TCdev Snacks : Requirements

Product Documentation

Application Name:	TCdev Snacks
Document Description:	Requirements
Item Reference:	SNK_001

Version History

Version	Author	Status	Description	Date
V0.1	Thom Bedford	Initial Draft	Outlined Overview	03-21
V0.2	Thom Bedford	Draft	Defined User Requirements	03-23
V0.3	Thom Bedford	Draft	Defined Constraints, Assumptions and Dependencies, Limitations and External Interface Requirements	03-26
V0.4	Thom Bedford	Draft	Defined Software System Attributes	03-27
V0.5	Thom Bedford	Draft	Defined Contract Requirements	04-08
V1.0	Thom Bedford	Release	Reviewed and Released	04-10

Table of Contents

1. Overview

1.1. Synopsis

TCdev Snacks is a small resale business that specialises in the purchase and sale of snack food, either for vending machines or to sell behind the bar at pubs. Currently all records are paper-based or held on a series of spreadsheets, which can result in a lack of data consistency and integrity. The primary function of the proposed system is to simplify all existing processes within the business, and replace any spreadsheets with a database back-end and application front-end.

1.2. System Objective

The proposed system must integrate many areas of the business, including:

1. Supplier Records
2. Customer Records
3. Stock (Including Purchases and Sales)
4. Finance (Including Assets, Expenses, Creditors, Debtors etc)

Once implemented, the system would replace any existing paper-based or spreadsheet-based systems, simplifying the process for all actors.

With this in mind, the overall objective of this system is to reduce the time spent on data upkeep, giving actors more time to spend on acquiring new customers, managing deliveries, and finally increasing profit.

1.3. Benefits of the Proposed System

The benefits of the proposed system will be evident throughout the business.

1. Time Saving
    - 1.1. Existing paper-based and spreadsheet-based systems to be replaced with a fully integrated system reducing the time taken filling in various pieces of paperwork
    - 1.2. When receiving a phone order from a customer, the time taken to respond and record the details of the order will be much faster with implementation of the proposed system
    - 1.3. Due to the integration of documentation, the risk of failed data integrity is greatly reduced, avoiding a large amount of auditing time
2. Information Integrity
    - 2.1. The reduction of separate processes will result in a single central source of information that is available to all actors
3. Reduction in Effort and Errors
    - 3.1. With only a single system to maintain, the effort required throughout the business is significantly reduced
    - 3.2. With more time freed up from implementation of the new system, more care can be put in to the data upkeep, resulting in less errors
4. Reporting
    - 4.1. Once the system is fully implemented, the ability to create reports from data will become available, allowing the users to analyse buying trends and stock flow
5. Financial
    - 5.1. With less time required on data upkeep, actors can work on increasing the delivery frequency, eventually increasing the customer base
    - 5.2. A more structured approach to the ordering system will result in greater customer satisfaction, increasing repeat custom

1.4. Current Process Overview

1.4.1. Supplier Records

The process of adding a supplier is relatively simple and short, as no background checks are required.

1.4.2. Customer Records

When adding a new customer, various security checks must be made to ensure that the customer will keep up with any outstanding payments. Such security checks must be done through liaising with banks, therefore these are out of scope for this system.

1.4.3. Stock

Within the existing business model, stock lists are not maintained, meaning that any stock check for an order process must be done manually on receiving each individual order.

Any record of a stock purchase or stock sale is held in a new spreadsheet (copied from a template), meaning that the order repository currently holds hundreds of different spreadsheets.

Stock Purchase

Purchasing stock involves phoning a supplier. Once the order is approved, financial records of the transaction are written up and flagged that the stock is not yet delivered.

On delivery of the stock, the relevant financial records are updated and archived.

Stock Sale

Selling stock involves receiving a call from a customer (existing or new), checking stock levels, writing up a record of the transaction and flagging it as unpaid and undelivered.

On delivery of a stock sale, these records must be updated.

Payments for orders are generally received in bulk (for example bi-weekly or monthly), meaning that if a client has three outstanding stock orders, they may simply transfer one amount of cash across to the business, in which case all previous sale records must be reviewed and maintained.

1.4.4. Finances

In addition to the finances involved with stock purchase and sale, other sheets must be maintained including assets and expenses. Customers and Suppliers at this stage of the process are referred to as Debtors and Creditors respectively.

It is important that financial records are summarised for reporting purposes.

