****

**BN002/013/104 Year 2 Software Engineering and Testing**

**Assessment 2: Requirements Document**

**Submitted by: Names, Student numbers**

**Submission date**

**Declaration**

I herby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_

Author: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_

**Table of Contents**

# Title:

# Client: retailers

# Project Overview

The project is to develop an application for a generic till and counting system. This till application processes transactions. This till also includes the added feature of a till counter which makes calculating money easier. It also reduces the chance of errors while counting the till. The application then saves this information to a database for permanent storage and for future reference. An interface should be provided as the application will be used on touch screen monitors.

# Document Revision

Rev. 1.0 February 8, 2018– initial version

# Scope

This software will be designed and developed to take transactions in a retail environment. It will be compatible with any basic monitor system with a touch screen. The user of the till will enter what the customer will like to buy. The total is summed up, the cash is entered and change is calculated. The till will have an option to accept sterling notes, which enables this till to be used all over the country and in airports. This till contains an added counting feature for end of day summarization. It is a basic transaction based software that automates the counting process as much as possible. This will benefit employees, save time and lower risk of any mistakes. This till also accommodates permanent storage, by saving transactions and balances to the till.

1. **Walkthrough Scenarios**

*User:*

User should be able to select an item based on a button and an image located on the GUI. After all the items have been selected the user has the option to select subtotal or cancel the transaction. If the user clicks the cancel button, all the items will be voided. If the user clicks subtotal, the total cost of the transaction is displayed. The user can now enter the money given (to be converted if necessary) into the system, which then calculates the change. At the end of the Users shift, they can click the “End of Day” button, which begins the till counting process. The user will then be prompted to enter the money into the till. After user enters the money, they click calculate and the balance will be saved to the database.

1. **Software Requirements Analysis:**

***Functional Requirements:***

These are statements of services the system should provide – how the system should react to particular inputs and how it should behave in particular situations. Explicitly state what the system should do. Every major scenario should be represented by a use case. Diagrams are encouraged. UML Use case diagrams, Use case specifications (as legible screen dumps, typed listings or activity diagrams)

Can approach them from a *user* and *system* point of view.

*User* – high level abstract requirements, readable by someone with no detailed technical knowledge.

*System* – detailed description of what the system should do. Targeted at technical staff and project managers…

e.g.

# 5.1 User Requirements

# 5.2 System Requirements

5.2.1 Use Cases

5.2.2 Use Case Specification

5.2.3 Activity Diagrams

# 5.3 Non-functional Requirements:

These are constraints on the service or functions offered by the system e.g. timing constraints

# Graphical User Interface Design

# Technical Requirements and Feasibility:

System models – UML

Development language – Java

Persistent storage – database?

Interface & Software / Hardware APIs

## Conclusion (1-2 paragraphs)

Your conclusions and recommendations (feasibility of the proposed project)

Additional sections: Table of Contents, executive summary, Index

Checklist: Is your document complete and correct?

*Content:*

* Do the requirements state the customers’ needs
* Are you satisfied with all parts of the document
* Do you believe all parts are possible to implement
* Is each part of the document in agreement with all other parts
* Do the requirements avoid specifying a solution
* Do the requirements avoid specifying a design

*Completeness*:

* Are all the necessary interfaces specified – this includes input and output
* Are the specifications precise enough
* Are all sections from the document template included – if changed, why?

*Clarity*:

* Are all requirements reasonable?
* Is the level of details for each requirements appropriate?
* Are the requirements written in a language appropriate to the reader?
* Are all items clear and unambiguous?