**Domino’s Pizza® Point of Sales Application**

**Software Design Description**

**Version 1.0**

**10/07/2014**

**Mismanagement, Inc.**

Tonya Brenner

Jennifer Hoffman

Michael Reinitz

Michael Schmidt

Table of Contents

[1. Introduction](#h.tzb50ny4b4ng)

[1.1 Purpose](#h.kqan234dowm7)

[1.1.1 Data Design](#h.13r0j3zdlfwq)

[1.1.2 Architecture Design](#h.lno4brxqqtue)

[1.1.3 Interface Design](#h.dci5gzd3kkso)

[1.1.4 Procedural Design](#h.gbz9l8tzstrb)

[1.2 Scope](#h.s4t8mssmryv)

[1.3 Definitions, Acronyms, and Abbreviations](#h.s3jszalp6mn3)

[1.4 References](#h.mblkupfy87cp)

[1.5 System Overview](#h.u4yw7ctrgp09)

[2. Data Design](#h.wnu1p2u5f0uh)

[2.1 Internal Data Structures](#h.654ehikymli7)

[2.2 Data Flow Diagrams](#h.ffsmvyrrnh71)

[2.3 Data Dictionary](#h.ph11zcnbhnh0)

[3. Architecture Design](#h.yy5hyhhb158m)

[3.1 Program Structure](#h.ie98xbulh7fw)

[4. Interface Design](#h.a3r15qslnk2d)

[4.1 Internal System Interfaces](#h.1y2um3qlmyo)

[4.2 External System Interfaces](#h.73xl1t3gat)

[4.3 User Interfaces](#h.2f7hsrfd3u4j)

[5. Procedural Design](#h.c4xhbglxj53m)

[6. Miscellaneous](#h.s9420u46cbth)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author(s) |
| 10/07/2014 | 1.0 |  | Tonya Brenner,  Jennifer Hoffman,  Michael Reinitz,  Michael Schmidt |

# 1. Introduction

## 1.1 Purpose

The Software Design Document (SDD) is a comprehensive software design model consisting of four distinct but interrelated activities: data design, architectural design, interface design, and procedural design. The document is used as a tool to communicate preliminary design concepts to the customer and among the software engineering team members.

The SDD contains the following sections:

* Data Design
* Architecture Design
* Interface Design
* Procedural Design

### 1.1.1 Data Design

The Data Design describes data structures that reside within the software. Attributes and relationships between data objects dictate the choice of data structures. The Data Flow Diagrams (DFD) display input, processing, and output along separate modules.

### 1.1.2 Architecture Design

The Architecture Design uses information flow characteristics, and maps them into the program structure. Transformation mapping method is applied to exhibit distinct boundaries between incoming and outgoing data.

### 1.1.3 Interface Design

The Interface Design describes internal and external program interfaces as well as the design of human interface. Internal and external interface designs are based on the information obtained from the analysis model.

### 1.1.4 Procedural Design

The Procedural Design describes structured programming concepts using graphical, tabular, and textual notations. These design mediums enable the designer to represent procedural detail that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering work.

## 1.2 Scope

This SDD describes the entirety of the Domino’s Pizza® Point of Sales software design. Included in the scope of this document are data design, architecture design, interface design, and procedural design.

## 1.3 Definitions, Acronyms, and Abbreviations

DFD Data Flow Diagrams

GUI Graphical User Interface

Hashtable An object in the Java computer language used to store data

PIN Personal Identification Number

SDD Software Design Description

SRS Software Specification Document

Slurm A type of soda first thought of by the writers of Futurama

## 1.4 References

Domino’s Website (for slogan), 10/6/2014

Software Requirements Specification Document, 10/22/2014, Mismanagement, Inc.

Software Engineering A Practitioner's Approach 7th edition, 9/23/2014, Roger S. Pressman

Procedural Flow Design document (Mismanagement\_SDD\_ProceduralFlowDesign\_Ver1.0.docx), 10/5/2014, Mismanagement, Inc.

Domino's Pizza® Point of Sales Interface Diagram Book (Mismanagement\_SDD\_InterfaceDiagram\_Ver1.0.xlsx), Version 1.0 10/7/2014

## 1.5 System Overview

The software is intended for the use of employees at Domino’s Pizza establishments. The software allows employees to take orders, create receipts, configure prices and employee information.

# 2. Data Design

The data design specifies how the data of the program is going to be stored. This includes all major data structures and external file structures. For each data structure, the actions which can be performed on it are specified.

## 2.1 Internal Data Structures

### 2.1.1 Configurable Data

The program will read data from the following text file: prices.txt, tax.txt, and employeeInfo.txt.

The data from prices.txt will be stored in a one dimensional array where the index represents the item and the element represents the price.

The data from tax.txt will be stored as a variable.

The data from employeeInfo.txt will be stored in a variable length container, potentially a Hashtable, where each row contains two elements, the employee’s name and PIN.

### 2.1.2 Static Data

The program will contain a pizza data structure, which contains one pizza size and zero or more toppings. This data structure may exist in three different configurations: fully specified by user selections, predefined toppings for Meat Lovers specialty pizza with user specified size, and predefined toppings for Veggie Lovers specialty pizza with user specified size.

