Bun Talk Business

(Automation System)

Software Requirements Specification

Prepared by: Trishan Kanishka

**Prepared for**

**CSE 4002 Fundamentals of Programming**

**International College of Business & Technology**

Table of Contents

[1. Introduction 3](#_Toc146101699)

[1.1Purpose 3](#_Toc146101700)

[1.2 Scope 3](#_Toc146101701)

[2. Overall Description 3](#_Toc146101702)

[2.1 Product Perspective 3](#_Toc146101703)

[2.2 User Classes and Characteristics 3](#_Toc146101704)

[2.3 Operating Environment 3](#_Toc146101705)

[2.4 Design and Implementation Constraints 3](#_Toc146101706)

[3. Functional Requirements 3](#_Toc146101707)

[3.1 Authentication 3](#_Toc146101708)

[3.1.1 Login 3](#_Toc146101709)

[3.1.2 User Registration 4](#_Toc146101710)

[3.2 Main Menu 4](#_Toc146101711)

[3.2.1 Display Breakfast Items 4](#_Toc146101712)

[3.2.2 Select Items 4](#_Toc146101713)

[3.2.3 Print Bill 4](#_Toc146101714)

[3.2.4 Help 4](#_Toc146101715)

[4. Non-Functional Requirements 4](#_Toc146101716)

[4.1 Usability 4](#_Toc146101717)

[4.2 Performance 4](#_Toc146101718)

[4.3 Security 4](#_Toc146101719)

[4.4 Reliability 4](#_Toc146101720)

[5. System Interfaces 5](#_Toc146101721)

[5.1 ER Diagram Of the system 7](#_Toc146101722)

[5.2 flowchart Diagram 8](#_Toc146101723)

[5.3 Pseudo codes for functionalities 9](#_Toc146101724)

[6. Design Overview 11](#_Toc146101725)

# Introduction

1.1Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the Bun Talk Automation System. This system aims to automate bakery billing procedures to provide faster and more reliable service.

1.2 Scope

The Bun Talk Automation System will include functionalities for user authentication, displaying breakfast items, calculating and printing bills, providing help information, and allowing users to exit the system.

# Overall Description

2.1 Product Perspective

The Bun Talk Automation System is a standalone application that interacts with users to automate billing processes. It does interact with SQL Server database to data transactions.

2.2 User Classes and Characteristics

Admin: Responsible for using the system to create menu items.

Users: Could login to the system and place an order.

2.3 Operating Environment

The system will be designed to run on standard desktop or laptop computers with a modern operating system (e.g., Windows, macOS, Linux).

2.4 Design and Implementation Constraints

The system should be designed with modularity to allow for future enhancements and maintainability. It should also ensure data security and user privacy.

# Functional Requirements

## 3.1 Authentication

## 3.1.1 Login

* Users must provide a valid username and password.
* The system shall authenticate users based on their credentials.
* Successful login grants access to the main menu.

## 3.1.2 User Registration

New users could register in to the system by providing Name, Address, Phone number, and Email.

## 3.2 Main Menu

## 3.2.1 Display Breakfast Items

Users can view a list of breakfast items categorized by type (Foods and Beverages).

Items shall display their name, price, and description.

## 3.2.2 Select Items

Users can select one or more items from the menu.

## 3.2.3 Print Bill

* The system shall calculate the total cost of selected items.
* It shall generate an itemized bill with details of each item, quantity, and total cost.

## 3.2.4 Help

Users can access help section to get information to assist with using the application.

# 4. Non-Functional Requirements

## 4.1 Usability

The user interface shall be intuitive and user-friendly.

Help information shall be easily accessible.

## 4.2 Performance

The system shall respond to user interactions promptly.

## 4.3 Security

User credentials shall be securely stored and transmitted.

Access control shall be enforced based on user roles.

## 4.4 Reliability

The system shall be available for use during Bun Talk's operating hours.

It shall provide reliable and accurate billing information.

# 5. System Interfaces

The system have a graphical user interface (GUI) for user interaction.

