

# Software Requirements Specification

---

## Business Accounting and Payroll

Anita Kotagiri (kot) and Will Hardy (wehardy)

September 24, 2007

### **Abstract**

This document contains the software requirements specification for the Business Accounting System. The goal of this document is to clearly specify the requirements for the system to be developed.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Scope . . . . .	1
1.3	Product . . . . .	1
1.4	The client . . . . .	2
1.5	The development team . . . . .	2
1.6	Definitions and acronyms and abbreviations . . . . .	2
1.6.1	Acronyms . . . . .	2
1.6.2	Definition of general terms . . . . .	3
1.6.3	Universal Resource Identifiers . . . . .	3
1.7	References . . . . .	3
1.8	Overview . . . . .	3
<b>2</b>	<b>Overall Description</b>	<b>5</b>
2.1	Product context . . . . .	5
2.2	Existing systems . . . . .	5
2.2.1	Manual system . . . . .	5
2.2.2	Spreadsheet . . . . .	6
2.2.3	Commercial software . . . . .	6
2.2.4	Open source software . . . . .	7
2.3	Proposed System . . . . .	7
2.3.1	Aims . . . . .	8
2.3.2	System Functionality . . . . .	8
2.4	Users . . . . .	10
2.5	Accounting Concepts . . . . .	10
2.5.1	What is accounting? . . . . .	10

2.5.2	Users of accounting information . . . . .	10
2.5.3	The double-entry system . . . . .	12
2.5.4	Accounts . . . . .	12
2.5.5	The accounting equation . . . . .	13
2.5.6	Financial statements . . . . .	13
<b>3</b>	<b>Functional Requirements</b>	<b>15</b>
3.1	Overview . . . . .	15
3.1.1	Layout . . . . .	15
3.1.2	Rankings . . . . .	15
3.1.3	Interface requirements . . . . .	15
<b>4</b>	<b>Functional Requirements: Global</b>	<b>16</b>
4.1	General . . . . .	16
[FR-1]	Undo functionality . . . . .	16
[FR-2]	Audit trail . . . . .	16
[FR-3]	Save . . . . .	16
[FR-4]	Localisation requirements . . . . .	17
[FR-5]	Internationalisation requirements (e.g. BSB). . . . .	17
4.2	Field lengths . . . . .	17
[FR-6]	Definitions . . . . .	17
[FR-7]	Names . . . . .	18
[FR-8]	Generic field length . . . . .	18
[FR-9]	Number field length . . . . .	18
[FR-10]	Comment lengths . . . . .	18
[FR-11]	Currency numbers. . . . .	19
4.3	Input / output . . . . .	19
[FR-12]	Copy and paste . . . . .	19

[FR-13]	Import file formats . . . . .	19
[FR-14]	Export file formats . . . . .	19
[FR-15]	Backup management. . . . .	20
4.4	Design Guidelines . . . . .	20
[FR-16]	Authorisation. . . . .	20
[FR-17]	Shortcuts . . . . .	20
[FR-18]	Spell check . . . . .	21
[FR-19]	Calculator . . . . .	21
[FR-20]	Date calendar. . . . .	21
4.5	Help . . . . .	22
[FR-21]	Help framework . . . . .	22
4.6	Data entry . . . . .	22
[FR-22]	Auto completion . . . . .	22
4.7	Search . . . . .	23
[FR-23]	Searching transactions . . . . .	23
[FR-24]	Expert search. . . . .	23
[FR-25]	Global searching . . . . .	24
[FR-26]	Natural language searches . . . . .	24
4.8	Assumptions . . . . .	25
[FR-27]	Tax Assumption . . . . .	25
<b>5</b>	<b>Functional Requirements: Company Data</b>	<b>26</b>
5.1	Company Details . . . . .	26
[FR-28]	Company details . . . . .	26
[FR-29]	Company details interface . . . . .	26
[FR-30]	GST registration . . . . .	27
[FR-31]	GST registration interface . . . . .	27

- [FR-32] Custom company details . . . . . 28
- 5.2 Chart of Accounts . . . . . 28
  - [FR-33] Account types . . . . . 28
  - [FR-34] Create new accounts . . . . . 29
  - [FR-35] Special accounts . . . . . 29
  - [FR-36] Special category . . . . . 30
  - [FR-37] Account categories . . . . . 30
  - [FR-38] Edit account category . . . . . 30
  - [FR-39] Account numbers . . . . . 30
  - [FR-40] Account names . . . . . 31
  - [FR-41] Account interface . . . . . 31
  - [FR-42] Edit accounts . . . . . 31
  - [FR-43] Edit accounts interface . . . . . 32
  - [FR-44] View chart of accounts interface . . . . . 32
  - [FR-45] User codes . . . . . 32
- 5.3 Default Accounts & Categories . . . . . 32
  - [FR-46] Default categories . . . . . 33
  - [FR-47] Default accounts . . . . . 33
  - [FR-48] Default accounts for a service business . . . . . 34
  - [FR-49] Default accounts for a wholesale business . . . . . 35
- 5.4 General . . . . . 35
  - [FR-50] Multiple companies . . . . . 35
- 5.5 Customer list . . . . . 36
  - [FR-51] Import customer list . . . . . 36
  - [FR-52] Import customer list interface . . . . . 36
- 5.6 Supplier list . . . . . 36
  - [FR-53] Import supplier list . . . . . 36

[FR-54]	Import supplier list interface . . . . .	37
<b>6</b>	<b>General Journal</b>	<b>37</b>
6.1	General . . . . .	37
[FR-55]	Transactions . . . . .	37
[FR-56]	Entries . . . . .	38
[FR-57]	Tax transaction accounting . . . . .	38
[FR-58]	General Journal Interface . . . . .	39
<b>7</b>	<b>Functional Requirements: Customers</b>	<b>40</b>
7.1	Debtors . . . . .	40
[FR-59]	Debtor details . . . . .	41
[FR-60]	Add new debtor functionality . . . . .	41
[FR-61]	Add new debtor interface . . . . .	42
[FR-62]	Update existing debtor functionality . . . . .	42
[FR-63]	Inactive debtors . . . . .	42
[FR-64]	List of debtors functionality . . . . .	43
[FR-65]	List of debtors interface: . . . . .	43
[FR-66]	Balance of all accounts receivable interface . . . . .	43
[FR-67]	Delete debtor functionality . . . . .	44
[FR-68]	Relinking debtor transactions . . . . .	44
[FR-69]	Delete debtor interface . . . . .	44
7.2	Quotes . . . . .	45
[FR-70]	Quote functionality . . . . .	45
[FR-71]	Quote interface . . . . .	45
[FR-72]	Quote printing . . . . .	46
7.3	Sales orders . . . . .	46
[FR-73]	Sales order functionality . . . . .	46

[FR-74]	Sales order interface . . . . .	46
[FR-75]	Sales order printing . . . . .	47
7.4	Invoice . . . . .	47
[FR-76]	Invoice functionality . . . . .	47
[FR-77]	Invoice interface . . . . .	48
[FR-78]	Credit sales accounting . . . . .	48
[FR-79]	Cash sales accounting . . . . .	48
[FR-80]	Invoice printing . . . . .	49
7.5	Receive payments from debtors . . . . .	49
[FR-81]	Receive Payment Functionality . . . . .	49
[FR-82]	Record debtor payment interface . . . . .	50
[FR-83]	Payment accounting . . . . .	50
7.6	Credit notes . . . . .	51
[FR-84]	Credit note functionality . . . . .	51
[FR-85]	Credit note interface . . . . .	51
[FR-86]	Credit note printing . . . . .	51
7.7	Customer deposits . . . . .	51
[FR-87]	Customer deposit account . . . . .	52
[FR-88]	Customer deposit accounting . . . . .	52
[FR-89]	Receipt of deposit . . . . .	52
<b>8</b>	<b>Functional Requirements: Suppliers</b>	<b>53</b>
8.1	Creditor details . . . . .	53
[FR-90]	Creditor details . . . . .	53
[FR-91]	Add new creditor functionality . . . . .	54
[FR-92]	Add new creditor interface . . . . .	54
[FR-93]	Update existing creditor functionality . . . . .	54
[FR-94]	Inactive creditors . . . . .	55

[FR-95]	List of creditors functionality . . . . .	55
[FR-96]	List of creditors interface . . . . .	56
[FR-97]	Balance of all accounts payable interface . . . . .	56
[FR-98]	Delete creditor functionality . . . . .	56
[FR-99]	Relinking creditor transactions . . . . .	57
[FR-100]	Delete creditor interface . . . . .	57
<b>8.2</b>	<b>Purchase order . . . . .</b>	<b>57</b>
[FR-101]	Purchase order functionality . . . . .	58
[FR-102]	Purchase order interface . . . . .	59
[FR-103]	Purchase order printing. . . . .	59
<b>8.3</b>	<b>Bill . . . . .</b>	<b>59</b>
[FR-104]	Bill functionality . . . . .	60
[FR-105]	Bill interface . . . . .	61
[FR-106]	Credit purchase accounting . . . . .	61
[FR-107]	Cash purchase accounting . . . . .	62
[FR-108]	Bill printing . . . . .	62
[FR-109]	List of bills . . . . .	63
[FR-110]	List of bills interface. . . . .	63
<b>8.4</b>	<b>Make payments to creditors . . . . .</b>	<b>64</b>
[FR-111]	Make payment functionality . . . . .	64
[FR-112]	Creditor payment accounting . . . . .	64
[FR-113]	Make payment interface . . . . .	65
[FR-114]	Print details of payment . . . . .	65
[FR-115]	List payment history. . . . .	65
[FR-116]	List payment history interface . . . . .	66
<b>9</b>	<b>Functional Requirements: Banking</b>	<b>67</b>
<b>9.1</b>	<b>Bank accounts . . . . .</b>	<b>67</b>



[FR-117]	Bank account . . . . .	67
[FR-118]	Bank account interface . . . . .	67
[FR-119]	Institution details . . . . .	68
[FR-120]	Institution details interface . . . . .	68
[FR-121]	List of bank accounts functionality . . . . .	69
[FR-122]	List of bank accounts interface . . . . .	69
[FR-123]	List of transactions . . . . .	70
[FR-124]	List of transactions interface . . . . .	70
<b>9.2</b>	<b>Bank reconciliation . . . . .</b>	<b>70</b>
[FR-125]	Reconciliation statement . . . . .	70
[FR-126]	Reconciliation with multiple accounts . . . . .	71
[FR-127]	Reconciliation dates . . . . .	71
[FR-128]	Reconciliation opening balance . . . . .	71
[FR-129]	Separation of transaction types . . . . .	71
[FR-130]	Reconciled transactions . . . . .	72
[FR-131]	Additional transactions. . . . .	72
[FR-132]	Final reconciliation . . . . .	73
[FR-133]	Reconciliation adjustments . . . . .	73
[FR-134]	Post reconciliation . . . . .	73
[FR-135]	Reconciliation interface . . . . .	74
[FR-136]	Reconciliation printing. . . . .	74
<b>10</b>	<b>Functional Requirements: Reporting</b>	<b>75</b>
<b>10.1</b>	<b>Balance sheet . . . . .</b>	<b>75</b>
[FR-137]	Balance sheet functionality . . . . .	75
[FR-138]	Balance sheet interface. . . . .	76
<b>10.2</b>	<b>Profit and loss statement . . . . .</b>	<b>76</b>
[FR-139]	Profit and loss functionality . . . . .	76

[FR-140]	Profit and Loss interface . . . . .	77
<b>11 Functional Requirements: Taxation</b>		<b>78</b>
11.1	Taxes . . . . .	78
[FR-141]	Australian taxes. . . . .	78
[FR-142]	Updating tax tables . . . . .	78
[FR-143]	Updating tax tables from a file. . . . .	79
[FR-144]	Tax reporting period . . . . .	79
[FR-145]	International Taxation . . . . .	79
11.2	Tax Reports . . . . .	80
[FR-146]	GST report. . . . .	80
[FR-147]	GST report interface. . . . .	80
[FR-148]	GST report printing . . . . .	80
[FR-149]	BAS report. . . . .	81
[FR-150]	BAS report interface. . . . .	81
[FR-151]	BAS report printing . . . . .	81
<b>12 Additional requirements</b>		<b>82</b>
12.1	Customers . . . . .	82
[FR-152]	Delete quote . . . . .	82
[FR-153]	Delete quote interface . . . . .	82
[FR-154]	Delete sales order . . . . .	82
[FR-155]	Delete sales order interface . . . . .	82
[FR-156]	Delete invoice . . . . .	83
[FR-157]	Delete invoice interface . . . . .	83
[FR-158]	Credit note accounting . . . . .	83
<b>13 Non-Functional Requirements</b>		<b>84</b>
13.1	Installation . . . . .	84

[NFR-1]	Windows packaged installation . . . . .	84
[NFR-2]	Linux and Mac packaged installation . . . . .	84
[NFR-3]	Installation time . . . . .	84
[NFR-4]	Deinstallation . . . . .	84
13.2	Performance . . . . .	85
[NFR-5]	Fast start-up . . . . .	85
[NFR-6]	Reporting results from searches . . . . .	85
[NFR-7]	Responsive. . . . .	85
13.3	Security . . . . .	85
[NFR-8]	Zero tolerance data loss . . . . .	86
[NFR-9]	Authentication . . . . .	86
13.4	Platform . . . . .	86
[NFR-10]	Windows compatibility. . . . .	86
[NFR-11]	Apple OS X compatibility . . . . .	86
[NFR-12]	Future Apple OS X compatibility . . . . .	87
[NFR-13]	Linux compatibility . . . . .	87
[NFR-14]	Future Linux compatibility . . . . .	87
13.5	Minimum Hardware Requirements . . . . .	87
[NFR-15]	Minimum Hardware Requirements . . . . .	87
13.6	Extensibility . . . . .	88
[NFR-16]	Plug-ins . . . . .	88
13.7	Custom report modules . . . . .	88
[NFR-17]	Basic reporting language . . . . .	88
[NFR-18]	Interpreted reporting language. . . . .	88
[NFR-19]	End user report interface . . . . .	88
13.8	Environment . . . . .	89
[NFR-20]	Multiple users . . . . .	89

[NFR-21]	Client/server . . . . .	89
[NFR-22]	Peer to peer . . . . .	89
13.9	Licence . . . . .	89
[NFR-23]	Licence . . . . .	89
13.10	Internationalisation . . . . .	90
[NFR-24]	Back-end language independence . . . . .	90
[NFR-25]	Internationalisation . . . . .	90
[NFR-26]	Localisation . . . . .	90
13.11	File formats . . . . .	90
[NFR-27]	File formats . . . . .	90
13.12	Language . . . . .	91
[NFR-28]	Chosen development language. . . . .	91
[NFR-29]	Acceptable databases . . . . .	91
[NFR-30]	Database transaction support . . . . .	91
[NFR-31]	Database support . . . . .	92
<b>14</b>	<b>Use cases</b>	<b>93</b>
14.1	Small business supplier . . . . .	93
14.1.1	Overview . . . . .	93
14.1.2	Process . . . . .	93
14.1.3	Reports . . . . .	93
14.2	Lessor . . . . .	94
14.2.1	Overview . . . . .	94
14.2.2	Process . . . . .	94
14.2.3	Reports . . . . .	94
14.3	Bulk data entry . . . . .	94
14.4	Process . . . . .	94
14.4.1	Traits of data entry . . . . .	95

<i>CONTENTS</i>	xii
<b>15 Acceptance Criteria</b>	<b>96</b>
15.1 SRS Acceptance . . . . .	96
15.2 Product Acceptance . . . . .	96
15.3 Product Deliverables . . . . .	96
<b>16 Client Sign-off</b>	<b>97</b>
16.1 Changing Requirements . . . . .	97
16.2 SRS change management . . . . .	97
16.3 SRS Sign-off Iteration 1 . . . . .	98
16.4 SRS Sign-off Iteration 2 . . . . .	98
16.5 Product Acceptance . . . . .	99
<b>Appendix A Requirement Ranking</b>	<b>100</b>
<b>Appendix B Glossary</b>	<b>101</b>

**List of Tables**

1	Acronyms used in this document . . . . .	2
2	Definitions of general terms . . . . .	3

**List of Figures**

1	Document Hierarchy . . . . .	1
---	------------------------------	---

# 1 Introduction

## 1.1 Purpose

This document will define an agreement between the client Cybersource and the developers Team Chilli, by formally outlining the functional requirements and non-functional requirements (performance constraints) for the Business Accounting System. These requirements form boundaries and constraints for the project's design process as well as providing a reference for the development of a test process.

The document also provides the design team with informative context, drawn from the domain experience of various contributors. In doing this, the SRS will house all domain knowledge required for the development of the final product.

Figure 1 shows the hierarchy of documents descending from this SRS.

Figure 1: Document Hierarchy

The intended readers of this Software Requirements Specification are Cybersource, Team Chilli and the Team Chilli Supervisor Chris Edwards.