2. Stakeholders and Actors

2.1. Identification

The company TCdev Snacks comprises of two employees:

- Eddie Bedford – Manages the financial aspect of the business, the orders, and the maintenance of records.
- Peter Bedford – Handles deliveries of orders, and the acquisition of new customers and suppliers.

These are the two primary stakeholders (and actors) with regards to the proposed system.

There are also a number of secondary stakeholders which include all suppliers and customers to the business, whom will be indirectly affected by the implementation of the proposed system.

2.2. Analysis

Due to the size of the company, the roles described above are interchangeable at present, meaning that both primary stakeholders require full access to the proposed system. During Joint Requirements Development (JRD) sessions, Eddie will take the lead as the key stakeholder as he currently maintains all business records.

3. Software Requirements Specification

3.1. User Requirements / Use Cases

For the use cases displayed in this section, the following fields have been omitted:

- Version (as there is only one version of each use case, see document version history for any information regarding document changes)
- Actors (due to the size of the business as described in section 2, there is no need to identify specific actors)
- Stakeholders (as above)
- Author (there is only one author of this document)
- Date (there is no time constraint on completion of this project)

The following is a list of required high-level functions of the proposed system:

Use Case:	**Review Finance**	**Reference:**	01
Goal:	The user must have the ability to review the financial records at any time within the system.		
Triggers:	The user will be able to press a button to bring up the finance review screen. There are no preconditions for this other than being logged on to the system.		
Course of Events:	N/A		
Alternative Paths:	N/A		
Post-Conditions:	Once closing the finance review screen, the system will be in the exact state it was before triggering the event.		
Business Rules:	1. The financial records must never allow alteration by any user 2. The financial review screen must allow exporting to a spreadsheet		
Notes:	None		

Use Case:	**Review Customer Records**	**Reference:**	02
Goal:	The user must have the ability to review the customer records at any time within the system.		
Triggers:	The user will be able to press a button to bring up the customer review screen. There are no preconditions for this other than being logged on to the system.		
Course of Events:	N/A		
Alternative Paths:	✘ The customer records must be accessible through a sale review ✘ The customer records must be accessible through the financial review		
Post-Conditions:	Once closing the customer review screen, the system will be in the exact state it was before triggering the event.		
Business Rules:	N/A		
Notes:	✘ If there are no customer records, the user should be asked if they would like to add one		

Use Case:	**Review Supplier Records**	**Reference:**	03
Goal:	The user must have the ability to review the supplier records at any time within the system.		
Triggers:	The user will be able to press a button to bring up the supplier review screen. There are no preconditions for this other than being logged on to the system.		
Course of Events:	N/A		
Alternative Paths:	✘ The supplier records must be accessible through an purchase review ✘ The supplier records must be accessible through the financial review		
Post-Conditions:	Once closing the supplier review screen, the system will be in the exact state it was before triggering the event.		
Business Rules:	N/A		
Notes:	✘ If there are no supplier records, the user should be asked if they would like to add one		

Use Case:	**Review Product Records**	**Reference:**	04
Goal:	The user must have the ability to review the product records at any time within the system.		
Triggers:	The user will be able to press a button to bring up the product review screen. There are no		

Use Case:	**Review Product Records**	**Reference:**	04
	preconditions for this other than being logged on to the system.		
Course of Events:	N/A		
Alternative Paths:	✘ The product records must be accessible through a supplier record		
Post-Conditions:	Once closing the product review screen, the system will be in the exact state it was before triggering the event.		
Business Rules:	N/A		
Notes:	✘ The product review screen must be categorised by supplier ✘ If there are no product records (either overall or for a specific supplier), the user should be asked if they would like to add one		

Use Case:	**Create a Stock Purchase Order**	**Reference:**	05
Goal:	The user must have the ability to create a new stock purchase order at any time within the system. The process of filling out the purchase form will take no longer than 1 minute (as this is done while on the phone to the supplier).		
Triggers:	The user will be able to press a button to bring up the stock purchase order screen. There are no preconditions for this other than being logged on to the system.		
Course of Events:	1. Open Form 2. Select Supplier From List 3. Amend Order Details (If Required): 3.1. Date of Order (Default: Current Date) 3.2. Date of Delivery (Default: Blank) 3.3. Payment Sent (Default: False) 4. Add Products To Order: 4.1. Select Product From List 4.2. Enter Quantity (Default: 1) 4.3. Enter Purchase Price (Default: Price Value From Product Record) 5. Submit Order		
Alternative Paths:	✘ The stock purchase order screen must be accessible through a supplier record, when navigating to the screen using this method the Supplier information (point 2 above) will be automatically filled in		
Post-Conditions:	Once submitting the order, the following actions will occur: 1. Update Stock (If Delivery Date Set) 2. Update Creditor Record (If Payment Not Sent) 3. Close form Following this, the system will appear in the same state as before the form was opened.		
Business Rules:	N/A		
Notes:	✘ Setting the date of delivery will flag the order as complete and will update stock records to reflect this ✘ The user must have the ability to fill in all fields and set the purchase order to having been delivered and paid for. This is so that the user can enter stock items in to the system that may have been purchased at a wholesaler for example		

