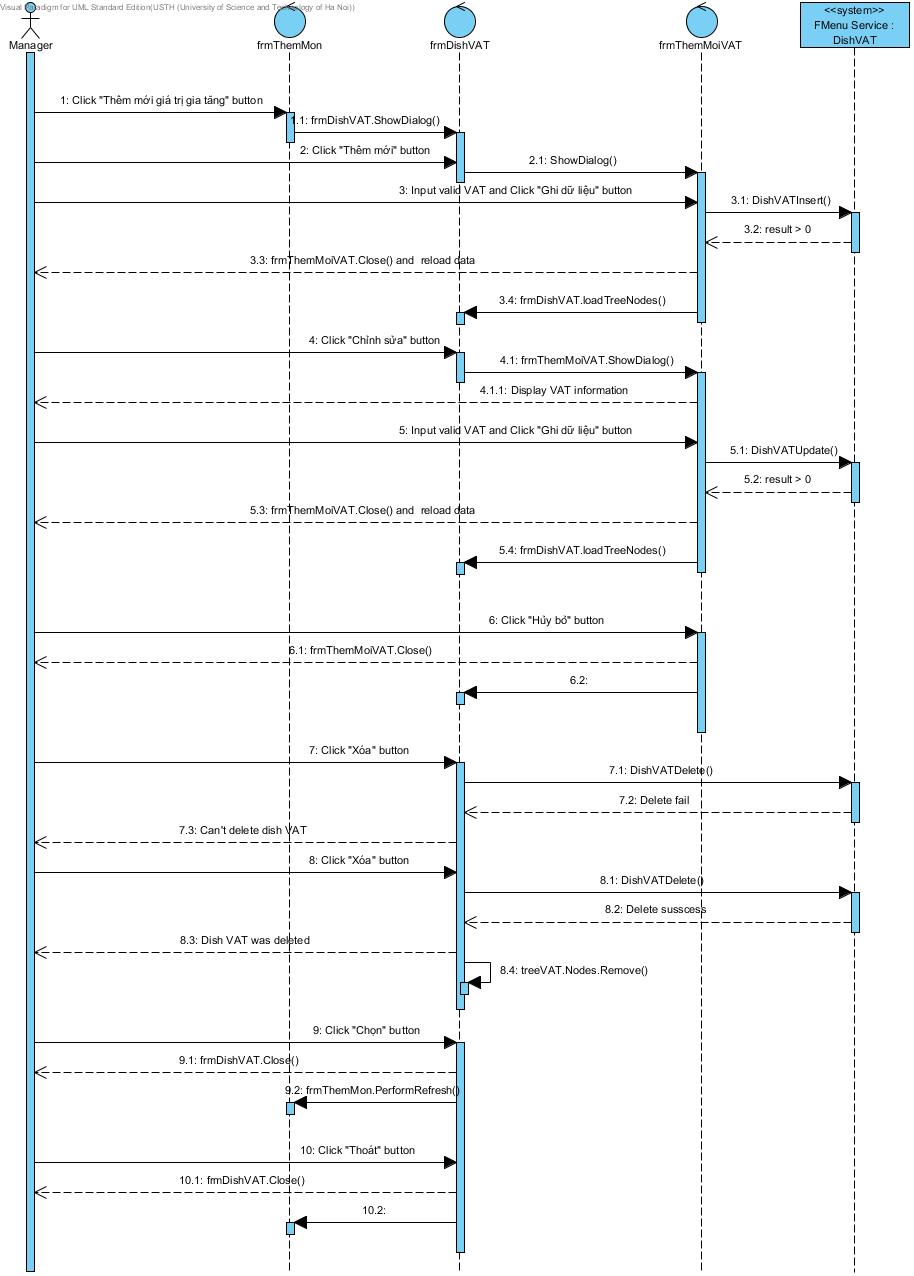


Figure D-IV-2-6: Dish VAT sequence diagram

##### Manage Dish Category

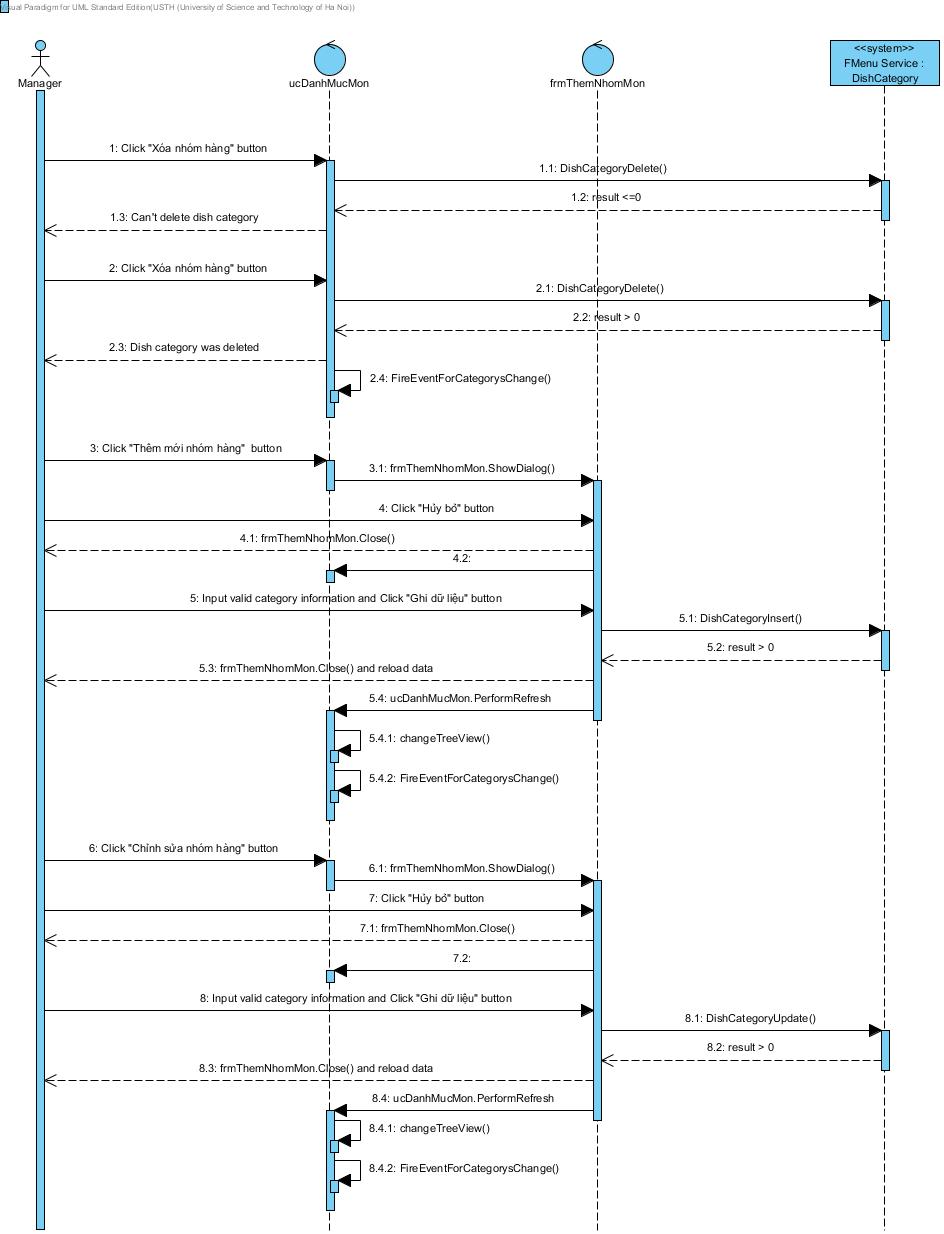


Figure D-IV-2-7: Manage Dish Category sequence diagram

##### Login

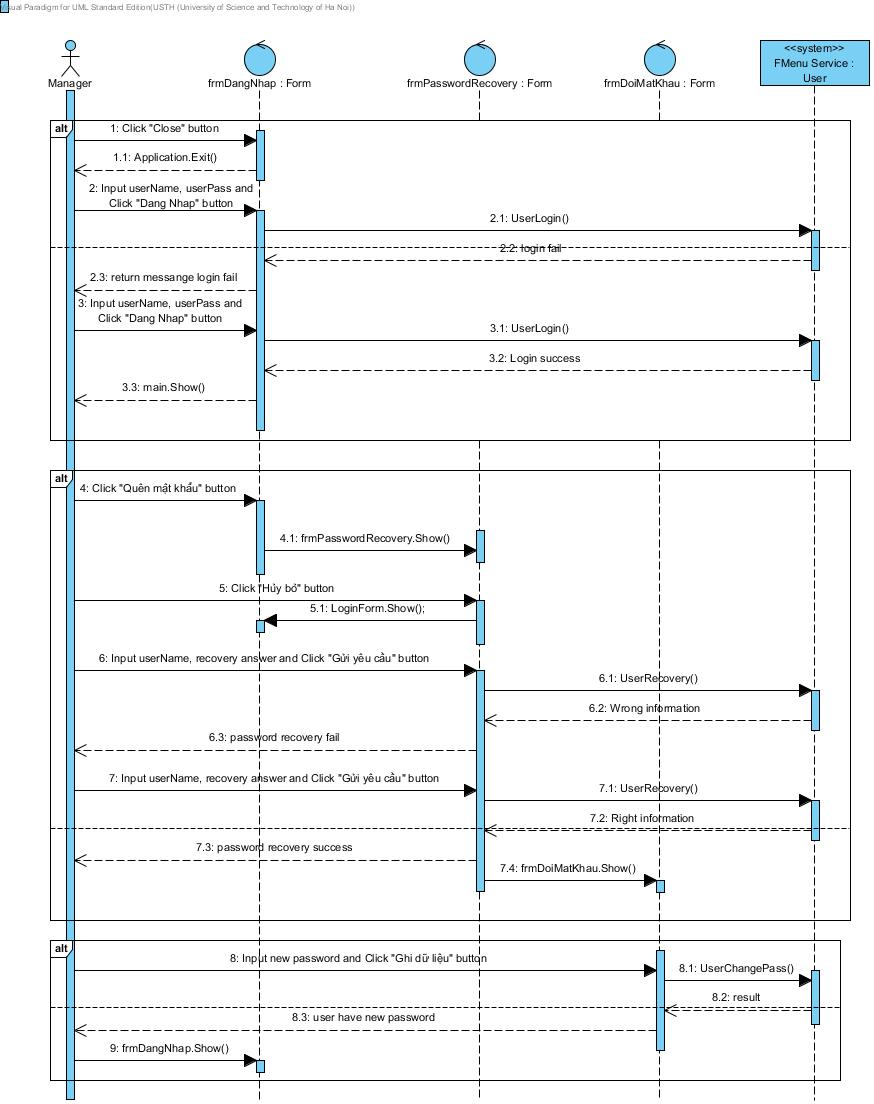


Figure D-IV-2-8: Login sequence diagram

##### Main menu

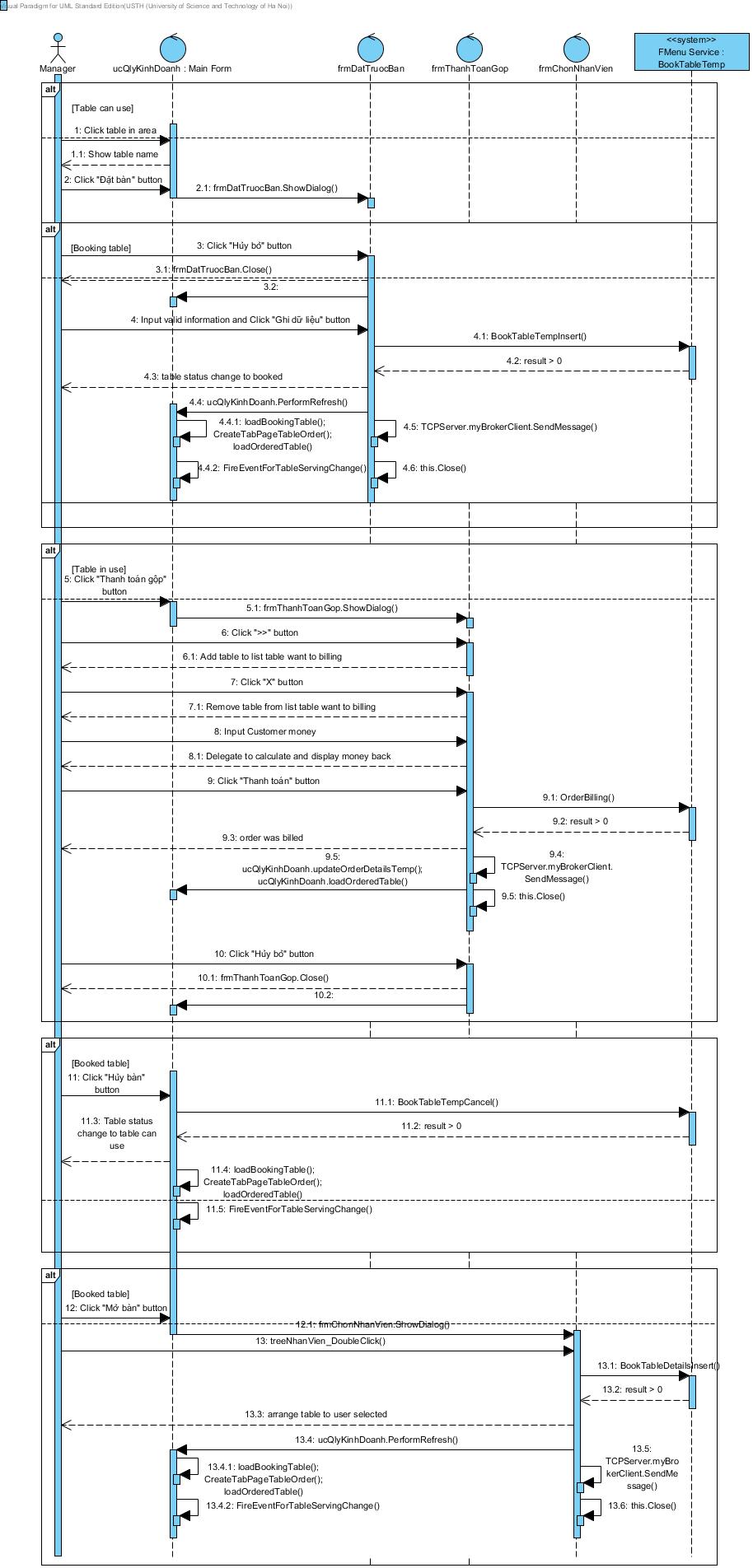


Figure D-IV-2-9: Main menu sequence diagram

##### Table Area

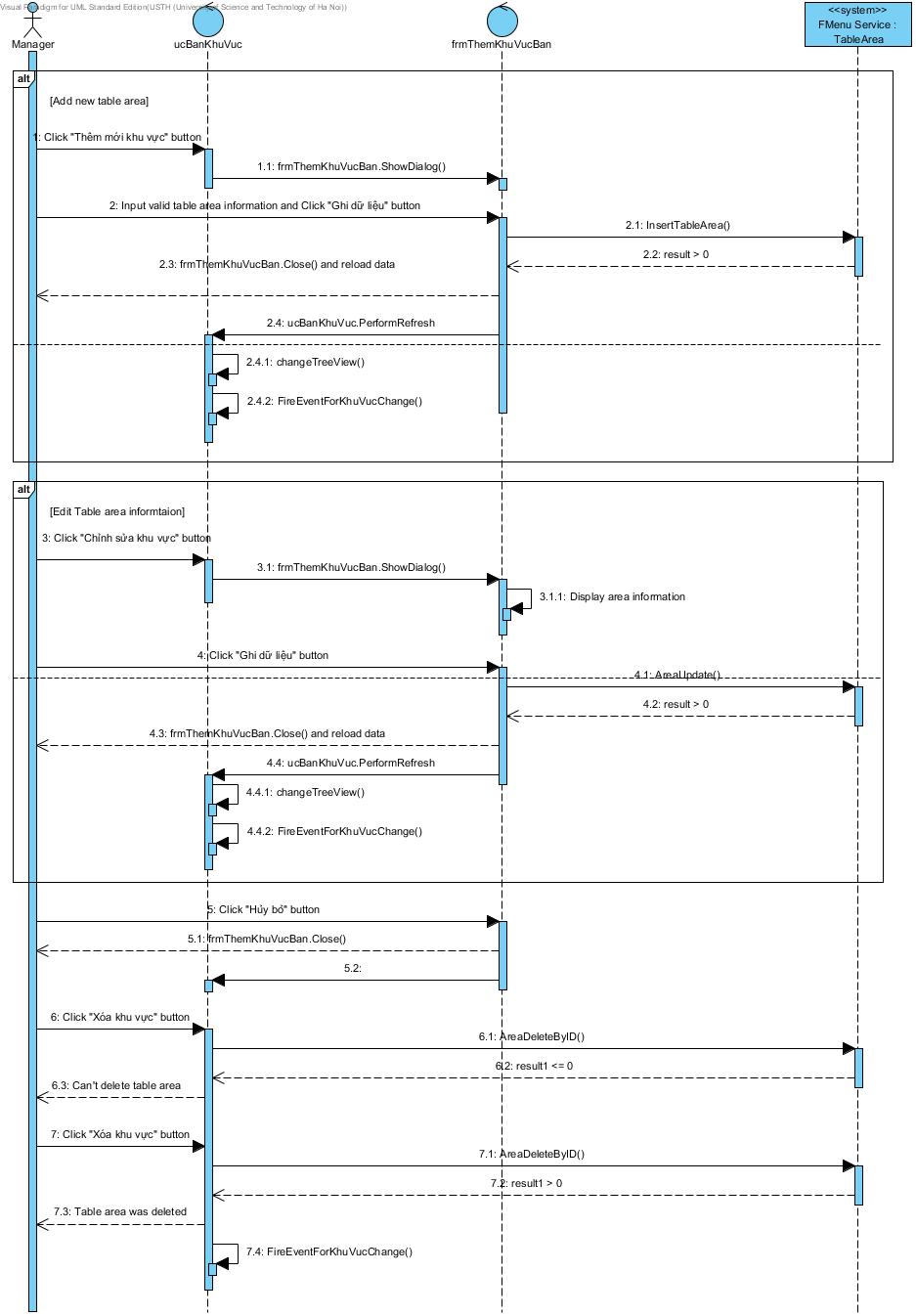


Figure D-IV-2-10: Table area sequence diagram

##### Table

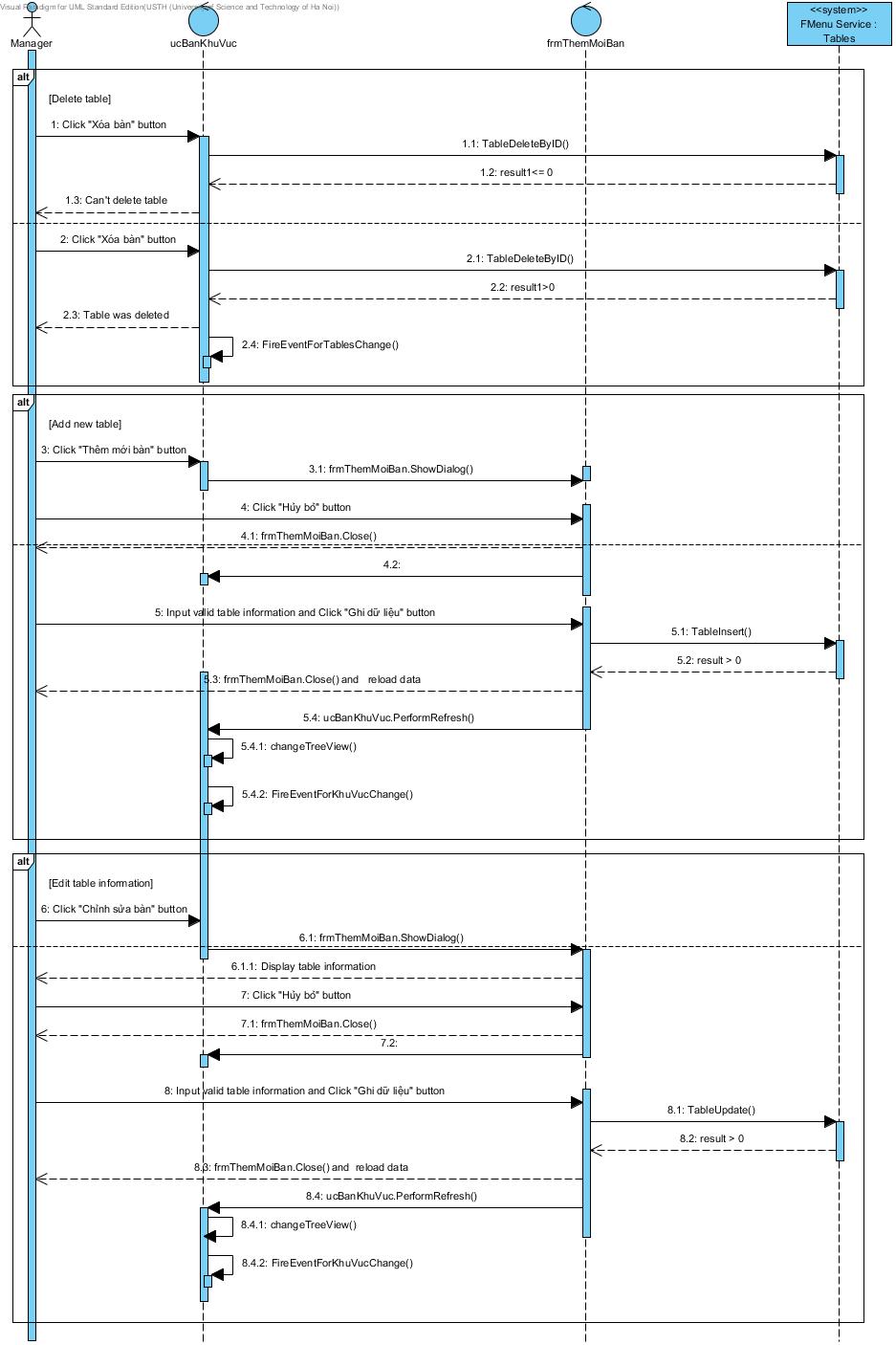


Figure D-IV-2-11: Table sequence diagram

### F-Menu Waiter

#### Class Diagram

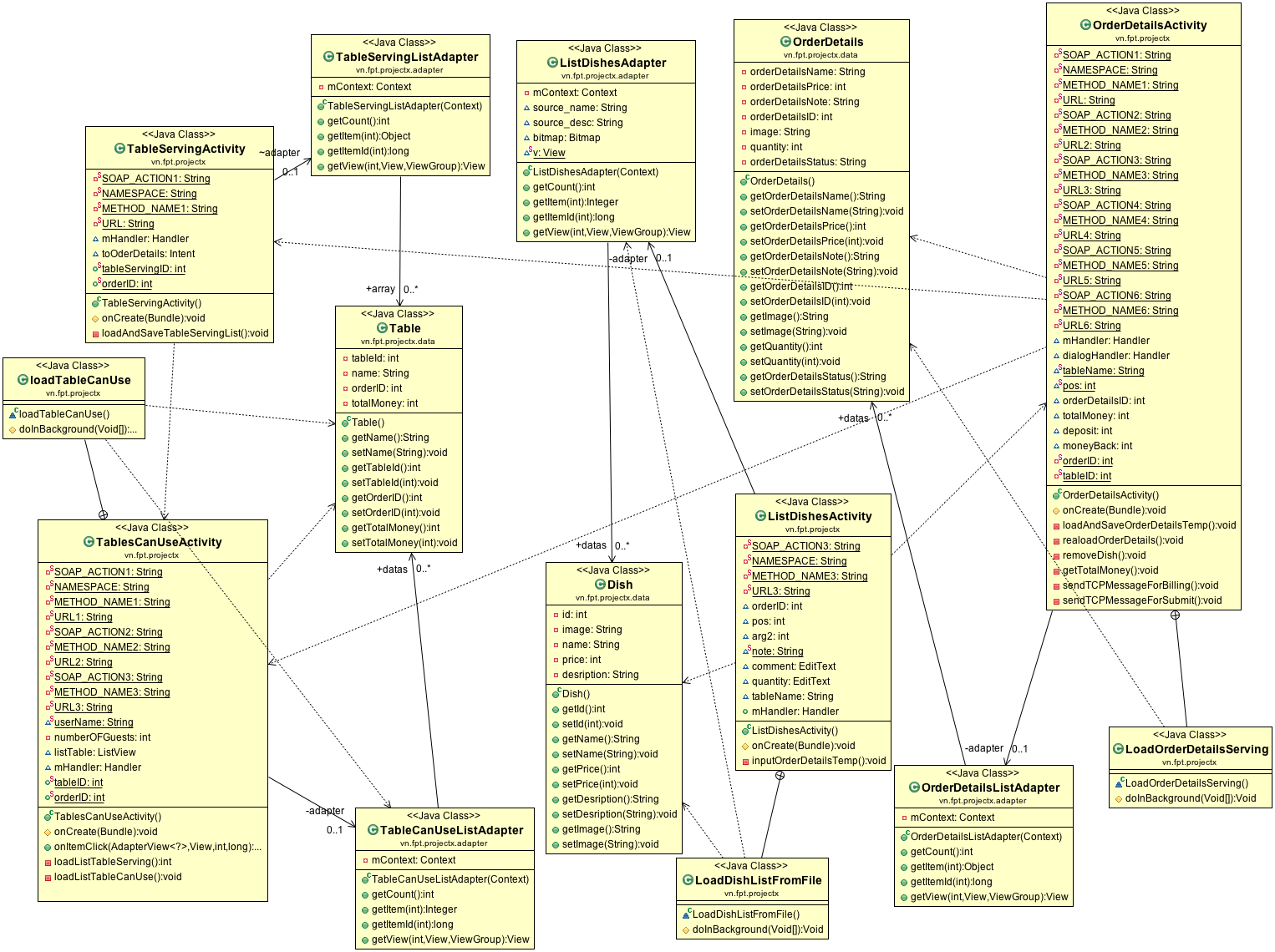


Figure D-IV-3-1: F-Menu Waiter Class Diagram

#### Class Description

##### Class TableCanUseActivity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **TableCanUseAcivity** | | | |
| **Description** | List of Table that waiter can select and use to create new Orders. | | | |
| **Base Class** | Activity | | | |
| **Constructor** | TableCanUseActivity() | | | |
| **Prototype** | Class tablescanuse\_layout.xml | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/TableCanUseActivity.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | |
| userName | String | Store string when user login | |
| numberOFGuests | Int | Store number of guest of a table | |
| listTable | ListView | Create a list view in view | |
| mHandler | Handler | A handler for execute some actions from another thread | |
| adapter | TableCanUseListAdapter | A adapter for listTable uses | |
| tableID | int | Store tableID use from TableCanUseActivity or TableServingActivity | |
| orderID | int | Store orderID use from TableCanUseActivity or TableServingActivity | |
| **Methods** | Name | Input | Output | Description |
| onCreate() | Bundle savedInstanceState | Void | Generate item forlist view and run loadListTable() |
| loadListTableServing |  | int | Load list table serving from Webservice. |
| onItemClick | AdapterView<?>adp, View v, final intpos,final long id | void | Action when user clicks each item in list view. |
| loadListTableCanUse |  | void | Load list table can use from webservice and save file local. |
| loadTableCanUse |  | AsyncTask<Void, Void, Void> | Thread for loading database from file saved. |

Table D-IV-3-1: Class TableCanUseActivity

##### Class Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Table** | | | |
| **Description** | Class to store value load from file to variables. | | | |
| **Base Class** | Object | | | |
| **Constructor** | Table() | | | |
| **Prototype** |  | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/Table.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | | |
| tableId | int | Store tableID from file saved. | | |
| name | String | Store name of each table from file saved | | |
| orderID | int | Store orderID for table selected from file saved. | | |
| totalMoney | int | Store total of money for table selected from file saved. | | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |

Table D-IV-3-2: Class Table

##### Class TableServingActivity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **TableServingActivity** | | | |
| **Description** | List table that a user serving a table. | | | |
| **Base Class** | Activity | | | |
| **Constructor** | TableServingActivity() | | | |
| **Prototype** | Class tableserving\_layout.xml | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/Table.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | | |
| mHandler | Handler | A handler for execute some actions from another thread | | |
| toOderDetails | Intent | Intent to pass value from TableServingActivity to OrerDetailsActivity. | | |
| tableServingID | int | Store tableServingID from file saved. | | |
| orderID | int | Store orderID from file saved. | | |
| **Methods** | Name | Input | Output | Description |
| onCreate() | Bundle savedInstanceState | Void | Generate item for list view and run loadAndSaveTableServingList. |
| loadAndSaveTableServingList | Menu menu | Void | Load list table serving from Webservice |
| LoadListTableServing |  | AsyncTask<Void, Void, Void> | Thread for loading list of table serving from file saved. |

Table D-IV-3-3: Class TableServingActivity

##### Class TableCanUseListAdapter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **TableCanUseListAdapter** | | | |
| **Description** | Adapter using for list view from TabeCanUseActivity. | | | |
| **Base Class** | BaseAdapter | | | |
| **Constructor** | TableCanUseListAdapter() | | | |
| **Prototype** | Class tablecanuse\_row.xml | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/TableCanUseAdapter.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | | |
| mContext | Context | Contain context of this class. | | |
| datas | ArrayList<Table> | Contain class Table in ArrayList. | | |
| **Methods** | Name | Input | Output | Description |