## 1.2 Scope

This Software Requirement Specification outlines the functional and non-functional requirements and constraints of the system, ranking them as *essential*, *desirable* or *optional*. It does not provide any design specifications, except for constraints explicitly requested by the client.

## 1.3 Product

The proposed product will referred to in this document as the Business Accounting System Product or the Product.

## 1.4 The client

The client for this project is:

Cybersource  
 Level 4, 10 Queen Street  
 Melbourne 3000 Australia  
 PH: (03) 9621 2377

The contacts for Cybersource are Steven D'Aprano, and Con Zymaris.

## 1.5 The development team

- Will Hardy
- Andrew Homer
- Anita Kotagiri
- Derek Lee Hon Chong
- Chin Yee Lee
- Lisa Luu
- Goran Stefkovski
- Sreedhar Valicherla
- Ronald Siswara Wong
- Alex Wong Tchen Ee
- Dennis Guang Yang

## 1.6 Definitions and acronyms and abbreviations

### 1.6.1 Acronyms

CLO	Client Liaison Officer
IEEE	Institute of Electrical and Electronic Engineers
SDD	Software Design Document
SPMP	Software Project Management Plan
SQAP	Software Quality Assurance Plan
SRS	Software Requirements Specification
TP	Test Plan
UD	User Documentation

Table 1: Acronyms used in this document



### 1.6.2 Definition of general terms

Term	Definition
433-440	The forth year Software Engineering Project subject taught by the University of Melbourne. The Business Accounting System project is being undertaken as part of this subject.
Client	The client for this project is Cybersource. See Section 1.4.
Product	Everything deliverable to the Client at the completion of the Project, known as Business Accounting System. See section 15.3 for a list of deliverables.
Project	The project as a whole, including the Product, processes and any relevant documentation to the development of the product. The project in this sense will cease on completion of the subject 433-440.
System	The system as a whole, including the software and any surrounding processes that integrate the software into the business environment.
Team Supervisor	Chris Edwards. His role and responsibilities can be found in the SQAP.

Table 2: Definitions of general terms

### 1.6.3 Universal Resource Identifiers

svn:/        Repository  
 wiki:/      Wiki  
 group:/    CSSE group directory  
 share:/    chilliShare directory  
 www:/      Group website

## 1.7 References

1. IEEE Std. 830-1998 “*IEEE Recommended Practice for Software Requirements Specifications*”
2. IEEE Std. 730.1-1995 “*IEEE Standard for Software Quality Assurance Planning*”
3. Team Chilli, “*433-440 Software Quality Assurance Plan (SQAP) 2006*”
4. Team Chilli, “*433-440 Software Process Management Plan (SPMP) 2006*”

## 1.8 Overview

The following sections of the SRS contain:

- **An overall description of the Product**

This section is *Informative*. It present the context of the proposed system by exploring the existing systems and outlines the goals of the proposed System.

- **A formal description of the functional requirements**  
This section is *Normative*. This section outlines the requirements that specify the functionality of the system.
- **A formal description of the non-functional requirements**  
This section is *Normative*. It lays down the broader system requirements, design constraints, performance constraints and documentation requirements for the Business Accounting System System.
- **Use cases**  
This section is *Informative*. It explains the System functionality with the help of use cases.
- **Acceptance criteria**  
This section is *Normative*. It lays out the specific criteria that the Business Accounting System System needs to satisfy in order to be formally accepted by the Cybersource.
- **A change management procedure**  
This section is *Normative*. It defines the processes required to modify or remove any of the requirements outlined in this document.

## 2 Overall Description

This section will give a broad overview of the product, articulating the various goals of the project and the solution the product aims to provide. Here, we provide context for the formal requirements that follow, and provide a reference for any discussion or modification of the requirements as they develop throughout the project.

- Product context. Here we describe the environment in which the system will operate, and with which it will need to integrate.
- Existing systems. Here we will take a look at the existing systems and try to explain why this product is needed and help define what goals the project aims to achieve.
- Proposed system. After looking at the existing systems we will map the general features of the proposed system.

### 2.1 Product context

The area of interest for this product will be small business management. The final client will be a small business manager or accountant and the end users will be the same small business manager or accountant and other employees, assigned to accounting tasks.

Documents produced by this system will be read by customers and suppliers of the small business as well as any external accountants, consultants, auditors, tax officers and company investors/purchasers.

### 2.2 Existing systems

All small businesses are required to keep track of accounting details. There are various software and non-software accounting solutions available to small businesses, the major approaches are summarised below.

#### 2.2.1 Manual system

A manual system involves documenting all accounting related information by hand.

##### Advantages

- no computer skills necessary
- inexpensive

##### Disadvantages

- error prone
- inflexible

- inefficient and time consuming
- difficult to organise and maintain
- requires accounting expertise
- expensive if training is required
- no built-in audit trail
- no authentication or authorisation
- very difficult to backup

### 2.2.2 Spreadsheet

Spreadsheet software is used to manage data and provide various calculations.

#### Advantages

- good integration with other office software
- very flexible and customisable

#### Disadvantages

- not scalable for larger businesses
- requires high level of spreadsheet skills
- requires accounting expertise
- lack of automation
- expensive if training is required
- no built-in audit trail

### 2.2.3 Commercial software

Accounting software such as Quickbooks and MYOB are used.

#### Advantages

- only a high level accounting knowledge required
- efficient
- good built-in audit trail
- scalable for larger businesses
- software is updated with new taxation laws

**Disadvantages**

- software licences may be expensive
- outdated software may be unusable, due to changes in taxation law
- software is tied to a specific platform (Windows, Mac)
- product training may be required and is often expensive
- closed source code means it is not customisable or internationalisable

**2.2.4 Open source software**

Open source accounting software such as GNUcash, Grisbi, KMyMoney and SQLedger are used.

**Advantages**

- freely available
- accessible to users
- easy to install
- software is customisable
- always improving

**Disadvantages**

- most packages are primarily concerned with providing personal financial management with insufficient attention to small businesses needs
- the user interface is often less polished than commercial systems
- limited functionality
- most packages are not cross platform
- no focus on the Australian taxation system

**2.3 Proposed System**

The team aims to build a new accounting management system that has much the same functionality as existing commercial software, but released under an open source licence and with no ties to a specific operating system.

### 2.3.1 Aims

The proposed system aims to deliver an open source (General Public Licence) desktop accounting package with the particular aim of servicing the Australian Small to Medium Enterprises (SMEs). The system will also address other perceived deficiencies of existing accounting packages, providing an open source, cross platform, scalable and user-friendly accounting management solution.

These characteristics will be individually explained in detail as follows:

#### **Cross-Platform**

The product does not require a specific operating system, allowing users to select their operating system of choice. If a business wishes to migrate its systems from one operating system to another, the product must allow the data to also be migrated and the interface must be sufficiently similar to minimise training costs.

#### **Scalable**

Engineered to cater for rapid growth in:

- **Functionality:** Allowing for integration of new features (e.g. Stock or Payroll Subsidiary Ledgers).
- **Changes to accounting reports:** Allowing the manager to change the way accounting information is reported by the system either due to business needs or evolving accounting standards.
- **Future internationalisation:** Allowing users in new regions to interact with the system in different languages.
- **Future localisation:** Allowing users in new countries to manage their accounting and produce reports relevant to their local accounting practices and laws.

#### **Usable**

Small business owners may not have detailed accounting knowledge or computer skills. The system needs to allow these users to work effectively and efficiently with little training costs.

The proposed system will provide existing small businesses with a competitive, flexible, open source and cheaper alternative to existing accounting management systems such as MYOB & Quickbooks.

### 2.3.2 System Functionality

The proposed system can be broken down into 8 separate modules of functionality, based on distinct accounting functions that small businesses require. 3 of these modules form the backbone and the remaining 5 add other core functionality.

The modules will be explained in further detail below.

### **Core functionality**

The following functionality is recognised as being the backbone of the system because by itself it forms, at the most basic level, an accounting management system.

**Company** Information specific to the accounting needs of the company needs to be maintained. This includes holding information such as company name, tax rates, financial year end dates, as well as maintaining a 'chart of accounts'. The chart of accounts is one of the most fundamental aspects of any accounting system, listing and categorising all the accounts in the general ledger. Different types of businesses have different needs, in terms of accounts, and it is important to properly customise the 'chart of accounts'.

**General Journal** The General Journal provides the connection between physical financial transactions and an accounting system. All financial transactions are recorded into their relevant accounts via the general journal. In fact, with both a General Ledger and a General Journal, we have the most basic accounting system, with an ability to record all the transactions of the business.

**Global** This module provides the general features of a system that will allow the end user to use the system. It provides additional help and support required by users to both operate and troubleshoot the functions of the system. The more integrated this is into the system as a whole, the better.

### **Other Core Modules**

To increase the speed, efficiency and accuracy of recording common transactions, an accounting system will provide other interfaces to the General Ledger than simply the General Journal. These interfaces provide very customised functionality for common tasks, and are grouped by functionality as follows:

**Customers** This module handles transactions relating to cash sales and credit sales. The module maintains a separate account for each new debtor. Financial transactions relevant to each debtor are recorded into their separate account. This provides for more detailed information analysis, than if only a generic 'debtor account' is maintained.

Additionally, the system recognises that every transaction with a debtor generates certain documentation such as quotations and invoices. The proposed system will standardise the process by generating the required documentation and then cross-reference these documents to a particular transaction. This allows for a seamless process to trace any financial transaction from beginning-to-end and vice-versa.

Finally, a summary of all debtor transactions need to be visible for a high-level view.

**Suppliers** This module handles transactions relating to cash purchases and credit purchases. As with the Customers module, the system will also provide functionality to simplify and enhance the process of recording creditor transactions.

**Bank** The accounting system needs to manage cash-flows, and a separate interface much like the creditor and debtors will be required. This module will also allow the user to complete bank reconciliations.

**Tax** This module maintains the tax tables and allows the user to complete Business Activity Statements.

**Reports** One of the key uses of an accounting systems is its ability to provide information to managers to help them make decisions. This module allows the user to produce financial reports such as the Balance Sheet and Profit and Loss statement.

## 2.4 Users

- Australian small businesses
- Potential international small businesses

The target end users will be Australian small businesses with minimal accounting knowledge. To this end, a few principles need to be kept in mind:

- detailed configuration may confuse the user
- flexibility is not important for this particular class of user. It is preferable to impose accounting standards on the user rather than allow the user to specify their own flavour of accounting practices.

## 2.5 Accounting Concepts

This section provides the basic overview of accounting, so that the concepts that are fundamental to the operation of the system can be properly understood.

### 2.5.1 What is accounting?

The purpose of accounting is to provide quantitative information, primarily financial in nature, about businesses. This information is useful in making decisions for the business. Accounting information allows users to make reasoned choices among alternative uses of scarce resources in the conduct of business and economic activities.

When the decision involves business and economic questions, accounting information is essential to the decision making because it provides quantitative information.

Bookkeeping is a process in accounting involving the recording of transactions and keeping records.

### 2.5.2 Users of accounting information

Management and external groups (example banks) have a direct financial interest in the business. Each group requires specific information for making decisions as to specific resource allocation and to evaluate the results of those decisions.



**Management**

Management have overall responsibility for operating the business and for achieving the business's goals. Such goals may include: increasing profitability, providing quality goods and services at low cost, creating new and improved products.

Profitability is the ability to make enough profit to attract and hold investment capital. Liquidity means having enough funds to pay debts when they fall due. Accounting aims to provide management with relevant and useful information.

A major function of accounting is to measure and report information about how a business has performed. Most businesses periodically publish a set of general-purpose financial statements that report on their success in meeting objectives of profitability and liquidity. Most business must borrow money for both long and short term operating needs. The creditors, who lend the money, are interested in whether the business will have the cash to pay the interest charges and repay the debt at the appropriate time. The business's liquidity and cash flow and profitability are important to a creditor.

**Taxing Authorities**

Taxing authorities collect taxes. Such taxes include income taxes, payroll taxes and sales taxes. Each tax requires special tax returns and are often complex. Proper reporting is generally a matter of the law and can be very complicated.

Business transactions are economic events that affect the financial position of a business and that can be reliably measured in dollars. Each transaction requires at least two accounts. Businesses may have hundreds to thousands of transactions each day. These transactions are the raw material used for producing accounting reports. All business transactions are recorded in monetary terms.

Depending on the type of business- sole proprietorship, partnership or company, the tax paid is calculated at different rates.

**Sole proprietorship**

Sole proprietorship is a business formed by one person. Legally, the proprietorship is the same economic unit as an individual. The individual receives all profits or losses and is liable for all obligations of the proprietorship. Most businesses in Australia are proprietorships and can be found among small retail, service, and individual-operated professional businesses.

**Partnership**

Similar to a proprietorship except that it involves more than one owner. A partnership is not a legal economic unit separate from the owners but an association that brings together the talents and resources of two or more people. The partners share profits and losses of the partnership according to an agreed-upon contract.

**Companies**

A business unit that is legally separate from its owners (shareholders).

A business is interested in measuring their financial position. This refers to the economic

resources belonging to a business and the claims against those resources at a point in time. A business can be viewed as a collection of assets and equities. Every business has two types of equities: creditors' equity and owners' equity. In accounting, creditor's equity is referred to as liabilities.  $A=L+P$ . This equation is known as the balance sheet equation because the two sides of the equation represent the two sides of the balance sheet and must always be "in balance".

### 2.5.3 The double-entry system

The double entry system is the backbone of accounting. It is based on the principal of duality, which means that all events have two aspects that offset or balance each other: effort and reward, sacrifice and benefit, sources and uses. In the double entry system each transaction must be recorded with at least one debit and one credit in such a way that the total dollar amount of debits and the total amount of credits equal each other. The rules of double entry are that every transaction affects at least two accounts. That is, there must be one or more accounts debited and one or more accounts credited.

### 2.5.4 Accounts

#### What is an account?

An account is the basic storage unit for data in accounting. An accounting system has separate accounts for each asset, each liability, and each component of owners' equity, including revenues and expenses. Consequently it is able to describe all changes occurring in each type of account during a specified period of time. A very small business may have only a few accounts, whereas a multinational corporation will have thousands. The accumulation of information in account form is useful to management in identifying all the events that took place relating to a specific account. This information allows the manager of a business to know how much cash is on hand at any one time and prepare for bills that need to be paid. The manager can foresee the need to borrow money by estimating future cash payments and cash receipts. Good planning for future cash needs requires the need for a record of past receipts and expenditures as well as the current cash balance. An account is often represented in the form of a t-account. A t-account is made up of two columns: debit column and credit column. Depending on the type of account, an account may increase or decrease on either the debit or credit side.

In a manual accounting system, each account is kept on a separate page. These are grouped together and are referred to as the "General Ledger". To find an account in the ledger easily and to identify the account, each account is often allocated a number. A list of these numbers with the corresponding account names is referred to as the "Chart of accounts"

#### Type of accounts

**Assets** These are economic resources owned by a business that are expected to benefit future operations. Certain kinds of assets are monetary items such as accounts receivables, non-monetary tangible items such as inventory(goods held for sale), land, buildings and equipment. An asset increases on the debit side of a t-account.

**Liabilities** Are probable future sacrifices of economic benefits arising from present obligations of a particular entity to transfer assets to provide services to other entities in

the future as a result of past transactions or events. These are debts of the business, amounts owed to creditors for goods or services bought on credit, salaries and wages owed to employees and taxes owed to the government. A liability increases on the the credit side of a t-account.

**Proprietorship/Owner's equity** Equity is the residual interest in the assets of an equity that remains after deducting its liabilities. It is the resources invested in the business by the owner. There are two types of transactions that affect owners' equity. These are revenues and expenses which either increase or decrease the owners' equity. Proprietorship increases on the credit side of a t-account.

**Revenue** Revenue results from the selling of goods or the performance of a service. They result in either an increase in cash or other asset. For example, if a business performs a service for \$200 and the customer pays \$200 cash, the cash account (which of the type asset) is increased thus the owners' equity in those assets has also increased. Revenue increases on the credit side of a t-account.

**Expense** Expenses are resource outflows needed to earn revenue. They are equal to the resulting decrease in assets and a decrease in the owners' equity in those assets. An expense increases on the debit side of a t-account.

### 2.5.5 The accounting equation

Accounting is based on the following equation:

$$\text{Assets} = \text{Liabilities} + \text{Proprietorship}$$

That is, the sum of the value of assets is equal to the total sum of the value of liabilities and proprietorship. Proprietorship is equal to the sum of the amount the owner contributed and the profit made from the business. Therefore,

$$\text{Proprietorship} = \text{Owner's contribution} + \text{Profit}$$

where

$$\text{Profit} = \text{Revenues} - \text{Expenses}$$

Therefore, the equation can be written as:

$$\text{Assets} = \text{Liabilities} + \text{Owner's contribution} + (\text{Revenues} - \text{Expenses})$$

This equation must always be balanced.