Use Case:	**Create a Stock Sale Order**	**Reference:**	06
Goal:	The user must have the ability to create a new stock sale order at any time within the system. The process of filling out the sale form will take no longer than 1 minute (as this is done while on the phone to the customer).		
Triggers:	The user will be able to press a button to bring up the stock sale order screen. There are no preconditions for this other than being logged on to the system.		
Course of Events:	1. Open Form 2. Select Customer From List 3. Amend Order Details (If Required): 3.1. Date of Order (Default: Current Date)		

Use Case:	**Create a Stock Sale Order**	**Reference:**	06
	3.2. Date of Delivery (Default: Blank) 3.3. Payment Received (Default: False) 4. Add Products To Order: 4.1. Select Product From List 4.2. Enter Quantity (Default: 1) 4.3. Enter Sale Price (Default: Price Value From Product Record) 5. Submit Order		
Alternative Paths:	✘ The stock sale order screen must be accessible through a customer record, when navigating to the screen using this method the Customer information (point 2 above) will be automatically filled in		
Post-Conditions:	Once submitting the order, the following actions will occur: 1. Update Stock (If Delivery Date Set) 2. Update Debtor Record (If Payment Not Received) 3. Close form Following this, the system will appear in the same state as before the form was opened.		
Business Rules:	✘ The user must be made aware of the current debt owed by the current customer		
Notes:	✘ Setting the date of delivery will flag the order as complete and will update stock records to reflect this ✘ The user must have the ability to fill in all fields and set the sale order to having been delivered and paid for. This is so that the user can enter stock sales that may have occurred outside the use of the system		

Use Case:	**Add an Asset**	**Reference:**	07
Goal:	The user must have the ability to add a new asset at any time within the system.		
Triggers:	The user will be able to press a button to bring up the asset management screen. There are no preconditions for this other than being logged on to the system.		
Course of Events:	1. Open Form 2. Select or Enter a Category 3. Enter Description 4. Enter Cost 5. Enter Date of Purchase (Default: Blank) 6. Enter Supplier 7. Commit Asset		
Alternative Paths:	N/A		
Post-Conditions:	Once committing the asset, the following actions will occur: 1. Update Financial Records (If Date of Purchase Complete) 2. Close form Following this, the system will appear in the same state as before the form was opened.		
Business Rules:	N/A		
Notes:	✘ Setting the date of purchase will flag the order as complete and will update financial records to reflect this		

3.2. Constraints

The following constraints will be apparent when developing the proposed system:

1. The system must be able to work offline on a standalone workstation (no internet or network)
2. The system must be able to work on low-end hardware (see Microsoft Windows XP Minimum Hardware Requirements)

3.3. Assumptions and Dependencies

The following can be assumed when developing the proposed system:

1. The users are computer literate, to the degree that they can easily navigate a Microsoft Operating System, and understand the capabilities of their hardware and software installed
2. A printer is available
3. An internet connection is not available

The implementation of the proposed system is dependant on the following:

1. A supported Microsoft Operating System installed and up-to-date

3.4. Limitations

The system will have the following limitations:

- ✘ Tax (Business Tax, VAT etc) will not be calculated by the finance system, this must be taken in to account by the user when managing data.
- ✘ Payroll will be left out of the system entirely.

3.5. External Interface Requirements

3.5.1. User Interface

The system will be produced for a single user, therefore only one interface is required.

The interface must include a large amount of validation to achieve maximum robustness and uptime, as well as being user-friendly in order to reduce the number of user-error occurrences to a maximum of 2 per day.

Following minimal training (1-2 hours), the user must be able to utilise all aspects of the system including creating a new stock purchase or sale order within 30 seconds.

3.5.2. Hardware Interface

The hardware interface is a PC or laptop.