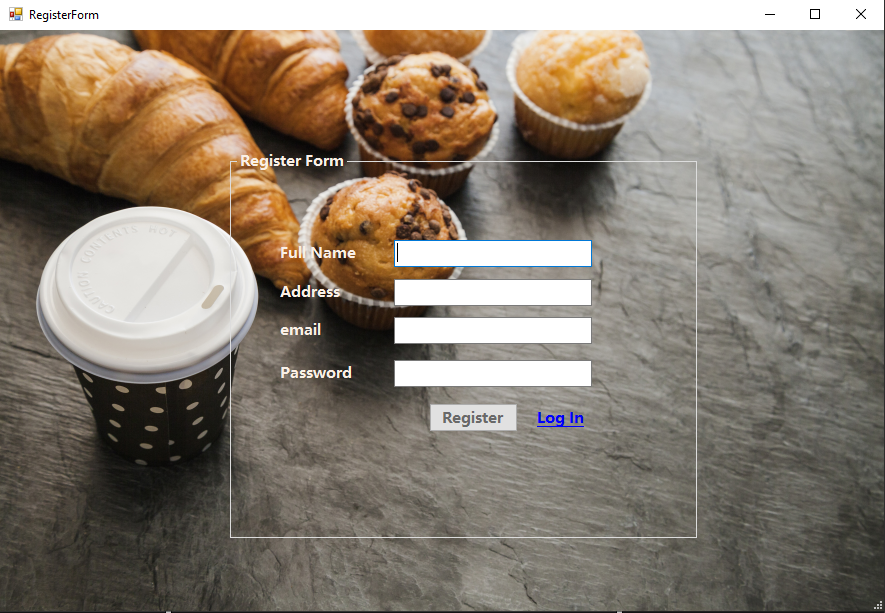


Figure 1 User Registration Form

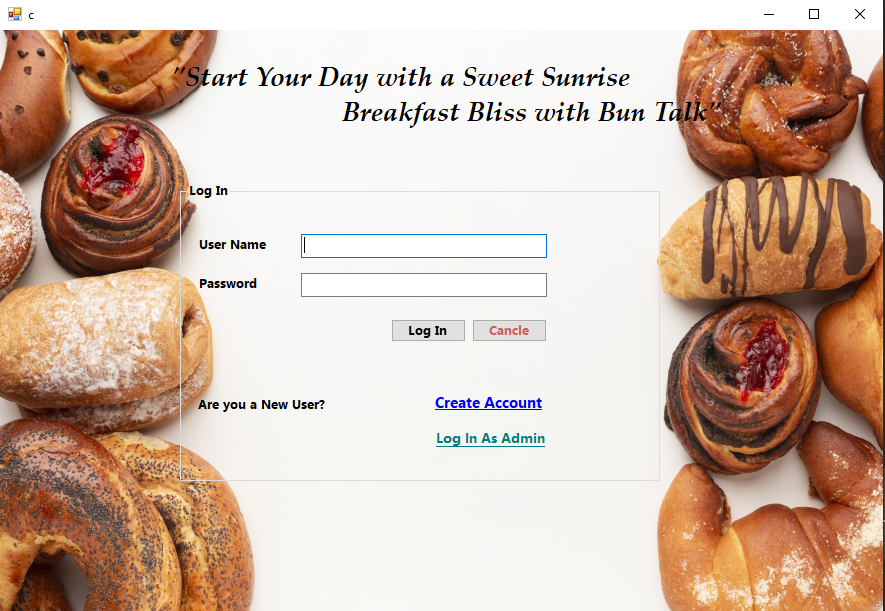


Figure 2 User Log in Form