### 2.5.6 Financial statements

Financial statements are a central feature of accounting because they are the primary means for communicating important accounting information to users. Common financial statements are the Balance sheet and the Profit and Loss statement.

#### Balance sheet

The Balance sheet shows the financial position of a business on a certain date. It presents a view of the business as the holder of resources or assets that are equal to the sources of or

claims against those assets. The sources or claims consist of the business's liabilities and the owners' equity in the business. The balance sheet is based on the equation:

$$\text{Assets} = \text{Liabilities} + \text{Proprietorship}$$

If this equation is not balanced, this means there has been error in the bookkeeping.

### **Profit and Loss statement**

The Profit and Loss statement summarises the amount of revenues earned and expenses incurred by a business over a period of time. When revenues exceed expenses, the difference between the two is called net profit and when expenses exceed revenues, the difference is called net loss. This statement is based on the equation:

$$\text{Profit} = \text{Revenues} - \text{Expenses}$$

### **The General Journal**

This is the most simplest and flexible type of journal and is the only one we will be concerned with in this system. The General Journal provides the information about each transaction. For each transaction, the date, the names of the accounts debited and credited, dollar amount debited and credited for each account, the explanation of the transaction and the account identification number if appropriate are recorded.

## 3 Functional Requirements

### 3.1 Overview

The requirements described in the following section define the required functionality for the Business Accounting System System. Each functional requirement is given an identifier, e.g.. [FR-1] , which stands for “Functional Requirement Number 1”.

#### 3.1.1 Layout

Each functional requirement is set out as follows:

[FR-X]	Requirement name
<b>Description:</b>	General information to help explain or give context to the requirement.
<b>Requirements:</b>	A precise and complete description of the requirement.
<b>Ranking:</b>	The ranking given to the requirement. See below for an overview of the rankings used.

#### 3.1.2 Rankings

Each functional requirement will be assigned a ranking of either: *Essential*, *Desirable*, or *Optional*

- A ranking of *Essential* implies that the Business Accounting System Product will not be accepted unless these requirements are provided in an agreed manner.
- A ranking of *Desirable* implies that these are requirements that would enhance the Business Accounting System Product but it would not make it unacceptable if they were absent.
- A ranking of *Optional* implies that these requirements hold no importance, except that they give Team Chilli the opportunity to provide a product that exceeds the SRS.

#### 3.1.3 Interface requirements

Ideally the interface functionality is made easily accessible by the user, for example by providing a single-click link or drop down list. However other design constraints that cannot be outlined in this requirements specification may need to be considered by the design team. In summary, these requirements do not attempt to prescribe how the interface functionality is made accessible to the user.

## 4 Functional Requirements: Global

### 4.1 General

#### [FR-1] Undo functionality

---

**Description:** The system must allow the user to undo any text input made into the system but before being processed. This does not apply to transactions.

**Requirements:** All text actions made must be reversible.

**Ranking:** Essential

#### [FR-2] Audit trail

---

**Description:** A user modification of any account must be logged. This is required for security and traceability.

**Requirements:**

1. The identification of the of the user who made the change
2. The date and time of the modification
3. The name of the account
4. The modification made

**Ranking:** Desirable

#### [FR-3] Save

---

**Description:** The system must continuously save its data, and should not require the user to manually save. If the system crashes or closes unexpectedly, no data must be lost.

**Requirements:**

1. The system must perform all save functionality automatically, without user command.
2. All changes in the system must be saved immediately

**Ranking:** Essential

**[FR-4] Localisation requirements**

---

**Description:** Localisation allows the user to specify the formatting of dates, times, currency and numbers at run-time. For example, in Australia short dates are formatted as 'dd/mm/yy' whereas in the US they are 'mm/dd/yy'. In Australia, number are formatted as '12,345.00' whereas in Germany they are '12.345,00'.

**Requirements:** The system must allow the user to specify at run-time the formatting of:

1. long dates
2. short dates
3. times
4. currency
5. numbers

**Ranking:** Desirable

**[FR-5] Internationalisation requirements (e.g. BSB)**

---

**Description:** Internationalisation allows the language to be changed on startup. The standard interface for translators is usually a flat file.

**Requirements:**

1. The system must allow new translations to be written in a separate file.
2. The system must allow the user to change the language at startup, if the desired translation exists.

**Ranking:** Essential

**4.2 Field lengths**

The field lengths specified below are the absolute minimum, for testing purposes. Larger lengths are perfectly acceptable if they are also appropriate from a design point of view.

**[FR-6] Definitions**

---

**Description:** The following sections require certain definitions of the following words

**Requirements:**

1. A character is a single character with ISO 8859-1 encoding

**Ranking:** Essential

[FR-7] **Names**

---

**Description:** This applies to account names, institution names, customer names, company names, supplier names, contact names, legal names

**Requirements:**

1. This includes only printable characters and the space character. (Essential)
2. The system must accept account names with any number of printable characters (Essential)
3. The system must accept UNICODE encoded names (Desirable)
4. The system must strip leading and trailing whitespaces (Desirable)
5. The system must accept BIG-5 encoded names (Optional)

**Ranking:** Refer to individual rankings above.

[FR-8] **Generic field length**

---

**Description:** This applies to email addresses and postal addresses.

**Requirements:**

1. All email address and postal address fields must store at least 200 characters

**Ranking:** Essential

[FR-9] **Number field length**

---

**Description:** The minimum lengths for various number based fields are specified below

**Requirements:** These are as follows:

1. Phone numbers: 25 characters. Phone numbers should allow the following punctuation: '+' '-' '(' ')'
2. Ledger account numbers: 7 digits
3. ABN, ACN number allow for at least 15 digits
4. Other numbers: 40 characters

**Ranking:** Essential

[FR-10] **Comment lengths**

---

**Description:** A comment is a free text field for the user to write abstract comments.

**Requirements:**

1. All comment fields must store at least 1000 characters.

**Ranking:** Essential



**[FR-11] Currency numbers**

---

**Description:** This applies to any number that represents a currency

**Requirements:**

1. All numbers that represent a currency amount, must store up to 12 significant figures, fixed at two decimal points.

**Ranking:** Essential

### 4.3 Input / output

**[FR-12] Copy and paste**

---

**Description:** Copy and paste functionality is described here

**Requirements:**

1. The system must allow all input and reported text and numbers to be copied to the desktop clipboard.
2. The system must allow all valid and compatibly encoded text to be pasted in any input field.

**Ranking:** Essential

**[FR-13] Import file formats**

---

**Description:** This allows the user to import data from other software

**Requirements:** The system must allow data stored in the following formats to be imported:

1. Quicken Interchange Format (QIF)
2. Open Financial Exchange format

**Ranking:** Desirable

**[FR-14] Export file formats**

---

**Description:** This allows the user to export data to other software The system must allow data stored in the following formats to be imported:

1. Open Financial Exchange format

**Ranking:** Desirable

**[FR-15] Backup management**

---

**Description:** The user may wish to store off-site backups of the data from the system.

**Requirements:**

1. The system must allow the user to export the system data to be stored on a separate disk (Essential)
2. The system must allow the user to import previously exported backup data (Essential)
3. The system must provide an interface to export the system data as described above (Essential)
4. The system must provide an interface to import system data as described above (Essential)
5. The system must allow periodic exports to be automatically performed (Optional)
6. Backup files should be in readable file format, using plain text and standard compression formats. (Essential)

**Ranking:** (individually ranked)

#### 4.4 Design Guidelines

**[FR-16] Authorisation**

---

**Description:** The system needs to permit authorisation of users for security purposes.

**Requirements:** The system must allow an administrative user to grant and revoke access for users.

**Ranking:** Desirable

**[FR-17] Shortcuts**

---

**Description:** It is desirable that the software allows the user to create a shortcut for any interface functionality. This will allow power users to avoid using the mouse and drive the system more quickly.

**Requirements:** The system must provide shortcuts for all distinct interface functionality.

**Ranking:** Optional

**[FR-18] Spell check**

---

**Description:** It would be convenient for the user if the system provided spell checking functionality.

**Requirements:**

1. Misspelled words must be brought to the users attention and an alternative for that word must be provided.
2. All alternative words provided by the spell checker must be located in the selected dictionary.
3. The application should have a custom user dictionary.
4. This dictionary should be in a format which can be edited by external tools.

**Ranking:** Optional

**[FR-19] Calculator**

---

**Description:** The system must provide a calculator facility to allow the user to make basic calculations.

**Requirements:** The system must provide an interface to allow simple equations to be evaluated.

**Ranking:** Desirable

**[FR-20] Date calendar**

---

**Description:** The system must allow the user to enter a date through using a GUI calendar and selecting the day. This should automatically input the date in the date field.

**Requirements:**

1. Any field that requires a date should accept both text input and GUI input using the calendar.

**Ranking:** Desirable

## 4.5 Help

### [FR-21] Help framework

---

- Description:** A framework should be developed to allow for the future development of the help navigator. A user guide will be produced as part of the delivery of the final product.
- Requirements:** The system must provide a framework to allow:
1. Each major functionality on all interfaces need to have a help page associated with it, detailing purpose, use and related functionality.
  2. each interface page to be linked to the relevant section of the user guide
  3. help pages should be extensible by sophisticated users, without needing to recompile.
- Ranking:** Essential

## 4.6 Data entry

### [FR-22] Auto completion

---

- Description:** Auto completion enables faster data-entry by anticipating what the user wishes to type and offering to complete the word or phrase automatically. This applies to fields that require restricted input such as customer and debtor names, account names and codes.
- Requirements:** The system must provide auto-completion for all fields that require restricted input.
- Ranking:** Desirable

## 4.7 Search

### [FR-23] Searching transactions

---

**Description:** The system must provide the functionality for searching for previous transactions.

**Requirements:** The system must allow the user to search according to:

1. invoice number
2. date
3. transaction amounts
4. customer name
5. text

The system must also allow any combinations of the above to be searched. Once a search has completed the following must be shown:

6. A list of matching transactions, which display:
  - (a) Transaction ID
  - (b) Transaction description/notes
  - (c) Entry details, see requirement [FR-56]

**Ranking:** Essential

### [FR-24] Expert search

---

**Description:** The system must provide an advanced search facility to allow users to type SQL queries to search for items.

**Requirements:**

1. The system must allow the user to perform SQL queries.
2. The system must not allow the user to perform SQL queries that modifies data.

**Ranking:** Optional

**[FR-25] Global searching**

---

**Description:** Global searches are a highly usable interface for searching, used by Google and Spotlight on Apple OS X. The interface is a single text entry row, where the user is free to enter what they want. Users are comfortable with this method of searching. It invariably involves creating an index of search strings for all data on a system and because of the use of indices, the search is almost instantaneous. On desktop systems, such searches are normally available at all times in either a fixed position, after keystroke or both. When displaying the data, the search terms are highlighted.

**Requirements:**

1. The system must provide a global search interface, allowing the user to search for data.
2. The interface must consist of a single search field and no visible options.
3. The system must take no longer than 1 second to present results for all single word searches.
4. The system must display results as they user types the query, not waiting for a submission command (optional).
5. The search interface must be available at all times, not necessarily visible, but available.
6. Matching names and document numbers should always appear first.
7. If only one item matches, the system should go directly to the relevant page.

**Ranking:** Desirable

**[FR-26] Natural language searches**

---

**Description:** The system needs to understand basic natural language query modifiers.

**Requirements:**

1. The system must understand date, expressed in all relevant formats for the given location.
2. The system must understand date modifier phrases including “between date and date”, “before date” and “after date”.
3. The system must understand basic amount modifier phrases including “less than 1234”, “more than 1234” and “between 1234 and 1234”, and search for matching transactions and balances depending on the context.

**Ranking:** Optional

## 4.8 Assumptions

### [FR-27] Tax Assumption

---

**Description:** It is assumed that from the backend perspective, all amounts, where a tax is applicable, includes the relevant tax amount. For example a stock purchase of \$110, if the user specifies a tax is applicable, then the system automatically assumes tax was included in the \$110. There is no case where the system will have to separately add tax to an entered amount.

**Requirements:**

1. All transactions passed to the backend must include the tax amount.
2. Any interface that involves tax must provide the user with the flexibility to select whether an amount is tax inclusive or not.

**Ranking:** Essential

## 5 Functional Requirements: Company Data

### 5.1 Company Details

#### [FR-28] Company details

---

**Description:** Certain details need to be maintained for each company. Any or all of the following fields may be left blank by the user.

**Requirements:** The system must allow the following company details to be stored:

1. The type of business: Service or Wholesale
2. Company name
3. Legal name (e.g. Pty Ltd)
4. GST Registration (see requirement [FR-30] )
5. Web page
6. Financial year end (see requirement [FR-4] for date format)
7. Company logo (optional)
8. Base currency (e.g. AUD, USD etc. - This is only used for displaying the currency denomination when producing reports)

The following contact details may have zero or more values associated:

9. Address (See requirement [FR-8] )
10. Postal address (See requirement [FR-8] )
11. Phone number (See requirement [FR-9] )
12. Fax number (See requirement [FR-9] )
13. Email address (See requirement [FR-8] )

**Ranking:** Essential

#### [FR-29] Company details interface

---

**Description:** The user must be able to view and edit all company details

**Requirements:** The system must provide an interface to display company details including:

1. All details in [FR-28]

**Ranking:** Essential



[FR-30] **GST registration**

---

**Description:** Australian companies that are registered for GST are required to collect GST on behalf of the Australian Tax Office (ATO). The company must be able to notify the system of this registration when creating the company. A company must be able to register for GST at a later date. Once a company has elected that they are registered for GST, this cannot be reversed.

**Requirements:** If the business is registered for GST upon creation of the company:

1. Once the user has notified the system of the GST registration, the system must allow the selection to be reversed at a later date
2. The system must create a an account called "GST Clearing"
3. This account is a "special" account (see [FR-35] )
4. This account is of type "current liability"
5. The opening account balance is 0
6. The account number is not specified.
7. A default tax table is created. See [FR-141] . These are GST tax codes directly related to the GST Clearing account.
8. The system must allow the user to update the existing tax codes. See requirement [FR-142] .

If the business is not registered for GST:

9. The system must allow the user to register for GST at a later date. See above if they do register at a later date.
10. The system must not create a "GST Clearing" account.
11. The system must not have any default GST tax codes.

**Ranking:** Desirable

[FR-31] **GST registration interface**

---

**Description:** The user must be able to view and edit the GST registration page.

**Requirements:** The interface must implement all of the functionality listed in [FR-30] .

**Ranking:** Desirable

**[FR-32] Custom company details**

---

**Description:** Modules and users may wish to attach other values to a company for future reference.

**Requirements:** The following custom details must also be maintained, each having zero or more values:

1. Module defined custom fields
2. User defined custom fields

**Ranking:** Optional

**5.2 Chart of Accounts****[FR-33] Account types**

---

**Description:** Refer to the glossary for definitions of the following terms.

**Requirements:** Every account must be one of the following types

1. Current Asset
2. Non-Current Asset
3. Current Liability
4. Non-Current Liability
5. Revenue
6. Expense
7. Proprietorship

**Ranking:** Essential

**[FR-34] Create new accounts**

---

**Description:** An account is the fundamental object for this system (refer to the glossary for a detailed description). The functional requirements for creating new accounts are described below:

**Requirements:** The user must specify the following:

1. Account type
2. Account name (see requirement [FR-40] )

The user has the option to specify the following:

3. Account number (see requirement [FR-39] )
4. Account balance (default is 0)
5. Default tax code (This only applies to revenue and expense account types)
6. An account may also be associated to a category (see requirement [FR-37] ).

The system must assign the following details:

7. Date and time of creation

**Ranking:** Essential

**[FR-35] Special accounts**

---

**Description:** Special accounts are accounts that the system relies on when completing certain automated transactions. All special accounts are to be created only by the system. Note that although the current system does not implement 'delete account' functionality, this may be developed in the future. The deletion requirement ensures that these accounts are not deleted when this happens.

**Requirements:**

1. The system must allow Special accounts to be edited as per requirement [FR-42] .
2. Refer to requirement [FR-48] for list of system special accounts.
3. Refer to requirement [FR-49] for list of system special accounts.
4. GST Clearing account, if created by the system, is also a special account
5. The system must not allow special accounts to be deleted

**Ranking:** Essential

**[FR-36] Special category**

---

**Description:** Special categories are categories that the system relies on when completing certain automated transactions. All special categories are to be created only by the system. Note that although the current system does not implement 'delete category' functionality, this may be developed in the future. The deletion requirement ensures that these categories are not deleted when this happens.

**Requirements:**

1. Refer to requirement [FR-46]
2. The system must not allow special categories to be deleted

**Ranking:** Essential

**[FR-37] Account categories**

---

**Description:** Account categories organise the accounts and are only used for reporting purposes (i.e. they do not behave like an actual account). They are simply a way of grouping accounts. The category also maintains a running balance that is the sum of all account balances in that category. Additionally, categories can not have sub-categories.