| getCount | void | int | Basic function of adapter |
| getItem | int | Object | Basic function of adapter |
| getItemID | int | long | Basic function of adapter |
| getView | int, View, ViewGroup | View | Basic function of adapter |

Table D-IV-3-4: Class TableCanUseListAdapter

##### Class TableServingListAdapter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **TableServingListAdapter** | | | |
| **Description** | Adapter using for list view from TabeServinActivity. | | | |
| **Base Class** | BaseAdapter | | | |
| **Constructor** | TableServingListAdapter() | | | |
| **Prototype** | Class tablecanuse\_row.xml | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/TableServingListAdapter.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | | |
| mContext | Context | Contain context of this class. | | |
| datas | ArrayList<Table> | Contain class Table in ArrayList. | | |
| **Methods** | Name | Input | Output | Description |
| getCount | void | int | Basic function of adapter |
| getItem | int | Object | Basic function of adapter |
| getItemID | int | long | Basic function of adapter |
| getView | int, View, ViewGroup | View | Basic function of adapter |

Table D-IV-3-4: Class TableServingListAdapter

##### Class ListDishesActivity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **ListDishesActivity** | | | |
| **Description** | List of Dishes in the Menu using. | | | |
| **Base Class** | Activity | | | |
| **Constructor** | ListDishesActivity() | | | |
| **Prototype** | Class orderlist\_layout.xml | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/ListDishesActivity.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | | |
| orderID | Int | Store orderID use from TableCanUseActivity or TableServingActivity | | |
| pos | Int | Store position when user click at each row. | | |
| note | String | Store string when user notes. | | |
| comment | EditText | Store comment when user notes in screen. | | |
| quantity | EditText | Store quantity of dish when user input in screen. | | |
| tableName | String | Store table name. | | |
| mHandler | Handler | A handler for execute some actions from another thread. | | |
| adapter | ListDishesAdapter | A adapter for list dishes uses. | | |
| **Methods** | Name | Input | Output | Description |
| onCreate() | Bundle savedInstanceState | Void | Generate item for list view and run LoadDishListFromFile. |
| LoadDishListFromFile |  | AsyncTask<Void, Void, Void> | Thread for loading database from file saved. |
| inputOrderDetailsTemp |  | void | send request on webservice with parameter to create new order detail. |

Table D-IV-3-5: Class ListDishesActivity

##### Class Dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Dish** | | | |
| **Description** | Class to store value load from file to variables. | | | |
| **Base Class** | Object | | | |
| **Constructor** |  | | | |
| **Prototype** |  | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/Dish.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | | |
| id | String | Store Dish’s ID | | |
| name | String | Store Dish’s name | | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |

Table D-IV-3-6: Class Dish

##### Class OrderDetailsActivity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **OrderDetailsActivity** | | | |
| **Description** | List of order details in tabler serving. | | | |
| **Base Class** | Activity | | | |
| **Constructor** | OrderDetailsActivity() | | | |
| **Prototype** | Class order\_details\_temp\_layout.xml | | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/OrderDetailsActivity.java | | | |
| **Namespace** | vn.fpt.projectx; | | | |
| **Attributes** | Name | Type | Description | | |
| mHandler | Handler | A handler for execute some actions from another thread | | |
| tableName | String | Store table name TableServingActivity | | |
| pos | int | Store position when user clicks at each row. | | |
| orderDetailsID | int | Store order details ID of a order details. | | |
| totalMoney | int |  | | |
| deposit | int |  | | |
| orderID | int | Store tableID use from TableCanUseActivity or TableServingActivity | | |
| tableID | int | Store orderID use from TableCanUseActivity or TableServingActivity. | | |
| adapter | OrderDetailsListAdapter | An adapter for list of order details uses. | | |
| **Methods** | Name | Input | Output | Description |
| onCreate() | Bundle savedInstanceState | Void | Generate item for list view and run LoadOrderDetailsServing. |
| loadAndSaveOrderDetailsTemp |  | void | Load list order details from Webservice and save file. |
| realoadOrderDetails | Void | Void | Reload list order details from Webservice and save file, and read file to fill into adapter. |
| removeDish |  | Void | Remove dish when click remove an order detail. |
| getTotalMoney |  | Void | Get total money for that table. |
| LoadOrderDetailsServing |  | AsyncTask<Void, Void, Void> | Thread for loading database from file saved. |