## 2.2 Data Flow Diagrams

### 2.2.1 Level 0 Data Flow Diagram

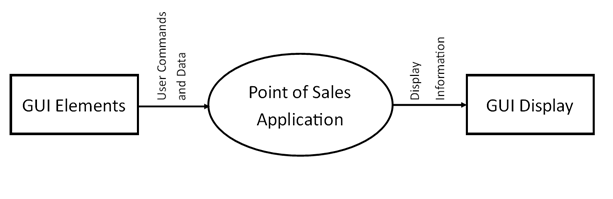


Figure 1.0 Level 0 Data Flow Diagram

### 2.2.2 Level 1 Data Flow Diagram

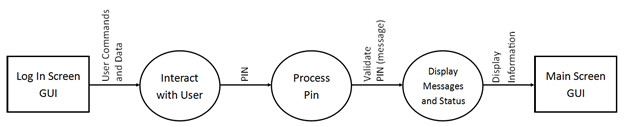


Figure 2.0 Level 1 Data Flow Diagram: Login Data

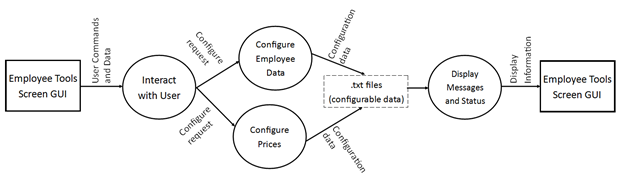


Figure 3.0 Level 1 Data Flow Diagram: Employee Data

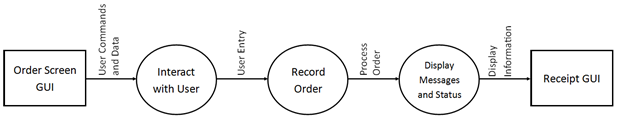


Figure 4.0 Level 1 Data Flow Diagram: Order Data

## 2.3 Data Dictionary

Employee Information Table (Hashtable) Fields:

* String for employee’s first and last name (key)
* String for employee’s PIN

Prices Array Fields:

* Index of array will indicate the item
* Double for item price

Topping Array Fields:

* String for topping name
* String (“0”, “1”) pseudo boolean for applicable topping to a given pizza

# 3. Architecture Design

The architectural design of a software project is simply the design of the entire software system. This includes the hierarchy of the modules and also which modules are present in the system. A good architectural design will create a clear and fair balance between cohesion (each module has only one distinct purpose), coupling (no two modules depend completely on each other), abstraction (seeing modules in full and not in detail), hierarchy (logical modules stem from others) and partitioning (logically grouping modules together) of the software modules.

## 3.1 Program Structure

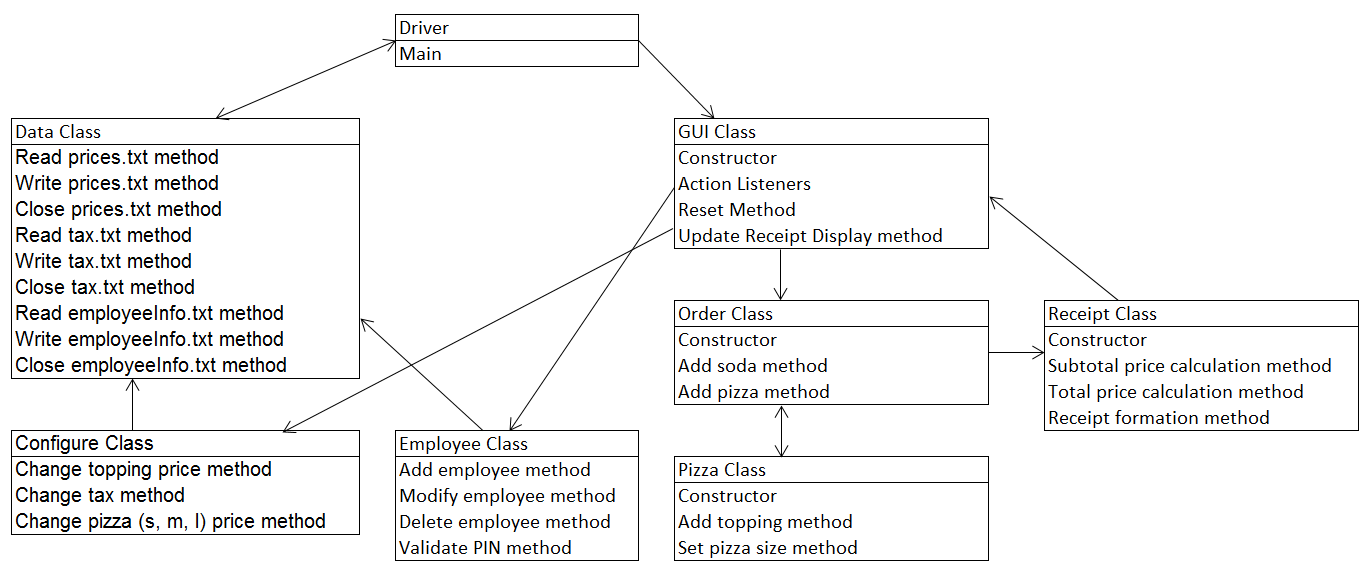


Figure 5.0 Program Structure Design

# 4. Interface Design

The Interface Design describes internal and external program interfaces as well as the design of human interface. Internal and external interface designs are based on the information obtained from the analysis model.

## 4.1 Internal System Interfaces

Our software is going to communicate with a text file in order to store and read in data. Classes and methods will then pull from the information provided by the text file in order to use prices, employee information, and tax percentage to calculate order costs.

## 4.2 External System Interfaces

Operating System (refer to Section 3.2 of the SRS document)

## 4.3 User Interfaces

Refer to Domino's Pizza® Point of Sales Interface Diagram Book (Mismanagement\_SDD\_InterfaceDiagram\_Ver1.0.xlsx), Version 1.0 10/7/2014.

# 5. Procedural Design

See Procedural Flow Design document (Mismanagement\_SDD\_ProceduralFlowDesign\_Ver1.0.docx), Version 1.0 10/7/2014.

# 6. Miscellaneous

There are no miscellaneous items.