**Requirements:** All account categories must have:

1. Category name. This name must be unique across all existing category and account names.
2. Account type
3. Running balance

**Ranking:** Essential

**[FR-38] Edit account category**

---

**Description:** The user must be able to edit account categories. Note that if the category is already linked to certain accounts or is a default category, it is the user's responsibility to re-name the category appropriately (i.e. not change its underlying meaning).

**Requirements:** The user can only edit the category name.

**Ranking:** Essential

**[FR-39] Account numbers**

---

**Description:** Valid account numbers are detailed below.

**Requirements:**

1. The system must accept any positive integer for an account number, ranging from zero (0) to at least one million (1,000,000).
2. All account numbers must be unique.

**Ranking:** Essential

**[FR-40] Account names**

---

**Description:** The following section outlines valid account names.

**Requirements:**

1. See requirement [FR-7]
2. The system must accept duplicate account names only if the account types are different.

**Ranking:** Essential

**[FR-41] Account interface**

---

**Description:** The system must provide the user with an interface for creating accounts.

**Requirements:** The interface must display:

1. all the functionality as defined in [FR-34]

**Ranking:** Essential

**[FR-42] Edit accounts**

---

**Description:** The user may wish to change certain details of an account. The system must allow this, without affecting previous transactions or the structure of the chart of accounts.

**Requirements:**

1. Account names must be editable (see requirement [FR-40] )
2. Account numbers must be editable (see requirement [FR-39] )
3. Tax code, if applicable, must be editable (see requirement [FR-34] ).
4. Account types must not be editable once an account is created.
5. Once an account has been linked to a category this can not be edited.
6. If an account has not been previously linked to a category, the system must allow the account to be now linked.
7. Pre-existing transactions must remain linked to the relevant account.

**NOTE:** If the account is already linked to transactions or is a default account, it is the user's responsibility to re-name the account appropriately (i.e. not change its underlying meaning).

**Ranking:** Essential

**[FR-43] Edit accounts interface**

---

**Description:** The system must provide the user with an interface for editing accounts.

**Requirements:** The interface must display:

1. All functionality defined in [FR-42]

**Ranking:** Optional

**[FR-44] View chart of accounts interface**

---

**Description:** The system must provide an interface for displaying the chart of accounts. The functional requirements are described below. Note that for all accounts associated to a category, they must be displayed as a group. The category will act as a heading for the group of accounts, by displaying the category name and balance.

**Requirements:** The interface must display the following for each account in the system:

1. Account name
2. Account type
3. Account balance

**Ranking:** Essential

**[FR-45] User codes**

---

**Description:** User codes are a way of referring to accounts, much like the account number, however they can be customised per user. Thus, one account can have many user codes, but each user code must only refer to one account. They are a way of allowing users to have their own short, relevant General Ledger (GL) codes.

**Requirements:**

1. The system must allow users to specify their own user codes, which refer to an account.
2. The system must allow many user codes to refer to a single account.
3. Each user code must only be valid, visible and usable for one user.
4. The system must accept any string of system supported characters of at least 32 characters.

**Ranking:** Optional

### 5.3 Default Accounts & Categories

This section describes the default Accounts and Categories, which the system provides to the end-users. Note that certain accounts and categories will be flagged as 'special ac-

count/category' (see [FR-35] ).

[FR-46] **Default categories**

---

<b>Description:</b>	The system must provide a set of default categories.		
	<u>Category name</u>	<u>Account type</u>	<u>Special category</u>
<b>Requirements:</b>	Equipment	Current Asset	NO
	Accounts Receivable	Current Asset	YES
	Accounts Payable	Current Liability	YES
<b>Ranking:</b>	Essential		

[FR-47] **Default accounts**

---

<b>Description:</b>	The system must provide a set of default accounts.		
<b>Requirements:</b>	<ol style="list-style-type: none"> <li>1. The system must provide two sets of default accounts depending on the type of business: Service business and Wholesale business.</li> <li>2. Default account definitions for each type of business must be defined in separate XML files residing in a single directory. This will in the future allow new default accounts to be defined using the same template.</li> </ol>		
<b>Ranking:</b>	Essential		

[FR-48] **Default accounts for a service business**

**Description:** The following lists the default accounts that are required for a service business.

	<b>Number</b>	<b>Name</b>	<b>Type</b>	<b>Category</b>	<b>Special</b>
	101	Bank	current asset	None	YES
	110	Petty Cash	current asset	Blank	NO
	120	Prepaid Expenses	current asset	None	NO
	130	Materials	current asset	None	NO
	140	Bank-Adjustments	current asset	None	YES
	210	Unearned Revenue	current liability	None	NO
	301	Capital	proprietorship	None	YES
	310	Drawings	proprietorship	None	YES
	401	Profit and Loss Summary	proprietorship	None	YES
	501	Credit Sales	revenue	None	YES
	502	Cash Sales	revenue	None	YES
<b>Requirements:</b>	601	Administration Expense	expense	None	NO
	602	Wages	expense	None	NO
	603	Advertising expense	expense	None	NO
	604	Rent expense	expense	None	NO
	605	Bad debts expense	expense	None	NO
	606	Bank fees	expense	None	NO
	607	Insurance expense	expense	None	NO
	608	Maintenance expense	expense	None	NO
	609	Telephone expense	expense	None	NO
	610	Water expense	expense	None	NO
	611	Gas expense	expense	None	NO
	612	Repairs expense	expense	None	NO
	613	Council rates expense	expense	None	NO
	614	Charity contributions	expense	None	NO
<b>Ranking:</b>	Essential				



[FR-49] **Default accounts for a wholesale business**

**Description:** The following lists the default accounts that are required for a wholesale business.

	<b>Number</b>	<b>Name</b>	<b>Type</b>	<b>Category</b>	<b>Special</b>
	101	Bank	current asset	None	YES
	110	Petty Cash	current asset	Blank	NO
	120	Prepaid Expenses	current asset	None	NO
	130	Inventory	current asset	None	NO
	140	Bank-Adjustments	current asset	None	YES
	210	Unearned Revenue	current liability	None	NO
	301	Capital	proprietorship	None	YES
	310	Drawings	proprietorship	None	YES
	401	Profit and Loss Summary	proprietorship	None	YES
	501	Credit Sales	revenue	None	YES
	502	Cash Sales	revenue	None	YES
	601	Administration Expense	expense	None	NO
<b>Requirements:</b>	602	Wages	expense	None	NO
	603	Advertising expense	expense	None	NO
	604	Rent expense	expense	None	NO
	605	Bad debts expense	expense	None	NO
	606	Bank fees	expense	None	NO
	607	Insurance expense	expense	None	NO
	608	Maintenance expense	expense	None	NO
	609	Telephone expense	expense	None	NO
	610	Water expense	expense	None	NO
	611	Gas expense	expense	None	NO
	612	Repairs expense	expense	None	NO
	613	Council rates expense	expense	None	NO
	614	Charity contributions	expense	None	NO
	615	Cost of Goods Sold	expense	None	YES

**Ranking:** Essential

**5.4 General**[FR-50] **Multiple companies**

**Description:** Many small business owners may like to use the software to manage the accounting for multiple businesses they run. The system needs to allow this, and at the same time, allow certain resources to be shared.

**Requirements:**

1. The system must be able to handle multiple companies
2. The system must allow the customer list from an existing company to be imported to a new company. (see section 5.5)
3. The system must allow the supplier list from an existing company to be imported to a new company. (see section 5.6)

**Ranking:** Desirable

## 5.5 Customer list

To improve usability and efficiency the user must be able to import all existing customers from an existing business on creation of a new business.

### [FR-51] Import customer list

---

**Description:** To import a customer list from a existing business the following requirements are required. Note that imported customers can be updated as per normal.

**Requirements:**

1. Import only customer details, see requirement [FR-59] .
2. Customer *account* details must not be imported.
3. For every customer imported, a new account must be created according to the 'debtor account details' in [FR-59] .

**Ranking:** Optional

### [FR-52] Import customer list interface

---

**Description:** The system must provide the user with an interface to import a customer list from an existing business.

**Requirements:**

1. The user must select which business customer list to import.

**Ranking:** Optional

## 5.6 Supplier list

To improve usability and efficiency the user must be able to import all existing suppliers from an existing business on creation of a new business.

### [FR-53] Import supplier list

---

**Description:** To import a supplier list from a existing business the following requirements are required. Note that imported suppliers can be updated as per normal.

**Requirements:**

1. Import only supplier details, see section ??.
2. Customer *account* details must not be imported.
3. For every supplier imported, a new account needs to be created according to the 'supplier account details' in ??.

**Ranking:** Optional

**[FR-54] Import supplier list interface**

---

**Description:** The system must provide the user with an interface to import a supplier list from an existing business.

**Requirements:**

1. The user must select which business supplier list to import.

**Ranking:** Optional

## 6 General Journal

### 6.1 General

**[FR-55] Transactions**

---

**Description:** A transaction is a set of related entries that reflect a change in two or more accounts. The set of related entries must balance, so that the sum of all debits always equals the sum of all credits. A transaction can be entered into either the general journal as a typical accounting entry, or into one of the other modules such as customers or suppliers. A transaction must be recorded as entries in at least two accounts. The user must specify whether the transaction includes tax. Refer to the glossary for detailed definitions of gross and net.

**Requirements:**

1. The system must allow a set of at least 2 entries to be entered, and marked as a single transaction.
2. If a transaction leads to a difference in the sum of credits and debits, the system must reject the transaction and rollback the accounts to their state before the transaction took place.
3. Each transaction must have a transaction ID
4. All entries within a transaction must have the same transaction ID.
5. The date of the transaction must be recorded
6. Must allow the user to enter a description for the transaction
7. The type of tax applicable (if any).
8. For each entry in a transaction the user must specify if tax is applicable by selecting the appropriate tax type. See requirement [FR-141]
9. See requirement [FR-57] for details relating to the processing of tax.

**Ranking:** Essential

[FR-56] **Entries**

**Description:** An entry is a single line of a transaction that displays the amount by which a specific account will be affected. Note that the account number and name referred to in an entry must remain current, even if the user has changed the account's number or name.

**Requirements:** Each entry must display:

1. Account name
2. Amount
3. The direction of exchange (either credit or debit)
4. Must somehow be related to a transaction

**Ranking:** Essential

[FR-57] **Tax transaction accounting**

**Description:** For any entry, within a transaction, where the user selects an applicable tax code, the system must process the tax component appropriately. See requirement ?? for assumptions relating to the system's handling of tax. Note that for entries related to the "GST Clearing" account, the system must also record the relevant GST tax code applied.

**Requirements:** If the balance for an entry is a debit:

1. The "GST Clearing" account is debited by:

$$\frac{\text{amount}}{1 + \text{tax code rate}}$$

2. The account specified by the account name is debited by:

$$\text{amount} - \frac{\text{amount}}{1 + \text{tax code rate}}$$

If the balance for an entry is a Credit:

3. The "GST Clearing" account is credited by:

$$\frac{\text{amount}}{1 + \text{tax code rate}}$$

4. The account specified by the account name is credited by:

$$\text{amount} - \frac{\text{amount}}{1 + \text{tax rate}}$$

**Ranking:** Essential

**[FR-58] General Journal Interface**

---

**Description:** The system must provide an interface for allowing the user to manually enter transactions

**Requirements:** The interface must:

1. Allow only one transaction to be entered at a time
2. Allow the user to enter a transaction date
3. Allow the user to enter a transaction description
4. Allow for a maximum of 50 separate entries per transaction. Each entry must allow the user to:
  - (a) Select an Account Name from all existing accounts
  - (b) Enter the amount (Debit / Credit)
  - (c) Applicable tax type by selecting from existing tax types
  - (d) Automatically display tax amount as follows:
  - (e) Automatically display zero in the tax field if no applicable tax is chosen
  - (f) Automatically display the actual tax in the tax field if an applicable tax is chosen (See also ??)
5. Display the total amount for all debit entries
6. Display the total amount for all credit entries
7. Display the total amount for all tax entries
8. Process transactions (see requirement ??)

**Ranking:** Essential

## 7 Functional Requirements: Customers

The following requirements are split as follows:

1. Debtors
2. Quotes
3. Sales orders
4. Invoices
5. Receive payments from debtors
6. Credit note

Note: Quotes, Sales orders, Invoices and Credit notes are actual account keeping documents. The above documents should be ideally linked, so that the user doesn't have to re-enter the details for the sales order and invoice. However, they are separate documents, and may vary slightly in content. For example, an item may appear on a quote but not the sales order. One implementation might allow a new sales order (with a new sales order ID) to be generated at any time from any quote.

### 7.1 Debtors

The system must maintain a list of debtors and provide functionality to select, update and create the list.

[FR-59] **Debtor details**

---

**Description:** A debtor is outlined here. Refer to the glossary for a detailed accounting based definition of a debtor.

**Requirements:**

1. A debtor is an account
2. The account name is independent from the debtor's legal name
3. The account type is always "current asset"
4. The account category is always "accounts receivable"

The user must specify the following:

5. Debtor's legal name

The user may choose to enter the following:

6. Debtor's ABN and ACN
7. Debtor's legal address
8. Debtor's phone numbers
9. Debtor's email addresses
10. Debtor's fax numbers
11. Any number of delivery details (Desirable)
12. Comments

**Ranking:** Essential

[FR-60] **Add new debtor functionality**

---

**Description:** The functional requirements for adding a new debtor are outlined below

**Requirements:**

1. The system must allow new debtors to be added
2. The system must allow the debtor's contact details to be specified see requirement [FR-59] .

A new account must be created with:

3. the account type set to 'asset'
4. the account category set to 'accounts receivable'

**Ranking:** Essential

**[FR-61] Add new debtor interface**

---

**Description:** The interface requirements for adding a new debtor are outlined below:

**Requirements:**

1. The system must provide an interface for allowing a user to create new debtors.
2. The interface must implement all of the functionality listed in [FR-60] .

**Ranking:** Essential

**[FR-62] Update existing debtor functionality**

---

**Description:** The functional requirements for editing an existing debtor are outlined below

**Requirements:**

1. The system must allow debtor details to be edited, see requirement [FR-59]
2. The debtor account type cannot be changed (from 'current asset')
3. The debtor account category cannot be changed (from 'accounts receivable')

**Ranking:** Essential

**[FR-63] Inactive debtors**

---

**Description:** A debtor can be flagged as inactive if the user wishes to stop making transaction with the debtor.

**Requirements:**

1. The system must allow debtors to be flagged as inactive
2. The system must allow inactive debtors to be flagged as active
3. Account balance must be zero before flagging an account as inactive
4. No new transactions can be made where it relates to an inactive debtor.

**Ranking:** Desirable



**[FR-64] List of debtors functionality**

---

**Description:** The functional requirements for listing debtors are outlined below. Activity date means that the debtor has made a purchase on a give date.

**Requirements:** The system must be able to produce a list of debtors, orderable by:

1. account number
2. account name
3. account balance
4. latest activity date (optional)
5. active/inactive/all

System must be able to filter debtors from the database who match a given criteria, including:

6. Threshold balance amount (i.e. less than \$x or greater than \$y)
7. Debtors that have purchased a given item (optional)
8. Activity date range (optional)

**Ranking:** Desirable

**[FR-65] List of debtors interface:**

---

**Description:** This interface will provide the user with a list of debtors, and any relevant details associated with that debtor.

**Requirements:** The system must provide an interface to display a list of debtors including:

1. Each debtor's account number, name, balance
2. Any user defined field from each debtor (i.e. phone number) (Optional)
3. The user must be able to reorder the list according to [FR-64]
4. The user must be able to filter the list according to [FR-64]

**Ranking:** Desirable

**[FR-66] Balance of all accounts receivable interface**

---

**Description:** The balance of all accounts receivable needs to be accessible somewhere by the user.

**Requirements:**

1. The system should display a list of all debtors, their account number and balance.
2. It should also display the final debtor balance.

**Ranking:** Essential

**[FR-67] Delete debtor functionality**

---

**Description:** Deletion is generally discouraged, as it may be more appropriate to flag the debtor as inactive. If the user really does want to delete a debtor with transactions associated, they must re-link the transactions with another debtor beforehand.

**Requirements:**

1. The system must allow debtors to be deleted
2. The deleted debtor account must not have any transactions associated with the account
3. All deleted debtors must be recoverable
4. The deleted account number must be reusable in future accounts

**Ranking:** Optional

**[FR-68] Relinking debtor transactions**

---

**Description:** Transactions that have been recorded under the wrong debtor may need to be edited. The system needs to provide a facility to do this for every transaction associated with a particular debtor.

**Requirements:**

1. The system must allow all transactions from a given debtor to be re-linked, or re-associated with another debtor.
2. Any relinking must be reversible (Desirable)

**Ranking:** Optional

**[FR-69] Delete debtor interface**

---

**Description:** From the user's perspective, deleting a debtor may require other tasks such as re-associating transactions before performing the actual delete. The user should be discouraged from performing deletions and guided towards flagging a debtor as inactive.