Table D-IV-3-7: Class OrderDetailsActivity

##### Class OrderDetailsListAdapter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **OrderDetailsListAdapter** | | | |
| **Description** | Adapter using for list view from OrderDetailsActivity. | | | |
| **Base Class** | BaseAdapter | | | |
| **Constructor** | OrderDetailsListAdapter() | | | |
| **Prototype** |  | | | |
| **Source File** | Food Advisor\src\com\rule\ UserCaculate.java | | | |
| **Namespace** | rule | | | |
| **Attributes** | Name | Type | Description | | |
| mContext | Context | Contain context of this class. | | |
| datas | ArrayList<OrderDetail> | Contain class OrderDetailinArrayList. | | |
| **Methods** | Name | Input | Output | Description |
| getCount | void | int | Basic function of adapter |
| getItem | int | Object | Basic function of adapter |
| getItemID | int | long | Basic function of adapter |
| getView | int, View, ViewGroup | View | Basic function of adapter |

Table D-IV-3-8: Class OrderDetailsListAdapter

##### Class OrderDetails

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **OrderDetails** | | |
| **Description** | Class to store value load from file to variables. | | |
| **Base Class** | Object | | |
| **Constructor** | OrderDetails() | | |
| **Prototype** |  | | |
| **Source File** | Fmenu\_waiter/src/vn/fpt/projectx/OrderDetails.java | | |
| **Namespace** | vn.fpt.projectx; | | |
| **Attributes** | Name | Type | Description | |
| orderDetailsName | String | Store OrderDetails’sname | |
| orderDetailsPrice | int | Store OrderDetails’sprice | |
| orderDetailsNote | String | Store OrderDetails’snote | |
| orderDetailsID | int | Store OrderDetails’s Id. | |
| image | String | Store OrderDetails’simage in based 64. | |
| quantity | Int | Store OrderDetails’squatity. | |
| orderDetailsStatus | String | Store OrderDetails’sstatus. | |