The system will not be available across any network.

3.5.3. Software Interface

The proposed system must operate on any supported Microsoft operating system from Windows XP onwards.

The proposed system must not require any other software to be installed on the machine; however the following elements can be assumed:

- ✘ .NET Framework
- ✘ Microsoft Office
- ✘ Printer Drivers

3.6. Software System Attributes

3.6.1. Reliability

The system must have a 95% uptime, with a semi-automated backup procedure (i.e. backups must be automatically created at varying intervals (daily and weekly), however the safe storage of these backups is the responsibility of the user.

3.6.2. Performance

Response times are very important as the user must have the ability to open the system and navigate to the order screen quickly after receiving a phone call.

The application should take no longer than 10 seconds to load.

3.6.3. Availability

The system must be available at all times when logged on to the computer, meaning that there can be no scheduled maintenance times. Any maintenance routines must be manually started by the user.

3.6.4. Security

The software itself will have no security. Security is the responsibility of the business.

3.6.5. Maintainability

The user should be allowed to manually update and/or delete any data within the system.

A maintenance routine must be made available to the user that allows a timescale to be specified for past order history, after this timescale has expired, selecting the maintenance routine will remove any orders older than the lifespan specified.

3.6.6. Database Requirements

A database will be required to store all the data. This must be a local database that can be transported easily.

Consideration must be given to software availability (see section 3.5.3).

4. Contract Requirements

This section details a list of contract style requirements which will be used to measure the success of the system.

4.1. Scoring

The requirements will be measured on review of the final system by their priority. A scoring system will be used here to determine implementation progress at any stage throughout development:

- Critical Priority: 5 points
- High Priority: 3 points
- Medium Priority: 2 points
- Low Priority: 1 point

4.2. Requirements

The requirements are broken down by application area:

4.2.1. Finance

Reference	Requirement Description	Priority
Finance_01	Ability to view basic balance sheet at any time	Critical
Finance_02	Ability to print balance sheet at any time	Critical
Finance_03	Ability to expand sections (i.e. expand 'Total Assets' to show Stock, Bank Accounts, Office Equipment etc.)	Medium
Finance_04	Must include information: Total Running Income	Critical
Finance_05	Must include information: Total Running Outgoings	Critical
Finance_06	Must include information: Total Profit / Loss	Critical
Finance_07	Must include information: Total Assets	Critical
Finance_08	Must include information: Total Running Balance	Critical
Finance_09	Must include information: Total Creditors (Suppliers / Money Owing)	Critical
Finance_10	Must include information: Total Debtors (Customers / Money Owed)	Critical
Finance_11	The finance sheet should be updated automatically when completing a transaction elsewhere within the system, however a refresh button would be acceptable	High
Finance_12	None of the finance information should be editable from this screen under any circumstance.	High
Finance_13	Any debts should be highlighted in red	Medium
Finance_14	Any assets should be highlighted in black or blue	Medium
Finance_15	The user should be alerted if the books are currently unbalanced	Low

4.2.2. Customer

Reference	Requirement Description	Priority
Customer_01	Ability to view customer details at any time	Critical
Customer_02	Ability to search customer lists	High
Customer_03	Ability to view customer order history	Critical

Reference	Requirement Description	Priority
Customer_04	Must include information: Company Name	Critical
Customer_05	Must include information: Representative Name	Critical
Customer_06	Must include information: Contact Number	Critical
Customer_07	Must include information: Money Owed	Critical
Customer_08	Must include information: Billing Address	Critical
Customer_09	Must include information: Delivery Address	Critical
Customer_10	From the customer screen, the user should be able to start the creation of a sale order, upon which the screen should be automatically populated with the customers details	High
Customer_11	A borrow limit should be recorded against each customer, and the user should be alerted when any orders would exceed this limit	Medium
Customer_12	The user should always be made aware of any outstanding debts the customer has	Low

4.2.3. Supplier

Reference	Requirement Description	Priority
Supplier_01	Ability to view supplier details at any time	Critical
Supplier_02	Ability to search supplier lists	High
Supplier_03	Ability to view supplier order history	Critical
Supplier_04	Ability to view product list	High
Supplier_05	Must include information: Company Name	Critical
Supplier_06	Must include information: Representative Name	Critical
Supplier_07	Must include information: Contact Number	Critical
Supplier_08	Must include information: Money Owed	Critical
Supplier_09	Must include information: Payment Address	Critical
Supplier_10	From the supplier screen, the user should be able to start the creation of a purchase order, upon which the screen should be automatically populated with the suppliers details	High
Supplier _11	The user should always be made aware of any outstanding debts the owed to the supplier	Medium

