

TUTOR ONLINE

DETAIL DESIGN DOCUMENT

Data design

**Project Code: TTO**

**Document Code: TTO-DD – v1.0**

**Hòa Lac, 9/6/2017**

Record of change

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 9/6/2017 | Create Data Design | A | Create Data Design | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

|  |  |  |
| --- | --- | --- |
| AUTHOR: | Nguyen Thi Khanh Huyen  Team member | 15/06/2017 |
| REVIEWERS: | Tran Viet Vuong  Project Manager (PM) | 16/06/2017 |
| Nong Thi Hoai Thuong  Team member | 16/06/2017 |
| APPROVAL: | Phan Truong Lam  Supervisor | 17/06/2017 |

TABLE OF CONTENTS

1 Introduction 4

1.1 Purpose 4

1.2 Definitions, Acronyms and Abbreviations 4

1.3 References 4

1.4 Overview 4

2 Database 4

2.1 ERDs 4

2.2 XXX table 4

3 File design 4

3.1.1 XXX file 4

4 Code Design 4

4.1 Customer Code 4

5 Other considerations 4

6 Appendix 4

6.1 Data format symbol 4

# Introduction

## Purpose

Tutor Online data design document describes structure of database and file structure of system including internal file structure as well as interface file structure.

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviations | Description | Comment |
| TTO | Tutor Online |  |
| Q&A | Question and Answer |  |

## References

List all the reference document such as: other document of the system, or the technical article,...

|  |  |
| --- | --- |
| Document Number | Title |
| Class design document | Class Design |
| Screen design document | Screen Design |
| Architecture Design document | Architecture Design |
| Software Requirement Specification document | Software Requirement Specification |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Overview

This document describes relationship between entities and properties of each entity.

# Database

## ERDs

<Entity relation ship diagrams>



<List of tables and brief description>

|  |  |  |
| --- | --- | --- |
| No | Table Name | Description |
|  | CustomerMaster |  |
|  | Order | Order made by customer to buy one or more products |
|  | OrderDetail | Detail information of each products in order |
|  | ProductMaster | Contains all products of the store |

## XXX table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Field Name | Type | Length | Size | Null | Unique | P/F Key | Default | Description |
| 01 | xxxCode | int | 4 |  |  | x | P |  | Code of xxx, Auto number |
| 02 | yyyCode | int | 4 |  | x |  | F |  | Code of yyy. |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

# File design

<File List>

|  |  |  |  |
| --- | --- | --- | --- |
| No | File Name | File type | Description |
|  | name of file | <Fixed length>  <CSV> |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### XXX file

#### Format of file

<Describe the file format with the sequential of field>

#### Fields

<if this file is CSV File>

|  |  |  |  |
| --- | --- | --- | --- |
| No | Field Name | Format | Description |
|  |  | special format such as date “mm/dd/yy” |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

<If this file is fixed length>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Field Name | Format | Start | End | Description |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Code Design

<Design the format of codes ex. Customer codes, product codes that are complicated and contains meanings. These codes are fields in database or files. >

<List of code>

|  |  |  |
| --- | --- | --- |
| No | Code | Description |
| 01 | Customer code | Code of customer in the system. |
|  |  |  |
|  |  |  |
|  |  |  |

## Customer Code

Customer code has the form of AAAMMYYYY000 which

AAA: 3 required characters stand for Province of customer such as HAN: Ha Noi, DAN: Da Nang, HCM: Ho Chi Minh City

MM: The month that customer registers; can be 01 to 12

YYYY: The year that customer registers; it is in 4 character form: ex. 2004

000 is the sequential number start from 0; each customer has one number: “000”; “001”

example of customer code: HAN121999001

# Other considerations

[This section provides a description of other design elements that were considered as alternatives in selection process for the above database design, i.e. a brief explanation of advantages and disadvantages of the selected entity relationships and/or database implementation in comparison with others. It should be a clear answer to the question why the above data design is selected for this system, not the others.]

# Appendix

## Data format symbol

Following symbols are used to describe format of data field in files as well as in tables.

|  |  |
| --- | --- |
| Symbol | Description |
| # | Digit placeholder. |
| . | Decimal placeholder. |
| , | Thousands separator. |
| : | Time separator. |
| / | Date separator. |
| \ | Treat the next character in the mask string as a literal. This allows you to include the '#', '&', 'A', and '?' characters in the mask. This character is treated as a literal for masking purposes. |
| > | Convert all the characters that follow to uppercase. |
| < | Convert all the characters that follow to lowercase. |
| A | Alphanumeric character placeholder (entry required). For example: a – z, A – Z, or 0 – 9. |
| a | Alphanumeric character placeholder (entry optional). |
| 9 | Digit placeholder (entry optional). For example: 0 – 9. |
| 0 | Digit placeholder (entry required). For example: 0 – 9. |
| C | Character or space placeholder (entry optional). Valid values for this placeholder are ANSI characters in the following ranges: 32-126 and 128-255. |
| & | Character placeholder (entry required). Valid values for this placeholder are ANSI characters in the following ranges: 32-126 and 128-255. |
| ? | Letter placeholder. For example: a – z or A – Z. |
| Literal | All other symbols are displayed as literals; that is, as themselves. bound in “” like “ABC” |