Table D-IV-3-9: Class OrderDetails

#### Sequence Diagram

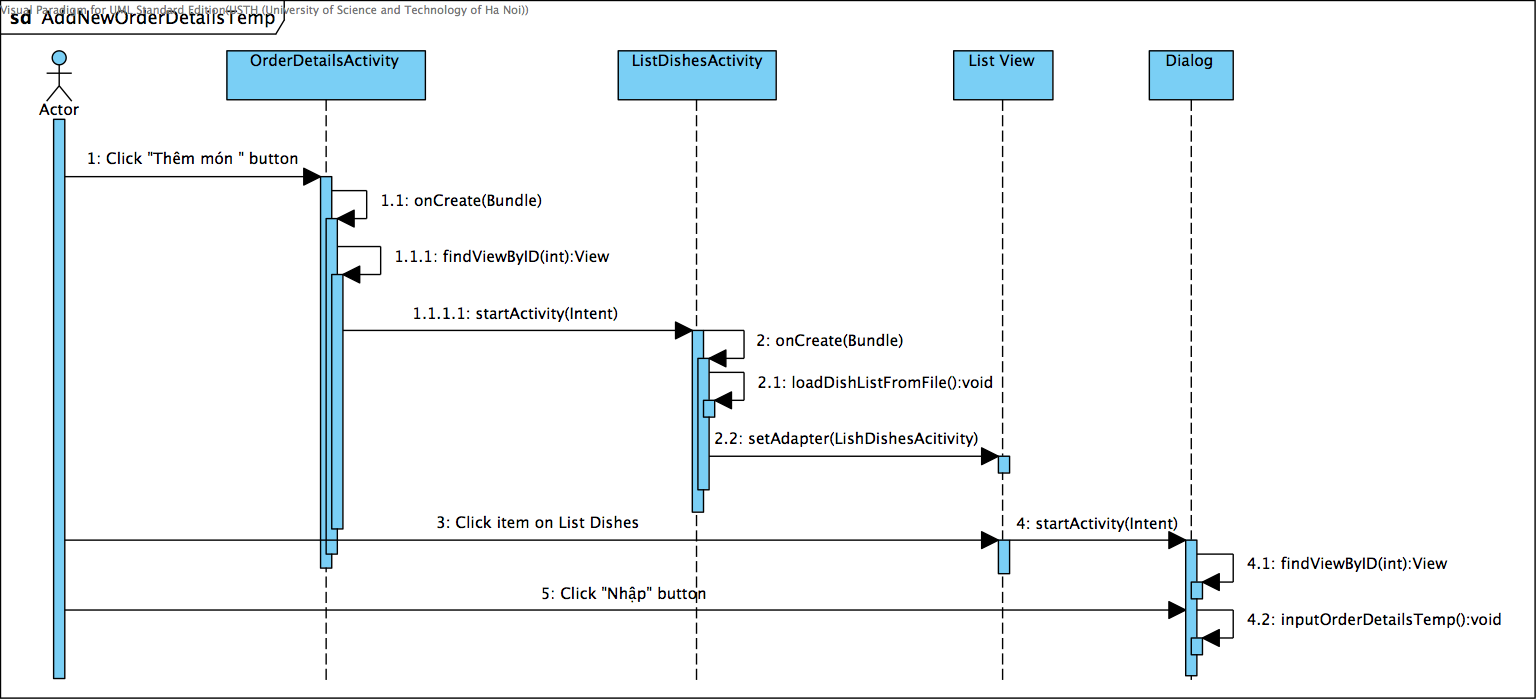


Figure D-IV-3-2: AddNewOrderDetailsTemp Sequence Diagram

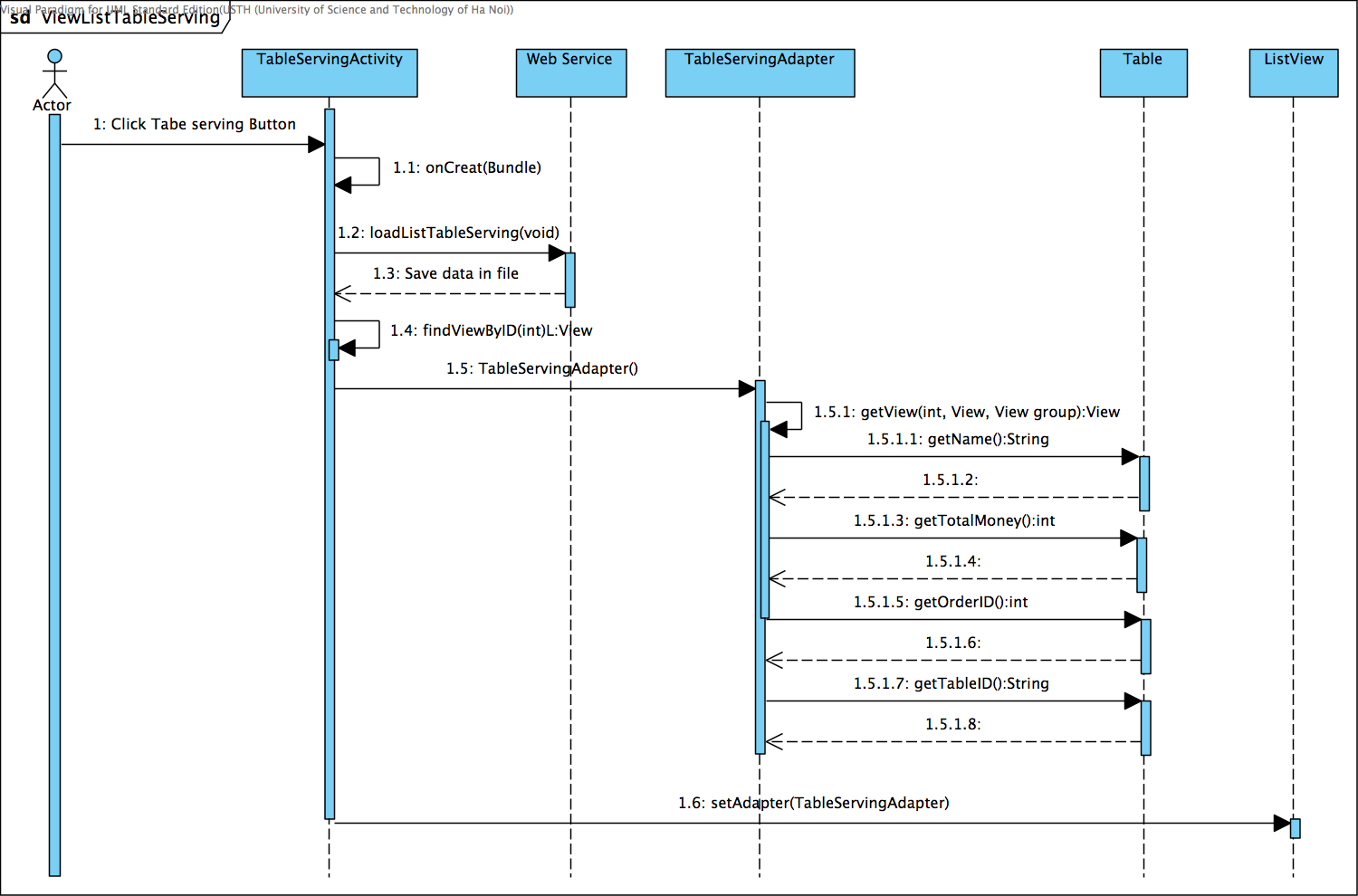


Figure D-IV-3-3: ViewListTableServing Sequenc Diagram

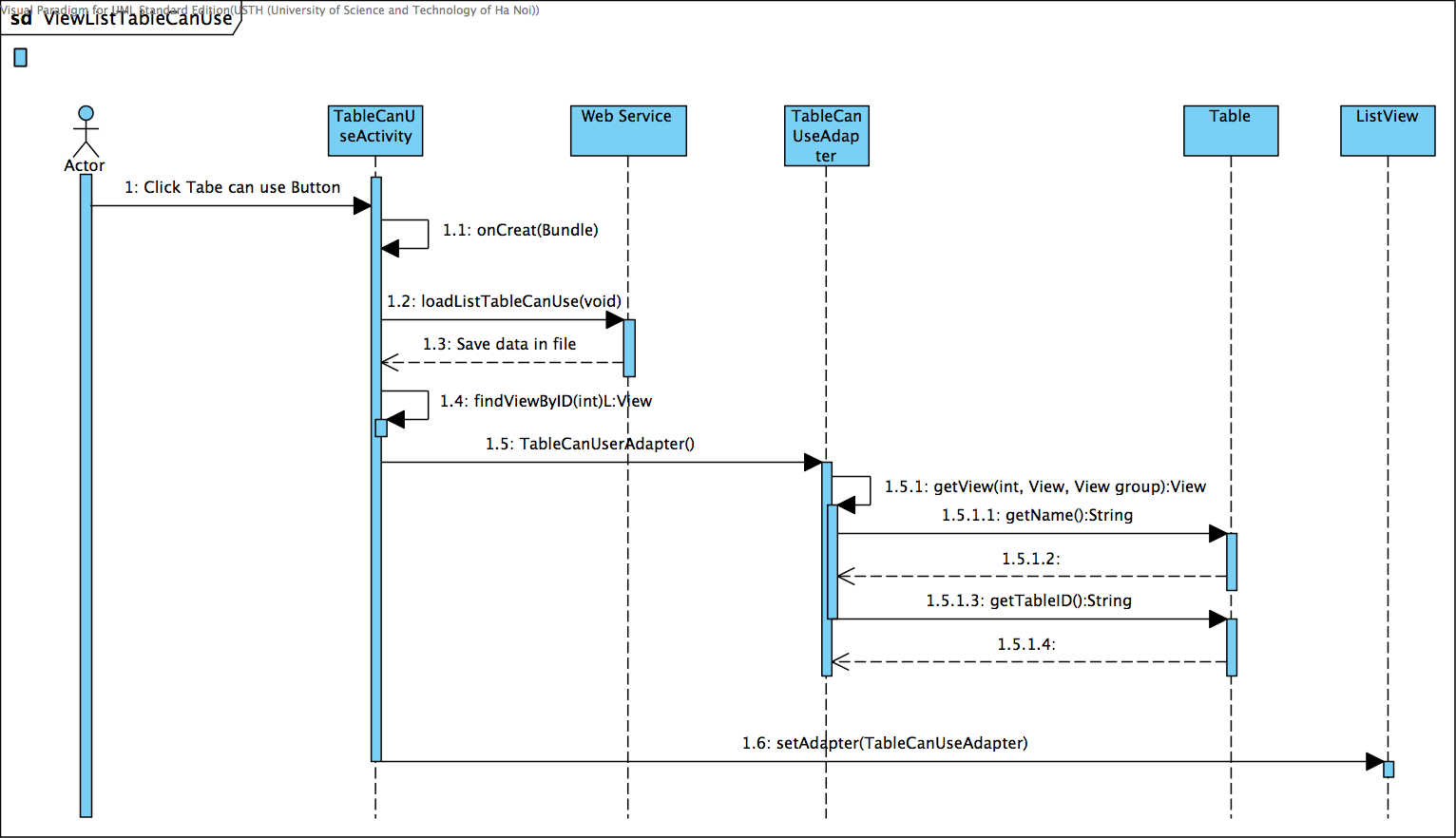


Figure D-IV-3-4: ViewListTableCanUse Sequence Diagram

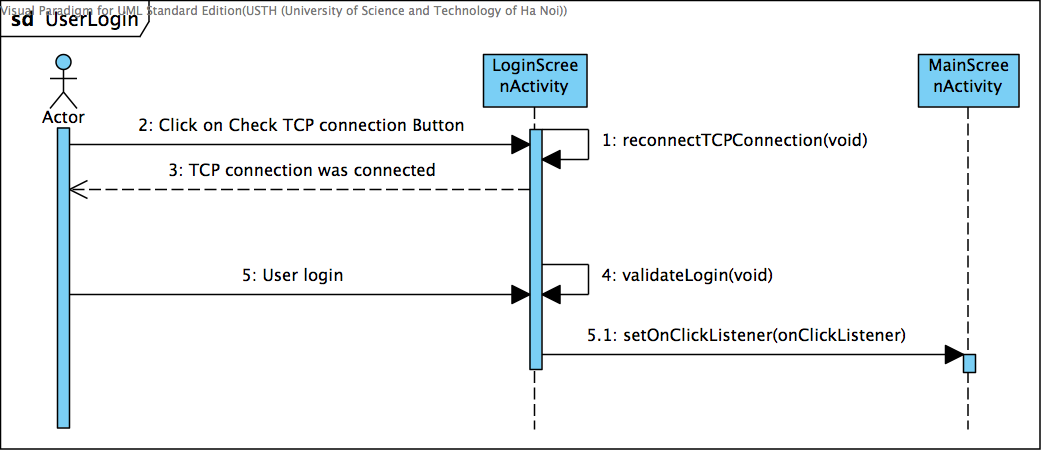


Figure D-IV-3-5: UserLogin Sequence Diagram

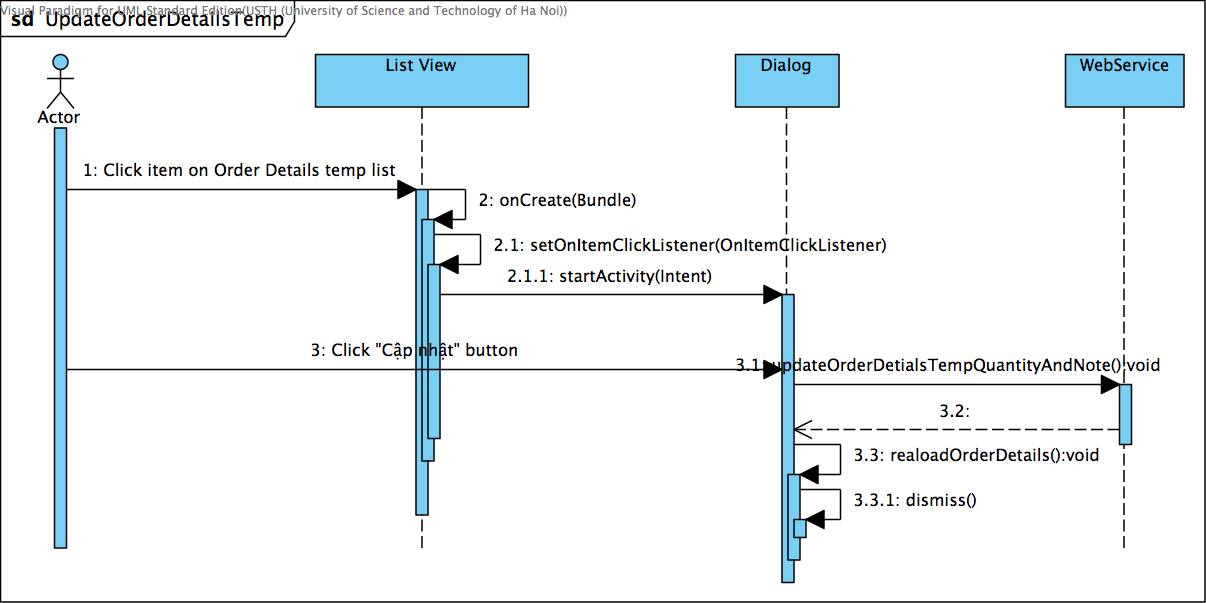


Figure D-IV-3-6: UpdateOrderDetailsTemp Sequence Diagram

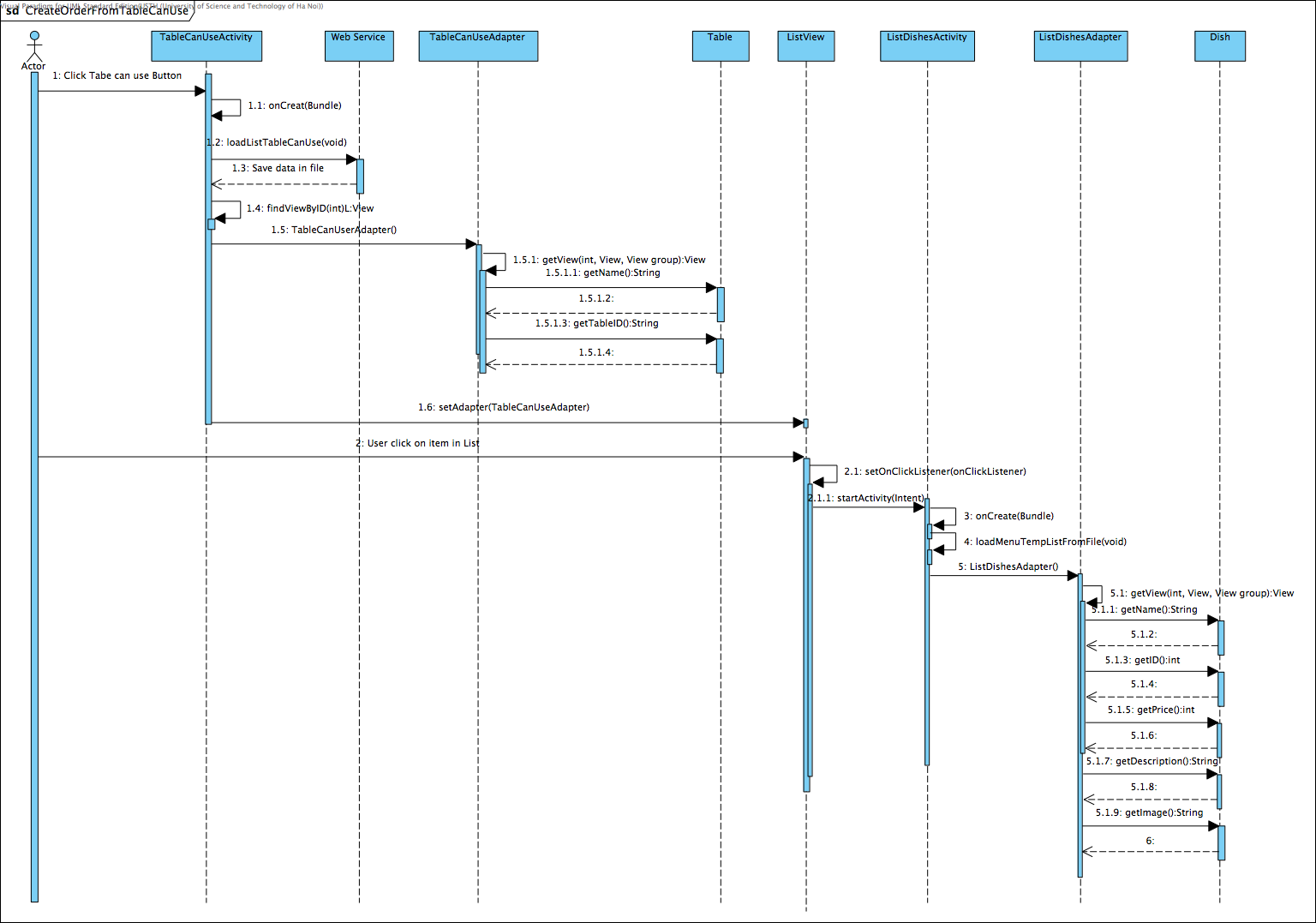


Figure D-IV-3-7: CreateOrderFromTableCanUse Sequence Diagram

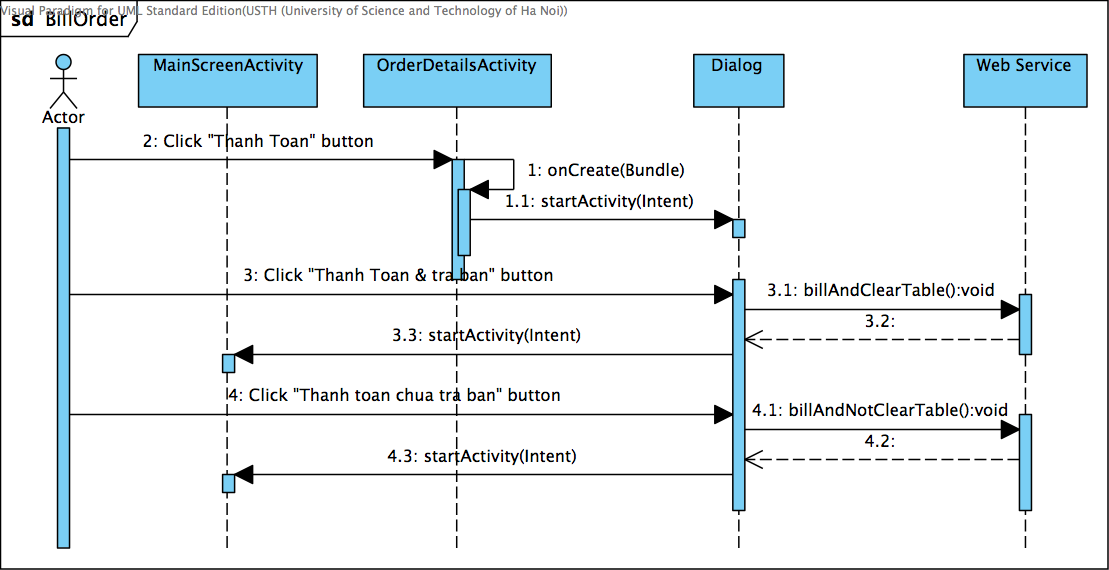


Figure D-IV-3-8: BillOrder Sequence Diagram

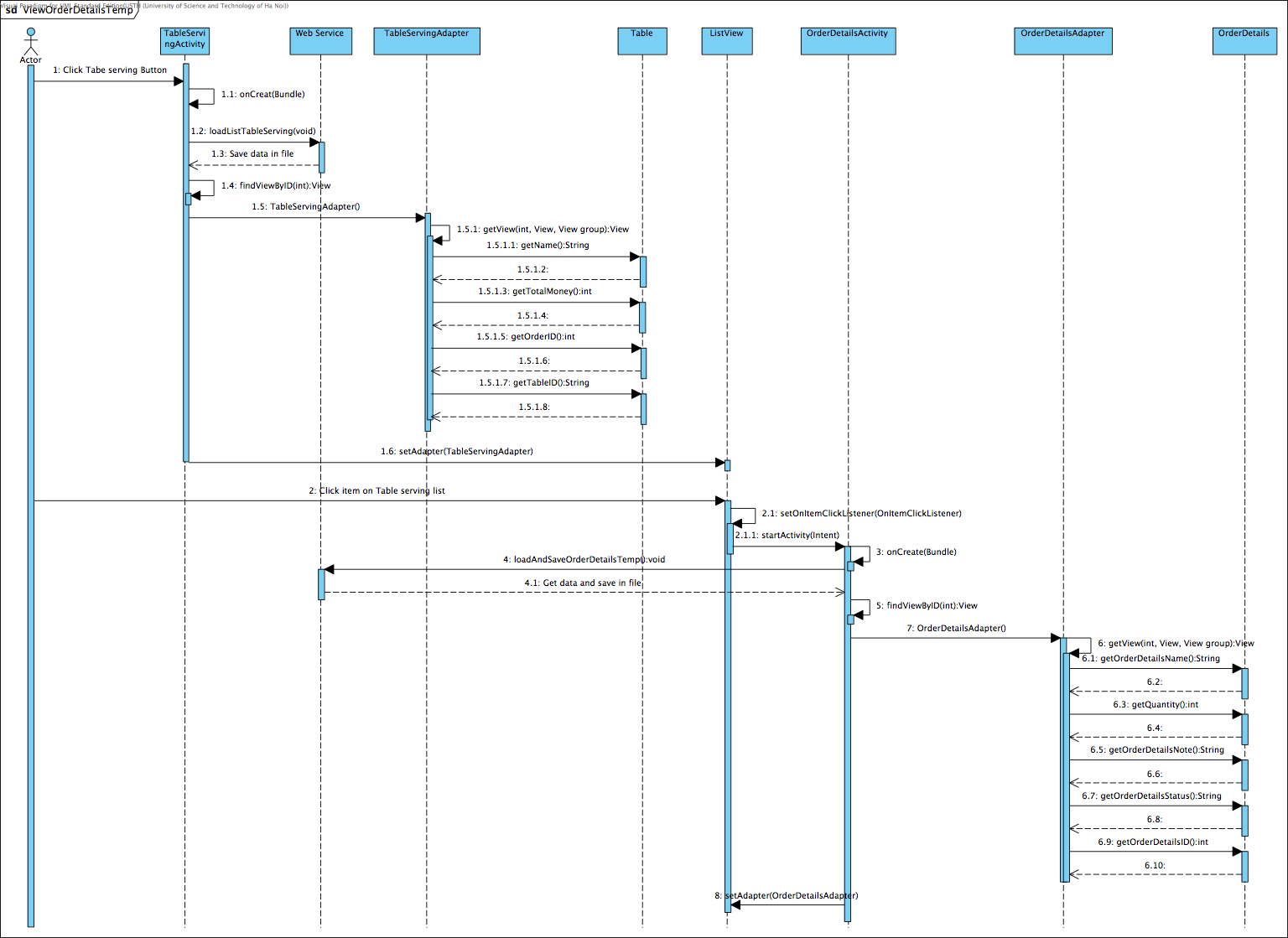


Figure D-IV-3-9: ViewOrderDetailsTemp Sequence Diagram

### F-Menu Chef

#### Class Diagram

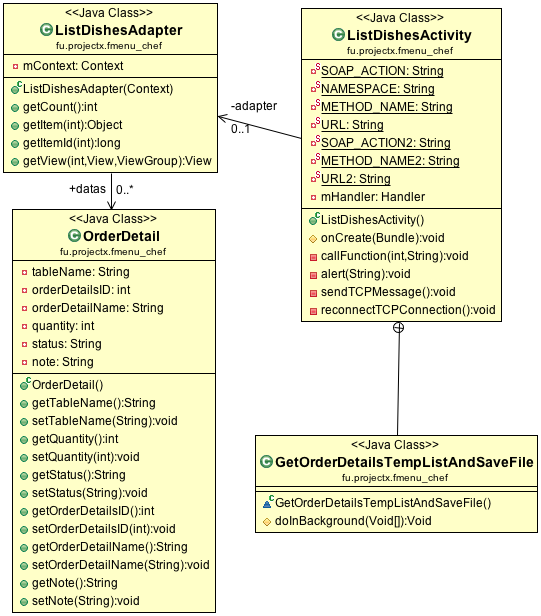


Figure D-IV-4-1: F-Menu Chef Class Diagram

#### Class Description

##### Class ListDishesActivity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **LishDishesAcivity** | | | |
| **Description** | List of Dishes that chef can handle status of each order details. | | | |
| **Base Class** | Activity | | | |
| **Constructor** | ListDishesActivity() | | | |
| **Prototype** | Class lish\_dish.xml | | | |
| **Source File** | Fmenu\_Chef/src/fu/projectx/fmenu\_chef/ListDishesActivity.java | | | |
| **Namespace** | fu.projectx.fmenu\_chef; | | | |
| **Attributes** | Name | Type | Description | |
| mHandler | Handler | A handler for execute some actions from another thread | |
| adapter | ListDishesAdapter | A adapter for ListDishesActivity uses | |
| **Methods** | Name | Input | Output | Description |
| onCreate() | Bundle savedInstanceState | Void | Generate item forlist view and run GetOrderDetailsTempListAndSaveFile (). |
| callFunction | **int**orderDetailsID, String message | callFunction | Load function from webservice when user presses specify button. |
| alert | void | String | Alert with dialog in specify situation. |
| sendTCPMessage |  | void | Send TCP message to winform or Client 1. |
| reconnectTCPConnection |  | Void | Reconnect TCP connection when it was disconnected. |
| GetOrderDetailsTempListAndSaveFile |  | AsyncTask<Void, Void, Void> | Thread for loading list of table serving from file saved. |

Table F-IV-4-1: Class ListDishesActivity

##### Class OrderDetail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **OrderDetail** | | | |
| **Description** | Class to store value load from file to variables. | | | |
| **Base Class** | Object | | | |
| **Constructor** | OrderDetail() | | | |
| **Prototype** |  | | | |
| **Source File** | Fmenu\_Chef/src/fu/projectx/fmenu\_chef/OrderDetail.java | | | |
| **Namespace** | fu.projectx.fmenu\_chef; | | | |
| **Attributes** | Name | Type | Description | | |
| tableName | String | Store table’s name. | | |
| orderDetailsID | int | Store order details’s ID. | | |
| orderDetailName | String | Store order detail’s name. | | |
| quantity | int | Store order detail’s quantity. | | |
| status | String | Store order detail’s status | | |
| note | String | Store order detail’s note | | |
| **Methods** | Name | Input | Output | Description |
|  |  |  |  |

Table F-IV-4-2: Class OrderDetail

##### Class ListDishesAdapter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **ListDishesAdapter** | | | |
| **Description** | Adapter using for list view from List dishes. | | | |
| **Base Class** | BaseAdapter | | | |
| **Constructor** | ListDishesAdapter() | | | |
| **Prototype** | Class row.xml | | | |
| **Source File** | Fmenu\_Chef/src/fu/projectx/fmenu\_chef/ListDishesAdapter.java | | | |
| **Namespace** | fu.projectx.fmenu\_chef; | | | |
| **Attributes** | Name | Type | Description | | |
| mContext | Context | Contain context of this class. | | |
| datas | ArrayList<OrderDetail> | Contain class OrderDetail in ArrayList. | | |
| **Methods** | Name | Input | Output | Description |
| getCount | void | int | Basic function of adapter |
| getItem | int | Object | Basic function of adapter |
| getItemID | int | long | Basic function of adapter |
| getView | int, View, ViewGroup | View | Basic function of adapter |

Table F-IV-4-3: Class ListDishesAdapter

#### Sequence Diagram

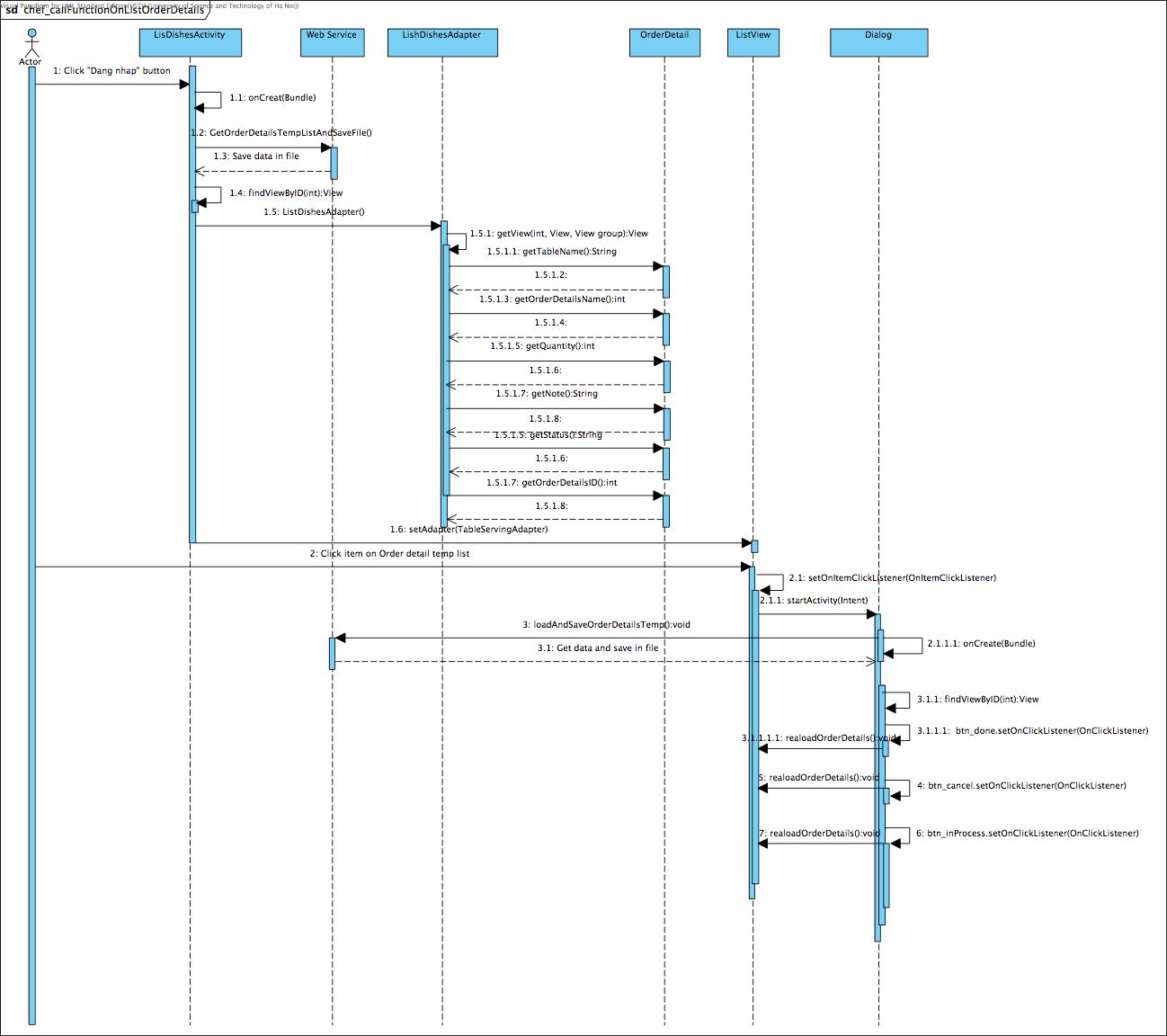


Figure D-IV-4-2: Chef main sequence diagram

# TEST DOCUMENTATION

## Introduction

### Purpose

The purpose of this document is to plan all the activities needed to perform testing for the application.