**Requirements:** If the deleted account has associated transactions, the system must ask the user if they would like to:

1. re-link the transactions to another debtor account and then delete
2. leave the transactions and flag the account as inactive (see [FR-63] ), or
3. prevent the delete from taking place

**Ranking:** Optional

## 7.2 Quotes

Quotes are simply a written offer for the sales contract, that only states the details of a hypothetical transaction. Including this functionality in the system will allow the user to re-use information from a quote.

### [FR-70] Quote functionality

---

**Description:** A detailed description of a quote can be found in the glossary. The following requirements outline the functionality of the quote. Quotes do not need to be necessarily addressed to a particular debtor. (i.e. Quotes have no accounting impact)

**Requirements:** The following information must be stored for each quote

1. Quote number
2. Date
3. Quote expiry date
4. Customer (if applicable)
5. Items (name, code, description, quantity, unit price, tax, amount)
6. Comments

**Ranking:** Desirable

### [FR-71] Quote interface

---

**Description:** The system needs to provide an interface for creating and viewing quotes

**Requirements:** The system must provide an interface for allowing a user to:

1. create new quotes.
2. view existing quotes.

The following is also applicable:

3. The interface must give access to all of the functionality listed in [FR-70] .
4. It must be possible to automatically fill in details of existing debtors
5. Any quotes generated must be editable.
6. The system must allow a sales order and/or invoice to be generated from the current quote. See requirement ??

**Ranking:** Desirable

**[FR-72] Quote printing**

---

- Description:** The system must allow quotes to be printed
- Requirements:** The printed quote must display:
1. The company details and ABN
  2. All details recorded in the quote (see [FR-70] )
- Ranking:** Desirable

**7.3 Sales orders**

A sales order is an internal document used by the business to ensure that there is sufficient stock for a customer order. This will have no accounting impact.

**[FR-73] Sales order functionality**

---

- Description:** A detailed description of a sales order can be found in the glossary. The following requirements outline the functionality of the sales order. Over the counter sales do not necessarily record the details of the customer. The system needs to allow anonymous orders.
- Requirements:**
1. Sales order number
  2. Date
  3. Customer (if applicable)
  4. Customer delivery address (if applicable)
  5. Items
- Ranking:** Desirable

**[FR-74] Sales order interface**

---

- Description:** The system needs to provide an interface for creating and viewing sales orders
- Requirements:**
1. The system must allow new sales orders to be generated.
  2. It must be possible to automatically fill in details of existing customers.
  3. If an existing customer has previous quotes, these might be made available to allow the sales order to be automatically generated from the quote.
  4. Any sales order generated must be editable.
- Ranking:** Desirable

**[FR-75] Sales order printing**

---

**Description:** The system must allow sales orders to be printed

**Requirements:** The printed sales order must display:

1. The company details and ABN
2. All details recorded in the sales order (see [FR-73] )

**Ranking:** Desirable

## 7.4 Invoice

An invoice is produced each time a sale takes place. This can be either a credit or cash sale.

**[FR-76] Invoice functionality**

---

**Description:** A detailed description of a invoice can be found in the glossary. The following requirements outline the functionality of the invoice.

**Requirements:** The following information must be stored for each invoice

1. Invoice number
2. Select cash or credit sale
3. Sales order number (if applicable)
4. Date
5. Terms of sale
6. Debtor
7. Custom delivery address (if applicable)
8. Items (name, code, description, quantity, unit price, tax, amount)
9. Comments

**Ranking:** Essential

**[FR-77] Invoice interface**

---

**Description:** The system needs to provide an interface for creating and viewing invoices

**Requirements:**

1. The system must allow new invoices to be generated.
2. It must be possible to automatically fill in details of existing customers.
3. If an existing customer has previous sales order or quote, these might be made available to allow the invoice to be automatically generated from the sales order or quote. This may be achieved by linking the customer or a given sales order/quote number.
4. Any invoice generated must be editable (optional).
5. The system must keep track of all the dates and times the invoice has been printed (desirable).

**Ranking:** Essential

**[FR-78] Credit sales accounting**

---

**Description:** When a credit sale is made, the debtor, tax and sales revenue accounts need to be updated. Note that as we are not implementing an Inventory Module to keep track of the cost of every inventory item (Perpetual inventory system), 'Cost of Goods Sold' will be performed using the Physical Inventory System method. See glossary for further details.

**Requirements:**

1. Sales revenue account is credited by the amount excluding tax.
2. GST clearing account needs to be credited for the tax amount.
3. Debtor account is debited by the amount of the sale including tax.

**Ranking:** Essential

**[FR-79] Cash sales accounting**

---

**Description:** When a cash sales invoice is made, the bank, tax and sales revenue accounts will need to be updated.

**Requirements:**

1. Sales revenue account is credited by the amount excluding tax.
2. GST clearing account needs to be credited for the tax amount.
3. Bank account is debited by the amount of the sale including tax.

**Ranking:** Essential

**[FR-80] Invoice printing**

---

**Description:** The system must allow invoices to be printed

**Requirements:** The printed invoice must display:

1. The company details and ABN
2. All details recorded in the invoice (see [FR-76] )

**Ranking:** Essential

## 7.5 Receive payments from debtors

The system needs to allow debtors to make payments for their credit purchases.

**[FR-81] Receive Payment Functionality**

---

**Description:** The following requirements outline the functionality for receiving payments. Note that the relevant accounting needs to occur, refer to ??

**Requirements:** The user must specify the following to record a payment:

1. The debtor
2. The invoice for which the payment is made to
3. The amount

The system must assume any invoice with a zero balance has been completely paid.

**Ranking:** Essential

**[FR-82] Record debtor payment interface**

---

**Description:** The system must provide an interface that allows the user to record payments made by the debtor.

**Requirements:** The following must be displayed

1. List of debtors
2. List of non-zero balance invoices for each debtor

Once the invoice has been selected, the user must specify:

3. The amount to be paid (this can exceed the outstanding balance on the invoice)
4. The date of payment
5. Comments

If the amount being paid exceeds the amount on the invoice, an 'unallocated payment' account which is of type 'liability' will need to be created for that particular debtor. Any excess payments will be transferred to this account. The next time the debtor receives an invoice, the amount in the 'unallocated payment' account will be used to pay for the invoice. The outstanding balance on the invoice will then need to be paid by the debtor.

**Ranking:** Essential

**[FR-83] Payment accounting**

---

**Description:** When a debtor makes a payment the following occurs:

**Requirements:** If the amount paid is the same as the invoiced amount, the following occurs:

1. The bank account is debited by the payment amount.
2. The debtor's account is credited by the payment amount.

If the amount paid exceeds the invoiced amount, the following occurs:

3. The bank account is debited by the payment amount.
4. The debtor's account is credited by the invoiced amount.
5. The unallocated payment account is credited by the excess amount.

When the debtor receives another invoice, the following occurs:

6. The debtor's account is credited by the amount in the unallocated payment account as long as it does not exceed the invoiced amount. If the invoiced amount exceeds the amount in the unallocated payment account, the debtor has an outstanding balance that needs to be paid.

**Ranking:** Essential



## 7.6 Credit notes

### [FR-84] Credit note functionality

---

**Description:** A detailed description of a credit note can be found in the glossary. The following requirements outline the functionality of the credit note.

**Requirements:** The following information must be stored for each credit note

1. Credit note number
2. Date
3. Customer (if applicable)
4. Comments

**Ranking:** Desirable

### [FR-85] Credit note interface

---

**Description:** The system needs to provide an interface for creating and viewing credit notes

**Requirements:**

1. The system must provide an interface for printing credit notes.
2. The system must allow new credit notes to be generated.

**Ranking:** Desirable

### [FR-86] Credit note printing

---

**Description:** The system must allow credit notes to be printed

**Requirements:** The printed credit note must display:

1. The company details and ABN
2. All details recorded in the credit note (see [FR-84] )

**Ranking:** Desirable

Refer to section Additional requirements for the credit note accounting requirement.

## 7.7 Customer deposits

A separate account needs to be created (Deposit trust account) to handle customer deposits.

[FR-87] **Customer deposit account**

---

**Description:****Requirements:**

1. If the customer deposit trust account does not already exist in the chart of accounts, the system must create it.
2. The account type is "Asset"
3. The account category is "current asset"

**Ranking:** Optional[FR-88] **Customer deposit accounting**

---

**Description:****Requirements:** When a deposit is made:

1. Deposit trust account is increased by the deposit amount
2. Prepaid revenue account is increased by the deposit amount

When the final purchase is made:

3. Deposit trust account is decreased by the deposit amount
4. Bank is increased by the deposit amount
5. Prepaid revenue is decreased by the deposit amount
6. Revenue is increased by the deposit amount

**Ranking:** Optional[FR-89] **Receipt of deposit**

---

**Description:****Requirements:** A receipt is printed, recording:

1. the reference for the deposit (deposit ID)
2. Customer name
3. Amount
4. Relevant invoice

**Ranking:** Optional

Refer to section Additional Requirements for delete quote, delete quote interface, delete sales order, delete sales order interface, delete invoice and delete invoice interface requirements.

## 8 Functional Requirements: Suppliers

The following requirements are split as follows:

1. Creditor details
2. Purchase order
3. Bill
4. Make payments to creditors

### 8.1 Creditor details

The system must maintain a list of creditors and provide functionality to create, update, delete and view the list.

[FR-90] **Creditor details**

---

**Description:** A creditor is outlined here. Refer to the glossary for a detailed accounting based definition of a creditor. BSB and IBAN are defined in the glossary.

**Requirements:**

1. A creditor is an account
2. The account name is independent from the creditor's legal name
3. The account type is always "liability"
4. The account category is always "accounts payable"

The following information can be stored:

5. Creditor's legal company name (the user must specify)
6. Creditor's ABN and ACN
7. Creditor's legal address
8. Creditor's phone numbers
9. Creditor's email addresses
10. Creditor's fax numbers
11. Creditor's bank details
  - (a) Account name
  - (b) BSB
  - (c) External Bank Account number
  - (d) IBAN
12. Comments

**Ranking:** Essential

**[FR-91] Add new creditor functionality**

---

**Description:** The functional requirements for adding a new creditor are outlined below

**Requirements:**

1. The system must allow new creditors to be added
2. The system must allow the creditor's details to be specified. See requirement [FR-90] .

A new account must be created with:

3. account type 'liability'
4. account category 'accounts payable'

**Ranking:** Essential

**[FR-92] Add new creditor interface**

---

**Description:** The interface requirements for adding a new creditor are outlined below

**Requirements:**

1. The system must provide an interface for allowing a user to create new creditors.
2. The interface must implement all of the functionality listed in requirement [FR-91] .

**Ranking:** Essential

**[FR-93] Update existing creditor functionality**

---

**Description:** The functional requirements for editing an existing creditor are outlined below

**Requirements:**

1. The system must allow creditor details to be edited
2. The creditor account type cannot be changed (from "liability")
3. The creditor account category cannot be changed (from "accounts payable")

**Ranking:** Essential

**[FR-94] Inactive creditors**

---

**Description:** A creditor can be flagged as inactive if the user wishes to stop making credit purchases with the creditor.

**Requirements:** The system must allow:

1. creditors to be flagged as inactive
2. inactive creditors to be flagged as active
3. account balance must be zero before flagging an account as inactive
4. no new transactions can be made where it relates to an inactive creditor

**Ranking:** Desirable

**[FR-95] List of creditors functionality**

---

**Description:** The functional requirements for listing creditors are outlined below.

**Requirements:** The system must be able to produce a list of all creditors, orderable by:

1. account number
2. account name
3. account balance
4. latest activity date (optional)
5. active/inactive/all

Note: Activity date means that the creditor has made a purchase on a give date.

The system must be able to filter creditors from the database who match a given criteria, including:

6. Threshold balance amount (i.e. less than \$x or greater than \$y)
7. creditors that have supplied a given item (optional)
8. Activity date range (optional)

**Ranking:** Desirable

**[FR-96] List of creditors interface**

---

**Description:** This interface will provide the user with a list of creditors, and any relevant details associated with that creditor.

**Requirements:** The system must provide an interface to display a list of creditors including:

1. each creditor's account number, name, balance See requirement ?? (Optional)
2. any user defined field from each creditor (i.e. phone number) (Optional)
3. The user must be able to reorder the list according to requirement [FR-95]
4. The user must be able to filter the list according to requirement [FR-95]

**Ranking:** Desirable

**[FR-97] Balance of all accounts payable interface**

---

**Description:** The balance of all accounts payable (creditors) needs to be accessible somewhere by the user.

**Requirements:**

1. The system must display a list all creditors, their account number and balance.
2. The system must display the final creditor balance.

**Ranking:** Essential

**[FR-98] Delete creditor functionality**

---

**Description:** Deletion is generally discouraged, as it may be more appropriate to flag the creditor as inactive. If the user really does want to delete a creditor with transactions associated, they must re-link the transactions with another creditor beforehand.

**Requirements:**

1. The system must allow creditors to be deleted
2. The deleted creditor account must not have any transactions associated with the account
3. All deleted creditors must be recoverable
4. The deleted account number must be reusable in future accounts

**Ranking:** Optional

**[FR-99] Relinking creditor transactions**

---

**Description:** Transactions that have been recorded under the wrong creditor may need to be edited. The system needs to provide a facility to do this for every transaction associated with a particular creditor.

**Requirements:**

1. The system must allow all transactions from a given creditor to be re-linked, or re-associated with another creditor.
2. Any relinking must be reversible (Desirable)

**Ranking:** Optional

**[FR-100] Delete creditor interface**

---

**Description:** From the user's perspective, deleting a creditor may require other tasks such as re-associating transactions before performing the actual delete. The user should be discouraged from performing deletions and guided towards flagging a creditor as inactive.

**Requirements:** If the deleted account has associated transactions, the system must ask the user if they would like to:

1. re-link the transactions to another creditor account and then delete
2. leave the transactions and flag the account as inactive (see [FR-94] ), or
3. prevent the delete from taking place

**Ranking:** Optional

## 8.2 Purchase order

Businesses usually maintain a generic order form, which they fill and send to their suppliers to order goods & services. Additionally, this form allows the user to indicate the price of goods but the final cost is determined by the actual supplier. Essentially, this is not a binding contract for any transaction, but is used to initiate a purchase transaction. It is useful for businesses to keep track of these documents because it allows them to verify their requests with the actual invoice a supplier might issue. This will have no accounting impact.

**[FR-101] Purchase order functionality**

---

**Description:** A detailed description of a purchase order can be found in the glossary. The following requirements outline the functionality of the purchase order. This is a stand-alone form and there is no accounting back-end. Over the counter purchases do not necessarily record the details of the supplier.

**Requirements:** A purchase order must have the following:

1. Unique purchase order number
2. Date
3. Supplier's details (name, address, phone number). See [FR-91] for name, address and phone number requirements.
4. Name and address of the user's business
5. List of items to be purchased:
  - (a) Item name
  - (b) Item description
  - (c) Item quantity
  - (d) Selection to indicate if all costs are 'exclusive of GST' or 'inclusive of GST'
  - (e) Item unit cost
  - (f) Selection of a type of tax (see requirement [FR-141] )
  - (g) Automatically generate GST component of line item. This calculation will be different based on whether the costs are GST inclusive or not.
  - (h) Generate total line item cost (unit cost of item x quantity)
6. Generate total cost of all line items
7. Generate total GST component of the ordered items
8. Generate total price of ordered items inclusive of GST. This will only be different to 'total cost of all line items' if the costs are exclusive of GST.
9. Comments

**Ranking:** Essential



**[FR-102] Purchase order interface**

---

**Description:** The system must provide an interface for creating and viewing purchase orders.

**Requirements:** The interface must allow the user to:

1. create new purchase orders
2. view existing purchase orders
3. give access to all of the functionality listed in [FR-101] .
4. must automatically fill in details of existing suppliers
5. allow purchase orders to be edited.(optional)
6. generate a bill from a purchase order. See requirement ??

**Ranking:** Essential

**[FR-103] Purchase order printing**

---

**Description:** The system must allow purchase orders to be printed

**Requirements:** The printed purchase order must display:

1. The company details and ABN
2. All details recorded in the purchase order. See [FR-101] )

**Ranking:** Essential

### 8.3 Bill

Once an order is received by the supplier (with or without a purchase order form), the business receives a bill for the purchase price (the bill is essentially an invoice from the suppliers point of view). Details of a bill must be recorded before any payments are made to a supplier. This ensures that a business has access to the details of any purchases for future references. The bill, unlike a purchase order, is an actual accounting transaction and so requires an accounting back-end to automatically perform the relevant ledger entries into the accounting system. Additionally, the bill can be transacted in 2 different ways: cash purchases or credit purchases.