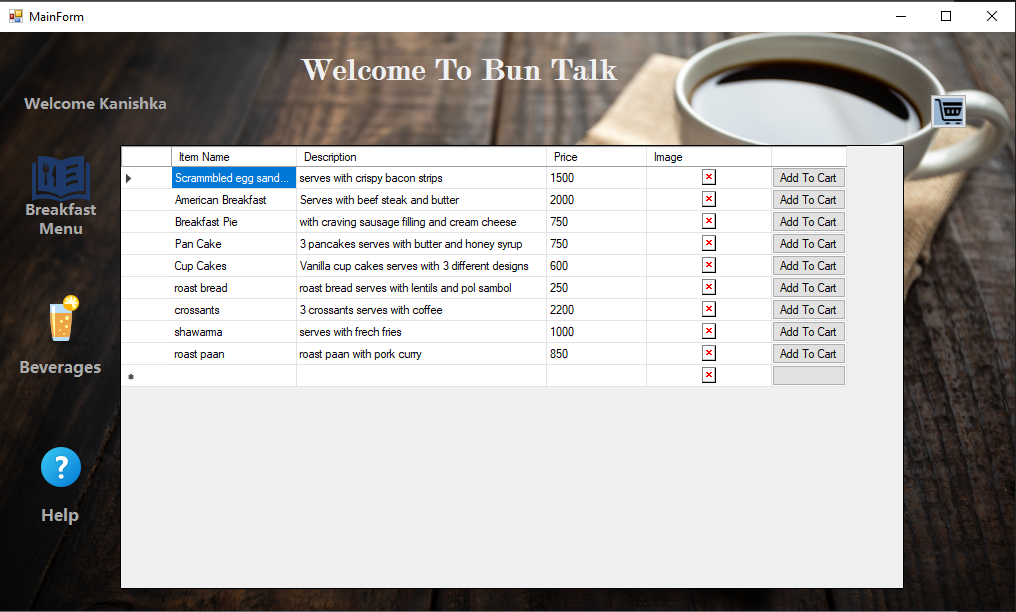


Figure 3 Menu Dashboard



Figure 4 Admin Dash Board

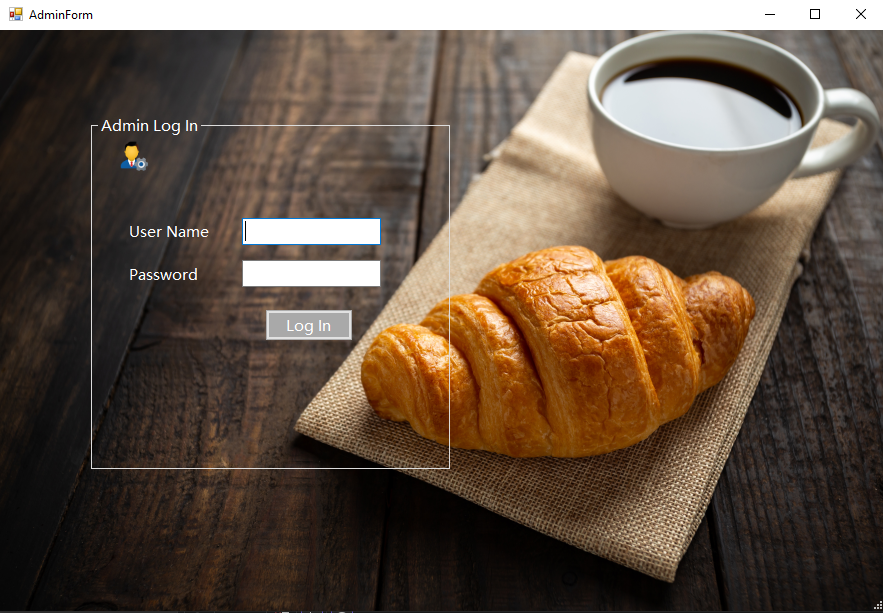
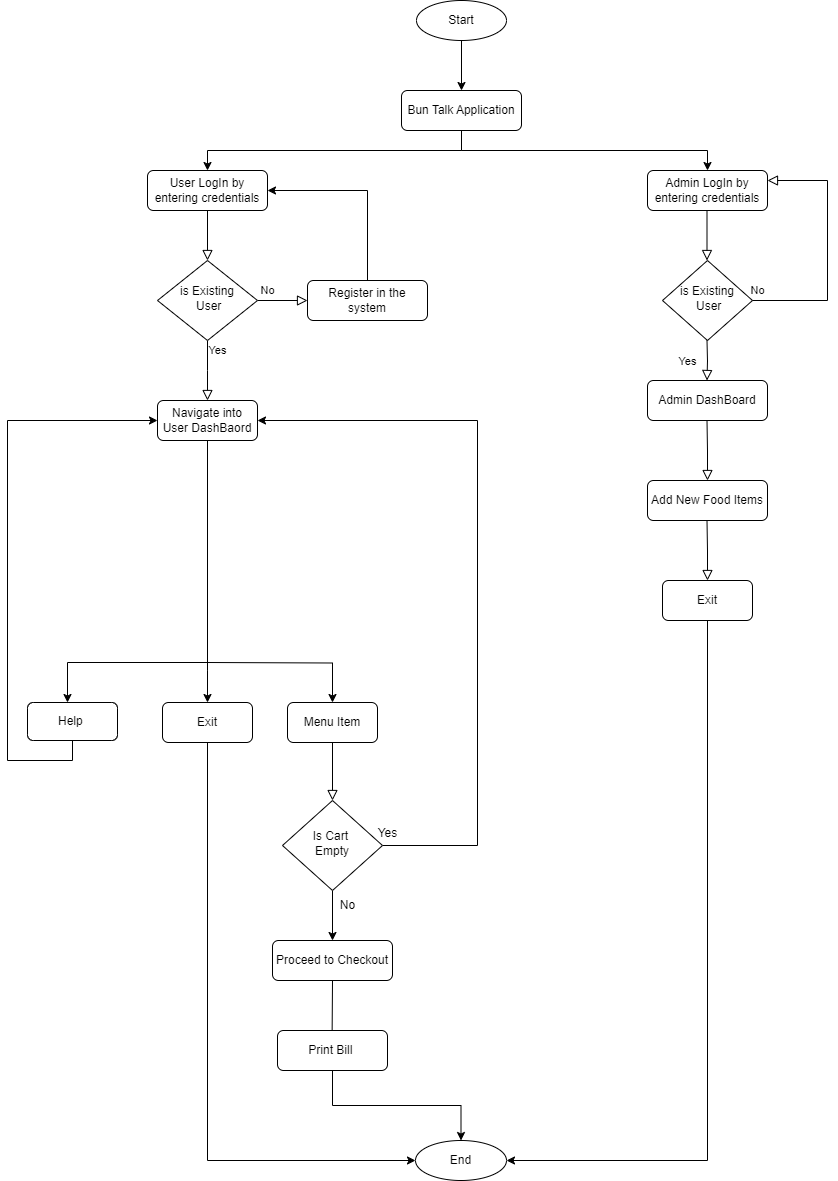