This document also defines scope of testing, requirement for testing and constraints…

### Test Approach

In the testing process of F-Menu, we all use end-user perspective when using the software. Each component is tested by running every use case of the system manually, to ensure each function performs correctly in the right bound of each object.

Most of the test cases are created based on system architecture design and system requirements. Some test cases are created based o user perspective to ensure that the system reach highest quality possible.

The test cases is created by test leader of the project, and reviewed by project manager. Because the time for testing is not enough to finish all the test cases, only some of the test cases are done.

## Test Plan

### Features to be tested

Here is the list of functions of the system that have to be tested in testing scope:

#### F-Menu Server

Including:

* Log in / out function
* Managing menu function
* Account management function
* Financial report function
* Table and area management function
* Menu and dishes management function
* Work management function
* Booking function
* Password recovery function

#### F-Menu Waiter

Including:

* Table arrangement function
* Creating orders function
* Managing orders function
* Billing function
* View paid cash function
* Log in/out function

#### F-Menu Chef

Including:

* Sending notifications function
* Requesting materials function

### Features not to be tested

Here is the list of the features that do not have to be tested in testing scope.

#### Performance

We are trying to ensure the system functions operates correctly first. The performance is not considered in this case.

#### Network Connection

In this test process, all of the test cases assume that the clients have stable LAN connection to the server over Wi-Fi. Only some specific test cases, the network connection is as mentioned.

## Testing tool and Environment

### Testing tool

* Visual Studio 2010
* Android Development Toolkit

We do not use any other special testing tool.

### Environment

* Windows XP Professional SP3 x86
* Windows 7 Ultimate x86 SP1
* Windows 7 Ultimate x64 SP1
* Windows 8 Professional x64
* Android 2.2 (Ice Cream Sandwich)

## Test cases

### Test Cases List

#### Management Console

##### Log in / log out function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Sign In** | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
| MC1.1 | Sign In successfully | **Valid account**  1. Input administrator username and password correctly  2. Click [Đăng nhập] button | 2. Main windows opened | Pass | ThanhHX |  |  |
| MC1.2 | Sign In fail | **Blank account**  1. Do not input username and password  2. Click [Đăng nhập] button | 1. Failed login notification appeared, main form is not opened |  |  |  |  |
| MC1.3 | **Blank username**  1. Input password, do not input username  2. Click [Đăng nhập] button |  |  |  |  |
| MC1.4 | **Blank password**  1. Input username, do not input password  2. Click [Đăng nhập] button |  |  |  |  |
| MC1.5 | **Invalid username**  1. Input wrong user name and password  2. Click [Đăng nhập] button |  |  |  |  |
| MC1.6 | **Invalid password**  1. Input correct username and wrong password  2. Click [Đăng nhập] button |  |  |  |  |
| MC1.7 | **Invalid authorization**  1. Input username and password of a non-administrator account  2. Click [Đăng nhập] button |  |  |  |  |
| MC1.8 | Cancel sign in | 1. Click close button | 1. F-Menu is closed |  |  |  |  |

#### Menu and dishes management function

##### Dish management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Dish management function** | | | | | | | |
| Step:   * Open Management Console * Input correct administrator username and password, main windows opened * Open [Danh mục món] | | | | | | | |
| MC2.1 | Menu open | **Dish management open correctly**  1. Click [Danh mục món] on [Quản lý] menu  2. Click [Danh mục món] on menu bar  3. Click [Danh mục món] close button  4. Click [Danh mục món] on menu bar again | 1. [Danh mục món] control is opened, list of dish and dish category is displayed correctly.  2. [Danh mục món] control is not double opened  3. [Danh mục món] control is closed  4. [Danh mục món] control is opened | Pass | ThanhHX |  |  |
| MC2.2 | Add dish category | **Add category successfully**  1. Click [Thêm mới nhóm hàng] button.  2. Input new dish category without [Ghi chú]  3. Click [Ghi dữ liệu]  4. Click new dish category | 1. [Thêm mới nhóm món] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới nhóm món] window closed, new dish category is displayed in the list with sort  4. No dish is shown on new category | Pass | ThanhHX |  |  |
| MC2.3 | **Add category successfully with description**  1. Select [Thêm mới] from context menu of dish category  2. Input new dish category with [Ghi chú]  3. Click [Ghi dữ liệu]  4. Click new dish category | 1. [Thêm mới nhóm món] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới nhóm món] window closed, new dish category with “Ghi chú” is displayed in the list  4. No dish is shown on new category | Pass | ThanhHX |  |  |
| MC2.4 | **Add maximum length category name**  1. Click [Thêm mới nhóm hàng] button.  2. Input new dish category with  127 character-length on name  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Thêm mới nhóm món] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. Cannot add any character more  4. [Thêm mới nhóm món] window closed, new dish category is displayed in the list | Pass | ThanhHX |  |  |
| MC2.5 | **Add maximum length category description**  1. Click [Thêm mới nhóm hàng] button.  2. Input new dish category with  1024 character-length on description  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Thêm mới nhóm món] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. Cannot add any character more  4. [Thêm mới nhóm món] window closed, new dish category is displayed in the list | Failed | ThanhHX |  |  |
| MC2.6 | **Add categoryfailed**  1. Click [Thêm mới nhóm hàng] button.  2. Input a current existing category  3. Add spaces before and after category name | 1. [Thêm mới nhóm món] window opened. [Ghi dữ liệu] button is disabled.  2. [Giá trị đã tồn tại] warning is displayed.  3. [Giá trị đã tồn tại] warning is displayed. |  |  |  |  |
| MC2.7 | **Add invalid category**  1. Click [Thêm mới nhóm hàng] button.  2. Input new dish category with these characters on name: \ / : \* " <> |  3. Clear category name field | 1. [Thêm mới nhóm món] window opened. [Ghi dữ liệu] button is disabled.  2. [Chứa kí hiệu không hợp lệ] warning is displayed. [Ghi dữ liệu] button is disabled.  3. [Ghi dữ liệu] button is disabled. | Failed |  |  |  |
| MC2.8 | **Cancel adding dish category**  1. Click [Thêm mới nhóm hàng] button.  2. Click [Hủy bỏ] button. | 1. [Thêm mới nhóm món] window opened.  2. [Thêm mới nhóm món] window closed |  |  |  |  |
| MC2.9 | Edit dish category | **Edit category successfully**  1. Select a dish category, click [Chỉnh sửa nhóm hàng] button  2. Change [Tên nhóm] and [Ghi chú] to new value, click [Ghi dữ liệu] | 1. [Chỉnh sửa nhóm hàng] window opened with existing data  2. [Chỉnh sửa nhóm hàng] window closed, Dish category list is updated |  |  |  |  |
| MC2.10 | **Edit to maximum length category name**  1. Click [Chỉnh sửa nhóm hàng] button.  2. Input new dish category with  127 character-length on name  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Chỉnh sửa nhóm hàng] window opened.  2. Input 127 characters successfully  3. Cannot add any character more  4. [Chỉnh sửa nhóm hàng] window closed, dish category list is updated |  |  |  |  |
| MC2.11 | **Edit to maximum length category description**  1. Click [Thêm mới] from context menu of dish category.  2. Input new dish category with  1024 character-length on description  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Chỉnh sửa nhóm hàng] window opened.  2. Input 1024 characters successfully  3. Cannot add any character more  4. [Chỉnh sửa nhóm hàng] window closed, dish category list is updated |  |  |  |  |
| MC2.12 | **Edit categoryfailed**  1. Click [Chỉnh sửa nhóm hàng] button.  2. Input a current existing category  3. Add spaces before and after category name | 1. [Chỉnh sửa nhóm hàng] window opened. [Ghi dữ liệu] button is disabled.  2. [Giá trị đã tồn tại] warning is displayed.  3. [Giá trị đã tồn tại] warning is displayed. |  |  |  |  |
| MC2.13 | **Edit invalid category**  1. Click [Chỉnh sửa nhóm hàng] button.  2. Input new dish category with these characters on name: \ / : \* " <> |  3. Clear category name field | 1. [Chỉnh sửa nhóm hàng] window opened. [Ghi dữ liệu] button is disabled.  2. [Chứa kí hiệu không hợp lệ] warning is displayed. [Ghi dữ liệu] button is disabled.  3. Warning disappeared. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC2.14 | **Edit invalid category**  1. Select [Tất cả] category | 1. [Chỉnh sửa nhóm hàng] button and [Chỉnh sửa] from context menu is disabled |  |  |  |  |
| MC2.15 | **Cancel editing dish category**  1. Click [Chỉnh sửa nhóm hàng] button.  2. Click [Hủy bỏ] button. | 1. [Chỉnh sửa nhóm hàng] window opened.  2. [Chỉnh sửa nhóm hàng] window closed |  |  |  |  |
| MC2.16 | Rename dish category | **Rename dish category**  1. Select a dish category, select [Đổi tên] from context menu  2. Input new name | 1. Name of dish category is editable  2. Dish category name is changed |  |  |  |  |
| MC2.17 | Delete dish category | **Delete category successfully**  1. Select a dish category, click [Xóa nhóm hàng] button  2. Select OK | 1. Warning & confirmation dialog box appeared correctly |  |  |  |  |
| MC2.18 | **Cancel deleting category**  1. Select a dish category, click [Xóa nhóm hàng] button in context menu  2. Select cancel | 1. Warning & confirmation dialog box appeared correctly  2. Warning & confirmation dialog box closed. |  |  |  |  |
| MC2.19 | **Delete invalid category**  1. Select [Tất cả] category | 1. [Xóa nhóm hàng] button and [Xóa] from context menu is disabled |  |  |  |  |
| MC2.20 | Add dish | **Add dish successfully**  1. Select [Thêm mới] button on [Danh mục món]  2. Input correct information of dish  3. Click [Ghi dữ liệu] button | 1. [Tạo mới món] window appeared. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Tạo mới món] window is disappeared, list of dishes is updated. |  |  |  |  |
| MC2.21 | **Add dish with least information**  1. Select [Thêm mới] option on [Danh mục món] context menu  2. Input dish name, leave other fields blank or default values  3. Click [Ghi dữ liệu] button | 1. [Tạo mới món] window appeared. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Tạo mới món] window is disappeared, list of dishes is updated. |  |  |  |  |
| MC2.22 | **Cancel adding dish**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Hủy bỏ button] | 1. [Tạo mới món] window appeared  2. [Tạo mới món] window disappeared |  |  |  |  |
| MC2.23 | **Add dish with new category**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Thêm mới nhóm hàng] button  3. Input dish category, clich [Ghi dữ liệu] button  4. Click [Ghi dữ liệu] button | 2. [Thêm mới nhóm món] is displayed.  3. New dish category is displayed in downdownlist.  4. New dish is displayed in list |  |  |  |  |
| MC2.24 | **Add dish with new unit**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Thêm mới đơn vị tính] button  3. Add a new Unit, click [Chọn] button  Click [Ghi dữ liệu] button | 2. [Đơn vị tính của món] window is displayed.  3. New unit is displayed  4. New dish is displayed in list |  |  |  |  |
| MC2.25 | **Add dish with edited unit**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Thêm mới đơn vị tính] button  3. Edit/delete an old Unit, click [Chọn] button  4. Click [Ghi dữ liệu] button | 2. [Đơn vị tính của món] window is displayed.  3. Unit is updated  4. New dish is displayed in list |  |  |  |  |
| MC2.26 | **Add dish with negative price**  1. Select [Thêm mới] button on [Danh mục món]  2. Input negative value to [Giá] field | 2. Negative change to 0 automatically when move cursor away |  |  |  |  |
| MC2.27 | **Add dish with new VAT**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Edit VAT] button  3. Create new VAT value, click [Ghi dữ liệu] button  4. Click [Ghi dữ liệu] button | 2. [Danh mục VAT] window displayed  3. VAT values is updated  4. New dish is displayed in list |  |  |  |  |
| MC2.28 | **Add dish with edited VAT**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Edit VAT] button  3. Edit/delete VAT value, click [Ghi dữ liệu] button  4. Click [Ghi dữ liệu] button | 2. [Danh mục VAT] window displayed  3. VAT values is updated  4. New dish is displayed in list |  |  |  |  |
| MC2.29 | **Add dish with new promotion**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Thêm mới khuyến mại] button  4. Click [Ghi dữ liệu] button | 2. [Khuyến mại] window displayed  3. New promotion values is updated  4. New dish is displayed in list |  |  |  |  |
| MC2.30 | **Add dish with edited promotion**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Thêm mới khuyến mại] button  3. Edit/delete promotion value, click [Ghi dữ liệu] button  4. Click [Ghi dữ liệu] button | 2. [Khuyến mại] window displayed  3. Promotion values is updated  4. New dish is displayed in list |  |  |  |  |
| MC2.31 | **Add dish with new materials**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Thêm Mới] button on Materials list  3. Input new material information, click [Ghi dữ liệu] button  Click [Ghi dữ liệu] button | 2. [Thêm mới nguyên liệu món] windows is displayed  3. New material is displayed  4. New dish is displayed in list |  |  |  |  |
| MC2.32 | **Add dish with edited materials**  1. Select [Thêm mới] button on [Danh mục món]  2. Select [Chỉnh sửa] button on Materials list  3. Edit material information, click [Ghi dữ liệu] button  4. Click [Ghi dữ liệu] button | 2. [Chỉnh sửa nguyên liệu món] windows is displayed  3. Current material’s information is displayed  4. New dish is displayed in list |  |  |  |  |
| MC2.33 | **Add dish failed**  1. Select [Thêm mới] button on [Danh mục món]  2. Input non-alphanumeric characters, with no dot (.), to dish name field.  3. Clear dish name field | 1. [Tạo mới món] window appeared. [Ghi dữ liệu] button is disabled.  2. Warning message is display, [Ghi dữ liệu] button is not enabled.  3. Warning message is disappeared. |  |  |  |  |
| MC2.34 | Edit dish | **Edit dish successfully**  1. Select [Chỉnh sửa] button on [Danh mục món]  2. Edit information of dish  3. Click [Ghi dữ liệu] button | 1. [Chỉnh sửa món] window appeared.  3. [Chỉnh sửa món] window is disappeared, list of dishes is updated. |  |  |  |  |
| MC2.35 | **Cancel edit dish**  1. Select [Chỉnh sửa] on [Danh mục món] context menu  2. Edit information of dish  2. Select [Hủy bỏ] button | 1. [Chỉnh sửa món] window appeared.  3. [Chỉnh sửa món] window is disappeared, list of dishes is not updated. |  |  |  |  |
| MC2.36 | **Edit dish failed**  1. Select [Chỉnh sửa] button on [Danh mục món]  2. Enter an existing dish name | 1. [Chỉnh sửa món] window appeared.  2. [Ghi dữ liệu] button is disabled. Warning message is displayed. |  |  |  |  |
| MC2.37 | Delete dish | **Delete dish successfully**  1. Select a dish, click [Xóa] button  2. Click Yes | 1. Confirmation dialog is shown  2. Dish is deleted |  |  |  |  |
| MC2.38 | **Delete dish failed**  1. Select a dish, select [Xóa] from context menu  2. Click No | 1. Confirmation dialog is shown  2. Dish is not deleted |  |  |  |  |