4.2.4. Product

Reference	Requirement Description	Priority
Product_01	Ability to view products lists at any time	Critical
Product_02	Ability to search product lists	Medium
Product_03	Ability to order product lists	Low
Product_04	Must include information: Description	Critical
Product_05	Must include information: Buying Price	Critical
Product_06	Must include information: Number in Stock	Critical
Product_07	An 'Average Weekly Units' (AWU) should be generated for each product based on sales data	Medium
Product_08	The ability to rank products against each other in terms of popularity from past orders should be available to the user	Low
Product_09	A weekly profit should be generated based on the product price and AWU	Low

Reference	Requirement Description	Priority
Product_10	The ability to rank products against each other in terms of profit from past orders should be available to the user	Low
Product_11	Stock order recommendations should be made to the user based on sales data	Low

4.2.5. Stock Purchase

Reference	Requirement Description	Priority
Purchase_01	Ability to view stock purchase records at any time	Critical
Purchase_02	Ability to search stock purchase records	High
Purchase_03	Ability to order stock purchase records	High
Purchase_04	Must include information: Date of Order (Order Level)	Critical
Purchase_05	Must include information: Date of Delivery (Order Level)	Critical
Purchase_06	Must include information: Payment Sent (Order Level)	Critical
Purchase_07	Must include information: Product (Item Level)	Critical
Purchase_08	Must include information: Quantity (Item Level)	Critical
Purchase_09	Must include information: Purchase Price (Item Level)	Critical
Purchase_10	The user should have the ability to either leave a purchase order incomplete, undelivered, or complete depending on how the purchase has been processed (for example the stock may have been purchased at a wholesaler in which case the order would be delivered and paid for)	High
Purchase_11	The user should have the ability to pay for multiple orders simultaneously (meaning that one transaction should be made to a supplier, which would then automatically complete multiple purchase orders)	Medium

4.2.6. Stock Sale

Reference	Requirement Description	Priority
Sale_01	Ability to view stock sale records at any time	Critical
Sale_02	Ability to search stock sale records	High
Sale_03	Ability to order stock sale records	High
Sale_04	Must include information: Date of Order (Order Level)	Critical
Sale_05	Must include information: Date of Delivery (Order Level)	Critical
Sale_06	Must include information: Payment Received (Order Level)	Critical
Sale_07	Must include information: Product (Item Level)	Critical
Sale_08	Must include information: Quantity (Item Level)	Critical
Sale_09	Must include information: Sale Price (Item Level)	Critical
Sale_10	The user should have the ability to either leave a sale order incomplete, undelivered, or complete depending on how the sale has been processed (for example the stock may have been sold outside of the system, in which case the order would be delivered and paid for)	High
Sale_11	The user should have the ability to receive payment for multiple orders simultaneously (meaning that one transaction should be made from a customer, which would then automatically complete multiple sale orders)	Medium

4.2.7. Assets

Reference	Requirement Description	Priority
Assets_01	Ability to view asset lists at any time	High
Assets_02	Ability to categorise assets	Medium
Assets_03	Ability to view assets by category	Medium
Assets_04	Ability to search assets	Low
Assets_05	Must include information: Description	High
Assets_06	Must include information: Cost	High
Assets_07	Must include information: Date of Purchase	High
Assets_08	Must include information: Supplier	High
Assets_09	Must include information: Category	Medium

4.2.8. Maintenance

Reference	Requirement Description	Priority
Maintenance_01	Any historical orders over a certain period should be flagged to the user as historic, in addition any client that has not made an order within a specified timeframe should also be flagged as no longer active	Medium
Maintenance_02	The user must have the ability to manually run maintenance routines that will clear out historical transactions	Medium

4.3. Summary

From the scoring system laid out previously, we have the following requirement priorities (from a total of 80 requirements):

- Critical Priority (42): 210
- High Priority (18): 54
- Medium Priority (14): 28
- Low Priority (8): 8

Total Potential: 300 points

5. Authorisation

The below table will detail any alterations required prior to document authorisation. If no alterations are listed and a signature is provided, this document version should be considered final.

Version	Stakeholder	Alterations Required	Signature
V1.0	Eddie Bedford		
V1.0	Pete Bedford		