[FR-104] **Bill functionality**

**Description:** A description of a bill can be found in the glossary.

**Requirements:** The following information is recorded in a bill:

1. a unique bill number
2. identify whether it is a credit transaction or cash transaction
3. supplier's name
4. due date of payment (if it is a credit purchase)

The system must allow the following to be recorded but it is not necessary for the user to enter these details:

5. Invoice number (the number on the bill received from the supplier)
6. an option to link existing purchase order forms to this bill (a purchase order form can be linked to more than one bill). When such a link is made, all common details are to be automatically filled from the selected purchase order. This will assist the user by not having to re-enter details.
7. Date the bill was issued
8. Terms of purchase comments
9. Delivery form number

The system must allow the user to specify details of items purchased. The following details must be specified or automatically generated from the purchase order:

10. Item name
11. Item description
12. Item quantity
13. Selection to indicate if all costs are 'exclusive of GST' or 'inclusive of GST'
14. Item unit cost
15. Selection of a type of tax (see requirement [FR-141] )
16. Automatically generate GST component of line item. This calculation will be different based on where the costs are GST inclusive or not.
17. Generate total line item cost (unit cost of item x quantity)
18. Generate total cost of all line items
19. Generate total GST component of the ordered items
20. Generate total price of ordered items inclusive of GST. This will only be different to 'total cost of all line items' if the costs are exclusive of GST.
21. comments

**Ranking:** Essential

**[FR-105] Bill interface**

---

**Description:** The system needs to provide an interface for creating and viewing bills

**Requirements:**

1. The system must allow new bills to be generated.
2. It must be possible to automatically fill in the details of existing suppliers.
3. All bills generated must be editable. (optional)
4. The interface must clearly indicate whether this bill is being transacted on credit or cash.

**Ranking:** Essential

**[FR-106] Credit purchase accounting**

---

**Description:** When a Bill is entered into the system the following accounting entries need to be entered into the ledgers in order for a credit purchase transaction to take place:

**Requirements:**

1. The creditor's account must be credited for the total price (including tax)
2. The inventory account must be debited for the total amount exclusive of GST.
3. The chosen tax type account must be debited for only the total tax amount
4. The completion of the above actions must indicate that the related bill has not been paid

**Ranking:** Essential

**[FR-107] Cash purchase accounting**

---

**Description:** When a bill is created, the following accounting entries need to be entered into the ledgers in order for a cash purchase transaction to take place.

**Requirements:**

1. The bank account must be credited for the total price (including tax)
2. The inventory account must be debited for the total amount exclusive of GST.
3. The chosen tax type account must be debited for only the total tax amount.
4. The completion of the above actions must flag that the related bill has been paid.

**Ranking:** Essential

**[FR-108] Bill printing**

---

**Description:** The system must allow bills to be printed.

**Requirements:** The printed bill must display:

1. All details recorded in the bill (see requirement [FR-105] )

**Ranking:** Essential

**[FR-109] List of bills**

---

**Description:** The functional requirements for listing bills saved in the system are described below.

**Requirements:**

1. The system must allow bills to be listed under the following search parameters:
  - (a) Unpaid credit purchase bills
  - (b) Overdue credit purchase bills (compared to the user's computer date and time)
  - (c) Bills of type credit, type cash or both
  - (d) Bills which are paid, unpaid, or both
2. The list must show these high-level details:
  - (a) bill number
  - (b) Purchase Order number
  - (c) Invoice number
  - (d) Net Total Amount (exclusive of Tax)
  - (e) Total Amount (inclusive of Tax)
  - (f) Supplier name
  - (g) Date of bill
  - (h) Due date if an unpaid credit purchase bill
3. Additionally listed bills (as per above requirements), may be linked to the actual bill. (see requirement ??)

**Ranking:** Desirable

**[FR-110] List of bills interface**

---

**Description:** The interface requirements for the list of bills interface are described below.

**Requirements:**

1. The system must provide an interface for viewing bills
2. The system must implement all of the functionality listed in [FR-109] .

**Ranking:** Desirable

## 8.4 Make payments to creditors

### [FR-111] Make payment functionality

---

**Description:** When a business purchases goods on credit from its suppliers, the business will need to pay the supplier. This action results in a transaction affecting relevant ledgers.

**Requirements:** The user must specify the following to make a payment to a creditor:

1. creditor's account name
2. bill number
3. date of payment
4. whether the payment is:
  - (a) In full (essential)
  - (b) Partial (optional)
5. Payment transaction must be entered into the relevant ledger accounts, see requirement [FR-112]
6. For partial payments, the system must indicate a partial payment made and the amount already paid next time the bill is viewed. (optional)
7. For full payments the system must indicate that a bill has been fully paid next time the bill is viewed.
8. The amount being paid. All payments are made inclusive of tax

**Ranking:** Essential

### [FR-112] Creditor payment accounting

---

**Description:** The functional requirements detailing the accounting back-end for payments made to creditors are as follows:

**Requirements:**

1. The creditor's account must be debited by the amount being paid (including tax).
2. The Bank account must be credited for the total amount being paid.

**Ranking:** Essential

**[FR-113] Make payment interface**

---

**Description:** The interface requirements for making payments to creditors are outlined below.

**Requirements:**

1. The system must provide an interface for allowing payments to be recorded.
2. The interface must allow all the functionality as described in [FR-111] .
3. The interface must show partial payments and the outstanding amount of each bill. If a full payment has been made, the bill should be flagged as paid and should not be visible in the list of outstanding bills.

**Ranking:** Essential

**[FR-114] Print details of payment**

---

**Description:** The system must allow payments made to be printed.

**Requirements:** The printed payment must display:

1. creditor's account name
2. bill number
3. date of payment
4. whether the payment is:
5. The amount being paid. All payments are made inclusive of tax
  - (a) In full (essential)
  - (b) Partial (optional)

**Ranking:** Essential

**[FR-115] List payment history**

---

**Description:** The system must allow the user to view payments made.

**Requirements:** The system must provide a list of all payments within a given date range specified by the user. The following details need to be displayed:

1. supplier's name
2. amount paid
3. date of payment
4. whether the payment was a partial payment or full payment

**Ranking:** Optional

[FR-116] **List payment history interface**

---

**Description:** The interface requirements for viewing payment history are described below:

**Requirements:**

1. The interface must implement all the functionality described in requirement [FR-115] .

**Ranking:** Optional



## 9 Functional Requirements: Banking

### 9.1 Bank accounts

#### [FR-117] **Bank account**

---

**Description:** A description of a bank account can be found in the glossary. The following requirements outline the details of a bank account. The system must allow for the creation of numerous bank accounts. The system must also store the date the user last completed a bank reconciliation and the closing balance of the bank account at the time. Refer to the glossary for a detailed description of bank reconciliation.

**Requirements:**

1. A bank account is an account
2. The account type is always “current asset”
3. The category is always “bank”

In addition to the attributes of a regular account, a bank account stores:

4. Institution
5. Bank’s account number
6. IBAN
7. Account limits
8. Account type (credit, cheque, savings, trust)
9. Comments
10. Closed flag(if a bank account has been closed and is not active)
11. Date of last reconciliation or no date if never reconciled
12. Closing balance of the account at last reconciliation

**Ranking:** Essential

#### [FR-118] **Bank account interface**

---

**Description:** The system must allow the user to view and edit all details of a bank account.

**Requirements:**

1. The system must provide an interface for the creation of new bank accounts.
2. The system must allow the user to modify all details of an existing bank account.

**Ranking:** Essential

**[FR-119] Institution details**

---

**Description:** The system must allow the institutional details to be shared for multiple accounts. This prevents the user from having to enter the details of an institution more than once.

**Requirements:**

1. Institution details should have a single point of control where multiple accounts share the same institution.
2. Institution branch
3. Institution contact details (phone number, address, email)
4. BSB
5. Comments

**Ranking:** Optional

**[FR-120] Institution details interface**

---

**Description:** The system must provide an interface that allows the user to enter/modify the details of a banking institution.

**Requirements:**

1. The system must provide an interface for adding new institutions. It must allow the user to enter all the details outlined in [FR-119] .
2. The system must allow the user to modify all details of an existing banking institution.

**Ranking:** Essential

**[FR-121] List of bank accounts functionality**

---

**Description:** The functional requirements for listing bank accounts are outlined below.

**Requirements:** The system must be able to produce a list of bank accounts, orderable by:

1. Account number
2. Account name
3. Account balance
4. Institution
5. Account type (credit, cheque, savings, trust)

The system must be able to filter bank accounts from the database according to the criteria below:

6. Selected institution
7. Threshold balance amount (i.e. less than \$x or greater than \$y) (optional)

**Ranking:** Essential

**[FR-122] List of bank accounts interface**

---

**Description:** This interface will provide the user with a list of bank accounts, and any relevant details associated with that bank account.

**Requirements:** The system must provide an interface to display a list of bank accounts including:

1. Each bank account's ledger account number, bank (institution's) account number, name, balance, institution, branch, BSB, account type (cheque or savings).
2. The user must be able to reorder the list according to ??
3. The user must be able to filter the list according to ??

**Ranking:** Essential

**[FR-123] List of transactions**

---

**Description:** The system needs to show a list of transactions for a given bank account, which may be useful for reconciling with a statement from an institution.

**Requirements:** The following needs to be displayed for each transaction:

1. Date
2. Corresponding transaction account (in double entry accounting)
3. Amount of transaction, shown in a column appropriate to the nature of the transaction: that is, either credit or debit
4. Any taxes appropriate to each transaction.
5. Running balance

**Ranking:** Essential

**[FR-124] List of transactions interface**

---

**Description:** The system must display a list of transactions for a given bank account.

**Requirements:** The system must provide an interface to display a list of transactions including:

1. Date
2. Corresponding transaction account (in double entry accounting)
3. Amount of the transaction shown in a column appropriate to the nature of the transaction: that is, either credit or debit
4. Any taxes appropriate to each transaction.
5. Running balance

**Ranking:** Essential

## 9.2 Bank reconciliation

**[FR-125] Reconciliation statement**

---

**Description:** A bank reconciliation statement allows the user to reconcile all of the transactions in a bank account according to his/her record with the bank statement provided by the banking institution. The following requirements outline the functionality of the bank reconciliation statement.

**Requirements:** The system must provide a bank reconciliation report.

**Ranking:** Desirable

---

**[FR-126] Reconciliation with multiple accounts**

---

- Description:** Users may have multiple bank accounts and so the system needs to allow them to reconcile each one separately.
- Requirements:** If a user has access to multiple bank accounts, the user must be able to select which bank account they would like to reconcile if there are numerous accounts.
- Ranking:** Desirable

---

**[FR-127] Reconciliation dates**

---

- Description:** The starting date of the period being reconciled needs to be shown.
- Requirements:** The bank reconciliation report must display the starting date of the period being reconciled.
1. The starting date must be one day after the date the selected bank account was last reconciled.
  2. The user must be able to edit the starting date of the reconciliation period.
- Ranking:** Desirable

---

**[FR-128] Reconciliation opening balance**

---

- Description:** The opening balance of the period being reconciled needs to be shown. This must be the same as the closing balance from the last reconciliation. If the last reconciliation has not been performed, the system must allow the user to enter the opening balance.
- Requirements:** The bank reconciliation report must display the opening balance
1. The balance must be the same as the closing balance of the selected account when it was last reconciled.
  2. If the previous period has not been reconciled the user must be able to enter the opening balance of the bank account for the purposes of reconciliation.
- Ranking:** Desirable

---

**[FR-129] Separation of transaction types**

---

- Description:** Deposits and payments that have not yet been reconciled, need to be separated. These may be displayed in a tabbed interface.
- Requirements:** All transactions that have not yet been reconciled in the bank account must be split into two groups: deposits and payments.
- Ranking:** Optional

**[FR-130] Reconciled transactions**

---

**Description:** During the reconciliation process, transactions will move through three states: Not reconciled, ticked, reconciled. The state will allow the system to track the progress of reconciliation, and allow a user to return to an unfinished reconciliation session. The system needs to record the state of all transactions, note that this does not need to be explicit, and an unhandled transaction can be assumed to have a non-reconciled state.

**Requirements:** The system will keep track of the state of all transactions for the purposes of the reconciliation process. The system must attribute one of the following states to every transaction:

1. Not reconciled (not ticked or reconciled)
2. Ticked but not complete (ticked but not reconciled-this could be because the user had to leave and was unable to complete the reconciliation process)
3. Complete (ticked, reconciled and confirmed)

**Ranking:** Desirable

**[FR-131] Additional transactions**

---

**Description:** Bank statements are not just a statement of all transactions, they are also a record of additional transactions such as bank fees and interest. These will not yet be recorded in the system and so the system will need to allow these additional transactions to be accounted for.

**Requirements:**

1. The system must allow for any extra bank-related charges appearing in the bank statement to be entered.
2. The system must sum the reconciled entries and these extra charges and produce a total to be added to the bank reconciliation.
3. The user must be able to enter the closing balance based on the Banking institution's official statement.

**Ranking:** Desirable

**[FR-132] Final reconciliation**

---

**Description:** Upon reconciling, the system needs to check the difference between the total sum calculated and the closing balance entered by the user. This difference is calculated by subtracting the institution's closing balance from the system's calculated closing balance.

**Requirements:** Upon reconciling, the system must check the difference between the calculated sum of transactions and the closing balance entered by the user. If there is a discrepancy, the system must:

1. Report the fact that the balances don't match and the amount of the discrepancy.
2. Allow the user to create an automatic adjustment.

**Ranking:** Desirable

**[FR-133] Reconciliation adjustments**

---

**Description:** Requirements relating to adjustments are outlined below.

**Requirements:**

1. Adjustments must be made to a default account called "Adjustments"
2. The "Adjustments" account is of type "asset"
3. If this account does not exist, the user must be prompted to create it.
4. If a discrepancy is negative (i.e. the institution's balance is greater than the system's balance), the system must debit the Adjustment account.
5. If the discrepancy is positive (i.e. the institution's balance is less than the system's balance), the system must credit the Adjustment account.

**Ranking:** Desirable

**[FR-134] Post reconciliation**

---

**Description:** The following requirements relate to post-reconciliation actions

**Requirements:**

1. Once the system knows that the bank account has been reconciled, the system must flag all entries that have been reconciled as complete.
2. The system must record the date of reconciliation and the final balance which will later be used for the next reconciliation.

**Ranking:** Desirable

**[FR-135] Reconciliation interface**

---

**Description:** The system needs to provide an interface for completing bank reconciliations.

**Requirements:**

1. The system must allow a bank reconciliation form to be created.
2. The user must be able to view and edit the starting and end date of the reconciliation period.
3. The user must be able to enter the closing balance based on the Banking institution's official statement.
4. The user must be able to flag each transaction by ticking or marking the transaction.
5. The bank reconciliation form must be implemented using a tabbed interface between deposits and payments. This will readily allow the user to alternate between these pages. These are usually at the top of each page.
6. The user must be able to enter any extra charges such as bank charges, interest charges etc.
7. There must be a button to reconcile the bank account once all transactions have been entered.

**Ranking:** Desirable

**[FR-136] Reconciliation printing**

---

**Description:** The system must allow completed bank reconciliation statements to be printed.

**Requirements:** The printed bank reconciliation statement must display:

1. The company details and ABN
2. Bank account number being reconciled
3. One of the following options:
  - (a) A summary statement containing a total of all deposits and a total of all payments.
  - (b) A detailed statement containing a list of every transaction and its amount.

**Ranking:** Desirable



## 10 Functional Requirements: Reporting

### 10.1 Balance sheet

[FR-137] **Balance sheet functionality**

---

**Description:** The balance sheet is a report based on the equation  $A = L + P$ . Refer to the glossary for full definitions of 'balance sheet', 'T-ledger', 'asset', 'liability' and 'proprietorship'.

**Requirements:** The output must display:

1. the title 'Balance Sheet'
2. the company's legal name
3. the relevant time period
4. a T-ledger
5. on the left side of the T-ledger, accounts of type asset under the heading 'Assets', with a total at the end
6. on the right side of the T-ledger, accounts of type liability on the right under the heading 'Liabilities', with a total at the end
7. on the left side of the T-ledger, a list of accounts of type proprietorship under the heading 'Proprietorship', with a total at the end
8. profit must be displayed under proprietorship
9. the sum of all assets, liabilities, proprietorship
10. a clear warning, if the system detects that  $A \neq L + P$  do not balance.
11. if the user has defined a custom view (optional)

**Ranking:** Desirable

**[FR-138] Balance sheet interface**

---

**Description:** Several suggested time period shortcuts are specified based on what are commonly used time periods.

**Requirements:** The user must be able to specify:

1. the required time period
2. the required detail of accounts and categories.
3. whether current and non-current assets and liabilities are displayed separately
4. shortcut for selecting the past month as the time period (optional)
5. shortcut for selecting three months as the time period (optional)
6. shortcut for selecting the past 6 months as the time period (optional)
7. shortcut for selecting the past 12 months as the time period (optional)
8. shortcut for selecting the financial year to date as the time period (optional)

**Ranking:** Desirable

## 10.2 Profit and loss statement

**[FR-139] Profit and loss functionality**

---

**Description:**

**Requirements:** The output must display:

1. the title 'Profit and loss statement'
2. the company's legal name
3. the relevant time period
4. a list of accounts of type revenue under the heading 'Revenues' with a total at the end
5. a list of accounts of type expense under the heading 'Expenses' with a total at the end
6. a net profit, calculated as revenues less expenses under the heading 'Net profit'

**Ranking:** Desirable

**[FR-140] Profit and Loss interface**

---

**Description:** Several suggested time period shortcuts are specified as options, based on what are commonly used time periods.