##### Menu management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Menu management function** | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
| MC2.39 | Menu open | **Menu open correctly**  1. Click [Quản lý menu] on [Quản lý] menu  2. Click [Quản lý menu] on menu bar  3. Click [Quản lý menu] close button  4. Click [Quản lý menu] on menu bar again | 1. [Quản lý menu] control is opened, list of dish and dish category is displayed correctly.  2. [Quản lý menu] control is not double opened  3. [Quản lý menu] control is closed  4. [Quản lý menu] control is opened | Pass | ThanhHX |  |  |
| MC2.40 | Add menu | **Add menu successfully**  1. Click [Tạo mới] button on [Thực đơn] list.  2. Input new menu without [Ghi chú]  3. Click [Ghi dữ liệu]  4. Click new menu | 1. [Thêm mới thực đơn món] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới thực đơn món] window closed, new dish category is displayed in the list with sort  4. No dish is shown on new menu | Pass | ThanhHX |  |  |
| MC2.41 | **Add menu successfully with description**  1. Select [Thêm mới] from context menu of menu list  2. Input new menu with [Ghi chú]  3. Click [Ghi dữ liệu]  4. Click new menu | 1. [Thêm mới thực đơn món] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới thực đơn món] window closed, new dish category with “Ghi chú” is displayed in the list  4. No dish is shown on new menu | Pass | ThanhHX |  |  |
| MC2.42 | **Add maximum length category name**  1. Click [Thêm mới] button on [Menu] list.  2. Input new menu with  127 character-length on name  3. Add another character  4. Click [Ghi dữ liệu] | 2. [Ghi dữ liệu] button is enabled.  3. Cannot add any character more  4. [Thêm mới thực đơn món] window closed, new dish category is displayed in the list | Pass | ThanhHX |  |  |
| MC2.43 | **Add maximum length category description**  1. Click [Thêm mới] button on [Menu] list.  2. Input new menu with  1024 character-length on description  3. Add another character  4. Click [Ghi dữ liệu] | 2. [Ghi dữ liệu] button is enabled.  3. Cannot add any character more  4. [Thêm mới thực đơn món] window closed, new menu is displayed in the list | Failed | ThanhHX |  |  |
| MC2.44 | **Add menufailed**  1. Click [Thêm mới] button on [Menu] list.  2. Input a current existing menu name  3. Add spaces before and after menu name | 2. [Giá trị đã tồn tại] warning is displayed.  3. [Giá trị đã tồn tại] warning is displayed. |  |  |  |  |
| MC2.45 | **Add menufailed**  1. Click [Thêm mới] button on [Menu] list.  2. Input a new menu name, and blank at [Ngày] field. | 2. Warning message is displayed. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC2.46 | **Add invalid menu**  1. Click [Thêm mới] button on [Menu] list.  2. Input new dish category with these characters on name: \ / : \* " <> |  3. Clear category name field | 1. [Thêm mới thực đơn món] window opened. [Ghi dữ liệu] button is disabled.  2. [Chứa kí hiệu không hợp lệ] warning is displayed. [Ghi dữ liệu] button is disabled.  3. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC2.47 | **Cancel adding menu**  1. Click [Thêm mới] button on [Menu] list.  2. Click [Hủy bỏ] button. | 1. [Thêm mới thực đơn món] window opened.  2. [Thêm mới thực đơn món] window closed |  |  |  |  |
| MC2.48 | Edit menu | **Edit menu successfully**  1. Select a menu, click [Chỉnh sửa] button on menu list  2. Change to new valid value, click [Ghi dữ liệu] | 1. [Chỉnh sửa thực đơn] window opened with existing data  2. [Chỉnh sửa thực đơn] window closed, menu list is updated |  |  |  |  |
| MC2.49 | **Edit to maximum length menu name**  1. Select a menu, click [Chỉnh sửa] button on menu list  2. Input new menu with  127 character-length on name  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Chỉnh sửa thực đơn] window opened.  2. Input 127 characters successfully  3. Cannot add any character more  4. [Chỉnh sửa thực đơn] window closed, dish menu list is updated |  |  |  |  |
| MC2.50 | **Edit to maximum length menu description**  1. Click [Chỉnh sửa] from context menu of menu list.  2. Input new menu with  1024 character-length on description  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Chỉnh sửa thực đơn] window opened.  2. Input 1024 characters successfully  3. Cannot add any character more  4. [Chỉnh sửa thực đơn] window closed, menu list is updated |  |  |  |  |
| MC2.51 | **Edit menufailed**  1. Click [Chỉnh sửa] from context menu of menu list.  2. Input a current existing menu name  3. Add spaces before and after menu name | 1. [Chỉnh sửa thực đơn] window opened. [Ghi dữ liệu] button is disabled.  2. [Giá trị đã tồn tại] warning is displayed.  3. [Giá trị đã tồn tại] warning is displayed. |  |  |  |  |
| MC2.52 | **Edit invalid menu**  1. Click [Chỉnh sửa] from context menu of menu list.  2. Input new menu with these characters on name: \ / : \* " <> |  3. Clear category name field | 1. [Chỉnh sửa thực đơn] window opened. [Ghi dữ liệu] button is disabled.  2. [Chứa kí hiệu không hợp lệ] warning is displayed. [Ghi dữ liệu] button is disabled.  3. Warning disappeared. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC2.53 | **Edit invalid menu**  1. Select [Tất cả] category | 1. [Chỉnh sửa thực đơn] button and [Chỉnh sửa] from context menu is disabled |  |  |  |  |
| MC2.54 | **Cancel editing menu**  1. Click [Chỉnh sửa] from context menu of menu list.  2. Click [Hủy bỏ] button. | 1. [Chỉnh sửa thực đơn] window opened.  2. [Chỉnh sửa thực đơn] window closed |  |  |  |  |
| MC2.55 | Rename menu | **Rename menu**  1. Select a menu, select [Đổi tên] from context menu  2. Input new name, then [Enter] | 1. Name of menu is editable  2. Menu name is changed |  |  |  |  |
| MC2.56 | Delete menu | **Delete menu successfully**  1. Select a menu, click [Xóa] button  2. Select OK | 1. Warning & confirmation dialog box appeared correctly |  |  |  |  |
| MC2.57 |  | **Cancel deleting menu**  1. Select a menu, click [Xóa] button in context menu  2. Select cancel | 1. Warning & confirmation dialog box appeared correctly  2. Warning & confirmation dialog box closed. |  |  |  |  |
| MC2.58 | Add dish to menu | **Add dish successfully**  1. Select [Thêm món] button on [Danh mục món]  2. Select dish to add  3. Click [Ghi dữ liệu] button | 1. [Thêm món] window appeared.  3. [Thêm món] window is disappeared, list of dishes is updated. |  |  |  |  |
| MC2.59 | **Cancel adding dish**  1. Select [Thêm mới] button on [Danh mục món] context menu  2. Select [Hủy bỏ] button | 1. [Thêm món] window appeared  2. [Thêm món] window disappeared with no change |  |  |  |  |
| MC2.60 | **Add dish failed**  1. Select [Thêm mới] button on [Danh mục món]  2. Select current existing dish.  3. Click [Ghi dữ liệu] button | 3. Error message appeared. |  |  |  |  |
| MC2.61 | Delete dish | **Delete dish successfully**  1. Select a dish, click [Xóa] button  2. Click Yes | 1. Confirmation dialog is shown  2. Dish is deleted |  |  |  |  |
| MC2.62 |  | **Delete dish failed**  1. Select a dish, select [Xóa] from context menu  2. Click No | 1. Confirmation dialog is shown  2. Dish is not deleted |  |  |  |  |

##### Account management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Account management function** | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
| MC3.1 | Open User management function | **Window open**  1. Click [Quản lý người dùng] on [Quản lý] menu  3. Click [Quản lý người dùng] close button  4. Click [Người dùng] on menu bar | 1. [Quản lý người dùng] window appeared.  1. [Quản lý người dùng] window closed  1. [Quản lý người dùng] window appeared. | Pass | ThanhHX |  |  |
| MC3.2 | Add user | **Add user successfully**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Click [Thêm mới] button on [Quản lý người dùng] window  3. Input valid account information  4. Click [Ghi dữ liệu] button | 2. [Thêm mới tài khoản] window appeared, [Ghi dữ liệu] button is disabled.  3. No warning message displayed, [Ghi dữ liệu] button is enabled.  4. New user is displayed in the list |  |  |  |  |
| MC3.3 | **Add existing user**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Click [Thêm mới] menu on [Quản lý người dùng] context menu  3. Input valid information with existing user name | 2. [Thêm mới tài khoản] window appeared  3. Warning message is displayed. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC3.4 | **Add blank user information**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Click [Thêm mới] button on [Quản lý người dùng] window  3. Input valid account information, then clear [Tên tài khoản] field  4. Input [Tên tài khoản], clear [Mật khẩu] field  5. Input [Mật khẩu], clear [Xác nhận mật khẩu] field | 3. Warning message is displayed. [Ghi dữ liệu] button is disabled.  4. Warning message is displayed. [Ghi dữ liệu] button is disabled.  5. Warning message is displayed. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC3.5 | **Add invalid user**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Click [Thêm mới] button on [Quản lý người dùng] window  3. Input username with non-space-alpha-numeric characters | 3. Warning message is displayed. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC3.6 | **Add invalid user password**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Click [Thêm mới] button on [Quản lý người dùng] window  3. Input 5 characters to [Mật khẩu] field | 3. Warning message is displayed. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC3.7 | **Cancel add user**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Click [Thêm mới] button on [Quản lý người dùng] window  3. Click [Hủy bỏ] button | 3. [Thêm mới người dùng] window disappeared. Nothing changes. |  |  |  |  |
| MC3.8 | Edit user | **Edit user successfully**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Choose an existing user, click [Chỉnh sửa] button  3. Edit user information  4. Click [Ghi dữ liệu] button | 2. [Chỉnh sửa tài khoản người dùng] window is displayed.  4. User is updated in the list |  |  |  |  |
| MC3.9 | **Cancel edit user**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Choose an existing user, select [Chỉnh sửa] from context menu  3. Click [Hủy bỏ] button | 2. [Chỉnh sửa tài khoản người dùng] window is displayed.  3. [Chỉnh sửa tài khoản người dùng] window disappeared. Nothing changes. |  |  |  |  |
| MC3.10 | **Edit user with new password recover question**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Choose an existing user, click [Chỉnh sửa] button  3. Add another password recovery question  4. Input password recovery answer and click [Ghi dữ liệu] | 3. New question is displayed in question list  4. Question and answer updated successfully |  |  |  |  |
| MC3.11 | **Edit user failed**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Choose an existing user, click [Chỉnh sửa] button  3. Input password and password confirmation wrong | 3. Warning message is displayed. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC3.12 | Delete User | **Delete user successfully**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Choose an existing user, click [Xóa] button  3. Click [Yes] button | 2. Confirmation dialog appeared  3. User is deleted. |  |  |  |  |
| MC3.13 | **Cancel deleting user**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Choose an existing user, select [Xóa] from context menu  3. Click [No] button | 2. Confirmation dialog appeared  3. User is not deleted. |  |  |  |  |
| MC3.14 | Reset user password | **Reset password successfully**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Select user, select [Cấp lại mật khẩu] on context menu  3. Input new username and password, click [Ghi dữ liệu] | 2. [Đổi mật khẩu] box is displayed  3. User log in with new password successfully and not able to login by old password |  |  |  |  |
| MC3.15 | **Cancel reseting password**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Select user, select [Cấp lại mật khẩu] on context menu  3. Input new username and password, click [Hủy bỏ] | 3. User log in with old password successfully, and not able to login by new password |  |  |  |  |
| MC3.16 | **Resting password failed**  1. Click [Quản lý người dùng] on [Quản lý] menu  2. Select user, select [Cấp lại mật khẩu] on context menu  3. Input invalid password and password confirmation | 3. Warning message is displayed. [Ghi dữ liệu] button is disabled. |  |  |  |  |

##### Financial report function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Financial Report Function** | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
| MC4.1 | Menu Open | **Financial Report open correctly**  1. Click [Báo cáo doanh thu] on [Quản lý] menu  2. Click [Báo cáo doanh thu] on menu bar  3. Click [Báo cáo doanh thu] close button  4. Click [Báo cáo doanh thu] on menu bar again | 1. [Báo cáo doanh thu] control is opened, [Từ] and [Đến] field display current day.  2. [Báo cáo doanh thu] control is not double opened  3. [Báo cáo doanh thu] control is closed  4. [Báo cáo doanh thu] control is opened |  |  |  |  |
| MC4.2 | Load financial data | **Valid day combination**  1. Open [Báo cáo doanh thu] control  2. Click [Từ] dropdown button, 3. Select a previous day.  4. Click [Đến] dropdown button, 5. Select a later day. | 2. A calendar appeared, the selected day is also display in the text field.  3. [Danh sách phiếu đã phục vụ] và [Danh sách nhóm đã phục vụ] is refresh automatically.  4. A canlenday appeared, the selected day is also displayed  5. [Danh sách phiếu đã phục vụ] và [Danh sách nhóm đã phục vụ is refresh automatically. | Pass | ThanhHX | 16/4/13 |  |
| MC4.3 | **Invalid day combination**  1. Open [Báo cáo doanh thu] control  2. Click [Từ] dropdown button, 3. Select a later day.  4. Click [Đến] dropdown button, 5. Select a previous day. | 5. [Danh sách phiếu đã phục vụ] và [Danh sách nhóm đã phục vụ is refresh automatically.  5. [Danh sách phiếu đã phục vụ] và [Danh sách nhóm đã phục vụ is refresh automatically. |  |  |  |  |
| MC4.4 | Print Financial Data | Print financial data successfully  1. Open [Báo cáo doanh thu] control  2. Select report to print, click [Print] button  3. Click [Print] button | 2. Print preview dialog opened.  3. Computer starts to print the document |  |  |  |  |
| MC4.5 |  | Print financial data canceled  1. Open [Báo cáo doanh thu] control  2. Select report to print, click [Print] button  3. Click [Close] button | 2. Print preview dialog opened.  3. Print preview dialog closed. |  |  |  |  |