Figure 5 Admin Log In

# 5.1 ER Diagram Of the system

# 5.2 flowchart Diagram



# 5.**3 Pseudo** codes for functionalities

1. Pseudo codes for User Login

function userLogin():

display("Welcome to Bun Talk!")

username = getInput("Enter your username: ")

password = getInput("Enter your password: ")

if authenticateUser(username, password):

display("Login successful.")

mainMenu()

else:

display("Invalid credentials. Please try again.")

userLogin()

end;

1. Pseudo codes for Add to Cart

function displayBreakfastItems():

breakfastItems = getBreakfastMenu() // Retrieve breakfast items from the database

if isEmpty(breakfastItems):

display("Sorry, no breakfast items available.")

else:

display("Breakfast Menu:")

for item in breakfastItems:

display(item.name + " - $" + item.price)

end;

1. Pseudo codes Bill Calculation

function calculateBill(selectedItems):

totalCost = 0

for item in selectedItems:

totalCost = totalCost + item.price

return totalCost

end;

1. Pseudo codes for Admin Login

function adminLogin():

display("Admin Login:")

adminUsername = getInput("Enter admin username: ")

adminPassword = getInput("Enter admin password: ")

if authenticateAdmin(adminUsername, adminPassword):

display("Admin login successful.")

adminDashboard()

else:

display("Invalid admin credentials. Please try again.")

adminLogin()

end;

1. Pseudo codes add items to menu

function addItemToDatabase():

display("Admin Dashboard - Add Item:")

itemName = getInput("Enter the item name: ")

itemPrice = getInput("Enter the item price: ")

itemType = getInput("Enter the item Type: ")

itemDescription = getInput("Enter the item description: ")

if isNotEmpty(itemName) and isNotEmpty(itemPrice) and isNotEmpty(itemCategory):

newItem = createNewItem(itemName, itemPrice, itemCategory, itemDescription)

if saveItemToDatabase(newItem):

display("Item added successfully.")

else:

display("Failed to add the item to the database.")

else:

display("Item information incomplete. Please provide all required details.")

end;

# 6. Design Overview

The application consists of two main modules

* User Module
* Admin Module.
* User Module

1. Login Process

* User enters their username and password.
* The system verifies the credentials against a local database.
* If authentication is successful, the user proceeds to the Main Menu.
* If authentication fails, an error message is displayed.

1. Main Menu

Users are presented with options:

* Display Breakfast Items
* Select Items
* Calculate Bill
* Help
* Exit

1. Display Breakfast Items

* The system retrieves breakfast items from the database.
* It displays the menu items with their names, prices, and categories.

1. Select Items

* Users can select items clicking buttons.
* The selected items are stored in a temporary order in List structure.

1. Calculate Bill

* The system calculates the total cost of selected items.
* It displays the itemized bill to the user.

1. Help

* Users can access help information, which provides instructions on using the system.

1. Exit

* Users can exit the application.
* Admin Module

1. Admin Login

* Admins enter their admin username and password.
* The system verifies the admin credentials.
* If authentication is successful, the admin proceeds to the Admin Dashboard.
* If authentication fails, an error message is displayed.

1. Admin Dashboard
2. Add Item to Menu

* Admins can add new bakery items to the menu.
* They provide item details like name, price, item type, and description.
* The system validates and saves the new item to the database.

1. Database

* The system uses a local database to store breakfast items and user/admin account information.

1. Error Handling

* The system provides appropriate error messages and handles exceptions gracefully to ensure a user-friendly experience.

1. Simplifications

* The design focuses on core functionalities, excluding advanced features.
* The user interface is text-based for ease of implementation.
* User and admin management are handled within the application without external authentication services.