**Requirements:** The user must be able to specify:

1. the required time period
2. the required detail of accounts and categories.
3. shortcut for selecting the past month as the time period (optional)
4. shortcut for selecting the past 6 months as the time period (optional)
5. shortcut for selecting the past 12 months as the time period (optional)
6. shortcut for selecting the financial year to date as the time period (optional)

**Ranking:** Desirable

## 11 Functional Requirements: Taxation

### 11.1 Taxes

#### [FR-141] Australian taxes

---

**Description:** The system must support all Australian taxes relating to the sale and purchase of goods and services. Each tax must have a tax code and rate (%). The tax code is used to uniquely identify each GST tax. The tax rate is used to calculate the amount of a transaction that is taxed (e.g. GST = 10.0%).

**Requirements:** The following GST tax codes must be supported:

1. Capital Acquisitions - GST-Free (Tax rate = 0.0%)
2. Capital Acquisitions - GST (Tax rate = 10.0%)
3. Capital Acquisitions - For making Input Taxed Supplies (0.0%)
4. GST-Free Exports (Tax rate = 0.0%)
5. GST-Free Supplies (Tax rate = 0.0%)
6. GST (Tax rate = 10.0%)
7. Non-Capital Acquisitions - GST-Free (Tax rate = 0.0%)
8. Non-Capital Acquisitions - GST (Tax rate = 10.0%)
9. Non-Capital Acquisitions - For making Input Taxed Supplies (0.0%)

Each type of tax must have:

10. A tax code (A unique integer identifier)
11. Name (The name of the tax - as listed above)
12. The tax rate (Percentage)

**Ranking:** Essential

#### [FR-142] Updating tax tables

---

**Description:** The user may wish to edit the contents in the tax table. This must be easily accessible to the user. A user must also be able to enter additional taxes that may be applicable to their business.

**Requirements:**

1. The user must be able to modify existing data in the tax table. The modifiable data includes:
  - (a) The tax name.
  - (b) The applicable tax rate.

**Ranking:** Essential

**[FR-143] Updating tax tables from a file**

---

**Description:** The system must allow the tax tables to be updated from a file.

**Requirements:**

1. The file with which the tax tables must be updated should be a CSV file or XML file with each tax code, name and rate.
2. The system must not allow the user to input a tax table for a different country than the one existing in the system.

**Ranking:** Optional

**[FR-144] Tax reporting period**

---

**Description:** The user must be able to select their taxation period.

**Requirements:** The user must enter the following details:

1. The frequency of the tax reporting period (monthly, quarterly, biannually, annually)
2. The start date of each period
3. The end date of each period

**Ranking:** Essential

**[FR-145] International Taxation**

---

**Description:** The system must be extensible so as to allow other countries' tax systems to be incorporated.

**Requirements:** The system must provide a framework for replacing the existing Australian taxation module with a country-specific taxation module.

**Ranking:** Desirable

## 11.2 Tax Reports

### [FR-146] GST report

---

**Description:** The user must be able to produce a GST report. A GST report shows the amount of tax that the company has collected, as well as the amount of tax that the company has paid.

**Requirements:**

1. The system should be able to display the GST report.
2. The system must display the GST component of each transaction from sales and purchases. (\$)
3. The system must calculate the net total of GST payable/refundable to the ATO. (\$)

**Ranking:** Optional

### [FR-147] GST report interface

---

**Description:** The user must be able to view a GST report.

**Requirements:** The system must provide an interface to display a GST report including:

1. The company name and ABN
2. The interface must implement all of the functionality listed in [FR-146] .

**Ranking:** Optional

### [FR-148] GST report printing

---

**Description:** The system must allow the GST report to be printed.

**Requirements:** The printed GST report must display:

1. The company details and ABN
2. All details in [FR-146] .

**Ranking:** Optional

[FR-149] **BAS report**

---

**Description:** The user must be able to produce a BAS report (a tax liability report). The BAS report shows the user the amounts to enter into each section on their BAS. This report only covers the GST amounts the company owes the ATO from sales and the GST amounts the ATO owes the company from purchases.

**Requirements:** The BAS report must conform to the guidelines on a BAS provided by the ATO. The BAS requires the following information :

1. Starting date of the tax period
2. End date of the tax period
3. Total sales including GST (\$)
4. Export sales (\$)
5. Other GST free sales (\$)
6. Total GST on sales (the amount owed to the ATO)(\$)
7. Capital purchases including GST (\$)
8. Non-capital purchases including GST (\$)
9. Total GST on purchases (the amount the ATO owes the business) (\$)
10. Net GST payable/refund-able. This is the difference between the total GST on purchases and the total GST on sales. (\$)

**Ranking:** Optional

[FR-150] **BAS report interface**

---

**Description:** The user must be able to view a BAS report.

**Requirements:** The system must provide an interface to display a BAS report including:

1. The company details and ABN
2. The interface must implement all of the functionality listed in [FR-149] .

**Ranking:** Optional

[FR-151] **BAS report printing**

---

**Description:** The system must allow the BAS report to be printed.

**Requirements:** The printed BAS report must display:

1. The company details and ABN
2. All details in [FR-149] .

**Ranking:** Optional

## 12 Additional requirements

### 12.1 Customers

#### [FR-152] Delete quote

---

**Description:** The functional requirements for deleting a quote are described below:

**Requirements:**

1. The system must allow saved quotes to be deleted
2. If a quote is linked to a sales order or an unpaid invoice, the system must then reject the delete action and inform the user.

**Ranking:** Optional

#### [FR-153] Delete quote interface

---

**Description:** The system must provide an interface for deleting saved quotes.

**Requirements:**

1. The interface must provide the functionality as described in requirement [FR-152]

**Ranking:** Optional

#### [FR-154] Delete sales order

---

**Description:** The functional requirements for deleting a sales order are described below:

**Requirements:**

1. The system must allow saved sales orders to be deleted
2. If a sales order is linked to an unpaid invoice, the system must then reject the delete action and inform the user.

**Ranking:** Optional

#### [FR-155] Delete sales order interface

---

**Description:** The system must provide an interface for deleting saved sales orders.

**Requirements:**

1. The interface must provide the functionality as described in requirement [FR-154]

**Ranking:** Optional



**[FR-156] Delete invoice**

---

**Description:** The functional requirements for deleting an invoice are described below:

**Requirements:**

1. The system must allow saved invoices to be deleted
2. If an invoice is unpaid, the system must then reject the delete action and inform the user.

**Ranking:** Optional

**[FR-157] Delete invoice interface**

---

**Description:** The system must provide an interface for deleting saved invoices.

**Requirements:**

1. The interface must provide the functionality as described in requirement [FR-156]

**Ranking:** Optional

**[FR-158] Credit note accounting**

---

**Description:** When a credit note is issued (due to a debtor returning goods), the following must occur:

**Requirements:**

1. The sales revenue account is debited by the amount to be paid by the debtor.
2. The debtor's account is credited by the amount that was purchased on credit.
3. The cost of goods sold account is credited by the cost of the goods returned.
4. The inventory account is debited by the cost of the goods returned.

**Ranking:** Desirable

## 13 Non-Functional Requirements

### 13.1 Installation

#### [NFR-1] Windows packaged installation

---

- Description:** Easy installation is a must for the sake of the non-technical Windows end users.
- Requirements:** The final product must be made available as a single file package, ready for automated installation on Windows based systems.
- Ranking:** Essential

#### [NFR-2] Linux and Mac packaged installation

---

- Description:** Easy installation for other operating systems is desirable.
- Requirements:** The final product must be made available as a single file package, ready for automated installation on Debian, Red Hat and Mac OS X based systems.
- Ranking:** Optional

#### [NFR-3] Installation time

---

- Description:** There needs to be a limit for the amount of time the installation process takes.
- Requirements:** The user must be able to be install the software in under 5 minutes on a host system that meets the Minimum System Requirements.
- Ranking:** Desirable

#### [NFR-4] Deinstallation

---

- Description:** It is important to ensure a clean, easy exit. This means that the process is quick and host system is left in a clean state. An easy exit is also aided by the ability to export the data in a usable format, see also [FR-14] .
- Requirements:**
1. The user must be able to be uninstall the software in under 5 minutes on a host system that meets the Minimum System Requirements.
  2. The system must remove all replaceable distribution files.
  3. The system must not remove any system data unless the user has given permission to do so.
- Ranking:** Desirable

## 13.2 Performance

For a desktop program which is going to be used 8 hours a day, performance is a central concern.

### [NFR-5] Fast start-up

---

- Description:** Waiting for an application to load can be frustrating for an end user. To them, a quick start up is extremely important.
- Requirements:** The System must have a start up time of less than 5 seconds, on a system that meets the Minimum System Requirements.
- Ranking:** Desirable

### [NFR-6] Reporting results from searches

---

- Description:** The system must report results from searches within a specified period of time.
- Requirements:** The following minimal performance requirements must be met for a reasonably fast machine:
1. When the system searches through 1 year's worth of transactions, the results from the search must be reported within 30 seconds.
  2. When the system searches through the entire database, the results from the search should be reported within 3 minutes.
- Ranking:** Essential

### [NFR-7] Responsive

---

- Description:** As financial transactions have the potential to consume system resources, the system needs to keep processes under control, and keep the user interface responsive at all times.
- Requirements:**
1. The System must react instantly to all user input (optional).
  2. The user must be able to cancel any processes that may potentially take a long time to finish.
- Ranking:** Essential

## 13.3 Security

Back-end reliability is supreme; no data should ever be lost as a result of software failure. To guard against hardware failure, the user may wish to create backups of their accounting data. The software needs to facilitate this. Encryption is not necessary, as it can be handled on the filesystem level.

**[NFR-8] Zero tolerance data loss**

---

**Description:** Data loss must be avoided at all costs. Using a mature existing relational database system is one way to ensure this.

**Requirements:**

1. If the software crashes or system experiences power failure, no data is lost or corrupted.
2. User must be able to backup all data in one step
3. User must be able to automate the backup process

**Ranking:** Essential

**[NFR-9] Authentication**

---

**Description:** Authentication may be required where the user wants to prevent other users from accessing the system. Note that not all users will want this, as it may entail entering a password every time the program is run.

**Requirements:** Simple user and role based authentication must be an option to the end user

**Ranking:** Desirable

## 13.4 Platform

The software is expected to run under Windows, Linux and Mac OS with no more than a re-compile. We don't expect you to test the software under all three operating systems, but you must choose an application framework which is multi-platform.

**[NFR-10] Windows compatibility**

---

**Description:** Most small businesses run some form of Windows. Windows 95 support is not necessary.

**Requirements:** The software must compile and run successfully on a Windows 98 and Windows XP machine.

**Ranking:** Essential

**[NFR-11] Apple OS X compatibility**

---

**Description:** Some small businesses may use Apple OS X. Supporting Apple OS X is important step towards achieving the goal of being cross-platform.

**Requirements:** The software must compile and run successfully on a Mac OS X machine.

**Ranking:** Optional

**[NFR-12] Future Apple OS X compatibility**

---

**Description:** Future development may provide full compatibility with Apple OS X  
**Requirements:** The system must not use any libraries or components which preclude future Apple compatibility.  
**Ranking:** Essential

**[NFR-13] Linux compatibility**

---

**Description:** There is little pre-existing accounting software available for Linux. As this project attempts to provide one, it is critical that the software runs on Linux.  
**Requirements:** The software must compile and run successfully on a machine running Debian, Ubuntu, SUSE or Red Hat.  
**Ranking:** Optional

**[NFR-14] Future Linux compatibility**

---

**Description:** Future development may provide full compatibility with Linux  
**Requirements:** The system must not use any libraries or components which preclude future Linux compatibility.  
**Ranking:** Essential

## 13.5 Minimum Hardware Requirements

**[NFR-15] Minimum Hardware Requirements**

---

**Description:** Not all small businesses can afford modern hardware. The software needs to run on machines that are not top of the line.  
**Requirements:** The software must be able to be run on a simple desktop, with the following specifications:  

1. 256MB RAM

  
**Ranking:** Desirable

## 13.6 Extensibility

### [NFR-16] Plug-ins

---

**Description:** Plug-ins allow third parties to extend the software without editing the source code, or in some cases recompiling. As the scope is quite large, this will be necessary for a complete and comprehensive implementation of small business accounting software.

**Requirements:**

1. The system must document a plug-in framework to allow third-party development of plug-ins.
2. The plug-ins need to be writable and understandable by Mid-Level Coders.

**Ranking:** Desirable

## 13.7 Custom report modules

This project will not attempt to second-guess and implement every possible report that users will want to see. It would be desirable to allow users to customise reports. The following three requirements present three different approaches to doing this.

### [NFR-17] Basic reporting language

---

**Description:** An easy to understand language would enable the end users themselves to write reports.

**Requirements:** The system must provide an easy to understand, basic programming language

**Ranking:** Optional

### [NFR-18] Interpreted reporting language

---

**Description:** A more powerful programming language would allow technical consultants to write reports for clients (users).

**Requirements:** Providing an interface for a powerful interpreted programming language to be used.

**Ranking:** Desirable

### [NFR-19] End user report interface

---

**Description:** A drag and drop graphical interface would allow the end user to very easily create and modify reports.

**Requirements:** The system must provide a non-technical graphical interface for creating and customising reports.

**Ranking:** Optional

## 13.8 Environment

### [NFR-20] Multiple users

---

- Description:** The system should be able to give access to more than one user, not necessarily simultaneously, but with varying privileges.
- Requirements:** System must be able to support multiple users.
- Ranking:** Optional

### [NFR-21] Client/server

---

- Description:** A client server setup will allow small businesses with a centralised server, to provide access to the system from multiple workstations simultaneously. This improves productivity where businesses have 2-3 people in administration.
- Requirements:** The system must allow data to reside in a single server, with access being obtained through any number of clients simultaneously.
- Ranking:** Optional

### [NFR-22] Peer to peer

---

- Description:** A peer to peer system would allow multiple users access to the data simultaneously. The data would be synchronised between the peers.
- Requirements:** The system must allow multiple users to manipulate data simultaneously without using a centralised server.
- Ranking:** Desirable

## 13.9 Licence

### [NFR-23] Licence

---

- Description:** The software needs to release the source code to the world. Note that the following requirements do not preclude the use of a dual commercial/open source licence.
- Requirements:**
1. All components must have a GPL compatible licence.
  2. The system source code must be made available under a GPL licence.
- Ranking:** Essential

### 13.10 Internationalisation

#### [NFR-24] Back-end language independence

---

- Description:** The back-end must be language independent, and must not need to be updated to add support for a new language. For example, error messages must be identified by a reference string, and any accompanying text must only exist purely as a development aid.
- Requirements:** The back-end must not produce language specific output.
- Ranking:** Essential

#### [NFR-25] Internationalisation

---

- Description:** The language of the software interface needs to be easily interchangeable.
- Requirements:** The software must allow the interface to be translated into multiple languages
- Ranking:** Desirable

#### [NFR-26] Localisation

---

- Description:** Different countries will have different customs, standards and laws that will require the software to operate completely differently.
- Requirements:** The software must allow itself to be adapted to conform to customs, standards and laws of any given country.
- Ranking:** Optional

### 13.11 File formats

#### [NFR-27] File formats

---

- Description:** To promote interoperability of data between software.
- Requirements:** All native file formats used must be open and documented
- Ranking:** Essential



## 13.12 Language

### [NFR-28] Chosen development language

---

**Description:** The longevity of the software relies on the language it is written in remaining modern, popular and actively developed.

**Requirements:** The Client needs to approve the chosen development languages. The Client may reject the chosen language only on the basis that they believe it is not likely to have active development in 5 years time. The following languages already have approval:

1. C
2. C++
3. Python
4. GTK2
5. Qt4
6. WxWidgets

**Ranking:** Essential

### [NFR-29] Acceptable databases

---

**Description:** The selection of an appropriate database needs to be approved by the client.

**Requirements:** The following database system must be used. Any other databases must be approved by the Client.

1. SQLite

**Ranking:** Essential

### [NFR-30] Database transaction support

---

**Description:** The accounting system must support a large number of transactions at a time.

**Requirements:** The system must deal with an average of 500 000 transactions at a time.

**Ranking:** Essential

**[NFR-31] Database support**

---

**Description:** Users may wish to choose which database back-end they would like to use. This would be made possible if the system supported multiple databases.