##### Table and area management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
| MC5.1 | **Menu open** | **Table and Area management open correctly**  1. Click [Danh mục bàn, khu vực] on [Quản lý] menu  2. Click [Danh mục bàn, khu vực] on menu bar  3. Click [Danh mục bàn, khu vực] close button  4. Click [Danh mục bàn, khu vực] on menu bar again | 1. [Danh mục bàn, khu vực] control is opened, list of tables and area is displayed correctly.  2. [Danh mục bàn, khu vực] control is not double opened  3. [Danh mục bàn, khu vực] control is closed  4. [Danh mục bàn, khu vực] control is opened |  |  |  |  |
| MC5.2 | **Add area** | **Add area successfully**  1. Click [Thêm mới khu vực] button.  2. Input new area without [Ghi chú]  3. Click [Ghi dữ liệu]  4. Click new area | 1. [Thêm mới khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới khu vực] window closed, new area is displayed in the list with sort  4. No table is shown on new area |  |  |  |  |
| MC5.3 | **Add area successfully with description**  1. Select [Thêm mới] from context menu of area  2. Input new area with [Ghi chú]  3. Click [Ghi dữ liệu]  4. Click new area | 1. [Thêm mới khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới khu vực] window closed, new area with “Ghi chú” is displayed in the list  4. No dish is shown on new area |  |  |  |  |
| MC5.4 | **Add maximum length areaname**  1. Click [Thêm mới khu vực] button.  2. Input new area with  127 character-length on name  MC5.3. Add another character  4. Click [Ghi dữ liệu] | 1. [Thêm mới khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. Cannot add any character more  4. [Thêm mới khu vực] window closed, new area is displayed in the list |  |  |  |  |
| MC5.5 | **Add maximum length areadescription**  1. Click [Thêm mới nhóm hàng] button.  2. Input new area with 1024 character-length on description  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Thêm mới khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. Cannot add any character more  4. [Thêm mới khu vực] window closed, new area is displayed in the list |  |  |  |  |
| MC5.6 | **Add areafailed**  1. Click [Thêm mới khu vực] button.  2. Input a current existing area  3. Add spaces before and after area name | 1. [Thêm mới khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Giá trị đã tồn tại] warning is displayed.  3. [Giá trị đã tồn tại] warning is displayed. |  |  |  |  |
| MC5.7 | **Add invalid area**  1. Click [Thêm mới khu vực] button.  2. Input new area with these characters on name: \ / : \* " <> |  3. Clear area name field | 1. [Thêm mới khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Chứa kí hiệu không hợp lệ] warning is displayed. [Ghi dữ liệu] button is disabled.  3. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC5.8 | **Cancel adding area**  1. Click [Thêm mới khu vực] button.  2. Click [Hủy bỏ] button. | 1. [Thêm mới khu vực] window opened.  2. [Thêm mới khu vực] window closed |  |  |  |  |
| MC5.9 | **Edit area** | **Edit areasuccessfully**  1. Select an area, click [Chỉnh sửa khu vực] button  2. Change [Khu vực] and [Ghi chú] to new value, click [Ghi dữ liệu] | 1. [Chỉnh sửa khu vực] window opened with existing data  2. [Chỉnh sửa khu vực] window closed, area list is updated |  |  |  |  |
| MC5.10 | **Edit to maximum length areaname**  1. Click [Chỉnh sửa khu vực] button.  2. Input new area with  127 character-length on name  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Chỉnh sửa khu vực] window opened.  2. Input 127 characters successfully  3. Cannot add any character more  4. [Chỉnh sửa khu vực] window closed, area list is updated |  |  |  |  |
| MC5.11 | **Edit to maximum length areadescription**  1. Click [Thêm mới] from context menu of area.  2. Input new area with 1024 character-length on description  3. Add another character  4. Click [Ghi dữ liệu] | 1. [Chỉnh sửa khu vực] window opened.  2. Input 1024 characters successfully  3. Cannot add any character more  4. [Chỉnh sửa khu vực] window closed, area list is updated |  |  |  |  |
| MC5.12 | **Edit areafailed**  1. Click [Chỉnh sửa khu vực] button.  2. Input a current existing area  3. Add spaces before and after area name | 1. [Chỉnh sửa khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Giá trị đã tồn tại] warning is displayed.  3. [Giá trị đã tồn tại] warning is displayed. |  |  |  |  |
| MC5.13 | **Edit invalid area**  1. Click [Chỉnh sửa khu vực] button.  2. Input new area with these characters on name: \ / : \* " <> |  3. Clear area name field | 1. [Chỉnh sửa khu vực] window opened. [Ghi dữ liệu] button is disabled.  2. [Chứa kí hiệu không hợp lệ] warning is displayed. [Ghi dữ liệu] button is disabled.  3. Warning disappeared. [Ghi dữ liệu] button is disabled. |  |  |  |  |
| MC5.14 | **Edit invalid area**  1. Select [Tất cả] category | 1. [Chỉnh sửa khu vực] button and [Chỉnh sửa] from context menu is disabled |  |  |  |  |
| MC5.15 | **Cancel editing area**  1. Click [Chỉnh sửa khu vực] button.  2. Click [Hủy bỏ] button. | 1. [Chỉnh sửa khu vực] window opened.  2. [Chỉnh sửa khu vực] window closed |  |  |  |  |
| MC5.16 | **Rename area** | **Rename area**  1. Select an area, select [Đổi tên] from context menu  2. Input new name, then Enter | 1. Name of area is editable  2. Area name is changed |  |  |  |  |
| MC5.17 | **Delete area** | **Delete areasuccessfully**  1. Select an area, click [Xóa khu vực] button  2. Select OK | 1. Warning & confirmation dialog box appeared correctly |  |  |  |  |
| MC5.18 | **Cancel deleting area**  1. Select an area, click [Xóa area] button in context menu  2. Select cancel | 1. Warning & confirmation dialog box appeared correctly  2. Warning & confirmation dialog box closed. |  |  |  |  |
| MC5.19 | **Delete invalid area**  1. Select [Tất cả] area | 1. [Xóa khu vực] button and [Xóa] from context menu is disabled |  |  |  |  |
| MC5.20 | **Add table** | **Add table successfully**  1. Select [Thêm mới bàn] button on [Danh sách bàn]  2. Input correct information of table  3. Click [Ghi dữ liệu] button | 1. [Thêm mới bàn] window appeared. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới bàn] window is disappeared, list of dishes is updated. |  |  |  |  |
| MC5.21 | **Add table with least information**  1. Select [Thêm mới bàn] option on [Danh sách bàn] context menu  2. Input table name, leave other fields blank or default values  3. Click [Ghi dữ liệu] button | 1. [Thêm mới bàn] window appeared. [Ghi dữ liệu] button is disabled.  2. [Ghi dữ liệu] button is enabled.  3. [Thêm mới bàn] window is disappeared, list of table is updated. |  |  |  |  |
| MC5.22 | **Cancel adding table**  1. Select [Thêm mới bàn] button on [Danh sách bàn]  2. Select [Hủy bỏ button] | 1. [Thêm mới bàn] window appeared  2. [Thêm mới bàn] window disappeared |  |  |  |  |
| MC5.23 | **Add table with new area**  1. Select [Thêm mới bàn] button on [Danh sách bàn]  2. Select [Thêm mới khu vực] button  3. Input area, click [Ghi dữ liệu] button  4. Click [Ghi dữ liệu] button | 2. [Thêm mới khu vực] window is displayed.  3. New area is displayed in downdownlist.  4. New table is displayed in list |  |  |  |  |
| MC5.24 | **Add table with negative size**  1. Select [Thêm mới bàn] button on [Danh sách bàn]  2. Input negative value to [Số chỗ] field | 2. Negative change to 0 automatically when move cursor away |  |  |  |  |
| MC5.25 | **Addtable with maximum length name**  1. Select [Thêm mới bàn] button on [Danh sách bàn]  2. Input 127 characters to [Tên bàn] field  3. Input 128 characters to [Tên bàn] field  4. Click [Ghi dữ liệu] button | 1. [Thêm mới bàn] window opened.  2. Input available  3. Cannot add any character more  4. [Thêm mới bàn] window closed, new table is displayed in the list |  |  |  |  |
| MC5.26 | **Add table with maximum size**  1. Select [Thêm mới bàn] button on [Danh sách bàn]  2. Input 99 value to [Số chỗ] field  3. Input 100 value to [Số chỗ] field | 2. Number change to 10 auomatically when moving cursor away  3. Cannot input 3 digits |  |  |  |  |
| MC5.27 | **Addtable with maximum length notice**  1. Select [Thêm mới bàn] button on [Danh sách bàn], input 0 to [tên bàn] field  2. Input 1024 characters to [Ghi chú] field  3. Input 1025 characters to [Ghi chú] field  4. Click [Ghi dữ liệu] button | 1. [Thêm mới bàn] window opened.  2. Input available  3. Cannot add any character more  4. [Thêm mới bàn] window closed, new table is displayed in the list |  |  |  |  |
| MC5.28 | **Addtable with maximum length description**  1. Select [Thêm mới bàn] button on [Danh sách bàn], input 0 to [tên bàn] field  2. Input 1024 characters to [Mô tả] field  3. Input 1025 characters to [Mô tả] field  4. Click [Ghi dữ liệu] button | 1. [Thêm mới bàn] window opened.  2. Input available  3. Cannot add any character more  4. [Thêm mới bàn] window closed, new table is displayed in the list |  |  |  |  |
| MC5.29 | **Add table failed**  1. Select [Thêm mới bàn] button on [Danh sách bàn]  2. Input non-alphanumeric characters, with no dot (.), to table name field.  3. Clear table name field | 1. [Thêm mới bàn] window appeared. [Ghi dữ liệu] button is disabled.  2. Warning message is display, [Ghi dữ liệu] button is not enabled.  3. Warning message is disappeared. |  |  |  |  |
| MC5.30 | **Edit Table** | **Edittable successfully**  1. Select a free or break table  2. Select [Chỉnh sửa bàn] button on [Danh sách bàn]  3. Edit correct information of table  4. Click [Ghi dữ liệu] button | 2. [Chỉnh sửa bàn] window appeared.  4. [Chỉnh sửa bàn] window is disappeared, list of dishes is updated. |  |  |  |  |
| MC5.31 | **Cancel edit table**  1. Select a free or break table  2. Select [Chỉnh sửa] button on [Danh sách bàn] context menu  3. Edit correct information of table  4. Select [Hủy bỏ] button | 2. [Chỉnh sửa bàn] window appeared.  4. [Chỉnh sửa bàn] window is disappeared, list of dishes is not updated. |  |  |  |  |
| MC5.32 | **Edit table failed**  1. Select a free or break table  2. Select [Chỉnh sửa bàn] button on [Danh sách bàn]  3. Enter an existing table name | 3. [Ghi dữ liệu] button is disabled. Warning message is displayed. |  |  |  |  |
| MC5.33 | **Edit table failed**  1. Select a table in use / in booking  2. Select [Chỉnh sửa bàn] button on [Danh sách bàn] | 2. Error message is displayed |  |  |  |  |
| MC5.34 | **Delete Table** | **Delete table successfully**  1. Select a table, click [Xóa] button  2. Click Yes | 1. Confirmation dialog is shown  2. Table is deleted |  |  |  |  |
| MC5.35 | **Delete table failed**  1. Select a table, select [Xóa] from context menu  2. Click No | 1. Confirmation dialog is shown  2. Table is not deleted |  |  |  |  |

##### Work management function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
| MC6.1 | Open Work Management | **Work management open correctly**  1. Click [Quản lý công việc] on [Báo cáo] menu  2. Click [Quản lý công việc] on menu bar  3. Click [Quản lý công việc] close button  4. Click [Quản lý công việc] on menu bar again | 1. [Quản lý công việc] control is opened, [Từ] and [Đến] field display current day. [Danh sách nhân viên] is loaded correctly. [Danh sách bàn user đang phục vụ] is loaded automatically with first user in the list.  2. [Quản lý công việc] control is not double opened  3. [Quản lý công việc] control is closed  4. [Quản lý công việc] control is opened again |  |  |  |  |
| MC6.2 | Load Work Management Data | **Valid day combination**  1. Open [Quản lý công việc] control  2. Click [Từ] dropdown button, 3. Select a previous day.  4. Click [Đến] dropdown button, 5. Select a later day. | 2. A calendar appeared, the selected day is also display in the text field.  3. [Danh sách user đang phục vụ], [Danh sách món đang phục vụ] and [Danh sách món đã phục vụ] is refresh automatically.  4. A canlenday appeared, the selected day is also displayed  5. [Danh sách user đang phục vụ], [Danh sách món đang phục vụ] and [Danh sách món đã phục vụ] is refresh automatically. |  |  |  |  |
| MC6.3 |  | **Invalid day combination**  1. Open [Quản lý công việc] control  2. Click [Từ] dropdown button, 3. Select a later day.  4. Click [Đến] dropdown button, 5. Select a previous day. | 3. [Danh sách user đang phục vụ], [Danh sách món đang phục vụ] and [Danh sách món đã phục vụ] is refresh automatically.  5. [Danh sách user đang phục vụ], [Danh sách món đang phục vụ] and [Danh sách món đã phục vụ] have no infomation. |  |  |  |  |
| MC6.4 | Priting | **Print work history data successfully**  1. Open [Quản lý công việc] control  2. Select report to print, click [Print] button  3. Click [Print] button | 2. Print preview dialog opened correctly with template.  3. Computer starts to print the document |  |  |  |  |
| MC6.5 |  | Print financial data canceled  1. Open [Quản lý công việc] control  2. Select report to print, click [Print] button  3. Click [Close] button | 2. Print preview dialog opened correctly with template.  3. Print preview dialog closed. |  |  |  |  |

##### Booking function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
| MC7.1 | **Menu open** | **Business management open correctly**  1. Click [Quản lý kinh doanh] on [Quản lý] menu  2. Click [Quản lý kinh doanh] on menu bar  3. Click [Quản lý kinh doanh] close button  4. Click [Quản lý kinh doanh] on menu bar again | 1. [Quản lý kinh doanh] control is opened, list of tables and area is displayed correctly. All buttons except [Thanh Toán Gộp] is disabled.  2. [Quản lý kinh doanh] control is not double opened  3. [Quản lý kinh doanh] control is closed  4. [Quản lý kinh doanh] control is opened |  |  |  |  |
|  | **Booking statistic open correctly**  1. Click [Thống kê bàn đặt trước] on [Báo cáo] menu  2. Click [Thống kê bàn đặt trước] on menu bar  3. Click [Thống kê bàn đặt trước] close button  4. Click [Thống kê bàn đặt trước] on menu bar again | 1. [Thống kê bàn đặt trước] control is opened, [Từ] and [Đến] field display current day.  2. [Thống kê bàn đặt trước] control is not double opened  3. [Thống kê bàn đặt trước] control is closed  4. [Thống kê bàn đặt trước] control is opened |  |  |  |  |
|  | **Open booked table** | Open booked table successfully |  |  |  |  |  |
|  | Open booked table canceled |  |  |  |  |  |
|  | Open booked table failed: no waiter |  |  |  |  |  |
|  | **Book table** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **Cancel booked table** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **Multiple Payment** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **Load booked data** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **Print booked data** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

### Waiter Client

##### Log in/out function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Sign In** | | | | | | | |
| Step:   * Open Waiter Client | | | | | | | |
| MC1.1 | Sign In successfully | **Valid account**  1. Input administrator username and password correctly  2. Tap [Đăng nhập] button | 2. Main form opened | Pass | ThanhHX |  |  |
| MC1.2 | Sign In fail | **Blank account**  1. Do not input username and password  2. Tap [Đăng nhập] button | 1. Failed login notification appeared, main form is not opened |  |  |  |  |
| MC1.3 | **Blank username**  1. Input password, do not input username  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.4 | **Blank password**  1. Input username, do not input password  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.5 | **Invalid username**  1. Input wrong user name and password  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.6 | **Invalid password**  1. Input correct username and wrong password  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.7 | **Invalid authorization**  1. Input username and password of a non-administrator account  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.8 | Cancel sign in | 1. Tap close button | 1. F-Menu is closed |  |  |  |  |

##### Table arrangement function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

##### Creating orders function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

##### Managing orders function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

##### Billing function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

##### View paid cash function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

### Chef Client

##### Log in/out function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
| **Sign In** | | | | | | | |
| Step:   * Open Chef Client | | | | | | | |
| MC1.1 | Sign In successfully | **Valid account**  1. Input administrator username and password correctly  2. Tap [Đăng nhập] button | 2. Main form opened | Pass | ThanhHX |  |  |
| MC1.2 | Sign In fail | **Blank account**  1. Do not input username and password  2. Tap [Đăng nhập] button | 1. Failed login notification appeared, main form is not opened |  |  |  |  |
| MC1.3 | **Blank username**  1. Input password, do not input username  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.4 | **Blank password**  1. Input username, do not input password  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.5 | **Invalid username**  1. Input wrong user name and password  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.6 | **Invalid password**  1. Input correct username and wrong password  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.7 | **Invalid authorization**  1. Input username and password of a non-administrator account  2. Tap [Đăng nhập] button |  |  |  |  |
| MC1.8 | Cancel sign in | 1. Tap close button | 1. F-Menu is closed |  |  |  |  |

##### Sending notifications function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

##### Requesting materials function

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Operation** | **Expected Output** | **Result** | **Tester** | **Date** | **Note** |
|  | | | | | | | |
| Step:   * Open Management Console | | | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

### Checklists

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Question** | **Yes** | **No** | **N/A** |
| **User interface** | | | | |
|  | Is windows’ resizing ability correct? |  |  |  |
|  | Are all thelabels, buttons spelt correctly? |  |  |  |
|  | Are all the buttons are in standard format and size? |  |  |  |
|  | All fonts to be the same? Default font is font’s system. |  |  |  |
|  | Is the all error message text spelt correctly? |  |  |  |
|  | Assure that all dialog boxes have a consistent look and feel |  |  |  |
|  | Assure the system supports Unicode characters |  |  |  |
|  | Are fonts too large or too small to read? |  |  |  |
|  | Are names in command button & option box names are not abbreviations. |  |  |  |
|  | Are there any truncated character? |  |  |  |
|  | Are all words wraped and full displayed? |  |  |  |
|  | Are scroll bars displayed when necessary? |  |  |  |
|  | Are objects with different status having different icons? |  |  |  |
| **Validation** | | | | |
|  | Is submit button disabled as there is any field which failed from validation? |  |  |  |
|  | Is validation warning message displayed correctly? |  |  |  |
|  | For all numeric fields check the minimum and maximum values and also some mid-range values allowable? |  |  |  |
|  | For all character/alphanumeric fields check the field to ensure that there is a character limit specified and that this limit is exactly correct for the specified database size? |  |  |  |
|  | Does all character/alphanumeric fields not allow to input characters exceeding limit, and display warning message? |  |  |  |
|  | If any of the database columns don't allow null values then the corresponding screen fields must be mandatory. (If any field which initially was mandatory has become optional then check whether null values are allowed in this field.) |  |  |  |
|  | Do all mandatory fields require user input? |  |  |  |
| **Usability** | | | | |
|  | Are all the list of the system sorted correctly? Alphabetic sorting with [All] (if any) first is the default unless otherwise specified. |  |  |  |
|  | Are textboxes trim the space characters automatically? |  |  |  |
|  | Are checkboxes, radio button, dropdownlist… showing data correctly and updated? |  |  |  |
|  | Are cancel buttons close current windows/dialog/box and leave no changes? |  |  |  |
|  | Is all the functions still available when connection is lost? Is syncing process started automatically when connection is re-connected? |  |  |  |

## Test logs

### Test results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Function code** | **Number Pass** | **Number Fail** | **Untested** | **N/A** | **Number Total** |
|  | **Total** |  |  |  |  |  |
| **I** | **Management Console** |  |  |  |  |  |
| 1. Log in / log out function |  |  |  |  | 8 |
| 1. Menus and dishes management function |  |  |  |  | 62 |
| 1. Account management function |  |  |  |  | 16 |
| 1. Financial report function |  |  |  |  | 5 |
| 1. Table and area managing |  |  |  |  | 35 |
| 1. Work management function |  |  |  |  | 5 |
| 1. Booking function |  |  |  |  |  |
| **II** | **Waiter Clients** |  |  |  |  |  |
| 1. Log in / log out function |  |  |  |  |  |
| 1. Table arrangement function |  |  |  |  |  |
| 1. Creating orders function |  |  |  |  |  |
| 1. Managing orders function |  |  |  |  |  |
| 1. Billing function |  |  |  |  |  |
| 1. View paid cash function |  |  |  |  |  |
| **III** | **Chef Clients** |  |  |  |  |  |
| 1. Log in / log out function |  |  |  |  |  |
| 1. Sending notification function |  |  |  |  |  |
| 1. Requesting material function |  |  |  |  |  |

# USER’S MANUAL

## Waiter

### Home screen



Figure F-I-1-1:Home screen

This home screen has 2 buttons for users:

* Button “ĐĂNG NHẬP” is the major button that help users login in the system with their accounts. Only the account registered can log in the system.
* Button “KẾT NỐI TCP” is a button that help user reconnect the TCP connection if disconnected.

The text “Disconnect” indicates the status of TCP connection. If it is “Disconnect”, user must press “KẾT NỐI TCP” to reconnect.

### Add order detail

**Step 1:** Tap on button “ĐĂNG NHẬP” in Figure F-I-1-1 to login the system. If login successfully, the following screen will be displayed.



Figure F-I-1-2**:** Main Waiter menu

**Step 2:** Tap on button “MENU”

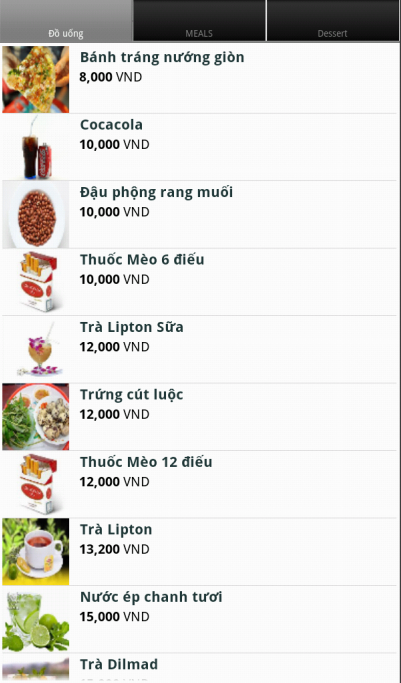


Figure F-I-1-3:List menu

**Step 3:** The list menu will be shown, and user can see all of the dishes with their detailed information. This menu has many tabs, if the user wants to see another category he will have to switch to another tab. This menu is only for show, if user taps on one item on the list, nothing will happen.

**Step 4:** After see the list menu, user can press Back button on the device to go back to Figure F-I-1-3 screen. If user taps on “DANH SÁCH BÀN TRỐNG”, the list of tables that are available will be shown like in Figure F-I-1-4.

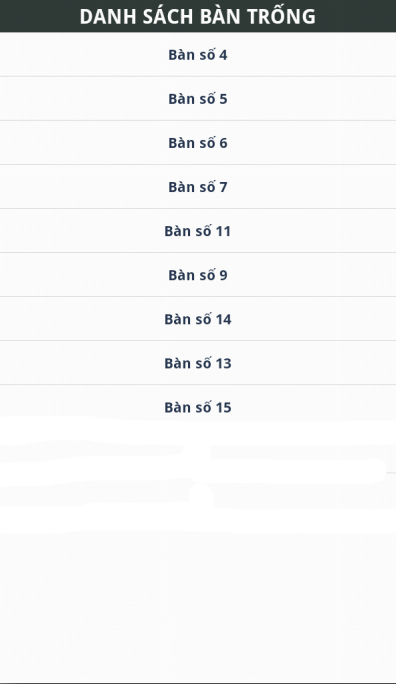


Figure F-I-1-4:List of available tables

**Step 5:** After seeing the list of available tables, if user tap on any item in the list, the dialog box will be displayed, and require user inputs the number of guests in that table. See Figure F-I-1-5.



Figure F-I-1-5: Input number of guests dialog

After **step 5**, the list of menu like figure F-I-1-3 will be displayed, but in this case, if user click on any item, another dialog will appear, see figure F-I-1-6.

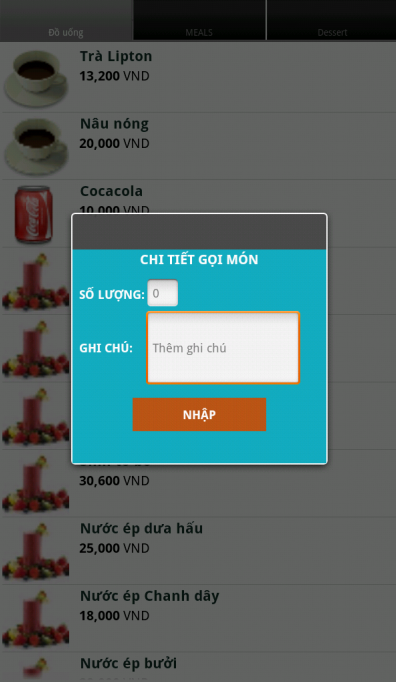


Figure F-I-1-6: Input quantity and note for an order detail.

**Step 6:** User must input the quantity, note is optional. After that, press “NHẬP” button to submit it.

These steps have shown you how to add an order detail for a specify table from list of available tables.

### Add more order detail in a table that is on currently serving list

**Step 1:** Click button “DANH SÁCH BÀN ĐANG SỬ DỤNG” on Figure F-I-1-1

**Step 2:** Click at any item in the list. The order detail for that table will show, see figure F-I-1-7.

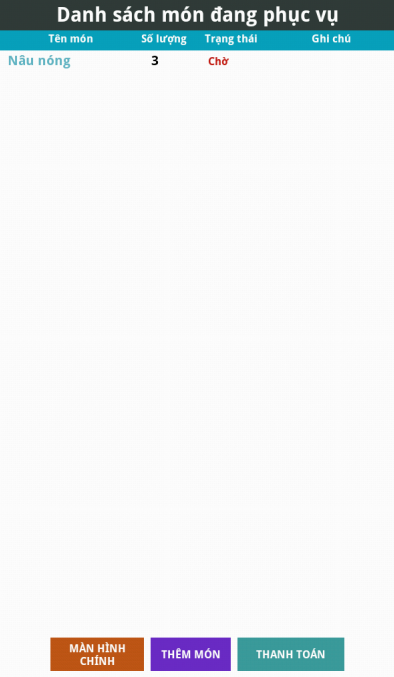


Figure F-I-1-7:Currently serving list

**Step 3:** Click button “THÊM MÓN” and the list menu atFigure F-I-1-3will be displayed. The procedures are like in Figure F-I-1-6. After that, the update of list of order details will be shown on Figure F-I-1-6.

### Update an order detail

**Step 1:** Click on an item that you want to update on the list in Figure F-I-1-7. The dialog will be shown with saved information, see Figure F-I-1-8.



Figure F-I-1-8:Delete and update order detail dialog

**Step 2:** User changes the value of quantity or note field, and finally taps on “CẬP NHẬT” button to update and “XOÁ” button to remove the order detail.

### Bill an order detail

**Step 1:** Tap on button “THANH TOÁN” at Figure F-I-1-7.

**Step 2:** After **Step 1,** the screen will show the dialog that show the total cost of that order, recipe (if it has). And simple calculator for waiter uses to calculate money back. See **Figure F-I-9**.



Figure F-I-1-9:Billing

**Step 3**: Click “TÍNH TIỀN THỪA” button to calculate the money back waiter must return for guest.

**Step 4:** 2 options:

* Click “Thanh toán & trả bàn” if guests want to bill and leave the table.
* Click “Thanh toán chưa trả bàn” if guests want to bill and don’t leave the table.

### Logout the app

**Step 1:** Tap on “THOÁT” button in Figure F-I-1-2to log out. A dialog will appear, asking for confirmation. See Figure F-I-1-10.



Figure F-I-1-10: Logout

**Step 2:** Click “OK” button to completely logout.

## Chef`

### Home screen



Figure F-II-1-1:Home screen

This home screen has 2 buttons for user.

* Button “ĐĂNG NHẬP” is the major button that help users login in the system with their accounts. Only the account registered can log in the system.
* Button “KẾT NỐI TCP” is a button that help user reconnect the TCP connection if disconnected.

The text “Disconnect” indicates the status of TCP connection. If it is “Disconnect”, user must press “KẾT NỐI TCP” to reconnect.

### Action in Main Screen

**Step 1:** After login, the Main screen will be shown, see Figure F-II-1-2.

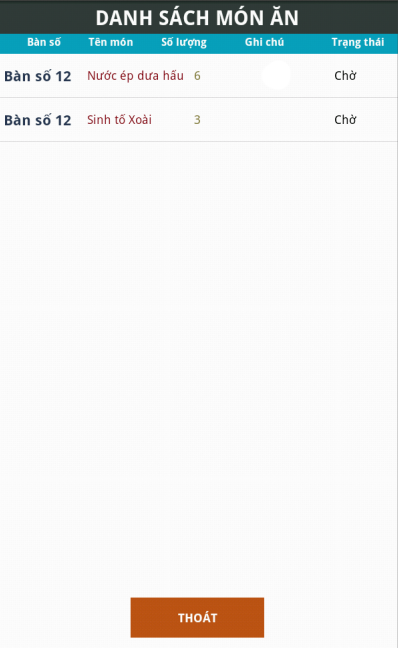


Figure F-II-1-2:Main screen

This will show the list of order detail that being served.

**Step 2:** User tap on an item on the list atFigure F-II-1-2**.** A dialog will be shown with 3 options like in Figure F-II-1-3.



Figure F-II-1-3: Update status of order detail dialog

* Click “XONG” button: order detail is done, and the order detail will change the status to “XONG”
* Click “HUỶ” button: order detail is cancel due to lack of materials, and the order detail will change the status to “HUỶ”
* Click “ĐANG CHẾ BIẾN” button: order detail is in process, and the order detail will change the status to “ĐANG CHẾ BIẾN”

# APPENDIX

## References

* + - <http://www.ipos.com.vn/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=39&category_id=1&vmcchk=1&option=com_virtuemart&Itemid=87>
    - <http://www.youtube.com/watch?v=r7_fA5UhyD4>
    - <http://www.adeptpros.com/products/geniusmenu/>
    - <http://elacarte.com/tablet/>
    - <http://www.emenuworld.com/>
    - Sommerville (2006), Software Engineering 8th Edition
    - Schwalbe (2008), Managing Information Technology Projects 6th Edition

## Meeting Minutes

### Meeting with customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Project kick off meeting | Date | 07/01/2013 | |
| **Location** | Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 18:00–20:00 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, TungNV, HieuHT | | | |
| **Absentees** |  | | | |
| Work | Content | | | Person in charge |
| Review last week’s work |  | | |  |
| This week’s work | Meeting with HieuHT, who is the first customer and the one came up with the initial idea. | | | Project team |
| Collect user requirements from customer. | | | Project team |

Table G-II-1: Meeting Minute 1

### Kick off meeting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Kickoff meeting | Date | 08/01/2013 | |
| **Location** | Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 14:00–16:00 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, TungNV | | | |
| **Absentees** |  | | | |
| Work | Content | | | Person in charge |
| Review last week’s work |  | | |  |
| This week’s work | Team building - Meeting team members at the first time and get to know each other. | | | Project team |
| Role assignment - Assign appropriate role for each member. | | | Project team |
| Creating rules - Set rules for project operation. | | | Project team |
| Vote team leader -  Voting on who will be the team leader. HieuNHD will be the leader. | | | Project team |

Table G-II-2: Meeting Minute 2

### Project meeting week 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subject** | Meeting week 2 | | Date | 14/01/2013 | |
| **Location** | Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | | Time | 14:00–17:00 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, TungNV | | | | |
| **Absentees** |  | | | | |
| Work | | Content | | | Person in charge |
| Review last week’s work | | * Review Customer requirements | | | Project team |
| * Review team members’ training progress | | | HieuNHD, LamPV, NghiaLT |
| This week’s work | | Assign training schedule and tasks for each member. | | | HieuNHD |

Table G-II-3: Meeting Minute 3

### Project meeting week 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Meeting week 3 | Date | 26/01/2013 | |
| **Location** | Room 105, Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 12:30–14:00 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, TungNV, Mr.LamPT | | | |
| **Absentees** |  | | | |
| Work | Content | | | Person in charge |
| Review last week’s work | * Meeting with instructor, review Use cases | | | HieuNHD |
| * Review SRS v1.0 | | | Project team |
| This week’s work | Assign training schedule and tasks for each member. | | | HieuNHD |

Table G-II-4: Meeting Minute 4

### Project meeting week 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Meeting week 5 | Date | 22/02/2013 | |
| **Location** | Room 105, Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 17:45–18:45 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, Mr.LamPT | | | |
| **Absentees** | TungNV | | | |
| Work | Content | | | Person in charge |
| Review last week’s work | * Meeting with instructor, review Report 1 | | | Project team |
| * Demonstrate software prototypes | | | LamPV, NghiaLT |
| This week’s work | Assign training schedule and tasks for each member. | | | HieuNHD |

Table G-II-5: Meeting Minute 5

### Project meeting week 7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Meeting week 7 | Date | 08/03/2013 | |
| **Location** | Meeting room,  Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 13:15–14:15 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, Mr.LamPT | | | |
| **Absentees** | TungNV | | | |
| Work | Content | | | Person in charge |
| Review last week’s work | * Meeting with instructor, review Report 1, 2 | | | Project team |
| * Demonstrate software | | | LamPV |
| * Review System Architecture | | | HieuNHD |
| This week’s work | Assign training schedule and tasks for each member. | | | HieuNHD |

Table G-II-6: Meeting Minute 6

### Project meeting week 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Meeting week 8 | Date | 15/03/2013 | |
| **Location** | Meeting room, Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 11:30–12:30 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, TungNV, Mr.LamPT | | | |
| **Absentees** |  | | | |
| Work | Content | | | Person in charge |
| Review last week’s work | * Meeting with instructor, review Report 4 (.vpp file) | | | Project team |
| * Demonstrate software | | | LamPV, NghiaLT |
| This week’s work | Assign training schedule and tasks for each member. | | | HieuNHD |

Table G-II-7: Meeting Minute 7

### Project meeting week 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Meeting week 9 | Date | 22/03/2013 | |
| **Location** | Meeting room, Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 14:00–15:00 | |
| Scriber | HieuNHD | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, Mr.LamPT | | | |
| **Absentees** | TungNV | | | |
| Work | Content | | | Person in charge |
| Review last week’s work | * Demonstrate whole system | | | LamPV, NghiaLT |
| * Review report 4 | | | HieuNHD |
| This week’s work | Assign training schedule and tasks for each member. | | | HieuNHD |

Table G-II-8: Meeting Minute 8

### Project meeting week 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | Meeting week 10 | Date | 30/03/2013 | |
| **Location** | Room 141, Detech Building  8, TônThấtThuyết street, CầuGiấy district, HàNội | Time | 16:00–17:00 | |
| Scriber | NghiaLT | |
| **Attendees** | HieuNHD, NghiaLT, ThanhHX, LamPV, Mr.LamPT | | | |
| **Absentees** | TungNV | | | |
| Work | Content | | | Person in charge |
| Review last week’s work | * Meeting with instructor, review Report 4 | | | HieuNHD |
| * Demonstrate whole system | | | LamPV, NghiaLT, ThanhHX |
| This week’s work | Assign training schedule and tasks for each member. | | | HieuNHD |
|  | Report to instructor about the problem with member TungNV. | | | Project team |

Table G-II-9: Meeting Minute 9