**Requirements:** All of the following databases must be connectable after compilation:

1. MySQL
2. Postgres
3. SQLite

**Ranking:** Desirable

## 14 Use cases

### 14.1 Small business supplier

#### 14.1.1 Overview

When a purchase is made, the business does the following:

- records customer and order details
- produces various documents
- records accounting details

These are the documents that are created in the process:

- order
- receipt of deposit
- invoice
- receipt

#### 14.1.2 Process

When a customer wishes to purchase an item, an order is made recording exactly what the company needs to supply. The customer pays a deposit on the total price, this money is not paid to the business, rather it is simply held by the business on trust in case the sale is never completed. A document is produced as evidence of the receipt of the deposit.

The company does what it needs to do to fulfill the order, this may take some time.

When the order is finally ready, an invoice is generated including the amount owed and possibly the details of the order to help explain how the amount was calculated. When the invoice has been issued, money is owed to the business and the business records this amount as expected income (accounts receivable). The amount of inventory that is supplied to the customer is also recorded in the business's accounts.

When the user pays the full purchase price, a receipt is issued as evidence of the transaction, recording the parties involved (the business and customer), the amount paid and the products delivered. The business records the reason for the income.

#### 14.1.3 Reports

The business manager needs to access sums of all assets, sales and accounts receivable at any time. Once a month the manager needs to generate a BAS statement and once a year tax statements. Continually, the manager will want to see various custom accounting based

statistics depending on the current need. Usually this is simply a sum of a group of accounts over a given period of time.

## **14.2 Lessor**

### **14.2.1 Overview**

The customers are constant and payments are identical and periodical. The business does the following:

- produce various documents
- record accounting details

These are the documents that are created in the process:

- invoice
- receipt of deposit/bond
- receipt

### **14.2.2 Process**

When a new customer signs a lease, a bond is paid which is held on trust by the lessor. A document is produced as evidence of receipt of the bond and the transaction needs to be recorded.. Rent is paid in advance at a fixed amount per month.

Each month, an almost identical invoice is produced by the lessor for each of the clients, detailing the amount owed. The money owed is recorded by the lessor as expected income (accounts receivable). The details, amount and terms on the monthly invoices are identical, only the date and invoice number changes. When the lessee pays the amount due, the lessor deposits the money in the bank and this transaction is recorded.

### **14.2.3 Reports**

The lessor needs to access sums of all assets, sales and accounts receivable at any time. Once a month the manager needs to generate a BAS statement and once a year tax statements.

## **14.3 Bulk data entry**

### **14.4 Process**

A pile of invoices and receipts is entered into the system as they are received and as they are paid. The user will work their way through the series of receipts and invoices, entering in

the date, amount, relevant accounts, direction and a comment. Most of the time, this will be a simple double entry transaction, with the same amount being entered twice. Sometimes, the user will need to enter a split transaction where money from one account is split into two accounts.

#### **14.4.1 Traits of data entry**

In the case of bills, the date of the transaction (when the bill was paid) will often be the same. This is a time consuming and unexciting job and the user will not want to repeat themselves, by repeatedly entering in the same date, same comment or same transaction amount. Many bills, such as rent payments, will always have the same amount, direction and comment and will have a predictable date (such as once a month). Often, transactions for exactly the same amount are for the same reason.

## 15 Acceptance Criteria

The deliverable product must meet certain criteria. If the criteria is not met, the Client need not accept the product. The Client will sign off the SRS as the requirements for each iteration are gathered.

### 15.1 SRS Acceptance

For the Software Requirements Specification to be accepted by the Client the document should be:

- Complete;
- Consistent;
- Clear;
- Concise, and
- Correct.

### 15.2 Product Acceptance

The software product will be accepted by the Client provided the following minimum standards are met:

- One hundred percent of the design of all essential requirements are completed.
- At least eighty percent of the implementation of essential requirements are completed.
- At least eighty percent of the design of all desirable requirements are completed.
- At least fifty percent of the implementation of all desirable requirements are completed.
- There are no expectations for the completion of optional requirements.

### 15.3 Product Deliverables

This system will be accepted by the Client provided all the following documents are delivered by the Team:

- Tar file containing source code, CVS repository and user documentation
- SRS document;
- SDD;
- Test Plan document, and
- Test cases and the test driver.

## 16 Client Sign-off

This section will be used by the Client to sign-off this document and the software product. Once the document has been signed by both parties, the document will act as a formal agreement to the expectations of the Accounting Management System between the Client and the Team. Any changes to the SRS following sign-off will need to follow the procedure outlined in section 16.2.

### 16.1 Changing Requirements

Any changes to requirements include:

- Addition of a new requirement
- Deletion of a requirement
- Modification of an existing requirement

Modifications include altering the rank, the priority or the wording of a requirement.

### 16.2 SRS change management

Changes may be proposed by either the Client or Team. If a change is to be carried through, the following procedures must be followed:

1. The party proposing the change will notify the other party through written communication (includes email) detailing the description of the change, reasons for the change and the expected outcome from the change.
2. Both the Team and Client must agree to the change in writing before the SRS is modified.
3. The new version of the SRS will be sent to the Client.
4. This revised document will then be used as the final contract between the Team and Client.

### 16.3 SRS Sign-off Iteration 1

The Client and Team agrees that the requirements outlined in this document are the required needs and expectations of the final product to be delivered.

I, the undersigned, do hereby agree that I have read and understood the contents of this SRS and that the contents of this SRS depict all the requirements required for the Accounting and Payroll System in a complete, correct and consistent manner.

Cybersource \_\_\_\_\_  
*Client*

Anita Kotagiri \_\_\_\_\_  
*Client Liaison Officer*

Will Hardy \_\_\_\_\_  
*Client Liaison Officer*

### 16.4 SRS Sign-off Iteration 2

The Client and Team agrees that the requirements outlined in this document are the required needs and expectations of the final product to be delivered.

I, the undersigned, do hereby agree that I have read and understood the contents of this SRS and that the contents of this SRS depict all the requirements required for the Accounting and Payroll System in a complete, correct and consistent manner.

Cybersource \_\_\_\_\_  
*Client*

Anita Kotagiri \_\_\_\_\_  
*Client Liaison Officer*

Will Hardy \_\_\_\_\_  
*Client Liaison Officer*



### 16.5 Product Acceptance

I, the undersigned do hereby agree that the delivered product meets the requirements as stated in the SRS and that the product is accepted and complete.

Cybersource \_\_\_\_\_  
*Client*

Anita Kotagiri \_\_\_\_\_  
*Client Liaison Officer*

Will Hardy \_\_\_\_\_  
*Client Liaison Officer*

## Appendix A Requirement Ranking

This section summarises the relative importance of the requirements for this project. The importance of a requirement is described by one of the following priority ranks:

**Essential** These requirements are crucial to the usefulness of the system. A product failing any of these requirements would be deemed unacceptable.

**Desirable** These requirements are very important to the usefulness of the system. A product failing to meet a significant number of these requirements would be deemed as unacceptable.

**Optional** These requirements would improve the usefulness of the system. However, these requirements are not necessary for the product to be acceptable.

The following table indicates the importance of each requirement specified in this document by assigning a priority rank to it. These ranks will be used as a guideline for setting milestones and development cycles during the design and development of the product.

## Appendix B Glossary

**Account** Used to maintain a detailed record of events for a particular asset, liability, proprietorship, revenue or expense.

**Account Number** Every account is associated to an account number. This is a numeric way of identifying a particular account. Furthermore, account numbers for assets, liabilities, proprietorship, revenue and expense accounts fall within separate account number ranges (for easy identification of these account types). The ranges have industry standards but can be company specified.

**Accounting Management System** A system that allows businesses to manage their accounting needs.

**Accounts Receivable** A current asset representing money owed to the business for performing sales or services on credit to an external entity of the business.

**Accounts Payable** Is a current liability representing money owed by the business for receiving goods or services on credit from an entity external to the business.

**Accounting Period** For reporting purposes the life of a business is broken down into many time periods. The length of the period is defined by each business, normally it is usually one month, and all transactions are associated to a particular period.

**Accrued Expense** Is a current-liability that recognises an expense incurred within a particular accounting period but the payment of which is postponed.

**Asset** An asset is anything owned by the business which can produce future economic benefit.

**ATO** Australian Tax Office.

**Australian Business Number (ABN)** Is a single 11-digit identifier for use in business dealings with the ATO and with other government agencies.

**Australian Company Number (ACN)** Is a unique, nine-digit number, assigned to every company by law in Australia.

**Bad Debt Account** An account established to record a subtraction from the Accounts Receivable account, to allow for those debts that might not be paid. In accounting you have a conservative approach, where as soon as you identify a reduction in the value of an account you should recognise that.

**Balance sheet** Is an itemised statement that lists the total assets and total liabilities of a given business to portray its net worth at a given moment in time.

**Balance** For any account, the total of all its entries in the debit and credit side will result in a final debit/credit balance.

**Bank Account** This tracks all the cash flow into and out of your bank (with an external provider).

**Bank Reconciliation** Is the verification of a bank statement balance (from an external provider) with that of the bank account in your Account Management System. Essentially, the two balances should be equal and any differences reconciled.

**Business Activity Statement (BAS)** Used to report tax obligations and entitlements, including the amount of GST the business has paid and collected.

**Bill** A statement of costs or charges.

**Bank State Branch (BSB)** Is a unique number used to identify the bank and branch of where your bank-account (external) is held.

**Cash flows Statement** Is a report providing detailed information on the cash inflows and outflows of a business over a period of time. It essentially explains the reasons for the change in the balance of your bank account over a period of time.

**Chart of Accounts** Is a list of general ledger account names and associated account numbers arranged in the order in which they normally appear in the financial statements. Usually in the order of: assets, liabilities, proprietorship, revenue and expenses.

**Company** Is an organised group of people to perform an activity or business.

**Control Account** For those accounts that experience a lot of entries it is sometimes more useful to have a separate subsidiary ledger maintain a detailed record of those entries and then summary transferred back to the control account. Examples are Accounts Receivable, Accounts Payable, stock etc.

**Corporation** A type of business organisation that is given many of the legal rights as a separate entity.

**Cost of Goods Sold** A figure representing the cost of buying materials and goods, which were then sold to an external customer.

**Credit** In accounting represents a decrease or increase to an account depending on whether it is an asset, liability, proprietorship, revenue or expense. A credit entry increases all liability, proprietorship and revenue accounts.

**Creditor** An external entity to the business that has a financial obligation to the business.

**Credit Note** Is issued to indicate a reduction of a liability in a creditor's account. Credit notes are issued for reasons such as overpayment or duplicate payments by the customer, or for damaged goods received by the customer.

**Creditor** An external entity to the business, who your business has an financial obligation to pay.

**Current Asset** Are those assets that represent a financial benefit to the business within one normal operating cycle of business (usually one year). Typically if it is expected that it will release cash, or is sold or consumed within one operating cycle.

**Current Liability** Opposite to current asset. Are those liabilities that represent a financial obligation to an external entity but which is due within one operating cycle of business (usually one year).

**Debit** In accounting a debt represents a decrease of increase to an account depending on whether it is an asset, liability, proprietorship, revenue or expense. A debit entry increases all asset and expense accounts.

**Debtor** An external entity to the business, who has a financial obligation to your business.

**Debtor Subsidiary Ledger** Is a ledger of accounts that are maintained separately from the main general ledger of accounts. This particular ledger will maintain the individual debtor accounts and the summary details transferred into the 'accounts receivable' control account of the general ledger.

**Delivery Notice** The note issued to the receiver of goods upon the deliver of goods. Signature of the delivery notice acknowledges the correct receipt of goods.

**Deposits** Can mean a variety of things; a payment given as a guarantee for an obligation that will be met in the future; or the act putting money into a bank account; or a partial payment made at the time of purchase, with the balance to be paid later.

**Deposit Trust Account** When ever a deposit is payed as a guarantee to a financial obligation, such entries are noted into the deposit trust account.

**Depreciation** When a non-current asset is purchased, typically it will not be expensed in whole in the current accounting period. This is so that such an action will not distort the profit in that particular period. Because a non-current asset provides benefits over multiple accounting periods, it makes sense to spread its cost as an expense over multiple periods as well. Depreciation is the periodic expense of the cost of a non-current asset.

**Entity** Is something that has a distinct and separate existence.

**Expense** Is the cost of doing business.

**Financial Transaction** Is an event or happening that changes the financial position and/or earnings of the business. A financial transaction will entail at least two accounts with offsetting debit and credit entries.

**Financial-Year-End** Defined as the end of one full financial year. In Australia a financial year typically extends from July of one year to June of the next year and will typically incorporate many accounting periods (e.g. if each accounting period is one-month). Usually at the financial-year-end all transactions are summarised to generate reports on financial performance (profit & loss) and position (balance sheet).

**General Journal** Is the most basic of journals. It is a chronological list of transactions.

**General Ledger** Is the record of all entries into each of the accounts as defined in the 'chart of accounts'.

**GL code** A unique account identifier.

**Gross** The entire amount of income before any deductions are made.

**Goods and Services Tax (GST)** As per Australian Tax laws all goods sold and services provide incorporate an additional 10% tax.

**Inventory** Includes raw materials, items available for sale or in the process of being made ready for sale.

**Invoice** Is a detailed list of the goods shipped or services rendered, with an account of all costs; an itemised bill.

**Invoice Number** A unique number used to identify an invoice.

**IBAN** International Bank Account Number

**Liability** A company's legal debts or obligations that arise during the course of business operations.

**Ledger** Is a book of accounts in which data from transactions recorded in journals are posted and thereby classified and summarised.

**Net** In general, is the figure remaining after all relevant deductions have been made from the starting or gross amount.

**Net Profit** Is the company's total earnings, reflecting revenues adjusted for costs of doing business, depreciation, interest, taxes and other expenses.

**Non-Current Asset** An asset that will provide economic benefit to the business over one accounting period.

**Non-Current Liability** A liability that represents a financial obligation to an external entity of the business but not due within one accounting period.

**Opening Balance** With the exception of 'temporary accounts', all other accounts will have a 'closing balance' at the end of an accounting period and so will become the opening balance in the new period. For any new accounts or temporary accounts, opening balances will be zero.

**Partnership** Is an unincorporated business that has more than one owner.

**Payroll Subsidiary Ledger** Is a ledger of separate accounts holding a detailed record of all payroll employees.

**Prepaid Revenue / Unearned Revenue** Revenue paid for goods or services in advance.

**Prepaid Expenses** Expenses paid for in advance.

**Profit & Loss Statement** A statement listing the businesses revenues and costs over one accounting period. The net effect of which will indicate where a profit (revenue exceeds costs) or a loss (costs exceed revenue).

**Proprietorship** Refers to the entity that has ownership in all or part of a business.

**Quote** A form detailing an indicative price for the specified goods or services.

**Receipt** Is a written acknowledgment that a specified sum of money has been received.

**Revenue** The amount of money that a company receives from its activities

**Sales Order** A form processed when a sales transaction is initiated. It indicates the purchaser, the goods/services, prices & total cost, as well as other details (specified by the selling company).

**Service Business** Is a form of business providing labour services as opposed to the provision of goods.

**Sole Proprietorship** Is a business structure where the owner of the business and the business itself is not separate for tax and liability purposes.

**Subsidiary Ledger** Is a group of related accounts where the sum of the balances are periodically transferred to a related control account in the general ledger.

**Supplier** An external entity who provides goods to a business which are either modified or directly re-sold by the business.

**T-Account** Is a symbolic representation of an 'account'. The entries in an account are divided between those that are a credit and those that are a debit. Symbolically, all debits are displayed on the left-hand side under the account name and all credit entries are displayed on the right-hand side under an account name.

**Tax Invoice** Essentially, is a receipt for the purchase and payment of goods.

**Tax Expense** Is an amount that is recognised by the business as the amount likely to be paid to the tax office under the current accounting methods. NOTE: for actual tax purposes businesses may use different accounting methods (such as depreciation) which results in a different amount for 'tax payable'.

**Tax Payable** This is the amount that is actually payable to the tax office. This is different from 'tax expense', which may or may-not be equal to the 'tax payable'.

**Tax Rate** This a percentage that is used when calculating the amount that needs to be paid as tax.

**Temporary Account** Accounts that have a lifespan of one accounting period. These accounts are closed at the end of every accounting period with their balances transferred to other permanent accounts. Effectively they start every new period with a zero balance, whereas permanent accounts start every new period with the balance at the end of the previous period. Revenue and Expense type accounts are temporary accounts, with their combined balance transferred to a proprietorship account at the end of every accounting period.

**Transaction** Every financial dealing of a business is described as a transaction. By the very nature of the accounting setup the combined debit entries must equate to the combined credit entries.

**Unearned Revenue** Revenue that has been received by the business but for services or goods yet to be delivered to an external entity.

**Wholesale** Is the selling of goods to retail merchants; usually in large quantities for resale to consumers.